

**HERRAMIENTA DE SIMULACION DE PROTOCOLOS DE ACCESO AL
MEDIO EN TOPOLOGÍAS DE RED DE ÁREA LOCAL**

**VICKY XIMENA BONILLA SILVA
ESPERANZA CLARET ROMERO ABAD**

**CORPORACION UNIVERSITARIA TECNOLOGICA DE BOLIVAR
FACULTAD DE INGENIERIA DE SISTEMAS
CARTAGENA DE INDIAS D.T. y C.**

2000

**HERRAMIENTA DE SIMULACIÓN DE PROTOCOLOS DE ACCESO AL MEDIO
EN TOPOLOGÍAS DE RED DE ÁREA LOCAL**

**VICKY XIMENA BONILLA SILVA
ESPERANZA CLARET ROMERO ABAD**

**Trabajo de Grado presentado como requisito parcial para optar el título de
Ingenieros de Sistemas**

**Director:
CARLOS FERRIOL
Lic. en Ciencias Computacionales**

**Asesor:
MOISES QUINTANA
Lic. en Ciencias Computacionales**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS D.T. y C.**

2000

BIBLIOGRAFIA

BLACK, Uyles. Redes de Computadores, Protocolos, Normas e Interfaces. 2ed. Madrid: Rama, 1995. Pág. 82-115.

CARBALLAR, José A. El libro de las comunicaciones del PC, técnica programación y aplicaciones. Madrid: Rama, 1996. Pág. 321-340.

GONZALEZ SAINZ, Nestor. Protocolos de Comunicaciones para Sistemas Abiertos. Argentina: Addison-Wesley Iberoamericana, 1996. Pág. 55-70.

TANENBAUM, Andrew S. Redes de Computadora. 3ed. México: Prentice Hall, 1997.

<http://www-mat.upc.es/~alcober/PAPERS/Alc97/slides/index.htm>

<http://www.inmesoft.com/ForRedes.htm>

<http://www.serinfo.es/guiacx.htm>

http://www.op.upm.es/tesmes/infraestructuras/doc/c/cred_1.html

<http://www.geocities.com/SiliconValley/Heights/6497/ejemplos.html>

<http://www.geocities.com/SiliconValley/Heights/6497/arquitectura.html>

<http://www.disc.ua.es/asignaturas/rc/trabajos/lan3/rc.html>

<http://sunsite.fisicc-ufm.edu/~obonilla/cn/l20/node9.html>

<http://sunsite.fisicc-ufm.edu/~obonilla/cn/l20/node8.html>

<http://sunsite.fisicc-ufm.edu/~obonilla/cn/l20/l20.html>

<http://www.disc.ua.es/asignaturas/rc/tema1.html>

<http://www.fisio.cinvestav.mx/fisio/index-lan.html>

<http://arraquis.dif.um.es/~rafa/8trab.htm>

<http://vgg.sci.uma.es/redes/topo.html>

<http://ftp.rhon.itam.mx/~extuni/sat/comp/redes/lantech.html>

<http://vgg.sci.uma.es/redes/bajos.html>

<http://vgg.sci.uma.es/redes/index.html>

<http://ftp.rhon.itam.mx/~extuni/sat/comp/redes/conceptos.html>

<http://www.facyt.uc.edu.ve/~rrodrigu/redes/clase4.html>

ANEXO B Manual del Sistema

En el siguiente documento quedara expuesto la mayor parte del código fuente del software *Simulador de Protocolos de Acceso al Medio*, así como también el conjunto de funciones de las cuales él consta.

El software esta diseñado en la poderosa herramienta de programación Visual Basic 5.0, la cual se utiliza para realizar cada una de las simulaciones que posee el programa.

El diseño del programa esta compuesto por un conjunto de funciones locales y globales al proyecto las cuales quedaran expuestas por simulaciones de protocolo.

Se recomienda que las personas que deseen leer este documento tengan al menos un poco de conocimiento acerca de la sintaxis de programación de Visual Basic 5.0.

CONTENIDO

	Pág.
INTRODUCCION	
1. MARCO TEÓRICO	23
1.1 REDES DE ÁREA LOCAL	23
1.1.1 Topología de redes de área local	24
1.1.1.1 Topología en bus	24
1.1.1.2 Topología en anillo	25
1.1.1.3 Topología estrella	26
1.2 PROTOCOLOS DE COMUNICACIÓN	27
1.3 PROTOCOLOS DE ACCESO AL MEDIO	28
1.4 ORGANISMOS DE ESTANDARIZACION	30
1.5 LAS DIRECTRICES DEL PROYECTO 802 DEL IEEE	31
1.5.1 IEEE 802.3 Acceso múltiple con detección de portadora y detección de colisión (CSMA/CD)	33
1.5.1.1 Método de operación	33
1.5.1.2 Visión general del estándar IEEE 802.3	36
1.5.1.3 Protocolo del subnivel MAC	37
1.5.1.3.1 Estructura de la trama MAC	37
1.5.1.3.2 Control de acceso al medio	39

1.5.2	IEEE 802.4 Paso de Testigo en Bus	40
1.5.2.1	Método de operación	40
1.5.2.2	Visión general de estándar IEEE 802.4	41
1.5.2.3	Protocolos del subnivel MAC	44
1.5.2.3.1	Estructura de la trama MAC	44
1.5.2.4	Gestión de acceso al medio	46
1.5.2.4.1	Iniciación del anillo	46
1.5.2.4.2	Inserción en el anillo	47
1.5.2.4.3	Eliminación del anillo	48
1.5.2.4.4	Recuperación del token	49
1.5.2.5	Esquema de prioridades	50
1.5.3	IEEE 802.5 Paso de Testigo en Anillo	51
1.5.3.1	Método de operación	51
1.5.3.1.1	Principios de operación	51
1.5.3.1.2	Conceptos generales de las redes en anillo	52
1.5.3.1.3	Método de control de acceso	55
1.5.3.2	Visión general del estándar IEEE 802.5	56
1.5.3.3	Protocolos del subnivel MAC	58
1.5.3.3.1	Estructura de la trama MAC	58
1.5.3.4	Problemáticas y gestión del anillo	61
1.5.3.4.1	Estación monitora y control del testigo	62

1.5.3.4.2	Notificación de estaciones contiguas	63
1.5.3.4.3	Recuperación de la estación monitora activa	63
1.5.3.4.4	Liberación anticipada del testigo	64
1.5.3.4.5	Sistema de prioridades	65
1.6	MODELO DE COMUNICACIÓN CLIENTE SERVIDOR	65
2.	EL PROBLEMA Y LOS OBJETIVOS DE LA INVESTIGACIÓN	66
2.1	ANTECEDENTES DEL PROBLEMA	66
2.2	PROBLEMA	67
2.3	DESCRIPCIÓN DEL PROBLEMA	67
2.4	JUSTIFICACIÓN	68
2.5	OBJETIVOS	68
2.5.1	Objetivo General	68
2.5.2	Objetivos Específicos	69
2.6	RESULTADOS DE LA INVESTIGACIÓN	70
3.	ENTORNO DEL DISEÑO DEL SOFTWARE	72
3.1	INTRODUCCION	72
3.2	DESTINATARIOS	73
3.3	PRESENTACION	73
3.4	AREAS DEL CONTENIDO	74
3.5	CONTEXTO	74
3.6	ANALISIS DE LOS RESULTADOS	75
3.7	COMPLEJIDAD	75
3.8	CONTROL INSTRUCCIONAL	76

3.9	AYUDAS	77
4.	DESCRIPCION DE ASPECTOS DEL DISEÑO DEL SOFTWARE	78
4.1	ASPECTOS GENERALES DE LA SIMULACION	78
4.2	CARACTERISTICAS DEL PROCESO SERVIDOR	79
4.2.1	Características de ejecución	79
4.2.2	Características de diseño	80
4.3	CARACTERISTICAS DEL PROCESO CLIENTE	81
4.3.1	Características de ejecución	81
4.3.2	Características de diseño	81
4.4	MENSAJES EN LA RED	81
4.5	ESTADOS DE LOS EQUIPOS	83
5	DESCRIPCION DE PROTOCOLOS DE ACCESO AL MEDIO EN TECNOLOGIAS DE AREA LOCAL (LAN)	84
5.1	IEEE 802.3 CSMA/CD	84
5.1.1	Funcionamiento	84
5.1.2	Características	85
5.2	IEEE 802.4 TOKEN BUS	85
5.2.1	Funcionamiento	85
5.2.2	Características	86
5.3	IEEE 802.5 TOKEN RING	87
5.3.1	Funcionamiento	87
5.3.2	Características	88
5.4	ASPECTOS A TENER EN CUENTA EN LA SIMULACIÓN	90

6.	<i>EVALUACION DEL RENDIMIENTO</i>	91
6.1	CSMA/CD	93
6.2	PASO DE TESTIGO EN BUS	94
6.3	PASO DE TESTIGO EN ANILLO	95
7.	DISEÑO DEL SOFTWARE	97
7.1	DESCRIPCION GENERAL DE LOS MODULOS PRINCIPALES	97
7.1.1	Definición modelo general de la aplicación	98
7.1.2	Modulo general de la aplicación	99
7.2	DESCRIPCION DETALLADA DE CADA MODULO PRINCIPAL	100
7.2.1	Modulo de Usuario Final	100
7.2.1.1	Modulo de Usuario Final (Cliente)	100
7.2.1.1.1	Definición de las Tareas del Modulo Usuario Final (Cliente)	101
7.2.1.2	Modulo de Usuario Final (Servidor)	102
7.2.1.2.1	Definición de las Tareas del Modelo Usuario Final (Servidor)	103
7.2.2	Modulo de Comunicaciones	104
7.2.2.1	Definición Modulo de Comunicaciones	105
7.2.3	Modulo de Resolución de Peticiones	106
7.2.3.1	Definición del Modulo de Petición de Resoluciones	107
7.2.4	Modulo de Tareas	109
7.2.4.1	Modulo de Tareas (Cliente)	109
7.2.4.2	Modulo de Tareas (Servidor)	110
8.	CONCLUSIONES	

BIBLIOGRAFÍA

ANEXOS

LISTA DE FIGURAS

Figura 1. Topología en Bus.....	5
Figura 2. Topología en Anillo.....	6
Figura 3. Topología en Estrella.....	6
Figura 4. Funcionamiento del estándar 802.3.....	13
Figura 5. Arquitectura IEEE 802.3.....	16
Figura 6. Formato del paquete MAC.....	18
Figura 7. Red paso de testigo en bus.....	22
Figura 8. Arquitectura IEEE 802.4.....	24
Figura 9. Estructura de la trama MAC IEEE 802.4.....	25
Figura 10. Marcos de control del token bus.....	26
Figura 11. Red paso de testigo en anillo.....	33
Figura 12. Arquitectura IEEE 802.5.....	38
Figura 13. Formato de trama IEEE 802.5.....	40
Figura 14. Marcos de control del token ring.....	41

LISTA DE ANEXOS

Anexo A. Manual del sistema

Anexo B. Manual del sistema

RESUMEN

La implementación y uso creciente de las redes de comunicación en la actualidad, da a conocer la necesidad de investigación y estudio acerca de todos los temas que de alguna u otra manera tienen relación con ellas.

Como parte determinante dentro del manejo de las comunicaciones se puede destacar el estudio de los protocolos de acceso al medio como uno de los aspectos fundamentales en la transmisión de información a través de una red.

El presente proyecto tiene como finalidad el desarrollo e implementación de una herramienta basada en la simulación de protocolos de acceso al medio: CSMA/CD Token Bus y Token Ring sobre redes de área local en las topologías: Anillo Bus y Estrella. Para su elaboración se llevó a cabo una intensa investigación, de donde se seleccionaron temas específicos que ayudaran a su estudio y análisis.

Por muy complejo que parezca, una red de ordenadores no es más que un conjunto de éstos, capaces de intercambiar información entre ellos mediante una serie de conexiones físicas que los une; facilitando el proceso de comunicación.

La elección de una topología en la implementación de una red se encuentra en gran parte influenciada por el tipo de acceso al medio utilizado, dado que determinadas topologías se adaptan mas naturalmente a unos métodos de acceso que a otros.

El control de acceso al medio define los protocolos o mecanismos que deben seguir las estaciones para compartir de manera satisfactoria un recurso común como es la capacidad de transmisión de la red. Para que dos entidades se comuniquen con éxito deben “hablar el mismo lenguaje”. Qué comunican, cómo se comunican y cuándo lo hacen constituyen un conjunto de reglas aceptadas mutuamente entre las entidades involucradas, denominadas Protocolos.

Existen varios métodos utilizados para transferir paquetes de datos desde la memoria de una estación hacia el enlace físico de la red. Cada método dependerá de la placa de red instalada en la estación y de la topología del sistema de cableado de la red.

Acceso múltiple por detección de portadora y colisión (CSMA/CD). (Carrier Sense Multiple Access/Collision detection) que quiere decir algo así como “Puedes transmitir cuando quieras, siempre que no haya nadie empleando el canal, pero controla tu transmisión asegurándote de que nadie lo está haciendo”.

Con este método, las estaciones que desean transmitir, primero “escuchan” el medio para determinar si ya hay alguien empleándolo. Si es así, esperan hasta que esté libre. Si el canal está vacío comienzan a transmitir sus paquetes, haciendo pausas por paquetes para dar oportunidad a otras estaciones a capturar el enlace (acceso múltiple por detección de portadora).

Mientras se transmite cada paquete, la estación comprueba continuamente que la información que aparece en el canal es lo que debía (detección de colisión). Si no es así, se supone que otra estación ha comenzado a transmitir a destiempo y ambas transmisiones se han mezclado produciéndose una colisión. En estas circunstancias se suspende la transmisión y espera un cierto tiempo antes de mandar de nuevo el paquete, mientras que la otra estación hará lo mismo.

Para que las dos estaciones no vuelvan a transmitir al mismo tiempo, el tiempo que cada una espera viene determinado por un generador de números al azar incluido en cada una.

Este es el método de acceso al cable utilizado por redes locales Ethernet con una topología lineal (bus).

Acceso al bus por paso de testigo (TOKEN BUS). Este sistema de control de acceso se emplea en topologías bus o árbol, pero haciéndola funcionar como si fuera un anillo lógico.

Partiendo de una red de topología bus o árbol. A cada estación se le asigna un orden determinado en secuencia cerrada, constituyendo un anillo lógico.

Cada estación sólo puede transmitir cuando recibe un paquete determinado de control llamado token (testigo) permaneciendo las demás a la escucha y durante un tiempo previamente establecido, pasando el token a la siguiente estación de la secuencia y así hasta la última que lo pasa a la primera cerrando el anillo.

Este sistema precisa de una serie de controles que permitan incluir o retirar una estación de la secuencia sin romper la cadena. También hay que prever que se pierda el token y poder regenerarlo.

Las redes ArcNet utilizan este método de acceso al medio con una topología en bus, pero se utilizan hubs para distribuir las estaciones de trabajo en una configuración de estrella.

Acceso al anillo por paso de testigo (TOKEN RING). Este método está diseñado para ser empleado en redes con topología en anillo. Se basa en un paquete especial llamado token que circula constantemente por el anillo. Si ninguna estación transmite, el token aparece marcado como libre.

Cuando una estación quiere transmitir, espera y captura el token marcándolo como ocupado y añadiendo la información que desee junto con la dirección de la estación de destino, enviando el token a la siguiente estación de la secuencia. Esta examina el token detectando que está ocupado y lo pasa a la siguiente.

Cuando el token llega a la estación de destino lo marca como leído y lo pasa a la siguiente estación hasta alcanzar la estación que lo originó, este detecta que el mensaje ha llegado a su destino y marca el token como libre para que pueda capturarlo cualquier estación.

Antes de montar una red hay que tener en cuenta consideraciones como cableado, topología y método de control de acceso. Por esto es necesario crear estándares, ya que las posibles combinaciones harían imposible la comunicación.

Generalmente, las placas de red y los sistemas de cableado están diseñados siguiendo las especificaciones del "Institute of Electrical and Electronics" (IEEE), que ha creado estándares de red en el que cada uno define una topología que ha de emplearse con un sistema de control de acceso al medio.

Para tal estandarización, el IEEE trabajó el proyecto 802 formado por varios comités cada uno de los cuales define métodos de controlar el acceso al medio físico de transmisión.

El Estándar IEEE 802.3

Define una familia de normativas para detección de portadora como la CSMA/CD de Ethernet, como sistema de control de acceso al medio.

El Estándar IEEE 802.4

Define un estándar de pase de testigo en topología bus (token bus) como sistema de control de acceso al medio.

El Estándar IEEE 802.5

Define un estándar de pase de testigo en una topología en anillo, como la red Token Ring de IBM para redes locales.

INTRODUCCION

Teniendo en cuenta la importancia y el auge de las redes de comunicación hoy día, el presente proyecto tiene por objeto proveer una sencilla herramienta en software, fácil, clara, pero a la vez completa y confiable, que asista en el estudio y evaluación de los protocolos de acceso al medio en redes de ámbito local.

La importancia de este proyecto radica en el conocimiento concreto y claro que deja la presente investigación en lo tocante a los protocolos de comunicación usados en las redes, en específico las de área local; ya que permite conocer las diferentes normas o reglas por las cuales se rige cada uno para establecer la comunicación; al mismo tiempo que muestra como la selección de la topología determina el buen desempeño de una red.

Este trabajo se llevó a cabo luego de una detallada y exhaustiva lectura y análisis de bibliografía, de una amplia consulta a personal especializado en el área y de un completo proceso de selección de la información recopilada; todo dirigido a encontrar patrones que nos permitieran parametrizar la tarea de analizar y evaluar los protocolos de comunicación en redes de área local. En un sentido general, este trabajo resume todos aquellos aspectos primordiales y determinantes en este tema.

Se tocan aspectos sencillos como las diferentes topologías que distinguen las redes de área local de otros tipos de redes, los protocolos de comunicación y los estándares para las LAN producidos por la IEEE, conocidos en conjunto como IEEE 802 (CSMA/CD, token bus y token ring).

Este documento que ahora presentamos, al igual que los manuales y el software producto de la investigación pueden ser de gran ayuda a todos aquellos que deseen abordar el estudio de los protocolos de acceso al medio en topologías de área local.

Cabe anotar que el conocimiento requerido para utilizar de manera óptima el software de la investigación es mínimo, en lo relativo a los protocolos de comunicación. Lo anterior se debe a que la aplicación tiene una clara orientación educativa, sin perder la rigurosidad en los resultados arrojados.

**HERRAMIENTA DE SIMULACION DE PROTOCOLOS DE ACCESO AL MEDIO
EN TOPOLOGÍAS DE RED DE ÁREA LOCAL**

VICKY XIMENA BONILLA SILVA

Cod. 9305035

ESPERANZA CLARET ROMERO ABAD

Cod. 9305532

CORPORACION UNIVERSITARIA TECNOLOGICA DE BOLIVAR

FACULTAD DE INGENIERIA DE SISTEMAS

CARTAGENA

1998

**HERRAMIENTA DE SIMULACIÓN DE PROTOCOLOS DE ACCESO AL MEDIO
EN TOPOLOGÍAS DE RED DE ÁREA LOCAL**

VICKY XIMENA BONILLA SILVA

Cod. 9305035

ESPERANZA CLARET ROMERO ABAD

Cod. 9305532

**Anteproyecto del Trabajo de Grado presentado como requisito parcial para
optar el título de Ingenieros de Sistemas**

CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR

FACULTAD DE INGENIERÍA DE SISTEMAS

CARTAGENA

1998

Cartagena, Diciembre 11 de 1998.

Señores:

COMITÉ DE EVALUACIÓN DE PROYECTOS
CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
L. C.

Respetados señores.

Por medio de la presente nos permitimos presentar a ustedes para que sea puesto a consideración el estudio y aprobación del anteproyecto de grado que lleva por nombre: **“HERRAMIENTA DE SIMULACIÓN DE PROTOCOLOS DE ACCESO AL MEDIO EN TOPOLOGÍAS DE RED DE ÁREA LOCAL”**, para optar el título de Ingenieros de Sistemas.

Agradeciendo de antemano la atención prestada a la presente.

Atentamente,

VICKY BONILLA SILVA
Cod. 9305035

ESPERANZA ROMERO ABAD
Cod. 9305532

CONTENIDO

	Pág.
1. MARCO TEÓRICO	7
1.1 REDES DE ÁREA LOCAL	7
1.1.1 Topología de redes de área local	8
1.1.1.1 Topología en Bus	8
1.1.1.2 Topología en Anillo	8
1.1.1.3 Topología Estrella	9
1.2 PROTOCOLOS DE COMUNICACIÓN	10
1.3 PROTOCOLOS DE ACCESO AL MEDIO	10
1.3.1 CSMA/CD	12
1.3.2 Token Bus	12
1.3.3 Token Ring	13
1.4 MODELO DE COMUNICACIÓN CLIENTE SERVIDOR	13
2. EL PROBLEMA Y LOS OBJETIVOS DE LA INVESTIGACIÓN	15
2.1 ANTECEDENTES DEL PROBLEMA	15
2.2 PROBLEMA	16
2.3 DESCRIPCIÓN DEL PROBLEMA	16
2.4 JUSTIFICACIÓN	16
2.5 OBJETIVOS	17

2.5.1	Objetivo General	17
2.5.2	Objetivos Específicos	18
2.6	RESULTADOS DE LA INVESTIGACIÓN	19
3.	ESTRATEGIA OPERATIVA ADMINISTRACIÓN Y CONTROL DE LA INVESTIGACIÓN	21
3.1	ESPECIFICACIÓN OPERACIONAL DE ACTIVIDADES	21
3.2	CRONOGRAMA DE ACTIVIDADES	22
4.	ASPECTOS DE INFRAESTRUCTURA	23
4.1	RECURSOS HUMANOS	23
4.2	DESCRIPCIÓN DE EQUIPOS Y/O LABORATORIOS REQUERIDOS	23

BIBLIOGRAFÍA

ANEXOS

1. MARCO TEÓRICO

1.1 REDES DE ÁREA LOCAL

Una red de área local es un medio de transmisión de información que proporciona la interconexión entre diversos ordenadores terminales y periféricos situados en un entorno reducido y perteneciente a una sola organización.

Entre sus características podemos anotar:

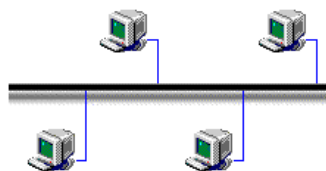
- ◆ El radio que abarca es de pocos kilómetros.
- ◆ Utilizan un medio privado de comunicación.
- ◆ Las velocidades de transmisión que más se manejan están alrededor de 2-10 Mbps.
- ◆ Pueden atender a cientos de dispositivos muy distintos entre sí (impresoras ordenadores, discos, teléfonos, módems etc.).

Una característica que distinguen las redes de área local de otros tipos de redes son las diferentes topologías que presenta.

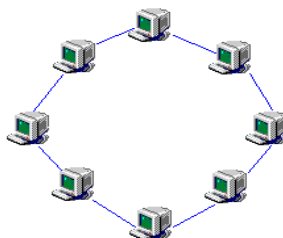
1.1.1 Topología de redes de área local. Se llama topología de una red al patrón de conexión física entre sus nodos es decir a la forma en que están interconectados los distintos nodos que la forman.

Entre las topologías que se acogen a los criterios expuestos tenemos:

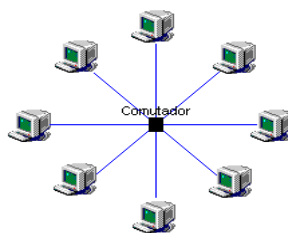
1.1.1.1 Topología en Bus. Una red en forma de Bus o Canal de difusión es un camino de comunicación **bidireccional** con puntos de terminación bien definidos. Cuando una estación transmite, la señal se propaga a ambos lados del emisor hacia todas las estaciones conectadas al Bus hasta llegar a las terminaciones del mismo.



1.1.1.2 Topología en Anillo. La topología en anillo se caracteriza por un camino *unidireccional* cerrado que conecta todos los nodos.



1.1.1.3 Topología Estrella. Se caracteriza por tener todos sus nodos conectados a un controlador central. Todas las transacciones pasan a través del nodo central, siendo éste el encargado de gestionar y controlar todas las comunicaciones. Por este motivo, el fallo de un nodo en particular es fácil de detectar y no daña el resto de la red pero un fallo en el nodo central desactiva la red completa.



Para que sea posible una comunicación debe existir un conjunto de reglas o mecanismo que faciliten este proceso, las cuales difieren de acuerdo a la topología de red empleada; por esto existe una estrecha relación entre la topología utilizada en una red de área local y los protocolos que ésta maneja; ya que la topología va a ser determinante en el diseño del protocolo.

1.2 PROTOCOLOS DE COMUNICACION

Los protocolos de comunicación son las reglas y procedimientos utilizados en una red para establecer la comunicación entre los nodos que disponen de acceso a la red.

Es indispensable para el intercambio de mensajes diseñar ciertos procedimientos o protocolos. El término protocolo se usa para describir las reglas o normas para el intercambio de información entre procesos (programas que se ejecutan en un hardware).

Las funciones más importantes que cumplen los protocolos son: *Control de errores*, protege integridad de los datos del usuario y de los mensajes de control; *Control de Flujo y Congestión*, permite a la red compartir sus recursos entre un gran número de usuarios, entregando a cada uno un servicio satisfactorio sin que sus operaciones corran peligro; y *Estrategias de*

enrutamiento, permite optimizar la utilización de los recursos de la red aumentando la disponibilidad de los servicios de la red al proveer caminos alternativos entre nodos terminales.

1.3 PROTOCOLOS DE ACCESO AL MEDIO

Por acceso al medio se puede entender la competencia existente entre los nodos de una red por acceder al mismo tiempo un medio físico.

Según el acceso al medio las redes pueden dividirse en dos categorías: las que usan conexiones punto a punto, que consisten en muchas conexiones entre pares individuales de máquinas (para ir del origen al destino, un paquete en este tipo de red puede tener que visitar primero una o más máquinas intermedias) y las que usan canales de difusión, que se caracterizan por tener un solo canal de comunicación compartido por todas las máquinas de la red.

Para el fin de nuestro proyecto nos limitaremos a estudiar los protocolos de acceso al medio en las redes de difusión.

En cualquier red de difusión, el asunto clave es la manera de determinar quién puede usar el canal cuando hay competencia por él. Cuando únicamente hay un canal, la determinación de quién tiene el turno es mucho más complicada. Se conocen muchos protocolos para resolver el problema. Los canales de difusión a veces se denominan canales multiacceso o canales de acceso aleatorio.

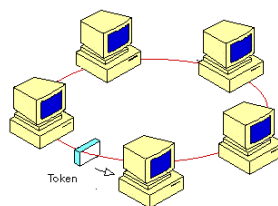
Los protocolos usados para determinar quién sigue en un canal multiacceso pertenecen a una subcapa de la capa de enlace de datos llamada subcapa de MAC (medium access control, control de acceso al medio). La subcapa MAC tiene especial importancia en las LAN, casi todas las cuales usan un canal multiacceso como base de su comunicación; con lo cual podemos determinar que los canales multiaccesos y las LAN están estrechamente relacionadas.

El IEEE ha producido varios estándares para las LAN. Estos estándares, conocidos en conjunto como IEEE 802, incluyen CSMA/CD, Token Bus y Token Ring. Los diferentes estándares difieren en la capa física y en la subcapa MAC, pero son compatibles en la capa de enlace de datos. Las partes 802.3 a 802.5 describen los tres estándares para LAN, CSMA/CD, Token Bus y Token Ring, respectivamente.

1.3.1 CSMA/CD. El CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*, *Acceso Múltiple con Detección de Portadora y Detección de Colisiones*), se usa ampliamente en redes LAN en la subcapa MAC. En éste cuando una estación quiere transmitir, escucha el cable. Si el cable está ocupado la estación espera hasta que se desocupa; de otra manera, transmite de inmediato. Si dos o más estaciones comienzan a transmitir simultáneamente por un cable inactivo, abra una colisión. Todas las estaciones en colisión terminan entonces su transición, esperan un tiempo aleatorio y repiten de nuevo todo el proceso.

1.3.2 Token Bus. Físicamente, el Token Bus (*Bus de Ficha*) es un cable lineal o en forma de árbol al que se conectan las estaciones. Las estaciones están organizadas lógicamente en forma de anillo donde cada estación conoce la dirección de la estación a su izquierda y a su derecha. Cuando se inicia el anillo lógico, la estación de número mas alto puede enviar el primer marco. Hecho esto, pasa el permiso a su vecino inmediato enviándole un marco de control especial llamado Token (*ficha*). La ficha se propaga alrededor del anillo lógico, teniendo permiso de transmitir marcos solo quien tiene la ficha. Dado que solo una estación a la vez tiene la ficha, no ocurren colisiones.

1.3.3 Token Ring. En un Token Ring (anillo con ficha), circula un patrón de bit especial, llamado Token (ficha) alrededor del anillo cuando todas las estaciones están inactivas (físicamente es una red en forma de anillo). Cuando una estación quiere transmitir un marco, debe tomar la ficha y retirarla del anillo antes de transmitir. Esta acción se lleva a cabo invirtiendo un solo bit de la ficha de 3 bytes, lo que instantáneamente lo convierte en los tres primeros bytes de un marco de 4 datos normal. Debido a que solo hay una ficha solo una estación



puede transmitir en un instante dado.

1.4 MODELO DE COMUNICACIÓN CLIENTE SERVIDOR

Es el concepto de interacción más común entre aplicaciones en una red que maneje comunicación entre procesos. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este concepto.

Un **Servidor** es un proceso que ofrece un servicio, es decir, que realiza una función específica. En ciertas circunstancias este término designará a una máquina; éste será el caso si dicha máquina está dedicada a un servicio particular.

Un **cliente** es un proceso que dirige una petición específica a un servidor que responde a una demanda de servicio.

2. EL PROBLEMA Y LOS OBJETIVOS DE LA INVESTIGACION

2.1 ANTECEDENTES DEL PROBLEMA

Dentro del campo de las investigaciones realizadas referentes al entorno que enmarca el tratamiento de las comunicaciones, los protocolos de acceso al medio juegan un papel importante dentro de la transmisión de información a través de una red, constituyéndose en un tema fundamental en el área de Comunicaciones y Redes.

Este aspecto dentro de las comunicaciones, a pesar de ser tan importante no ha contado con un campo de investigación dentro de la C.U.T.B. que desarrolle proyectos que permitan conocer el funcionamiento de los protocolos de Acceso al medio, dándole una visión real de cómo es el proceso de compartir un canal.

En la formación académica del estudiante de la Corporación Universitaria Tecnológica de Bolívar hace falta herramientas prácticas que apoyen la teoría vista dentro del área de Comunicaciones y Redes.

2.2 PROBLEMA

Desarrollar e implementar una herramienta de simulación de protocolos de acceso al medio en topologías de red de área local.

2.3 DESCRIPCION DEL PROBLEMA

En este proyecto se lleva a cabo el desarrollo e implementación de una herramienta basada en la simulación de protocolos de acceso al medio: CSMA/CD, Token Ring, Token Bus, sobre redes de área local en las topologías: Anillo, Bus y Estrella.

Dentro de éste se permitirá a los estudiantes interactuar en una red de computadores en la que podrá observar mediante una simulación características y aspectos de cómo cada protocolo maneja el acceso en la red.

La herramienta contemplará características tales como: un coordinar del sistema; el cual especifica la topología de la red y el protocolo, manejará un solo tipo de red a la vez, todos los usuarios serán informados de cuando entró y/o salió un usuario de la red y se proporcionarán ayudas .

2.4 JUSTIFICACIÓN

Teniendo en cuenta la importancia y evolución de las redes hoy en día, existe la necesidad de estudiar temas relacionados con los avances de su tecnología; dentro de los cuales encontramos conceptos que no brindan la posibilidad de observar su funcionamiento, tal es el caso de “Los Protocolos de acceso al medio”.

Este proyecto se basa en los conocimientos vistos en las áreas de Comunicaciones y Redes y Protocolos de Comunicación, las cuales nos permitirán tener las bases para desarrollar una herramienta de simulación que nos muestre y nos sirva de soporte sobre el papel de los protocolos en la administración y compartición de un canal dentro de una comunicación.

También es de destacarse la importancia que la elaboración de este proyecto representa para sus autores, ya que constituye una puesta en práctica de muchos de los conceptos y conocimientos adquiridos durante la carrera. Además de fortalecerse como base para ser utilizada y ampliada por generaciones venideras.

2.5 OBJETIVOS

2.5.1 Objetivo General. Desarrollar una herramienta de simulación de protocolos de acceso al medio en topología de red de área local, que brinde al estudiante la posibilidad de contar con una herramienta de software en una red de computadoras de la universidad, en la cual puedan realizar prácticas en las que observarán los pasos o reglas en que se fundamentan los protocolos para asignar un recurso.

2.5.2 Objetivos específicos.

- ◆ Proporcionar a la CUTB y en particular a la Facultad de Sistemas una herramienta que afiance de manera práctica el desarrollo teórico dado en las áreas de comunicaciones y redes.
- ◆ Este proyecto se constituye en una base fundamental para ser utilizado y ampliado por generaciones venideras, motivándolas a trabajar en proyectos de investigación que aumenten el nivel académico de la universidad.
- ◆ Brindar a los estudiantes la oportunidad de tener a mano una herramienta de apoyo para reforzar sus conocimientos teóricos, mediante la simulación de los protocolos de acceso al medio, implementada en una pequeña red de computadores.
- ◆ Diseñar e implementar una herramienta basada en la simulación de protocolos de acceso al medio: CSMA/CD, Token Ring, Token Bus, sobre redes de área local en las topologías: Anillo, Bus y Estrella.

- ◆ Permitir a los estudiantes interactuar entre sí en una red de computadores en la que podrá observar mediante diferentes opciones: reportes, características y aspectos de cómo se lleva a cabo el acceso en la red (características propias del funcionamiento de cada protocolo), además del tipo de red seleccionado, número de máquinas en la red, el nombre del protocolo, entre otros.

- ◆ Desarrollar una herramienta amigable e interactiva que abarque y describa el funcionamiento de los protocolos de acceso al medio en red de área local.

- ◆ La herramienta contemplará las siguientes características:
 - ◆ Habrá un coordinador del sistema, el cual especifica la topología de la red y el protocolo.
 - ◆ Existirá un solo tipo de red a la vez.
 - ◆ Todo usuario puede saber cuando entró y/o salió un usuario de la red.
 - ◆ La herramienta proporciona al usuario ayudas.
 - ◆ El coordinador y los usuarios deben conocer el número de máquinas conectadas en la red.

2.6 RESULTADOS DE LA INVESTIGACIÓN

Al culminar nuestro trabajo de grado, se entregará a la Corporación Universitaria Tecnológica de Bolívar, una herramienta en la cual se simula el funcionamiento de algunos Protocolos de Acceso al medio escogidos, dentro de las topologías de área local seleccionadas, por medio de la cual tanto docentes como estudiantes facilitarán la enseñanza y aprendizaje en las áreas de Comunicaciones y Redes. Junto con ésta herramienta de apoyo se entregará:

- ❖ Documento final del proyecto.
- ❖ Manual de usuario.
- ❖ Manual de operación e instalación.
- ❖ Librerías y software debidamente documentados.

3 ESTRATEGIA OPERATIVA, ADMINISTRACIÓN Y CONTROL DE LA INVESTIGACIÓN

3.1 ESPECIFICACIÓN OPERACIONAL DE ACTIVIDADES

La ejecución de nuestro trabajo de grado, implica la realización de una serie de actividades y/o pasos que tienen como propósito fundamental facilitar o posibilitar la consecución de los objetivos propuestos. Tales tareas son:

- ◆ **Etapa 1.** Recolección de información este paso implica documentarse sobre todos los posibles temas y/o recursos a utilizar en el desarrollo del proyecto.

- ◆ **Etapa 2.** Organización, análisis y estudio de los aspectos y conocimientos fundamentales de la información obtenida.

- ◆ **Etapa 3.** Realización de prácticas dentro de una red para familiarizarnos con la herramienta a trabajar.

- ◆ **Etapa 4.** Diseño de la estructura procedimental, en esta etapa se procede a especificar y hallar cada uno de los pormenores adjuntos que tienen estas actividades.

- ◆ **Etapa 5.** Implementación de la herramienta, aquí se desarrolla el software y todas sus librerías debidamente documentas.

- ◆ **Etapa 6.** Revisión y pruebas, en esta etapa se llevan a cabo todas pruebas necesarias para verificar si se han cumplido los objetivos establecidos.

- ◆ **Etapa 7.** Elaboración y entrega del documento final, en esta se elaborará y entregará el documento final del proyecto, en el que se plasmará de manera escrita y detallada toda la información referente a los Protocolos de Acceso al Medio.

3.2 CRONOGRAMA DE ACTIVIDADES

Aparece anexo

4. ASPECTOS DE INFRAESTRUCTURA

4.1 RECURSOS HUMANOS

Director: CARLOS VALENCIA

Ingeniero Electricista

Además se contará con el apoyo de los profesores de la CUTB con conocimientos en el área de Comunicaciones.

.....

4.2 DESCRIPCIÓN DE EQUIPOS Y/O LABORATORIOS REQUERIDOS

- ◆ Una red de computadores
- ◆ Software requerido.
- ◆ Cuenta de acceso a Internet a través de la CUTB con la cual se facilite el proceso de investigación.

BIBLIOGRAFIA

- <http://www-mat.upc.es/~alcober/PAPERS/Alc97/slides/index.htm>
- <http://www.inmesoft.com/ForRedes.htm>
- <http://www.serinfo.es/guiacx.htm>
- http://www.op.upm.es/tesmes/infraestructuras/doc/c/cred_1.html
- <http://www.geocities.com/SiliconValley/Heights/6497/ejemplos.html>
- <http://www.geocities.com/SiliconValley/Heights/6497/arquitectura.html>
- <http://www.disc.ua.es/asignaturas/rc/trabajos/lan3/rc.html>
- <http://sunsite.fisicc-ufm.edu/~obonilla/cn/l20/node9.html>
- <http://sunsite.fisicc-ufm.edu/~obonilla/cn/l20/node8.html>
- <http://sunsite.fisicc-ufm.edu/~obonilla/cn/l20/l20.html>
- <http://www.disc.ua.es/asignaturas/rc/tema1.html>
- <http://www.fisio.cinvestav.mx/fisio/index-lan.html>
- <http://arraquis.dif.um.es/~rafa/8trab.htm>
- <http://vgg.sci.uma.es/redes/topo.html>
- <http://ftp.rhon.itam.mx/~extuni/sat/comp/redes/lantech.html>
- <http://vgg.sci.uma.es/redes/bajos.html>
- <http://vgg.sci.uma.es/redes/index.html>
- <http://dsi.cem.itesm.mx/~ltrejo/REDES/Lan/node39.html>

- <http://ftp.rhon.itam.mx/~extuni/sat/comp/redes/conceptos.html>
- <http://www.facyt.uc.edu.ve/~rrodrigu/redes/clase4.html>
- TANENBAUM, Andrew S. Redes de Computadora. Editorial Prentice Hall. Tercera edición. México, 1997.
- CARBALLAR, José A. El libro de las comunicaciones del PC, técnica programación y aplicaciones, Editorial Rama. Edición original. Madrid – España, 1996.
- BLACK, Uyles. Redes de Computadores, Protocolos, Normas e Interfaces. Editorial Rama, Segunda edición. Madrid – España, 1995.
- GONZALEZ SAINZ, Nestor. Protocolos de Comunicaciones para Sistemas Abiertos, Editorial Addison-Wesley Iberoamericana, Primera edición. Argentina, 1996

ANEXO B MANUAL DEL SISTEMA

En el siguiente documento quedara expuesto la mayor parte del código fuente del software *Simulador de Protocolos de Acceso al Medio*, así como también el conjunto de funciones de las cuales él consta.

El software esta diseñado en la poderosa herramienta de programación Visual Basic 5.0, la cual se utiliza para realizar cada una de las simulacines que posee el programa.

El diseño del programa esta compuesto por un conjunto de funciones locales y globales al proyecto las cuales quedaran expuestas por simulaciones de protocolo.

Se recomienda que las personas que deseen leer este documento tengan al menos un poco de conocimiento acerca de la sintaxis de programación de Visual Basic 5.0.

VARIABLES GLOBALES

Variable	Definición
NServidor	Variable de tipo cadena que sirve para guardar el nombre del equipo servidor.
Nusuario	Variable de tipo cadena que sirve para guardar el nombre del equipo cliente
IntMax	Variable de tipo entero que sirve para guardar el numero de usuarios conectados.
EquiposActivos(8)	Vector de 8 posiciones de tipo byte que sirve para guardar el nombre de los equipos activos.
AnilloLogico	Matriz de tipo variant que contiene las posiciones iniciales de las lineas que dibujan el anillo logico.
Xi, Yi	Variables de tipo variant, guardan las posiciones del objeto info.
Sig	Variable de tipo variant que sirve para guardar el numero del equipo siguiente al que tiene el token.
Tkn	Variable de tipo variant que sirve para guardar el numero del equipo que tiene el token.
OpcionTr	Variable de tipo byte que sirve para guardar el numero de la simulación de tokenring que se va a realizar.
OpcionCs	Variable de tipo byte que sirve para guardar el numero de la simulación de CSMA / CD que se va a realizar.
Paso	Variable de tipo boleano que sirve para saber si se ha pasado por la función MoverEnBus.
Tiempo	Variable de tipo variant que sirve para guardar el tiempo en segundos que pasa el token circulando en el protocolo tokenring.

MODULO “Modulo”	
Este es un modulo global en el cual se tienen funciones que son de uso general, es decir, son funciones que se pueden utilizar en todos los formularios del proyecto(Funciones Globales). Las funciones que contiene este modulo son:	
Función	Descripción
CambiaToken	<p>Esta es la función que cambia el icono con el que aparecen los equipos activos por el icono PC con token. En síntesis, cambia el token al equipo especificado.</p> <p>Parámetros: Token(Equipo en donde quedará el token), T1(Equipo que tiene el token actualmente).</p>
ActualizaAnillo	<p>Esta es la función encargada de actualizar el anillo lógico de la red, para el protocolo token bus son topologías bus y estrella.</p> <p>Parámetros: Token(Equipo que tiene el Token), Equipo(Equipo que será insertado, eliminado de la red).</p>
Iniciar	<p>Esta es la función encargada de iniciar la red con los parámetros especificados. Por ejemplo, si se seleccionan en el formulario de datos que la red tendrá 4 equipos activos, la función iniciar será la encargada de activar dichos</p>

	equipos.
CargarAnillo	Esta es la función encargada de colocar los valores iniciales de las líneas utilizadas para formar el anillo lógico de la red. Esta sólo es utilizada en las simulaciones de Tokenbus con topologías estrella y bus.
BuscaSiguiete	Esta función es la encargada de buscar quién es el equipo consecutivo al que tiene el token actualmente. Tiene como parámetros Token(Equipo que posee el token),E (vector en donde están los equipos activos en la red), N (número de equipos en la red).
MoverInfo	Esta es la función encargada del movimiento de la información en la topología de bus. Recibe como parámetros el equipo origen y el equipo destino.
MoverInfo1	Esta es la función encargada del movimiento de la información en la topología estrella. Recibe como

	parámetros el equipo origen y el equipo destino.
Pausa	Es la función que detiene la ejecución del programa un intervalo de tiempo. Recibe como parámetro los segundos que desea detener la ejecución.
QuienToken	Esta es la función encargada de determinar quien tiene el token.
Código de las funciones del Modulo	

Sub CargarAnillo()

AnilloLogico(0, 0) = 405 'x

AnilloLogico(0, 1) = 855 'y

AnilloLogico(1, 0) = 900 'x

AnilloLogico(1, 1) = 585 'y

AnilloLogico(2, 0) = 1665 'x

AnilloLogico(2, 1) = 585 'y

AnilloLogico(3, 0) = 2250 'x

AnilloLogico(3, 1) = 855 'y

AnilloLogico(4, 0) = 2250 'x

AnilloLogico(4, 1) = 1350 'y

AnilloLogico(5, 0) = 1665 'x

AnilloLogico(5, 1) = 1530 'y

AnilloLogico(6, 0) = 990 'x

```
Sub Pausa(Numsegundos)
Anterior = Timer
Do
    Actual = Timer
Loop Until (Anterior + Numsegundos <= Actual)
End Sub
```

```
Sub MoverInfo(desde As Variant, hasta As Variant, desde1 As  
Shape, hasta1 As Shape, Mov As Byte)
```

```
    desde1.Visible = True
```

```
    Xi = desde1.Left
```

```
    Yi = desde1.top
```

```
    If (Mov >= 1) Then
```

```
        Do
```

```
            Key = True
```

```
            If (desde > 3) And (desde1.top - 2160 > 50) Then
```

```
                Desde1.top = desde1.top - 50 '20
```

```
                Key = False
```

```
            ElseIf (desde1.top < 2130) And (Abs(desde1.top - 2160) >  
50) Then
```

```
                Desde1.top = desde1.top + 50 '20
```

```
                Key = False
```

```
            End If
```

```
        Call Pausa(0.08)
```

```
    Loop Until (Key = True)
```

```
desde1.top = 2115
desde1.Left = Xi
End If
If (Mov >= 3) Then
Do
    Key = True
    If (desde1.Left < hasta1.Left) Then
        Desde1.Left = desde1.Left + 100
        Key = False
    ElseIf (desde1.Left - hasta1.Left > 100) Then
        Desde1.Left = desde1.Left - 100
        Key = False
    End If
    Call Pausa(0.08)
Loop Until (Key = True)
desde1.Left = hasta1.Left
End If
If (Mov = 3) Then
Do
    Key = True
```

```
Function QuienToken(ByRef Siguiete As Byte) As Byte
Mayor = 0
Menor = 7
For i = 0 To 7 Step 1
    If (Mayor < EquiposActivos(i)) Then
        Mayor = EquiposActivos(i)
        li = i
    End If
    If (Menor > EquiposActivos(i)) And (EquiposActivos(i) >=
0) Then
        Menor = EquiposActivos(i)
    End If
Next i
Siguiete = Menor
QuienToken = Mayor
End Function
```

```
Function BuscaSiguiente(Token, E(), N As Variant) As Variant
```

```
'ordenando el vector
```

```
N = 0
```

```
For i = 0 To 7 Step 1
```

```
  For j = i + 1 To 7 Step 1
```

```
    If (E(j) < E(i)) Then
```

```
      Aux = E(i)
```

```
      E(i) = E(j)
```

```
      E(j) = Aux
```

```
    End If
```

```
  Next j
```

```
  If (E(i) < 0) Then
```

```
    N = N + 1
```

```
  End If
```

```
Next i
```

```
For i = 0 To 7 Step 1
```

```
  If (Token = E(i) And (E(i) >= 0)) Then
```

```
    If (i < 7) Then
```

```

Sub MoverInfo1(desde, hasta, desde1 As Shape, hasta1 As Shape, Mov As Byte)
desde1.Visible = True
Xi = desde1.top
Yi = desde1.Left
'movimiento 1
'*****
If (Mov >= 1) Then
  C = 0
  Do
    If (desde = 0) Or (desde = 7) Then
      Desde1.Left = desde1.Left + 50
      C = C + 50
    ElseIf (desde = 1) Or (desde = 2) Then
      Desde1.top = desde1.top + 50
      C = C + 50
    ElseIf (desde = 3) Or (desde = 4) Then
      Desde1.Left = desde1.Left - 50
      C = C + 50
    Else
      Desde1.top = desde1.top - 50
      C = C + 50
    End If
    Call Pausa(0.1)
  Loop Until (C >= 810)
End If
'movimiento 2
If (Mov = 2) Then
  If (hasta = 1) Or (hasta = 2) Then
    desde1.top = hasta1.top + 810
    desde1.Left = hasta1.Left
  ElseIf (hasta = 3) Or (hasta = 4) Then
    desde1.top = hasta1.top
    desde1.Left = hasta1.Left - 810
  ElseIf (hasta = 5) Or (hasta = 6) Then

```

```
desde1.top = hasta1.top - 810
desde1.Left = hasta1.Left
Else
desde1.top = hasta1.top
desde1.Left = hasta1.Left + 810
End If
C = 0
Do
If (hasta = 0) Or (hasta = 7) Then
Desde1.Left = desde1.Left - 50
C = C + 50
ElseIf (hasta = 1) Or (hasta = 2) Then
Desde1.top = desde1.top - 50
C = C + 50
ElseIf (hasta = 3) Or (hasta = 4) Then
Desde1.Left = desde1.Left + 50
C = C + 50
Else
Desde1.top = desde1.top + 50
C = C + 50
End If
Call Pausa(0.1)
Loop Until (C >= 810)
End If
desde1.top = Xi
desde1.Left = Yi
desde1.Visible = False
End Sub
```

OBJETO TBESTRELLA	
Este objeto es el formulario encargado de la simulación Inicialización e Inserción de equipos en la red utilizando el protocolo TokenBus con Topología Estrella. En este, se pueden distinguir las siguientes funciones:	
Función	Descripción
Insertar_Click	Esta es la función encargada de insertar los nuevos equipos que se seleccionaron, después de haber hecho click en el botón Insertar. En esta, se hacen llamados a las funciones ActualizaAnillo, MoverInfo1 y pausa.
Timer1_Timer	Esta es la función que realiza la inicialización de la red, una vez presionado el botón Inicializar. Esta función es la que controla el objeto Timer, que es un controlador de tiempo que se ejecuta cada cierto intervalo.
Código de las funciones del Objeto TBESTRELLA	

```
Private Sub insertar_Click()  
Dim E(8)  
Sw = False  
NumEquipos = 0  
For i = 0 To 7 Step 1  
    If (Equipos(i).Enabled = True) And (Equipos(i).Value = 1)  
Then  
        Sw = True  
        NumEquipos = NumEquipos + 1  
    End If  
Next i  
If (Sw = False) Then  
    v = MsgBox("Debe seleccionar los equipos a insertar",  
vbCritical, "Error...")  
    Exit Sub  
End If  
Token = Tkn  
If (Tkn < 0) Then  
    Token = QuienToken(CByte(Siguiente))  
End If  
StatusBar1.Panels(1) = "Activo"
```



```
If (Equipos(i).Value = 1) And (Equipos(i).Enabled = True) Then

    StatusBar1.Panels(2).Text = "Insertando Equipo " & i + 1 & " a la red. - Enviando
Solicit-sucesor"

    Trama(2).Caption = "3: 00000001"

Call CambiaToken(Token, CByte(i))

    Color1 = Info(Token).FillColor

    Info(Token).FillColor = &HFF00FF

    Call MoverInfo1(Token, i, Info(Token), Info(i), 2)

    Info(Token).FillColor = Color1

    Color1 = Info(i).FillColor

    Info(i).FillColor = &HFF00&

    StatusBar1.Panels(2).Text = "Insertando Equipo " & i + 1 & " a la red. - Enviando Set-
Sucesor"

    Call MoverInfo1(i, Token, Info(i), Info(Token), 2)

    Info(i).FillColor = Color1

    Info(Token).FillColor = &HFFFF&

    Info(i).FillColor = &HFFFF&

    Trama(2).Caption = "3: 00001000"

    Frame1(4).Refresh

    StatusBar1.Panels(2).Text = "Insertando Equipo " & i + 1 & " a la red. - Transfiriendo
el Token a equipo " & i + 1

    Call MoverInfo1(Token, i, Info(Token), Info(i), 2)

    Info(Token).Visible = False

Color1 = Info(i).FillColor

    Info(i).FillColor = &HFF&
```

```
'***
```

```
Call CambiaToken(i, CByte(Token))
```

```
Info(Token).Visible = False
```

```
Call ActualizaAnillo(i, CByte(Token))
```

```
'***
```

```
NumEquipos = NumEquipos - 1
```

```
Info(Token).FillColor = Color1
```

```
Info(i).FillColor = Color1
```

```
Info(Token).Visible = False
```

```
Info(i).Visible = True
```

```
Equipos(i).Value = 0
```

```
Equipos(i).Enabled = False
```

```
Frame1(0).Refresh
```

```
Frame1(3).Refresh
```

```
Token = i
```

```
ElseIf (Equipos(i).Enabled = False) Then
```

```
Info(Token).FillColor = &HFFFF&
```

```
Info(i).FillColor = &HFFFF&
```

```
StatusBar1.Panels(2).Text = "Cambiano Token al Equipo "
```

```
Private Sub Timer1_Timer()  
For i = 0 To 7 Step 1  
If (EquiposActivos(i) >= 0) Then  
If (i = 0) Or (i = 7) Then  
    Info(i).Left = Info(i).Left + 50  
    k = i  
ElseIf (i = 3) Or (i = 4) Then  
    Info(i).Left = Info(i).Left - 50  
    k = i
```

```
ElseIf (i = 1) Or (i = 2) Then
    Info(i).top = Info(i).top + 50
    k = i
Else
Info(i).top = Info(i).top - 50
    k = i
End If
End If
Next i
C = C + 50
Sw = False
If (C >= 810) Then
    Sw = True
End If
If (Sw = True) Then
    Sw = False
    C = 0
    Do
        For i = 0 To 7 Step 1
            If (EquiposActivos(i) >= 0) Then
                If (i = 0) Or (i = 7) Then
                    Info(i).Left = Info(i).Left - 50
                    k = i
                ElseIf (i = 3) Or (i = 4) Then
```

```
Set Image1(Mayor).Picture = LoadPicture("server.ico")
Set Pc(Mayor + 1).Picture = LoadPicture("server.ico")
Image1(Mayor).ToolTipText = "Equipo con Token"
Info(Mayor).Visible = True
Timer1.Enabled = False
Inicializa.Enabled = False
Iniciar1.Enabled = False
insertar.Enabled = True
Insert.Enabled = True
StatusBar1.Panels(1).Text = "Inactivo"
StatusBar1.Panels(2).Text = "Red Inicializada... - Equipo " &
Mayor + 1 & " con Token"
Frame1(0).Refresh
Frame1(3).Refresh
Call Pausa(1)
' StatusBar1.Panels(2).Text = ""
End If
End Sub
```


OBJETO TBESTRELLA1	
Este objeto es el formulario encargado de la simulación de Salida de equipos de la red con protocolo TokenBus y Topología Estrella. Se pueden distinguir las siguientes funciones:	
Función	Descripción
Sacar_Click	Esta es la función encargada de eliminar de la red los equipos seleccionados, después de haber hecho click en el botón Sacar. En esta, se hacen llamados a las funciones ActualizaAnillo, MoverInfo1 , pausa, entre otras.
Código de Sacar_Click	

```
Private Sub Sacar_Click()  
  
Dim Siguiente As Byte  
  
Dim E(8)  
  
StatusBar1.Panels(1).Text = "Activo"  
  
StatusBar1.Panels(2).Text = "Sacando equipos..."  
  
NumEquipos = 0  
  
N = 0  
  
For i = 0 To 7 Step 1  
  
    E(i) = EquiposActivos(i)  
  
    If (Equipos(i).Value = 1) And (Equipos(i).Enabled = True) Then  
  
        NumEquipos = NumEquipos + 1  
  
        N = N + 1  
  
    ElseIf (Equipos(i).Enabled = True) Then  
  
        N = N + 1  
  
    End If  
  
Next i  
  
If (NumEquipos <= 0) Then  
  
    v = MsgBox("Debe seleccionar los equipos a sacar", vbCritical, "Error...")  
  
    Exit Sub  
  
End If  
  
If (Tkn < 0) Then  
  
    Token = QuienToken(CByte(Siguiente))  
  
Else
```

```
UToken = E(6)
If (UToken < 0) Then
    UToken = Token
End If
i = Token
Do
    Siguiente = BuscaSiguiente(Token, E, N)
    If (Equipos(i).Value = 1) And (Equipos(i).Enabled = True)
    Then
        StatusBar1.Panels(2).Text = "Sacando el equipo " & Token +
        1 & " de la red."
        Color1 = Info(Token).FillColor
        Info(Token).FillColor = &HFF00&
        StatusBar1.Panels(2).Text = "Sacando el equipo " & Token +
        1 & " de la red. - Enviando Set_Sucesor a equipo " & UToken +
        1
        Call MoverInfo1(Token, UToken, Info(Token),
        Info(UToken), 2)
        Info(Token).FillColor = Color1
        StatusBar1.Panels(2).Text = "Transfiriendo Token a Equipo
        " & Siguiente + 1
        Call MoverInfo1(Token, Siguiente, Info(Token),
        Info(Siguiente), 2)
```

```
Equipos(Token).Enabled = False

Equipos(Token).Value = 0

NumEquipos = NumEquipos - 1

'UToken = Token

i = Token

Token = Siguiete

ElseIf (Equipos(i).Enabled = True) And (Equipos(i).Value = 0) Then

    StatusBar1.Panels(2).Text = "Transfiriendo Token a Equipo
" & Siguiete + 1

    Call MoverInfo1(Token, Siguiete, Info(Token),
Info(Siguiete), 2)

    'enviamos el Ack

    Color1 = Info(Siguiete).FillColor

    StatusBar1.Panels(2).Text = "Enviando ACK a Equipo " &
Token + 1

    Info(Siguiete).FillColor = &HFF&

    Call MoverInfo1(Siguiete, Token, Info(Siguiete),
Info(Token), 2)

    Info(Siguiete).FillColor = Color1

    '*****

    Call CambiaToken(Siguiete, CByte(Token))
```


OBJETO TBESTRELLA2	
Este objeto es el formulario encargado de la simulación Recuperación del Token Caso 1, utilizando el protocolo TokenBus con Topología Estrella. Se pueden resaltar las siguientes funciones:	
Función	Descripción
Recuperar	<p>En esta función es donde realmente se realiza la simulación. Tiene como parámetros el Token (Equipo donde está el Token), Siguiente (Equipo siguiente al que tiene el Token), E (vector donde están los equipos a participar en la simulación), NumEquipos (Número de equipos activos en la red).</p> <p>Esta función, utiliza las funciones BuscaSiguiente, MoverInfo1, CambiaToken, entre otras.</p>
EnviarInfoTodos	Esta es la función encargada de enviar información del HUB a cada uno de los equipos activos en la red.
Código de las funciones principales del Objeto TBEstrella2	

```
Sub Recuperar(Token, Siguiente, E(), NumEquipos)
Dim Eq(8)
StatusBar1.Panels(1).Text = "Activo"
If (Tkn >= 0) Then
    Token = Tkn
End If
CFallo = 0
N = 0
For i = 0 To 7 Step 1
    Eq(i) = EquiposActivos(i)
    If (Eq(i) >= 0) Then
        N = N + 1
    End If
    If (E(i) = -2) Then
        CFallo = CFallo + 1
    End If
Next i
Siguiente = BuscaSiguiente(Token, Eq, N)
If (CFallo <= 0) Then
    v = MsgBox("Para esta simulación se necesita" & Chr(13) &
```

```
Do
  If (E(Siguiente) = -2) Then
    C = 1
    Do
      If (C < 3) Then
        StatusBar1.Panels(2).Text = "Enviando Token a equipo "
        & Siguiente + 1 & " - Número de veces = " & C
        StatusBar1.Refresh
        Call MoverInfo1(Token, Siguiente, Info(Token),
        Info(Siguiente), 2)
      End If
      C = C + 1
    Loop Until (C > 2)
    Us = Siguiente
    Siguiente = BuscaSiguiente(Siguiente, Eq, NumEquipos)
  StatusBar1.Panels(2).Text = "Enviando Who-Follow"
  Color1 = Info(Token).FillColor
  Info(Token).FillColor = &HFF00FF
  If (Token <> Siguiente) Then
    Call MoverInfo1(Token, Siguiente, Info(Token),
```

```
If (cont >= 1) Then
    s = False
    ii = Token
    Do
        valor = ii
        If (Eq(valor) >= 0) And _
            (Equipos(valor).Value = 0) And _
                (valor <> Token) Then
            s = True
        Else
            If (ii <= 6) Then
                ii = ii + 1
            Else
                ii = 0
            End If
            If (ii = Token) Then
                s = True
                valor = Token
            End If
        End If
    End If
```

```
'* set sucesor*

If (Siguiete <> Token) Then

    StatusBar1.Panels(2).Text = "Enviando Set-Sucesor"

    Info(Siguiete).FillColor = &HFF00&

    Call MoverInfo1(Siguiete, Token, Info(Siguiete),
Info(Token), 2)

    Info(Siguiete).FillColor = Color1

    StatusBar1.Panels(2).Text = "Transfiriendo Token al
equipo " & Siguiete + 1

    Color1 = Info(Token).FillColor

    Call MoverInfo1(Token, Siguiete, Info(Token),
Info(Siguiete), 2)

    'enviamos el Ack

    Info(Siguiete).FillColor = &HFF&

    StatusBar1.Panels(2).Text = "Enviando ACK al equipo " &
Token + 1

    Call MoverInfo1(Siguiete, Token, Info(Siguiete),
Info(Token), 2)

    Info(Siguiete).FillColor = Color1

    Call CambiaToken(Siguiete, CByte(Token))

End If

Call ActualizaAnillo(Siguiete, CByte(Token))
```

```
Tkn = Token
For i = 0 To 7 Step 1
    Equipos(i).Value = 0
    If (EquiposActivos(i) >= 0) And (i <> Token) Then
        Equipos(i).Enabled = True
    Else
        Equipos(i).Enabled = False
    End If
Next i
StatusBar1.Panels(1).Text = "Inactivo"
StatusBar1.Panels(2).Text = ""
End Sub
```

```
Sub EnviarInfoTodos()  
For i = 0 To 7 Step 1  
    Color1 = Info(i).FillColor  
    Info(i).FillColor = &HFFFFFF  
Next i  
For i = 0 To 7 Step 1  
    If (EquiposActivos(i) >= 0) Then  
        Info(i).Visible = True  
        If (i = 0) Or (i = 7) Then  
            Info(i).Left = 1620 + 810  
        ElseIf (i = 1) Or (i = 2) Then  
            Info(i).top = 900 + 810  
        ElseIf (i = 3) Or (i = 4) Then  
            Info(i).Left = 4815 - 810  
        Else  
            Info(i).top = 3285 - 810  
        End If  
    End If  
End For  
Next i  
C = 0
```


OBJETO TBESTRELLA3	
Este objeto es el formulario encargado de la simulación Recuperación del Token Caso 2, utilizando el protocolo TokenBus con Topología Estrella. Se pueden resaltar las siguientes funciones:	
Función	Descripción
Inicia_Click	En esta función, se inicia la simulación del segundo caso de recuperación del token. Utiliza las funciones Moverinfo1, CambiaToken, ActualizaAnillo, entre otras.
Código de la Función Inicia_Click	

```
Private Sub Inicia_Click()
Dim Eq(8) As Variant

Sw = False
C = 0
For i = 0 To 7 Step 1
  If (Equipos(i).Value = 1) Then
    Sw = True
    C = C + 1
  End If
Next i
If (Sw = False) Then
  v = MsgBox("Debe seleccionar" & Chr(13) & "los equipos en donde falla la
transmisión", vbCritical, "Error...")
  Exit Sub
ElseIf (C <> 2) Then
  v = MsgBox("Debe seleccionar 2 equipos...", vbCritical, "Error...")
  Exit Sub
End If

Time(0).Caption = 0
Time(0).Refresh
Time(1).Caption = 0
Time(1).Refresh

NumEquipos = 0
For i = 0 To 7 Step 1
  Eq(i) = EquiposActivos(i)
  If (EquiposActivos(i) >= 0) Then
    NumEquipos = NumEquipos + 1
  End If
Next i
```

```

StatusBar1.Panels(1).Text = "Activo"
tkn1 = QuienToken(CByte(Sig))
Tkn = tkn1
A = Timer
For i = 0 To NumEquipos - 1 Step 1
    Sig = BuscaSiguiente(tkn1, Eq, NumEquipos)
    StatusBar1.Panels(2).Text = "Calculando tiempo de regreso del token... - Transfiriendo
Token a equipo " & Sig + 1
    Call MoverInfo1(tkn1, Sig, Info(tkn1), Info(Sig), 2)
    Color1 = Info(Sig).FillColor
    Info(Sig).FillColor = &HFF&
    StatusBar1.Panels(2).Text = "Calculando tiempo de regreso del token... - Enviando ACK a
equipo " & tkn1 + 1
    Call MoverInfo1(Sig, tkn1, Info(Sig), Info(tkn1), 2)
    Info(Sig).FillColor = Color1
    Call CambiaToken(Sig, CByte(tkn1))
    Call ActualizaAnillo(Sig, CByte(tkn1))
    Time(0).Caption = Int(Timer - A)
    Frame1(0).Refresh
    Frame1(3).Refresh
    Time(0).Refresh
    Time(1).Refresh
    Beep
    tkn1 = Sig
Next i
Sw = False
Token = Tkn
A = Timer
Do
    Call Pausa(0.5)
    Time(1).Caption = Int(Timer - A)
    Time(1).Refresh
    If (CDec(Time(1).Caption) > CDec(Time(0).Caption)) Then
        StatusBar1.Panels(2).Text = "Tiempo de Respuesta Excedido"
        Beep
        Beep
        Call Pausa(1)
        Sw = True
StatusBar1.Panels(2).Text = "Iniciando la Red..."
    Call Inicializacion
    StatusBar1.Panels(2).Text = "Iniciando la Red..."
    Frame1(0).Refresh
    Frame1(3).Refresh
    Time(0).Caption = 0
    Time(1).Caption = 0
    Time(0).Refresh
    Time(1).Refresh
    Exit Sub
Else

```

```

Sig = BuscaSiguiente(Token, Eq, NumEquipos)
  If ((Equipos(Token).Value = 1) And (Equipos(Sig).Value = 1)) Then
    If (UToken <> Token) Then
      pos = Info(Token).top
      Pos1 = Info(Token).Left
      StatusBar1.Panels(2).Text = "Transfiriendo Token a equipo " & Sig + 1
      Call MoverInfo1(Token, Sig, Info(Token), Info(Sig), 1)
      StatusBar1.Panels(2).Text = "Transfiriendo Token a equipo " & Sig + 1 & " -
Esperando ACK"
      Info(Token).Visible = False
      Info(Token).top = pos
      Info(Token).Left = Pos1
      Info(Token).Refresh
      Frame1(0).Refresh
      Frame1(3).Refresh
    End If
    UToken = Token
  Else
    Color1 = Info(Sig).FillColor
    StatusBar1.Panels(2).Text = "Transfiriendo Token a equipo " & Sig + 1
    Call MoverInfo1(Token, Sig, Info(Token), Info(Sig), 2)
    Info(Sig).FillColor = &HFF&
    'Ack
    StatusBar1.Panels(2).Text = "Enviando ACK a equipo " & Token + 1
    Call MoverInfo1(Sig, Token, Info(Sig), Info(Token), 2)
    Info(Sig).FillColor = Color1
    Call CambiaToken(Sig, CByte(Token))
    StatusBar1.Panels(2).Text = "Token Transferido..."
    Token = Sig
  End If
End If
Loop Until (Sw = True)
End Sub

```

OBJETO TBESTRELLA4	
Este objeto es el formulario encargado de la simulación Transmisión Exitosa, utilizando el protocolo TokenBus con Topología Estrella. Se pueden resaltar las siguientes funciones:	
Función	Descripción
Inicia_Click	En esta función se realiza esta simulación. Utiliza las funciones MoverInfo1, CambiaToken, entre otras.
Código de la función Inicia_Click	
<pre> Private Sub Inicia_Click() Dim Origen, destino Dim E(8) If (EqDestino.Text = "") Then v = MsgBox("Debe seleccionar un destino...", vbCritical, "Error...") Exit Sub End If N = 0 For i = 0 To 7 Step 1 If (EquiposActivos(i) >= 0) Then N = N + 1 End If E(i) = EquiposActivos(i) Next i StatusBar1.Panels(1).Text = "Activo" Origen = CDec(Mid(EqOrigen.List(EqOrigen.ListIndex), 8, 1)) destino = CDec(Mid(EqDestino.List(EqDestino.ListIndex), 8, 1)) Trama(3).Caption = "4: PC " & destino Trama(4).Caption = "5: PC " & Origen Trama(5).Caption = "6: Mensaje" Trama(3).Refresh Trama(4).Refresh Trama(5).Refresh Sw = False If (Tkn >= 0) Then Token = Tkn Else Token = QuienToken(CByte(Sig)) End If </pre>	

```

Do
  If (Token = Origen - 1) Then
    StatusBar1.Panels(2).Text = "Enviando información de " & Origen & " a " &
destino
    Color1 = Info(Origen - 1).FillColor
    Info(Origen - 1).FillColor = &HC0C0FF
    Call MoverInfo1(Origen - 1, destino - 1, Info(Origen - 1), Info(destino - 1), 2)
    Info(Origen - 1).FillColor = &HC0C0FF
    Call MoverInfo1(Origen - 1, destino - 1, Info(Origen - 1), Info(destino - 1), 2)
    Info(Origen - 1).FillColor = Color1
    StatusBar1.Panels(2).Text = "Información enviada exitosamente..."
    Sw = True
  Else
    Sig = BuscaSiguiente(Token, E, N)
    'transfiriendo token de token a sig
    StatusBar1.Panels(2).Text = "Transfiriendo token a equipo " & Sig + 1
    Call MoverInfo1(Token, Sig, Info(Token), Info(Sig), 2)
    'enviamos el Ack
    Color1 = Info(Sig).FillColor
    StatusBar1.Panels(2).Text = "Enviando ACK a equipo " & Token + 1
    Info(Sig).FillColor = &HFF&
    Call MoverInfo1(Sig, Token, Info(Sig), Info(Token), 2)
    Info(Sig).FillColor = Color1
    *****
    Call ActualizaAnillo(Sig, CByte(Token))
    Call CambiaToken(Sig, CByte(Token))
    StatusBar1.Panels(2).Text = "Token Transferido"
    Frame1(0).Refresh
    Frame1(3).Refresh
    Call Pausa(0.5)
    Token = Sig
  End If
Loop Until (Sw = True)
Tkn = Token
StatusBar1.Panels(1).Text = "Inactivo"
StatusBar1.Panels(2).Text = ""
End Sub

```

OBJETO TOKENBUS1	
Este objeto es el formulario encargado de la simulación Inicialización e Inserción de equipos en la red utilizando el protocolo TokenBus con Topología bus. En este, se pueden distinguir las siguientes funciones:	
Función	Descripción
InsertarEquipo	Esta es la función encargada de insertar los nuevos equipos que se seleccionaron, después de haber hecho click en el botón Insertar. En esta, se hacen llamados a las funciones ActualizaAnillo, MoverInfo y pausa. Tiene como parámetro NumEquipos(Numero de equipos activos).
Timer1_Timer	Esta es la función que realiza la inicialización de la red, una vez presionado el botón Inicializar. Esta función es la que controla el objeto Timer, que es un controlador de tiempo que se ejecuta cada cierto intervalo.
Código de la funciones del Objeto TokenBus1	

```
Function InsertaEquipo(NumEquipos As Byte)

Dim Siguiete As Byte

StatusBar1.Panels(1).Text = "Activo"

StatusBar1.Panels(2).Text = "Insertando Equipos..."

If (Tkn < 0) Then

    Token = QuienToken(Siguiete) 'determina quien tiene el token

    Tkn = Token

End If

Token = Tkn

i = Token

Call Pausa(0.5)

Do

    If (i >= 7) Then

        i = 0

    Else

        i = i + 1

    End If

    If (Equipos(i).Value = 1) And (Equipos(i).Enabled = True) Then

        StatusBar1.Panels(2).Text = "Insertando Equipo " & i + 1 & " a la red. - Enviando Solicit-sucesor"

        Trama(2).Caption = "3: 00000001"

        Frame1(4).Refresh

    Call CambiaToken(Token, CByte(i))

    Color1 = Info(Token).FillColor

End Do
```

```
StatusBar1.Panels(2).Text = "Insertando Equipo " & i + 1 & " a
la red. - Enviando Set-Sucesor"
```

```
Color1 = Info(i).FillColor
```

```
Info(i).FillColor = &HFF00&
```

```
Call MoverInfo(i, Token, Info(i), Info(Token), 3)
```

```
Info(i).FillColor = Color1
```

```
'****
```

```
Trama(2).Caption = "3: 00001000"
```

```
Frame1(4).Refresh
```

```
StatusBar1.Panels(2).Text = "Insertando Equipo " & i + 1 & "
a la red. - Transfiriendo el Token a equipo " & i + 1
```

```
Call MoverInfo(Token, i, Info(Token), Info(i), 3)
```

```
'ack
```

```
StatusBar1.Panels(2).Text = "Insertando Equipo " & i + 1 & "
a la red. - Enviando ACK a equipo " & Token + 1
```

```
Color1 = Info(i).FillColor
```

```
Info(i).FillColor = &HFF&
```

```
Call MoverInfo(i, Token, Info(i), Info(Token), 3)
```

```
Info(i).FillColor = Color1
```

```
'****
```

```
Call CambiaToken(i, CByte(Token))
```

```
'ack

    StatusBar1.Panels(2).Text = "Enviando ACK a Equipo " &
Token + 1

    Color1 = Info(i).FillColor

    Info(i).FillColor = &HFF&

    Call MoverInfo(i, Token, Info(i), Info(Token), 3)

    Info(i).FillColor = Color1

'****

    Call CambiaToken(i, CByte(Token))

    Info(Token).Visible = False

    Info(i).Visible = True

    Frame1(0).Refresh

    Frame1(3).Refresh

    StatusBar1.Panels(2).Text = "Token Transferido..."

    Call Pausa(1)

    Token = i

End If

Loop Until (NumEquipos <= 0)

Tkn = Token

StatusBar1.Panels(1).Text = "Inactivo"
```

```
Private Sub Timer1_Timer()  
Dim Mayor As Variant  
  
If (Ejecutar.Enabled = False) And (Sw1 = True) Then  
    Sw = True  
    For i = 0 To 7 Step 1  
        If (EquiposActivos(i) >= 0) Then  
            Select Case EquiposActivos(i)  
                Case Is <= 3  
                    If (Info(EquiposActivos(i)).top < 2040) Then  
                        Info(EquiposActivos(i)).top =  
Info(EquiposActivos(i)).top + 25  
                        Sw = False  
                    End If  
                Case Is > 3  
                    If (Info(EquiposActivos(i)).top > 2205) Then  
                        Info(EquiposActivos(i)).top =  
Info(EquiposActivos(i)).top - 24  
                        Sw = False  
                    End If  
            End Select  
        End If  
    End For  
End Sub
```

```
Mayor = 0
If (Sw = True) Then
  For i = 0 To 7 Step 1
    If (i <= 3) Then
      Info(i).top = 855
    Else
      Info(i).top = 3330
    End If
    Info(i).Visible = False
    'buscamos PC mayor
    If (Mayor < EquiposActivos(i)) Then
      Mayor = EquiposActivos(i)
    End If
  Next i
  'identificamos el PC con Token
  Set Image1(Mayor).Picture = LoadPicture("Server.ico")
  Image1(Mayor).ToolTipText = "Equipo con Token"
  Set Pc(1 + Mayor).Picture = LoadPicture("Server.ico")
  Info(Mayor).Visible = True
  Beep
```


OBJETO TOKENBUS2	
Este objeto es el formulario encargado de la simulación de Salida de equipos de la red con protocolo TokenBus y Topología Bus. Se pueden distinguir las siguientes funciones:	
Función	Descripción
Sacar_Click	Esta es la función encargada de eliminar de la red los equipos seleccionados, después de haber hecho click en el botón Sacar. En esta, se hacen llamados a las funciones ActualizaAnillo, MoverInfo1 , pausa, entre otras.
Código de Sacar_Click	

```
Private Sub Sacar_Click()  
  
Dim Siguiente As Byte  
  
Dim E(8)  
  
StatusBar1.Panels(1).Text = "Activo"  
  
StatusBar1.Panels(2).Text = "Sacando equipos..."  
  
NumEquipos = 0  
  
N = 0  
  
For i = 0 To 7 Step 1  
  
    E(i) = EquiposActivos(i)  
  
    If (Equipos(i).Value = 1) And (Equipos(i).Enabled = True) Then  
  
        NumEquipos = NumEquipos + 1  
  
        N = N + 1  
  
    ElseIf (Equipos(i).Enabled = True) Then  
  
        N = N + 1  
  
    End If  
  
Next i  
  
If (NumEquipos <= 0) Then  
  
    v = MsgBox("Debe seleccionar los equipos a sacar", vbCritical, "Error...")  
  
    Exit Sub  
  
End If  
  
If (Tkn < 0) Then  
  
    Token = QuienToken(CByte(Siguiente))  
  
Else  
  
    Token = Tkn
```

Do

Siguiente = BuscaSiguiente(Token, E, N)

If (Equipos(i).Value = 1) And (Equipos(i).Enabled = True) Then

StatusBar1.Panels(2).Text = "Sacando el equipo " & Token + 1 & " de la red."

Color1 = Info(Token).FillColor

Info(Token).FillColor = &HFF00&

StatusBar1.Panels(2).Text = "Sacando el equipo " & Token + 1 & " de la red. -
Enviando Set_Sucesor a equipo " & UToken + 1

Call MoverInfo(Token, UToken, Info(Token), Info(UToken), 3)

Info(Token).FillColor = Color1

StatusBar1.Panels(2).Text = "Sacando el equipo " & Token + 1 & " de la red. -
Transfiriendo Token a Equipo " & Siguiente + 1

Call MoverInfo(Token, Siguiente, Info(Token), Info(Siguiente), 3)

Call CambiaToken(Siguiente, CByte(Token))

StatusBar1.Panels(2).Text = "Sacando el equipo " & Token + 1 & " de la red. -
Enviando ACK a Equipo " & Token + 1

Color1 = Info(Siguiente).FillColor

Info(Siguiente).FillColor = &HFF&

Call MoverInfo(Siguiente, Token, Info(Siguiente), Info(Token), 3)

Info(Siguiente).FillColor = Color1

Call ActualizaAnillo(Token, CByte(Siguiente))

StatusBar1.Panels(2).Text = "Equipo " & Token + 1 & " ha salido de la red..."

Frame1(0).Refresh

Frame1(3).Refresh

```
ElseIf (Equipos(i).Enabled = True) And (Equipos(i).Value = 0) Then
```

```
    StatusBar1.Panels(2).Text = "Sacando equipo de la red. -  
    Transfiriendo Token a Equipo " & Siguiente + 1
```

```
    Call MoverInfo(Token, Siguiente, Info(Token),  
    Info(Siguiente), 3)
```

```
    'ack
```

```
    StatusBar1.Panels(2).Text = "Sacando equipo de la red. -  
    Enviando ACK a Equipo " & Token + 1
```

```
    Color1 = Info(Siguiente).FillColor
```

```
    Info(Siguiente).FillColor = &HFF&
```

```
    Call MoverInfo(Siguiente, Token, Info(Siguiente),  
    Info(Token), 3)
```

```
    Info(Siguiente).FillColor = Color1
```

```
    '*****
```

```
    Call CambiaToken(Siguiente, CByte(Token))
```

```
    StatusBar1.Panels(2).Text = "Token Transferido..."
```

```
    Frame1(0).Refresh
```

```
    Frame1(3).Refresh
```

```
    Call Pausa(1)
```

```
    UToken = Token
```

```
    Token = Siguiente
```


OBJETO TOKENBUS3	
Este objeto es el formulario encargado de la simulación Recuperación del Token Caso 1, utilizando el protocolo TokenBus con Topología Bus. Se pueden resaltar las siguientes funciones:	
Función	Descripción
Recuperar	<p>En esta función es donde realmente se realiza la simulación. Tiene como parámetros el Token (Equipo donde está el Token), Siguiente (Equipo siguiente al que tiene el Token), E (vector donde están los equipos a participar en la simulación), NumEquipos (Número de equipos activos en la red).</p> <p>Esta función, utiliza las funciones BuscaSiguiente, MoverInfo, CambiaToken, entre otras.</p>
Código de la Función Recuperar	

```
Sub Recuperar(Token, Siguiente, E(), NumEquipos)
Dim Eq(8)

StatusBar1.Panels(1).Text = "Activo"
If (Tkn >= 0) Then
    Token = Tkn
End If

'funcion para recuperar el token caso 1
CFallo = 0
N = 0
For i = 0 To 7 Step 1
    Eq(i) = EquiposActivos(i)
    If (Eq(i) >= 0) Then
        N = N + 1
    End If
    If (E(i) = -2) Then
        CFallo = CFallo + 1
    End If
Next i
```

```
Do
  If (E(Siguiente) = -2) Then
    C = 1
    Do
      If (C < 3) Then
        StatusBar1.Panels(2).Text = "Enviando Token a equipo "
        & Siguiente + 1 & " - Número de veces = " & C
        StatusBar1.Refresh
        Call MoverInfo(Token, Siguiente, Info(Token),
        Info(Siguiente), 3)
      End If
      C = C + 1
    Loop Until (C > 2)
    Us = Siguiente
    Siguiente = BuscaSiguiente(Siguiente, Eq, NumEquipos)
  StatusBar1.Panels(2).Text = "Enviando Who-Follow"
  Color1 = Info(Token).FillColor
  Info(Token).FillColor = &HFF00FF
  Call MoverInfo(Token, Siguiente, Info(Token),
  Info(Siguiente), 3)
```

```
If (cont >= 1) Then
```

```
    s = False
```

```
    ii = Token
```

```
    Do
```

```
        valor = ii
```

```
        If (EquiposActivos(valor) >= 0) And _
```

```
            (Equipos(valor).Value = 0) And _
```

```
                (valor <> Token) Then
```

```
                    s = True
```

```
        Else
```

```
            If (ii <= 6) Then
```

```
                ii = ii + 1
```

```
            Else
```

```
                ii = 0
```

```
            End If
```

```
        If (ii = Token) Then
```

```
            s = True
```

```
            valor = Token
```

```
        End If
```

```
    End If
```

```
If (Siguiente <> Token) Then
    StatusBar1.Panels(2).Text = "Enviando Set-Sucesor"
    Info(Siguiente).FillColor = &HFF00&
    Call MoverInfo(Siguiente, Token, Info(Siguiente),
Info(Token), 3)
    Info(Siguiente).FillColor = Color1
    StatusBar1.Panels(2).Text = "Transfiriendo Token al
equipo " & Siguiente + 1
    Color1 = Info(Token).FillColor
    Call MoverInfo(Token, Siguiente, Info(Token),
Info(Siguiente), 3)
    'enviamos el Ack
    StatusBar1.Panels(2).Text = "Enviando ACK al equipo " &
Token + 1
    Info(Siguiente).FillColor = &HFF&
    Call MoverInfo(Siguiente, Token, Info(Siguiente),
Info(Token), 3)
    Info(Siguiente).FillColor = Color1
    Call CambiaToken(Siguiente, CByte(Token))
End If
Call ActualizaAnillo(Siguiente, CByte(Token))
```

```
Tkn = Token
For i = 0 To 7 Step 1
    Equipos(i).Value = 0
    If (EquiposActivos(i) >= 0) And (i <> Tkn) Then
        Equipos(i).Enabled = True
    Else
        Equipos(i).Enabled = False
    End If
Next i
StatusBar1.Panels(1).Text = "Inactivo"
StatusBar1.Panels(2).Text = ""
End Sub
```

OBJETO TOKENBUS4	
Este objeto es el formulario encargado de la simulación Recuperación del Token Caso 2, utilizando el protocolo TokenBus con Topología Bus. Se pueden resaltar las siguientes funciones:	
Función	Descripción
Inicia_Click	En esta función, se inicia la simulación del segundo caso de recuperación del token. Utiliza las funciones Moverinfo, CambiaToken, ActualizaAnillo, entre otras.
Código de la Función Inicia_Click	

```
Private Sub Inicia_Click()
Dim Eq(8) As Variant
Sw = False
cont = 0
For i = 0 To 7 Step 1
  If (Equipos(i).Value = 1) Then
    Sw = True
    cont = cont + 1
  End If
Next i
If (Sw = False) Then
  v = MsgBox("Debe seleccionar" & Chr(13) & "los equipos en donde falla la
transmisión", vbCritical, "Error...")
  Exit Sub
ElseIf (cont <> 2) Then
  v = MsgBox("Debe seleccionar 2 Equipos...", vbCritical, "Error...")
  Exit Sub
End If

Time(0).Caption = 0
Time(1).Caption = 0
Time(0).Refresh
Time(1).Refresh
NumEquipos = 0
For i = 0 To 7 Step 1
  Eq(i) = EquiposActivos(i)
  If (EquiposActivos(i) >= 0) Then
    NumEquipos = NumEquipos + 1
  End If
Next i
StatusBar1.Panels(1).Text = "Activo"
tkn1 = QuienToken(CByte(Sig))
Tkn = tkn1
A = Timer
```

```

For i = 0 To NumEquipos - 1 Step 1
  Sig = BuscaSiguiente(tkn1, Eq, NumEquipos)
  StatusBar1.Panels(2).Text = "Calculando tiempo de regreso del token... -
Transfiriendo Token a equipo " & Sig + 1
  Call MoverInfo(tkn1, Sig, Info(tkn1), Info(Sig), 3)
  Color1 = Info(Sig).FillColor
  Info(Sig).FillColor = &HFF&
  StatusBar1.Panels(2).Text = "Calculando tiempo de regreso del token... - Enviando
ACK a equipo " & tkn1 + 1
  Call MoverInfo(Sig, tkn1, Info(Sig), Info(tkn1), 3)
  Info(Sig).FillColor = Color1
  Call CambiaToken(Sig, CByte(tkn1))
  Call ActualizaAnillo(Sig, CByte(tkn1))
  Time(0).Caption = Int(Timer - A)
  Frame1(0).Refresh
  Frame1(3).Refresh
  Time(0).Refresh
  Time(1).Refresh
  Beep
  tkn1 = Sig
Next i
'desaparecemos el token ÷ los equipos marcados
Sw = False
Token = Tkn
UToken = -1
A = Timer
Do
  Call Pausa(0.1)
  Time(1).Caption = Int(Timer - A)
  Time(1).Refresh
  If (CDec(Time(1).Caption) > CDec(Time(0).Caption)) Then
    StatusBar1.Panels(2).Text = "Tiempo de Respuesta Excedido"
    Call Pausa(1)
    Sw = True
  'inicialización
  StatusBar1.Panels(2).Text = "Iniciando la Red..."
  Call Inicializacion
  Time(0).Caption = 0
  Time(1).Caption = 0
  Time(0).Refresh
  Time(1).Refresh
  StatusBar1.Panels(2).Text = "Iniciando la Red..."
  Frame1(0).Refresh
  Frame1(3).Refresh
Exit Sub

```

```

Else
  Sig = BuscaSiguiente(Token, Eq, NumEquipos)
  If ((Equipos(Token).Value = 1) And (Equipos(Sig).Value = 1)) Then
    If (UToken <> Token) Then
      StatusBar1.Panels(2).Text = "Transfiriendo Token a equipo " & Sig + 1
      Call MoverInfo(Token, Sig, Info(Token), Info(Sig), 2)
      StatusBar1.Panels(2).Text = "Transfiriendo Token a equipo " & Sig + 1 & " -
Esperando ACK"
      Frame1(0).Refresh
      Frame1(3).Refresh
    End If
    UToken = Token
  Else
    StatusBar1.Panels(2).Text = "Transfiriendo Token a equipo " & Sig + 1
    Call MoverInfo(Token, Sig, Info(Token), Info(Sig), 3)
    'Ack
    Color1 = Info(Sig).FillColor
    Info(Sig).FillColor = &HFF&
    StatusBar1.Panels(2).Text = "Enviando ACK a equipo " & Token + 1
    Call MoverInfo(Sig, Token, Info(Sig), Info(Token), 3)
    Info(Sig).FillColor = Color1
    Call CambiaToken(Sig, CByte(Token))
    Call ActualizaAnillo(Sig, CByte(Token))
    StatusBar1.Panels(2).Text = "Token Transferido..."
    Frame1(0).Refresh
    Frame1(3).Refresh
    Call Pausa(0.5)
    Token = Sig
  End If
End If
Loop Until (Sw = True)
End Sub

```

OBJETO TOKENBUS5	
Este objeto es el formulario encargado de la simulación Transmisión Exitosa, utilizando el protocolo TokenBus con Topología Bus. Se pueden resaltar las siguientes funciones:	
Función	Descripción
Inicia_Click	En esta función se realiza esta simulación. Utiliza las funciones MoverInfo, CambiaToken, entre otras.
Código de la función Inicia_Click	
<pre> Private Sub Inicia_Click() Dim Origen, destino Dim E(8) If (EqDestino.Text = "") Then v = MsgBox("Debe seleccionar un destino...", vbCritical, "Error...") Exit Sub End If N = 0 For i = 0 To 7 Step 1 If (EquiposActivos(i) >= 0) Then N = N + 1 End If E(i) = EquiposActivos(i) Next i StatusBar1.Panels(1).Text = "Activo" Origen = CDec(Mid(EqOrigen.List(EqOrigen.ListIndex), 8, 1)) destino = CDec(Mid(EqDestino.List(EqDestino.ListIndex), 8, 1)) Trama(3).Caption = "4: PC " & destino Trama(4).Caption = "5: PC " & Origen Trama(5).Caption = "6: Mensaje" Trama(3).Refresh Trama(4).Refresh Trama(5).Refresh Sw = False If (Tkn >= 0) Then Token = Tkn Else Token = QuienToken(CByte(Sig)) End If </pre>	

```

Do
If (Timer1.Enabled = False) Then
If (Token = Origen - 1) Then
    StatusBar1.Panels(2).Text = "Enviando información de " & Origen & " a " & destino
    Color1 = Info(Origen - 1).FillColor
    Info(Origen - 1).FillColor = &HC0C0FF
    Call MoverInfo(Origen - 1, destino - 1, Info(Origen - 1), Info(destino - 1), 1)
    Aux(0).Visible = True
    Aux(1).Visible = True
    Aux(0).Left = Line2(Origen - 1).X1 - 50
    Aux(1).Left = Line2(Origen - 1).X1 + 50
Do
    For i = 0 To 1 Step 1
        If (i = 0) Then
            Aux(i).Left = Aux(i).Left - 50
        Else
            Aux(i).Left = Aux(i).Left + 50
        End If
        If (Abs(Aux(i).Left - Info(destino - 1).Left) < 50) Then
            Timer1.Enabled = True
            GoTo salir
        End If
    Next i
    Call Pausa(0.07)
Loop Until (Aux(0).Left <= Line1(0).X1) And (Aux(1).Left >= Line1(0).X2)
salir:
    Dest = destino - 1
Else
    Sig = BuscaSiguiente(Token, E, N)
    'transfiriendo token de token a sig
    StatusBar1.Panels(2).Text = "Transfiriendo token a equipo " & Sig + 1
    Call MoverInfo(Token, Sig, Info(Token), Info(Sig), 3)
StatusBar1.Panels(2).Text = "Enviando ACK a equipo " & Token + 1
    Color1 = Info(Sig).FillColor
    Info(Sig).FillColor = &HFF&
    Call MoverInfo(Sig, Token, Info(Sig), Info(Token), 3)
    Info(Sig).FillColor = Color1
Call ActualizaAnillo(Sig, CByte(Token))
Call CambiaToken(Sig, CByte(Token))
    StatusBar1.Panels(2).Text = "Token Transferido"
    Call Pausa(0.5)
    Frame1(0).Refresh
    Frame1(3).Refresh
    Call Pausa(0.5)
    Token = Sig
End If
Else 'timer
    Sw = True
End If
Loop Until (Sw = True)
End Sub

```


PROTOCOLO TOKEN RING

FUNCIÓN CirculacionIndTR	
Esta función se encarga de realizar la simulación de circulación indefinida del tokenring en el protocolo tokenring y topología anillo.	
Variables	Definición
Desde	Es de tipo byte y se encarga de guardar el desde donde se va a transmitir el token.
Hasta	Es de tipo byte y se encarga de guardar el destino donde va dirigido el token.
Moni	Es de tipo byte y se encarga de guardar el numero de la estación monitora.
Band, band1	Son variable auxiliares de tipo booleano.
Código Fuente de la Función	
<pre> Sub CirculacionIndTR() Dim desde As Byte Dim hasta As Byte, moni As Byte Dim band, band1 As Boolean pantalla(0).Refresh If (equiposac.Text = "") Then equiposac.ListIndex = 0 End If band = False band1 = True For i = 0 To 7 Step 1 If ((band = True) And EquiposActivos(i) > -1) Then hasta = i band = False End If </pre>	

```
If (EquiposActivos(i) = 8) Then
    band = True
    moni = i
    desde = i
End If
Next i
hasta = CByte(Mid(desde2.Text, 4, 1)) - 1
info1(desde).Visible = False
If (desde = moni) Then
    Pc(desde).Picture = LoadPicture("Server2.ico")
Else
    Pc(desde).Picture = LoadPicture("Active.ico")
End If
Call MoverToken(desde, hasta)
For i = 0 To 7 Step 1
    Info(i).Shape = 3
Next i
Pc(hasta).Picture = LoadPicture("Active.ico")
desde = CByte(Mid(desde2.Text, 4, 1)) - 1
hasta = CByte(Mid(equiposac.Text, 4, 1)) - 1
OpcionTr = 4
info1(desde).FillColor = &HFF00FF
info1(desde).Visible = True
If (desde > hasta) Then
    For i = desde To 7
        Call MoverToken(i, i + 1)
        If (i + 1 = moni) Then
            If (i = 7) Then
                Pc(0).Picture = LoadPicture("server2.ico ")
            Else
                Pc(i + 1).Picture = LoadPicture("server2.ico ")
            End If
        End If
    Next i
End If
```

```
Else
  If (EquiposActivos(i + 1) >= 0) Then
    If (i = 7) Then
      Pc(0).Picture = LoadPicture("active.ico ")
    Else
      Pc(i + 1).Picture = LoadPicture("active.ico ")
    End If
  End If
End If
If (i + 1 = moni) Then
  Info(i + 1).BorderWidth = 3
End If
Next i
For i = 0 To hasta - 1
  Call MoverToken(i, i + 1)
  If (i + 1 = moni) Then
    If (i = 7) Then
      Pc(0).Picture = LoadPicture("server2.ico ")
    Else
      Pc(i + 1).Picture = LoadPicture("server2.ico ")
    End If
  Else
    If (EquiposActivos(i + 1) >= 0) Then
      If (i = 7) Then
        Pc(0).Picture = LoadPicture("active.ico ")
      Else
        Pc(i + 1).Picture = LoadPicture("active.ico ")
      End If
    End If
  End If
End If
```

```
If (i + 1 = moni) Then
    Info(i + 1).BorderWidth = 3
End If
Next i
Else
    For i = desde To hasta - 1
        Call MoverToken(i, i + 1)
        If (i + 1 = moni) Then
            Pc(i + 1).Picture = LoadPicture("server2.ico ")
        Else
            If (EquiposActivos(i + 1) >= 0) Then
                Pc(i + 1).Picture = LoadPicture("active.ico ")
            End If
        End If
    End If
    If (i + 1 = moni) Then
        Info(i + 1).BorderWidth = 3
    End If
Next i

End If
info1(hasta).Visible = False
OpcionTr = 2
'Call MoverToken1(hasta)
'Call MoverToken(0, moni)
Pc(moni).Picture = LoadPicture("Server2.ico")
pantalla(0).Refresh
Pausa (1)

Call MoverToken(hasta, moni)
If (Info(moni).BorderWidth = 1) Then
    Info(moni).BorderWidth = 3
    Pc(moni).Picture = LoadPicture("Server2.ico")
```

```
Call MoverToken(moni, moni)
End If
Info(moni).Visible = False
Pc(moni).Picture = LoadPicture("Server2.ico")
pantalla(0).Refresh
For j = 0 To 5 Step 1
    Beep
    Pausa (0.5)
Next j

For i = 0 To 7 Step 1
    Info(i).BorderWidth = 1
    Info(i).Shape = 0
Next i
Pc(moni).Picture = LoadPicture("monitkn.ico")
'Info(moni).Visible = True
End Sub
```

FUNCIÓN InicializarTR	
Esta función se encarga de realizar la simulación de inicialización del anillo en el protocolo tokenring	
Variables	Definición
desde	Es de tipo byte y se encarga de guardar el desde donde se va a transmitir el token.
hasta	Es de tipo byte y se encarga de guardar el destino donde va dirigido el token.
aleatorio	Es de tipo byte y sirve para guardar un valor aleatorio, el cual será la estación monitora
Ban, ban1, Swch, s	Son variable auxiliares de tipo booleano.
Código Fuente de la Función	
<pre> Sub InicializarTR() Dim desde As Byte Dim hasta As Byte Dim aleatorio As Byte Dim j As Variant ban = False ban1 = False For k = 0 To 7 Step 1 If ((Equipos(k).Value = 1) And (Equipos(k).Enabled = True)) Then ban = True End If If (EquiposActivos(k) = 8) Then ban1 = True End If Next k </pre>	

```
If (ban = False) And (ban1 = False) Then
  Swch = True
  For i = 0 To 7 Step 1
    If (EquiposActivos(i) >= 0) Then
      If (Swch = False) Then
        TokenRing.Pc(i).Picture = LoadPicture("Active.ico")
        TokenRing.Pc(i).ToolTipText = "Equipo Activo"
        TokenRing.Refresh
      Else
        TokenRing.Pc(i).Picture = LoadPicture("Active.ico")
        TokenRing.Pc(i).ToolTipText = "Equipo Activo"
        Info(i).FillColor = &HFF&
        Info(i).Visible = True
        TokenRing.Refresh
        desde = i
        Swch = False
      End If
    End If
  Next i
  Randomize (7)
  Do
    aleatorio = Int(Rnd * 7)
    Sw1 = False
    If (aleatorio = EquiposActivos(aleatorio)) Then
      Sw1 = True
    Else
      Sw1 = False
    End If
  Loop Until (Sw1 = True)
  For i = 0 To 7 Step 1
    If (Info(i).Visible = True) Then
      Info(i).Visible = False
    End If
```

```
hasta = aleatorio
If (OpcionTr <= 0) Then
  For i = 0 To 7 Step 1
    If (EquiposActivos(i) >= 0) Then
      Call MoverToken(i, i)
      Info(i).Visible = False
      desde = 0
    End If
  Next i
  'For i = 0 To 7 Step 1
  ' Info(i).FillColor = &HFF&
  'Next i
  Info(hasta).FillColor = &HC000&
  Info(hasta).Visible = True
  Call MoverToken(hasta, hasta)
End If
EquiposActivos(aleatorio) = 8
info1(aleatorio).Visible = True
Info(aleatorio).Visible = False
TokenRing.Pc(aleatorio).Picture = LoadPicture("Monitkn.ico")
TokenRing.Pc(aleatorio).ToolTipText = "Estación Monitora"
For i = 0 To 7 Step 1
  Info(i).Visible = False
  If (EquiposActivos(i) = 8) Then
    Info(i).Top = info1(i).Top
    Info(i).Left = info1(i).Left
    Info(i).FillColor = &HFFFF&
    Info(i).Visible = True

  End If
Next i
Ejecutar.Caption = "Insertar"
*** insertar equipo
```

```
**** insertar equipo
ElseIf (ban = True) Then
  For j = 7 To 0 Step -1
    If (Equipos(j).Enabled = True And Equipos(j).Value = 1) Then
      TokenRing.Pc(j).Picture = LoadPicture("active.ico")
      TokenRing.Pc(j).ToolTipText = "Equipo Activo"
      EquiposActivos(j) = j
      'Equipos(j).Enabled = False
      desde = j
    End If
  Next j
  For j = 0 To 7 Step 1
    If (Equipos(j).Value = 1 And Equipos(j).Enabled = True) Then
      Equipos(j).Enabled = False
      Call MoverToken(j, j)
      Info(j).Visible = False
      hasta = j
    End If
  Next j

  For i = 0 To 7 Step 1
    If (info1(i).Visible = True) Then
      aleatorio = i
    End If
  Next i
  Info(aleatorio).Top = info1(aleatorio).Top
  Info(aleatorio).Left = info1(aleatorio).Left
  Info(aleatorio).FillColor = info1(aleatorio).FillColor
  info1(aleatorio).Visible = False
  Info(aleatorio).Visible = True
```

```
desde = aleatorio
    Call MoverToken(desde, hasta)
    Info(hasta).Visible = False
    info1(hasta).Visible = True
Else
    s = False
    For i = 0 To 7 Step 1
        If (EquiposActivos(i) <= -1) Then
            v = MsgBox("Inserte un nuevo equipo al anillo", vbInformation, "Cuadro
de Dialogo...")
            s = True
            Exit For
        End If
    Next i
    If (s = False) Then
        v = MsgBox("No hay equipos disponibles para insertar en el anillo",
vbInformation, "Cuadro de Dialogo...")
        Ejecutar.Enabled = False
    End If
End If
End Sub
```

FUNCIÓN Liberacion AntTestigo	
Esta función se encarga de realizar la simulación de liberación anticipada del testigo en el protocolo tokenring	
Variables	Definición
desde	Es de tipo byte y se encarga de guardar el desde donde se va a transmitir el token.
hasta	Es de tipo byte y se encarga de guardar el destino donde va dirigido el token.
moni	Es de tipo byte y sirve para guardar el numero de la estación monitora.
Opciontr	Variable global de tipo byte que se encarga de guardar el numero de la simulación que se escoge.
S, s1	Son variable auxiliares de tipo boleano.
Código Fuente de la Función	

```
Sub LiberacionAntTestigo()  
  pantalla(0).Refresh  
  Pausa (2)  
  s1 = False  
  For i = 0 To 7 Step 1  
    If (Info(i).Visible = True) Then  
      desde = i  
      Exit For  
    End If  
  Next i  
  For i = 0 To 7 Step 1  
    If (EquiposActivos(i) = 8) Then  
      moni = i  
    End If  
  Next i
```

```
For i = desde + 1 To 7 Step 1
  If (EquiposActivos(i) > -1) Then
    hasta = i
    s1 = True
    Exit For
  End If
Next i
If (s1 = False) Then
  For i = 0 To 7 Step 1
    If (EquiposActivos(i) > -1) Then
      hasta = i
      Exit For
    End If
  Next i
End If
OpcionTr = 200
If (desde2.Text = "") Then
  desde2.ListIndex = 0
End If
hasta = CByte(Mid(desde2.Text, 4, 1)) - 1
info1(desde).Visible = False
Call MoverToken(desde, hasta)
If (hasta = moni) Then
  TokenRing.Pc(hasta).Picture = LoadPicture("Server2.ico")
Else
  TokenRing.Pc(hasta).Picture = LoadPicture("active.ico")
End If
OpcionTr = 4
Info(hasta).Shape = 3
info1(hasta).FillColor = &HFF00FF
info1(hasta).Visible = True
desde = hasta
s = False
```

Randomize (7)

Do

aleatorio = Int(Rnd * 7)

If (aleatorio <> hasta And EquiposActivos(aleatorio) > -1) Then

s = True

End If

Loop Until (s = True)

hasta = aleatorio

hasta = CByte(Mid(equiposac.Text, 4, 1)) - 1

pantalla(0).Refresh

Pausa (0.5)

Call MoverToken(desde, hasta)

If (hasta = moni) Then

TokenRing.Pc(hasta).Picture = LoadPicture("Server2.ico")

Else

TokenRing.Pc(hasta).Picture = LoadPicture("active.ico")

End If

info1(hasta).Visible = False

For i = 0 To 7 Step 1

info1(i).Visible = False

info1(i).FillColor = &HFFFF&

Next i

Pausa (0.5)

Info(hasta).Shape = 0

desde = hasta

**** para saber cual es el siguiente activo****

s1 = False

For i = desde + 1 To 7 Step 1

If (EquiposActivos(i) > -1) Then

hasta = i

s1 = True

Exit For

```
If (s1 = False) Then
  For i = 0 To 7 Step 1
    If (EquiposActivos(i) > -1) Then
      hasta = i
      Exit For
    End If
  Next i
End If
*****

OpcionTr = 200
Pausa (0.5)
Call MoverToken(desde, hasta)
OpcionTr = 4
Beep
End Sub
```

FUNCIÓN llenadestino	
Esta función se encarga de llenar la lista de equipos que puede ser origen de la transmisión del tokenring.	
Variables	Definición
i	Variable de tipo byte. Contador.
Código Fuente de la Función	
<pre> Sub llenadestino() desde2.Clear For i = 0 To 7 Step 1 If (EquiposActivos(i) >= 0) Then If (Info(i).Visible = False) Then TokenRing.desde2.List(TokenRing.desde2.ListCount) = "PC " & i + 1 End If End If End For Next i End Sub </pre>	

FUNCIÓN llenadestino	
Esta función se encarga de llenar la lista de equipos que puede ser destino de la transmisión del tokenring.	
Variables	Definición
i	Variable de tipo byte. Contador.
Código Fuente de la Función	
<pre> Sub llenahasta() equiposac.Clear For i = 0 To 7 Step 1 If (EquiposActivos(i) >= 0) Then vari = "PC" & Str\$(i + 1) If (desde2.Text <> vari) Then TokenRing.equiposac.List(TokenRing.equiposac.ListCount) = "PC " & i + 1 End If End If Next i TokenRing.equiposac.ListIndex = 0 Ejecutar.Enabled = True End Sub </pre>	

FUNCIÓN MoverToken	
Esta función se encarga de realizar el movimiento del tokenring desde un origen hasta un destino	
Parámetros de Entrada	Definición
desde	Es de tipo byte y se encarga de guardar el desde donde se va a transmitir el token.
hasta	Es de tipo byte y se encarga de guardar el destino donde va dirigido el token.
Variables	Definición
Desde1	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
Hasta1	Es de tipo byte y se encarga de guardar el destino donde va dirigido el token.
Tiempo	Es una variable global que se encarga de llevar el tiempo que circula el token por el anillo.
monitora	Es de tipo byte y sirve para determinar quien es la monitora
k	Es de tipo byte y sirve para determinar el signo de la pendiente.
Band, band1	Son variables auxiliares de tipo booleano.
Código Fuente de la Función	

```
Sub MoverToken(desde As Variant, hasta As Variant)
```

```
Dim desde1, hasta1 As Variant
```

```
tiempo = Timer
```

```
desde1 = desde
```

```
hasta1 = hasta
```

```
tem1.Text = Int(Timer - tiempo)
```

```
monitora = 9
```

```
For i = 0 To 7 Step 1
```

```
    If (EquiposActivos(i) = 8) Then
```

```
        monitora = i
```

```
    End If
```

```
Next i
```

```
If (monitora <> 9) Then
  If (Info(desde).Visible = True And Info(desde) = info1(desde)) Then
    If (monitora = desde) Then
      TokenRing.Pc(desde).Picture = LoadPicture("server2.ico")
    Else
      TokenRing.Pc(desde).Picture = LoadPicture("Active.ico")
    End If
  End If
End If
If (desde >= hasta) Then
  Call MoverToken1(desde)
  desde1 = 0
End If

For i = desde1 To hasta1 - 1 Step 1
  derecha = info1(i).Left
  arriba = info1(i).Top
  Info(i).Visible = True
  X1 = anillo(i).X1
  Y1 = anillo(i).Y1
  X2 = anillo(i).X2
  Y2 = anillo(i).Y2
  m = (Y2 - Y1) / (X2 - X1)
  If (m >= 0) Then
    k = 1
  Else
    k = -1
  End If
  If (Y1 > Y2) And Not (X1 > X2) Then
    k = 1
  ElseIf (Y1 > Y2) And (X1 > X2) Then
    k = -1
  End If
```

```
For j = X1 To X2 Step 50 * k
  Y = m * (j - X1) + Y1
  Info(i).Left = j
  Info(i).Top = Y
  If (OpcionTr = 0 And (Info(i).FillColor = &HFF& Or Info(i).FillColor =
&HC000&)) Then
    Pausa (0.05)
  Else
    Pausa (0.1)
  End If
  If (k >= 0) Then
    If (OpcionTr = 4 And j > X1) Then
      info1(i).Left = j - 50
      info1(i).Top = Y - 50
    End If
  Else
    If (OpcionTr = 4 And j < X1) Then
      info1(i).Left = j + 50
      info1(i).Top = Y + 50
    End If
  End If
Next j
If (i < 7) Then
  If (OpcionTr = 3 And Pc(i + 1).ToolTipText = "Equipo Activo") Then
    Info(i).FillColor = &HFF8080
    pantalla(0).Refresh
    Pausa (0.5)
    Info(i).FillColor = &HFF&
  End If
ElseIf (OpcionTr = 3 And Pc(0).ToolTipText = "Equipo Activo") Then
  Info(i).FillColor = &HFF8080
  pantalla(0).Refresh
  Pausa (0.5)
  Info(i).FillColor = &HFF&
```

```
End If
Info(i).Visible = False ' OJO
If (i < 7) Then
    Info(i + 1).Top = info1(i + 1).Top
    Info(i + 1).Left = info1(i + 1).Left
    Info(i + 1).Shape = Info(i).Shape
    Info(i + 1).FillColor = Info(i).FillColor
    Info(i + 1).BorderWidth = Info(i).BorderWidth
    Info(i + 1).Visible = True
    If (OpcionTr = 4) Then
        info1(i + 1).FillColor = info1(i).FillColor
        info1(i).Visible = False
        info1(i).Left = derecha
        info1(i).Top = arriba
        info1(i + 1).Visible = True
    End If
Else
    Info(0).Top = info1(0).Top
    Info(0).Left = info1(0).Left
    Info(0).Shape = Info(i).Shape
    Info(0).FillColor = Info(i).FillColor
    Info(0).BorderWidth = Info(i).BorderWidth
    Info(0).Visible = True
    If (OpcionTr = 4) Then
        info1(0).FillColor = info1(i).FillColor
        info1(i).Visible = False
        info1(i).Left = derecha
        info1(i).Top = arriba
        info1(0).Visible = True
    End If
End If
```

```
If (i < 7) Then
  If (Pc(i + 1).ToolTipText = "Equipo Activo" And Info(i).FillColor =
    &HFFFF&) Then
    If (i = hasta - 1) Then
      TokenRing.Pc(i + 1).Picture = LoadPicture("Server.ico")
      Beep
      pantalla(0).Refresh
    Else
      TokenRing.Pc(i + 1).Picture = LoadPicture("Server.ico")
      Beep
      pantalla(0).Refresh
      Pausa (0.4)
      TokenRing.Pc(i + 1).Picture = LoadPicture("Active.ico")
    End If
  End If
  If (Pc(i + 1).ToolTipText = "Estación Monitora" And Info(i).FillColor =
    &HFFFF&) Then
    If (i = hasta - 1) Then
      TokenRing.Pc(i + 1).Picture = LoadPicture("Monitkn.ico")
      Beep
      pantalla(0).Refresh
    Else
      TokenRing.Pc(i + 1).Picture = LoadPicture("Monitkn.ico")
      Beep
      pantalla(0).Refresh
      Pausa (0.4)
      TokenRing.Pc(i + 1).Picture = LoadPicture("Server2.ico")
    End If
  End If
```

```
End If
    pantalla(0).Refresh
    tem1.Text = Int(Timer - tiempo)
Next i
If (desde1 = hasta1) Then
    If (OpcionTr > 0) Then
        If (monitora = desde1) Then
            TokenRing.Pc(desde1).Picture = LoadPicture("monitkn.ico")
        Else
            TokenRing.Pc(desde1).Picture = LoadPicture("Server.ico")
        End If
    End If
End If
End If
End Sub
```

FUNCIÓN MoverToken1	
Esta función se encarga de realizar el movimiento del tokenring desde un origen hasta el equipo 8.	
Parámetros de Entrada	Definición
Desde	Es de tipo byte y se encarga de guardar el desde donde se va a transmitir el token.
Variables	Definición
Desde	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
Tiempo	Es una variable global que se encarga de llevar el tiempo que circula el token por el anillo.
k	Es de tipo byte y sirve para determinar el signo de la pendiente.
Band	Variable auxiliar de tipo booleano.
Código Fuente de la Función	
<pre> Sub MoverToken1(desde As Variant) tem1.Text = Int(Timer - tiempo) For k = 0 To 7 Step 1 If (EquiposActivos(k) = 8) Then monitora = k End If Next k For i = desde To 7 Step 1 tem1.Text = Int(Timer - tiempo) tem1.Refresh ban1 = False derecha = info1(i).Left arriba = info1(i).Top </pre>	

```
If (OpcionTr <= 0 And Info(i).Visible = False) Then
    Info(i).FillColor = &HFF&
End If
Info(i).Visible = True
X1 = anillo(i).X1
Y1 = anillo(i).Y1
X2 = anillo(i).X2
Y2 = anillo(i).Y2
m = (Y2 - Y1) / (X2 - X1)
If (m >= 0) Then
    k = 1
Else
    k = -1
End If
If (Y1 > Y2) And Not (X1 > X2) Then

    k = 1
ElseIf (Y1 > Y2) And (X1 > X2) Then

    k = -1
End If
For j = X1 To X2 Step 50 * k
    Y = m * (j - X1) + Y1
    Info(i).Left = j
    Info(i).Top = Y
    If (OpcionTr = 0 And (Info(i).FillColor = &HFF& Or Info(i).FillColor =
    &HC000&)) Then
        Pausa (0.05)
    Else
        Pausa (0.1)
    End If
End For
```

```

If (k >= 0) Then
    If (OpcionTr = 4) Then
        info1(i).Visible = True
        info1(i).Left = j - 50
        info1(i).Top = Y - 50
    End If
    ElseIf (OpcionTr = 4) Then
        info1(i).Visible = True
        info1(i).Left = j + 50
        info1(i).Top = Y + 50
    End If
Next j
If (i < 7) Then
    If (OpcionTr = 3 And Pc(i + 1).ToolTipText = "Equipo Activo") Then
        Info(i).FillColor = &HFF8080
        pantalla(0).Refresh
        Pausa (0.5)
        Info(i).FillColor = &HFF&
    End If
    ElseIf (OpcionTr = 3 And Pc(0).ToolTipText = "Equipo Activo") Then
        Info(i).FillColor = &HFF8080
        pantalla(0).Refresh
        Pausa (0.5)
        Info(i).FillColor = &HFF&
    End If
Info(i).Visible = False ' OJO
If (i < 7) Then
    Info(i + 1).Top = info1(i + 1).Top
    Info(i + 1).Left = info1(i + 1).Left
    Info(i + 1).FillColor = Info(i).FillColor
    Info(i + 1).Shape = Info(i).Shape
    Info(i + 1).BorderWidth = Info(i).BorderWidth
    Info(i + 1).Visible = True

```

```
If (OpcionTr = 4) Then
    info1(i + 1).FillColor = info1(i).FillColor
    info1(i).Visible = False
    info1(i).Left = derecha
    info1(i).Top = arriba
    info1(i + 1).Visible = True
End If
Else
    Info(0).Top = info1(0).Top
    Info(0).Left = info1(0).Left
    Info(0).FillColor = Info(i).FillColor
    Info(0).Shape = Info(i).Shape
    Info(0).BorderWidth = Info(i).BorderWidth
    Info(0).Visible = True
    If (OpcionTr = 4) Then
        info1(0).FillColor = info1(i).FillColor
        info1(i).Visible = False
        info1(i).Left = derecha
        info1(i).Top = arriba
        info1(0).Visible = True
    End If
End If
If (i < 7) Then
    If (Pc(i + 1).ToolTipText = "Equipo Activo" And Info(i).FillColor =
    &HFFFF&) Then
        TokenRing.Pc(i + 1).Picture = LoadPicture("Server.ico")
        Beep
        pantalla(0).Refresh
        Pausa (0.4)
        TokenRing.Pc(i + 1).Picture = LoadPicture("Active.ico")
    End If
    If (Pc(i + 1).ToolTipText = "Estación Monitora" And Info(i).FillColor =
    &HFFFF&) Then
        TokenRing.Pc(i + 1).Picture = LoadPicture("monitkn.ico")
```

```
Beep
    pantalla(0).Refresh
    Pausa (0.4)
    TokenRing.Pc(i + 1).Picture = LoadPicture("Server2.ico")
End If

Else
    If (Pc(0).ToolTipText = "Equipo Activo" And Info(0).FillColor = &HFFFF&)
Then
    TokenRing.Pc(0).Picture = LoadPicture("Server.ico")
    Beep
    pantalla(0).Refresh
    Pausa (0.4)
    TokenRing.Pc(0).Picture = LoadPicture("Active.ico")
End If
    If (Pc(0).ToolTipText = "Estación Monitora" And Info(0).FillColor =
&HFFFF&) Then
    TokenRing.Pc(0).Picture = LoadPicture("monitkn.ico")
    Beep
    pantalla(0).Refresh
    Pausa (0.4)
    TokenRing.Pc(0).Picture = LoadPicture("Server2.ico")
End If
End If
Next i

End Sub
```

FUNCIÓN PerdidaTR	
Esta función se encarga de realizar la simulación de perdida del tokenring.	
Variables	Definición
Desde	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
hasta	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
aleatorio	Es de tipo byte y se encarga de guardar en primera instancia el destino del tokenring
Tiempo	Variable global de tipo variant que sirve para contabilizar el tiempo que pasa el tokenring circulando por el anillo
Código Fuente de la Función	

```
Sub PerdidaTR()
Dim desde As Byte
Dim hasta As Byte
Dim aleatorio As Byte

tiempo1.Visible = True
tiempo = 0
pantalla(0).Refresh
If (desde2.Text = "") Then
    desde2.ListIndex = 0
End If
aleatorio = CByte(Mid(desde2.Text, 4, 1)) - 1
hasta = aleatorio
For i = 0 To 7 Step 1
    If (EquiposActivos(i) = 8) Then
        desde = i
        info1(i).Visible = False
    End If
Next i
```

```
For i = 0 To 7 Step 1
    Info(i).FillColor = &HFFFF&
Next i
destino1.Caption = "Perdida TR"
destino1.Visible = True
TokenRing.Refresh
Call MoverToken(desde, hasta)
monitora = desde
If (Info(hasta).Visible = True And Info(hasta) = info1(desde)) Then
    If (monitora = hasta) Then
        TokenRing.PC(hasta).Picture = LoadPicture("server2.ico")
    Else
        TokenRing.PC(hasta).Picture = LoadPicture("Active.ico")
    End If
End If
Info(hasta).Visible = False
TokenRing.Refresh
Do
    tem1.Text = Int(Timer - tiempo)
    Pausa (1)
    tem1.Refresh
Loop Until (CInt(tem1.Text) >= 20)
Pausa (1)
For i = 0 To 7 Step 1
' Info(i).Visible = False
If (EquiposActivos(i) = 8) Then
    Info(i).FillColor = &HFF&
    Info(i).Visible = True
    tiempo1.Visible = False
    moni = i
    Call MoverToken(i, i)
    Info(i).FillColor = info1(i).FillColor
```

```
tem.Text = 20
    tem1.Text = 0
    tiempo1.Visible = True
Exit For
' Info(i).FillColor = &HFFFF&
' Info(i).Visible = True
End If
Next i

TokenRing.PC(moni).Picture = LoadPicture("monitkn.ico")
End Sub
```

FUNCIÓN RecuperarMonitora	
Esta función se encarga de realizar la simulacion recuperación de la estacion monitora.	
Variables	Definición
Desde	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
hasta	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
aleatorio	Es de tipo byte y se encarga de guardar en primera instancia el destino del tokenring
moni	Es de tipo byte y se encarga de guardar quien es la monitora
hast	Es de tipo byte y se encarga de guardar quien será la nueva estación monitora
Código Fuente de la Función	

```
Sub RecuperarMonitora()
```

```
    pantalla(0).Refresh
```

```
    Pausa (1)
```

```
    For i = 0 To 7 Step 1
```

```
        If (Info(i).Visible = True Or info1(i).Visible = True) Then
```

```
            desde = i
```

```
        End If
```

```
    Next
```

```
    For i = 0 To 7 Step 1
```

```
        If (EquiposActivos(i) = 8) Then
```

```
            moni = i
```

```
        End If
```

```
    Next
```

Randomize (7)

Do

 aleatorio = Int(Rnd * 7)

 Sw1 = False

 If (aleatorio = EquiposActivos(aleatorio)) Then

 Sw1 = True

 Else

 Sw1 = False

 End If

Loop Until (Sw1 = True)

hasta = aleatorio

OpcionTr = 200

info1(desde).Visible = False

Pc(desde).Picture = LoadPicture("Server2.ico")

Call MoverToken(desde, hasta)

OpcionTr = 3

Info(hasta).Visible = False

pantalla(0).Refresh

Info(moni).FillColor = &HFF&

Call MoverToken(moni, moni)

TokenRing.Pc(moni).Picture = LoadPicture("Desac.ico")

TokenRing.Pc(moni).ToolTipText = "Equipo Inactivo"

EquiposActivos(moni) = -1

Info(moni).FillColor = &HFF8080

OpcionTr = 200

Randomize (7)

Do

 aleatorio = Int(Rnd * 7)

 Sw1 = False

 If (aleatorio = EquiposActivos(aleatorio) And Equipos(aleatorio) <> 8) Then

```
Sw1 = True
Else
Sw1 = False
End If
Loop Until (Sw1 = True)
hast = aleatorio
Call MoverToken(moni, hast)
If (hast = hasta) Then
Pc(hast).Picture = LoadPicture("Server.ico")
Else
Pc(hast).Picture = LoadPicture("Active.ico")
End If
Info(hast).FillColor = &HC000&
pantalla(0).Refresh
Pausa (1)
Call MoverToken(hast, hast)
Info(hast).Visible = False
Info(hast).FillColor = &HFFFF&
If (hast = hasta) Then
TokenRing.Pc(hast).Picture = LoadPicture("Monitkn.ico")
TokenRing.Pc(hast).ToolTipText = "Estación Monitora"
Else
TokenRing.Pc(hast).Picture = LoadPicture("Server2.ico")
TokenRing.Pc(hast).ToolTipText = "Estación Monitora"
End If
EquiposActivos(hast) = 8
info1(hasta).Visible = False
Info(hasta).FillColor = info1(hasta).FillColor
Info(hasta).Top = info1(hasta).Top
Info(hasta).Left = info1(hasta).Left
Info(hasta).Visible = True
OpcionTr = 3
End Sub
```


FUNCIÓN TExitosa	
Esta función se encarga de realizar la simulación de una transmisión exitosa en la topología tokenring.	
Variables	Definición
Desde	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
hasta	Es de tipo byte y se encarga de guardar desde donde se va a transmitir el token.
moni	Es de tipo byte y se encarga de guardar quien es la monitora
s1	Variable auxiliar de tipo boleano
Opciontr	Variable global de tipo byte que sirve para guardar el numero de la simulación
Código Fuente de la Función	

```
Sub TExitosa()  
Dim hasta As Variant  
  
For i = 0 To 7 Step 1  
    If (Info(i).Visible = True) Then  
        desde = i  
    End If  
Next i  
For i = 0 To 7 Step 1  
    If (EquiposActivos(i) = 8) Then  
        moni = i  
    End If  
Next i  
If (desde2.Text = "") Then  
    desde2.ListIndex = 1  
End If  
sigue = CByte(Mid(desde2.Text, 4, 1)) - 1  
Info(desde).Visible = False  
info1(desde).Visible = False
```

```
If (desde = moni) Then
    Pc(desde).Picture = LoadPicture("Server2.ico")
Else
    Pc(desde).Picture = LoadPicture("Active.ico")
End If
Call MoverToken(desde, sigue)
desde = sigue
Info(desde).Shape = 3
Pausa (2)
info1(desde).FillColor = &HFF00FF
info1(desde).Visible = True
hasta = CByte(Mid(equiposac.Text, 4, 1)) - 1
Label2(20).Caption = ""
Label2(21).Caption = ""
Label2(20).Caption = "4:PC" & Str$(desde + 1)
Label2(21).Caption = "5:PC" & Str$(hasta + 1)
If (desde = moni) Then
    Pc(desde).Picture = LoadPicture("Server2.ico")
Else
    Pc(desde).Picture = LoadPicture("Active.ico")
End If
OpcionTr = 4
Call MoverToken(desde, hasta)
OpcionTr = 5
info1(hasta).Visible = False
info1(hasta).FillColor = Info(hasta).FillColor
pantalla(0).Refresh
If (hasta = moni) Then
    TokenRing.Pc(hasta).Picture = LoadPicture("Server2.ico")
Else
    TokenRing.Pc(hasta).Picture = LoadPicture("active.ico")
End If
Pausa (1)
```

```
Call MoverToken(hasta, desde)
Info(desde).Shape = 0
Pausa (1)
**** para saber cual es el siguiente activo****
s1 = False
For i = desde + 1 To 7 Step 1
  If (EquiposActivos(i) > -1) Then
    sigue = i
    s1 = True
    Exit For
  End If
Next i
If (s1 = False) Then
  For i = 0 To 7 Step 1
    If (EquiposActivos(i) > -1) Then
      sigue = i
      Exit For
    End If
  Next i
End If
If (hasta = moni) Then
  TokenRing.Pc(hasta).Picture = LoadPicture("server2.ico")
Else
  TokenRing.Pc(hasta).Picture = LoadPicture("active.ico")
End If
Call MoverToken(desde, sigue)
If (sigue = moni) Then
  TokenRing.Pc(sigue).Picture = LoadPicture("monitkn.ico")
Else
  TokenRing.Pc(sigue).Picture = LoadPicture("server.ico")
End If
Beep
End Sub
```


CSMA / CD

Topología Bus

Funciones

FUNCION ColisionBus
Esta función activa un objeto timer de VisualBasic llamado mover el cual llama a la función moverseñal que se encarga de mover la señal de los PC a el bus.
Código de la Función
Sub ColisionBus() BarEstado.Panels(1).Text = "Simulando" mover.Enabled = True End Sub

FUNCION InicializarBus
Esta función sirve para inicializar los PC's que se encuentran en la modulo protocolo csma / cd y topología bus.

Código de la Función

```
Sub InicializarBus()

For i = 0 To 7 Step 1
  If (Equipos(i).Enabled = True) Then
    If (Equipos(i).Value = 0) Then
      pc2(i).Picture = LoadPicture("Active.ico")
      pc2(i).ToolTipText = "Equipo Activo"
    Else
      pc2(i).Picture = LoadPicture("Server.ico")
      pc2(i).ToolTipText = "Equipo Transmisor"
    End If
    CsmaBus.Refresh
    Beep
  Else
    pc2(i).Picture = LoadPicture("Desac.ico")
    pc2(i).ToolTipText = "Equipo Inactivo"
  End If
Next i

End Sub
```

FUNCION Ininfo	
Esta función sirve para inicializar los objetos que realizan el movimiento de las señales, en el protocolo de csma / cd y topología bus.	
Variables	Definición
Desde	Es de tipo byte y sirve para guardar el desde donde se va enviar la información.
Hasta	Es de tipo byte y sirve para guardar el destino de la transmisión.
Código de la Función	
<pre> Sub Ininfo() Ejecutar.Enabled = True desde = CByte(Mid(desde2.Text, 4, 1)) - 1 hasta = CByte(Mid(equiposac.Text, 4, 1)) - 1 If (desde < 4) Then Info(desde).top = Line2(desde).Y1 + 200 señali(desde).Left = Line2(desde).X1 - 100 señald(desde).Left = Line2(desde).X1 + 100 Else señali(desde).Left = Line2(desde).X1 - 100 señald(desde).Left = Line2(desde).X1 + 100 Info(desde).top = Line2(desde).Y2 - 150 End If For i = 0 To 7 Step 1 Info(i).FillColor = &HFFFF& señali(i).FillColor = &HFFFF& señald(i).FillColor = &HFFFF& Next i BarEstado.Panels(1).Text = "Inactivo" BarEstado.Panels(2).Text = "" End Sub </pre>	

FUNCION llenadestino
Esta función sirve para inicializar la lista de equipos que pueden ser transmisores.
Código de la Función
<pre>Sub llenadestino() desde2.Clear For i = 0 To 7 Step 1 If (EquiposActivos(i) >= 0) Then If (Info(i).Visible = False) Then CsmaBus.desde2.List(CsmaBus.desde2.ListCount) = "PC " & i + 1 End If End If Next i End Sub</pre>

FUNCIÓN MoverEnBus	
Esta función se encarga de mover la señal en el bus. Además es quien detecta las colisiones entre equipos.	
Parámetros de Entrada	Definición
Izquierda	Variable de tipo entero que sirve para limitar hasta donde llega la señal que se dirige hacia la izquierda.
Derecha	Variable de tipo entero que sirve para limitar hasta donde llega la señal que se dirige hacia la derecha.
Variables	Definición
s, s1	Variable de tipo booleano que sirve como swiche para saber si pasa por determinada parte del código del programa.
OpcionCs	Variable global de tipo byte que sirve para guardar la simulación a realizar.
Texto	Variable de tipo texto que sirve para guardar el mensaje de colisión entre equipos.
Código Fuente de la Función	
<pre> Sub MoverEnBus(izquierda As Integer, derecha As Integer) s = True s1 = False Do For i = 0 To 7 Step 1 If (señali(i).Left > izquierda) Then If (señali(i).Visible = True) Then señali(i).Left = señali(i).Left - 50 End If Else señali(i).Visible = False End If Next i For k = 0 To 7 Step 1 </pre>	

```

If (señali(k).Visible = True Or señald(k).Visible = True) Then
    s = True
    s1 = False
    Exit For
Else
    s = False
    s1 = True
End If
Next k
End If
Next i
For j = 0 To 7 Step 1
    If (señald(j).Left < derecha) Then
        If (señald(j).Visible = True) Then
            señald(j).Left = señald(j).Left + 50
        End If
    Else
        señald(j).Visible = False
    For k = 0 To 7 Step 1
        If (señald(k).Visible = True Or señali(k).Visible = True) Then
            s = True
            s1 = False
            Exit For
        Else
            s = False
            s1 = True
        End If
    Next k
    End If
Next j
Pausa (0.1)
For i = 0 To 3 Step 1
    For j = 7 To 7 - i Step -1
        If (j <> i) Then
            If (i = 0) Then
                If (señali(i).Left - 100 < señald(j).Left And señali(i).Visible = True And
señald(j).Visible = True) Then
                    s = False
                    texto = "Entre Eq. " & i + 1 & " y Eq. " & j + 1

```

```

End If
  ElseIf (i = 1) Then
    If (señali(i).Left - 100 < señald(j).Left And señali(i).Visible = True And
señald(j).Visible = True) Then
      s = False
      texto = "Entre Eq. " & i + 1 & " y Eq. " & j + 1

    End If
    If (señali(i).Left - 100 < señald(0).Left And señali(i).Visible = True And
señald(0).Visible = True) Then
      s = False
      texto = "Entre Eq. " & i + 1 & " y Eq. 1"

    End If
    ElseIf (i = 2) Then
      If (señali(i).Left - 100 < señald(j).Left And señali(i).Visible = True And
señald(j).Visible = True) Then
        s = False
        texto = "Entre Eq. " & i + 1 & " y Eq. " & j + 1

      End If
      If (señali(i).Left - 100 < señald(0).Left And señali(i).Visible = True And
señald(0).Visible = True) Then
        s = False
        texto = "Entre Eq. " & i + 1 & " y Eq. 1"

      End If
      If (señali(i).Left - 100 < señald(1).Left And señali(i).Visible = True And
señald(1).Visible = True) Then
        s = False
        texto = "Entre Eq. " & i + 1 & " y Eq. 2"

      End If
      ElseIf (i = 3) Then
        If (señali(i).Left - 100 < señald(j).Left And señali(i).Visible = True
And señald(j).Visible = True) Then
          s = False
          texto = "Entre Eq. " & i + 1 & " y Eq. " & j + 1
        End If

```

```

If (señali(i).Left - 100 < señald(0).Left And señali(i).Visible = True And
señald(0).Visible = True) Then
    s = False
    texto = "Entre Eq. " & i + 1 & " y Eq. 1"
End If
    If (señali(i).Left - 100 < señald(1).Left And señali(i).Visible = True
And señald(1).Visible = True) Then
        s = False
        texto = "Entre Eq. " & i + 1 & " y Eq. 2"
    End If
        If (señali(i).Left - 100 < señald(2).Left And señali(i).Visible = True
And señald(2).Visible = True) Then
            s = False
            texto = "Entre Eq. " & i + 1 & " y Eq. 3"
        End If
    End If
End If
Next j
Next i
For i = 4 To 7 Step 1
    For j = i + 1 To 7 Step 1
        If (i <> j) Then
            If (señali(i).Left - 100 < señald(j).Left And señali(i).Visible = True And
señald(j).Visible = True) Then
                s = False
                texto = "Entre Eq. " & i + 1 & " y Eq. " & j + 1
            End If
            If (i <= 4) Then
                If (señali(i).Left - 100 < señald(2).Left And señali(i).Visible = True And
señald(2).Visible = True) Then
                    s = False
                    texto = "Entre Eq. " & i + 1 & " y Eq. 3"
                End If
            End If
            If (i <= 5) Then
                If (señali(i).Left - 100 < señald(1).Left And señali(i).Visible = True And
señald(1).Visible = True) Then
                    s = False
                    texto = "Entre Eq. " & i + 1 & " y Eq. 2"
                End If
            End If
        End If
    End If
End For
End For

```

```

If (i <= 6) Then
    If (señali(i).Left - 100 < señal(0).Left And señali(i).Visible = True And
señal(0).Visible = True) Then
        s = False
        texto = "Entre Eq. " & i + 1 & " y Eq. 1"
        End If
    End If
    End If
    Next
Next i
Loop Until (s = False)
If (s1 = False) Then
    BarEstado.Panels(2).Text = "Se ha producido colisión"
    v = MsgBox("Colision " & texto, vbCritical, "CSMA/CD Bus")
    BarEstado.Panels(1).Text = "Inactivo"
    BarEstado.Panels(2).Text = ""
    For j = 0 To 7 Step 1
        señali(j).Visible = False
        señal(j).Visible = False
        señali(j).Left = Line2(j).X1 - 150
        señal(j).Left = Line2(j).X1 + 105
        Info(j).Visible = False
        If (j < 4) Then
            Info(j).top = Line2(j).Y1 + 150
        Else
            Info(j).top = Line2(j).Y2 - 200
        End If
    Next j
Else
    mover.Enabled = False
    For j = 0 To 7 Step 1
        señali(j).Visible = False
        señal(j).Visible = False
        señali(j).Left = Line2(j).X1 - 150
        señal(j).Left = Line2(j).X1 + 105
        Info(j).Visible = False
        If (j < 4) Then
            Info(j).top = Line2(j).Y1 + 150
        Else

```

```
End If
  Next j
Else
  mover.Enabled = False
  For j = 0 To 7 Step 1
    señali(j).Visible = False
    señald(j).Visible = False
    señali(j).Left = Line2(j).X1 - 150
    señald(j).Left = Line2(j).X1 + 105
    Info(j).Visible = False
    If (j < 4) Then
      Info(j).top = Line2(j).Y1 + 150
    Else
      Info(j).top = Line2(j).Y2 - 200
    End If
  Next j

  If (OpcionCs = 1) Then
    For i = 0 To 7 Step 1
      Info(i).FillColor = &HFF00FF
      señali(i).FillColor = &HFF00FF
      señald(i).FillColor = &HFF00FF

      Next i

      Call Transmitir

    End If
  End If

End Sub
```

FUNCIÓN moverseñal	
Esta función se encarga hacer mover la señal desde los equipos al bus	
Variables	Definición
Paso	Variable de tipo booleano que sirve como swiche para saber se pasa por mover señal cuando se transmite la información.
OpcionCs	Variable global de tipo byte que sirve para guardar la simulación a realizar.
Desde	Variable de tipo byte que sirve para guardar el equipo desde donde se va enviar la informació.
Hasta	Variable de tipo byte que sirve para guardar el equipo hasta donde se va enviar la información.
mov	Variable de tipo booleano para saber si se pasa por determinada parte del programa.
Código Fuente de la Función	

```
Sub moverseñal()
```

```
  If (OpcionCs = 0) Then
```

```
    BarEstado.Panels(2).Text = "Escuchando el canal"
```

```
    For i = 0 To 7 Step 1
```

```
      If (Equipos(i).Value = 1) Then
```

```
        Info(i).Visible = True
```

```
      End If
```

```
    Next i
```

```
  ElseIf (OpcionCs = 1) Then
```

```
    If (Paso = True) Then
```

```
      BarEstado.Panels(2).Text = "Transmitiendo Información"
```

```
      BarEstado.Refresh
```

```
    Else
```

```
      BarEstado.Panels(2).Text = "Escuchando el canal"
```

```
    End If
```

```
    i = CByte(Mid(desde2.Text, 4, 1)) - 1
```

```
    Info(i).Visible = True
```

```
If (Paso = True) Then
  For i = 0 To 7 Step 1
    Info(i).FillColor = &HFF00FF
    señali(i).FillColor = &HFF00FF
    señald(i).FillColor = &HFF00FF
  Next i
End If
End If
Mov = True
For i = 0 To 7 Step 1
  If (i = 0 Or i = 1 Or i = 2 Or i = 3) Then
    If (Info(i).top < 2100) Then
      If (Info(i).Visible = True) Then
        Info(i).top = Info(i).top + 100
      End If
    Else
      If (Info(i).Visible = True) Then
        Info(i).Visible = False
        señali(i).Visible = True
        señald(i).Visible = True
        Mov = False
      End If
    End If
  Else
    If (Info(i).top > 2200) Then
      If (Info(i).Visible = True) Then
        Info(i).top = Info(i).top - 100
      End If
    Else
      If (Info(i).Visible = True) Then
        Info(i).Visible = False
        señali(i).Visible = True
        señald(i).Visible = True
        Mov = False
      End If
    End If
  End If
End If
```

```
End If
Next i
If (Mov = False) Then
    mover.Enabled = False
End If
If (mover.Enabled = False) Then
    If (Paso = False) Then

        For i = 0 To 7
            señali(i).Visible = False
            señald(i).Visible = False
        Next i
        Call Transmitir

    Else
        For i = 0 To 7 Step 1
            If (Equipos(i).Value = 1) Then
                desde = i
                Exit For
            End If
        Next i
        desde = CByte(Mid(desde2.Text, 4, 1)) - 1
        Randomize
        Do
            aleatorio = Int(Rnd * 8)
            s1 = False
            For i = 0 To 7 Step 1
                If (aleatorio = EquiposActivos(i)) Then
                    s1 = True
                End If
            Next i
            If (aleatorio <> desde And s1 = True) Then
                s = True
            End If
        Loop Until (s = True)
        hasta = aleatorio
        hasta = CByte(Mid(equiposac.Text, 4, 1)) - 1
```

```
If (Line2(hasta).X1 > Line2(desde).X1) Then
    seÑald(desde).FillColor = Info(desde).FillColor
    seÑali(desde).Visible = True
    seÑald(desde).Visible = True
    movhasta.Enabled = True
    MovinfoenBus.Enabled = True

Else
    If (Line2(hasta).X1 < Line2(desde).X1) Then
        seÑali(desde).FillColor = Info(desde).FillColor
        seÑald(desde).Visible = True
        seÑali(desde).Visible = True
        movhasta.Enabled = True
        MovinfoenBus.Enabled = True

    End If
End If
*****fin

End If
End If
End Sub
```

Topología Estrella

FUNCIÓN Colisión	
Esta función se encarga de activar los timer rcolisión que detecta cuando hay colisión entre equipos.	
Variables	Definición
Vec(8)	Vector de tipo byte que sirve para guardar que equipos estan activos.
Código Fuente de la Función	

```
Sub Colision()  
    BarEstado.Panels(1).Text = "Simulando..."  
    Dim vec(8) As Byte  
    For i = 0 To 8 Step 1  
        vec(i) = 0  
    Next i  
    For i = 0 To 7 Step 1  
        If (Equipos(i).Value = 1) Then  
            vec(i) = 1  
        End If  
    Next i  
    For i = 0 To 7 Step 1  
        If (vec(i) = 1) Then  
            señal(i).Visible = True  
        End If  
    Next i  
    Rcolision.Enabled = True  
    Rcolision.Interval = 200  
    Randomize  
    BarEstado.Panels(2).Text = "Peticon de Transmision al Hub"  
    For i = 0 To 7 Step 1  
        If (vec(i) = 1) Then  
            Moverseñal(i).Interval = Int(Rnd * 3000)  
            Moverseñal(i).Enabled = True  
        End If  
    Next i  
End Sub
```

FUNCIÓN InicializarCSMA
Esta función se encarga de activar los equipos que estan activos en la simulación asi como tambien los objetos que hacen de señal.
Código Fuente de la Función
<pre> Sub InicializarCSMA() Rcolision.Enabled = False HubEquipo.Enabled = False For j = 0 To 7 Step 1 señal(j).Visible = False señal1(j).Visible = False moverseñal(j).Enabled = False Next j pantalla(1).Refresh For i = 0 To 7 Step 1 If (Equipos(i).Enabled = True) Then If (Equipos(i).Value = 0) Then PC1(i).Picture = LoadPicture("Active.ico") bl(i).ToolTipText = "PC " & i + 1 Else PC1(i).Picture = LoadPicture("Server.ico") bl(i).ToolTipText = "PC " & i + 1 End If PC1(i).ToolTipText = "Equipo Activo" bl(i).Visible = True lin(i).Visible = True CsamaEstrella.Refresh Beep End If Next i </pre>

```
señal(0).top = Line1(0).Y2
  señal1(0).top = Line1(0).Y1
señal(1).top = Line1(1).Y2
  señal1(1).top = Line1(1).Y1
señal(2).Left = Line1(2).X2
  señal1(2).Left = Line1(2).X1
señal(3).Left = Line1(3).X2
  señal1(3).Left = Line1(3).X1
señal(4).top = Line1(4).Y1
  señal1(4).top = Line1(4).Y2
señal(5).top = Line1(5).Y1
  señal1(5).top = Line1(5).Y2
señal(6).Left = Line1(6).X1
  señal1(6).Left = Line1(6).X2
señal(7).Left = Line1(7).X1
  señal1(7).Left = Line1(7).X2
  CsmaEstrella.Refresh
End Sub
```

OBJETO HubEquipo
Objeto de tipo timer que se encarga de mover la señal desde el hub hasta los equipos.
Código Fuente de la Función

```
Private Sub HubEquipo_Timer()  
    k = 1.714285714286  
    CsmasEstrella.Refresh  
    For i = 0 To 7 Step 1  
        señal1(i).Visible = True  
        If (i = 0 Or i = 1) Then  
            If (señal1(i).top > Line1(i).Y2 + 100) Then  
                señal1(i).top = señal1(i).top - 100  
            Else  
                HubEquipo.Enabled = False  
                señal1(i).Visible = False  
                señal1(i).top = Line1(i).Y1  
            End If  
        ElseIf (i = 2 Or i = 3) Then  
            If (señal1(i).Left < Line1(i).X2 - 100) Then  
                señal1(i).Left = señal1(i).Left + 100 * k  
            Else  
                HubEquipo.Enabled = False  
                señal1(i).Visible = False  
                señal1(i).Left = Line1(i).X1  
            End If  
  
        ElseIf (i = 4 Or i = 5) Then  
            If (señal1(i).top < Line1(i).Y1 - 100) Then  
                señal1(i).top = señal1(i).top + 100  
            Else  
                HubEquipo.Enabled = False  
                señal1(i).Visible = False  
                señal1(i).top = Line1(i).Y2  
            End If  
        End If  
    End For  
End Sub
```

```
ElseIf (i = 6 Or i = 7) Then
  If (señal1(i).Left > Line1(i).X1 + 100) Then
    señal1(i).Left = señal1(i).Left - 100 * k
  Else
    HubEquipo.Enabled = False
    señal1(i).Visible = False
    señal1(i).Left = Line1(i).X2
    If (OpcionCs = 1) Then
      For k = 0 To 7 Step 1
        señal(k).FillColor = &HFF00FF
        'señal1(k).FillColor = &HFF00FF
      Next k
      'señal(CByte(Mid(desde2.Text, 4, 1)) - 1).FillColor = &HFF00FF
      'señal1(CByte(Mid(equiposact.Text, 4, 1)) - 1).FillColor = &HFF00FF
      Transmission.Enabled = True
    End If
  End If
End If

End If
Next i
Pausa (0.1)
End Sub
```

OBJETO moverseñal
Objeto de tipo timer que se encarga de move la señal los equipos al hub para escuchar el canal.
Código Fuente de la Función
<pre> Private Sub moverseñal_Timer(Index As Integer) señal(Index).Visible = True If (Index = 0 Or Index = 1) Then If (señal(Index).top < Line1(Index).Y1 - 100) Then señal(Index).top = señal(Index).top + 100 Pausa (0.1) Else moverseñal(Index).Enabled = False señal(Index).Visible = False señal(Index).top = Line1(Index).Y2 HubEquipo.Enabled = True HubEquipo.Interval = 201 End If ElseIf (Index = 2 Or Index = 3) Then If (señal(Index).Left > Line1(Index).X1 + 100) Then señal(Index).Left = señal(Index).Left - 100 Pausa (0.1) Else moverseñal(Index).Enabled = False señal(Index).Visible = False señal(Index).Left = Line1(Index).X2 HubEquipo.Enabled = True HubEquipo.Interval = 201 End If </pre>

```
ElseIf (Index = 4 Or Index = 5) Then
  If (señal(Index).top > Line1(Index).Y2 + 100) Then
    señal(Index).top = señal(Index).top - 100
    Pausa (0.1)
  Else
    moverseñal(Index).Enabled = False
    señal(Index).Visible = False
    señal(Index).top = Line1(Index).Y1
    HubEquipo.Enabled = True
    HubEquipo.Interval = 201
  End If

  ElseIf (Index = 6 Or Index = 7) Then
    If (señal(Index).Left < Line1(Index).X2 - 100) Then
      señal(Index).Left = señal(Index).Left + 100
      Pausa (0.1)
    Else
      moverseñal(Index).Enabled = False
      señal(Index).Visible = False
      señal(Index).Left = Line1(Index).X1
      HubEquipo.Enabled = True
      HubEquipo.Interval = 201
    End If

  End If

End Sub
```

OBJETO Transmision	
Objeto de tipo timer que se encarga de transmitir la información de un equipo a otro.	
Variables	Definición
Desde	Variable de tipo byte que sirve para guardar desde que equipo se va a enviar la información.
Hasta	Variable de tipo byte que sirve para guardar hasta que equipo se va a enviar la información.
Código Fuente de la Función	

```

Private Sub Transmision_Timer()
Dim desde As Byte
Dim hasta As Byte

desde = CByte(Mid(desde2.Text, 4, 1)) - 1
hasta = CByte(Mid(equiposac.Text, 4, 1)) - 1
Label2(45).Caption = "3: PC " & desde + 1
Label2(46).Caption = "4: PC " & hasta + 1
*****
Frame2.top = PosY
Frame2.Left = PosX
Frame1(0).Visible = True
Me.Refresh
*****
aleatorio = hasta
BarEstado.Panels(2).Text = "Transmision de Informacion desde PC " & desde +
1 & " hasta PC " & hasta + 1
If (desde = 0 Or desde = 1) Then
    señal(desde).Visible = True
    For i = Line1(desde).Y2 To Line1(desde).Y1 Step 100
        señal(desde).top = i
        Pausa (0.3)
    Next i
    señal(desde).Visible = False
    señal(desde).FillColor = &HFFFF&
    señal(desde).top = Line1(desde).Y2
ElseIf (desde = 2 Or desde = 3) Then
    señal(desde).Visible = True
    For i = Line1(desde).X2 To Line1(desde).X1 Step -100
        señal(desde).Left = i
        Pausa (0.3)
    Next i
    señal(desde).Visible = False
    señal(desde).FillColor = &HFFFF&
    señal(desde).Left = Line1(desde).X2

```

```

ElseIf (desde = 4 Or desde = 5) Then
    señal(desde).Visible = True
    For i = Line1(desde).Y1 To Line1(desde).Y2 Step -100
        señal(desde).top = i
        Pausa (0.3)
    Next i
    señal(desde).Visible = False
    señal(desde).FillColor = &HFFFF&
    señal(desde).top = Line1(desde).Y1
    Transmision.Enabled = False
    ElseIf (desde = 6 Or desde = 7) Then
        señal(desde).Visible = True
        For i = Line1(desde).X1 To Line1(desde).X2 Step 100
            señal(desde).Left = i
            Pausa (0.3)
        Next i
        señal(desde).Visible = False
        señal(desde).FillColor = &HFFFF&
        señal(desde).Left = Line1(desde).X1
    End If
    *****
    If (hasta = 0 Or hasta = 1) Then
        señal1(hasta).FillColor = señal(hasta).FillColor
        señal1(hasta).Visible = True
        For i = Line1(hasta).Y1 To Line1(hasta).Y2 Step -100
            señal1(hasta).top = i
            Pausa (0.3)
        Next i
        señal1(hasta).Visible = False
        señal1(hasta).top = Line1(hasta).Y1
        For k = 0 To 7 Step 1
            señal1(k).FillColor = &HFF&
            señal(k).FillColor = &HFFFF&
        Next k
        señal1(hasta).FillColor = &HFF&
        Transmision.Enabled = False
    ElseIf (hasta = 2 Or hasta = 3) Then
        señal1(hasta).FillColor = señal(hasta).FillColor
        señal1(hasta).Visible = True

```

```

For k = 0 To 7 Step 1
    señal1(k).FillColor = &HFF&
    señal(k).FillColor = &HFFFF&
Next k
señal1(hasta).FillColor = &HFF&
Transmission.Enabled = False
ElseIf (hasta = 2 Or hasta = 3) Then
    señal1(hasta).FillColor = señal(hasta).FillColor
    señal1(hasta).Visible = True
    For i = Line1(hasta).X1 To Line1(hasta).X2 Step 100
        señal1(hasta).Left = i
        Pausa (0.3)
    Next i
    señal1(hasta).Visible = False
    señal1(hasta).Left = Line1(hasta).X1
    For k = 0 To 7 Step 1
        señal1(k).FillColor = &HFF&
        señal(k).FillColor = &HFFFF&
    Next k

    señal1(hasta).FillColor = &HFF&
    Transmission.Enabled = False
    ElseIf (hasta = 4 Or hasta = 5) Then
        señal1(hasta).FillColor = señal(hasta).FillColor
        señal1(hasta).Visible = True
        For i = Line1(hasta).Y2 To Line1(hasta).Y1 Step 100
            señal1(hasta).top = i
            Pausa (0.3)
        Next i
        señal1(hasta).Visible = False
        señal1(hasta).top = Line1(hasta).Y2
        For k = 0 To 7 Step 1
            señal1(k).FillColor = &HFF&
            señal(k).FillColor = &HFFFF&
        Next k

```

```
señal1(hasta).FillColor = &HFF&
    Transmision.Enabled = False
    ElseIf (hasta = 6 Or hasta = 7) Then
        señal1(hasta).FillColor = señal(hasta).FillColor
        señal1(hasta).Visible = True
        For i = Line1(hasta).X2 To Line1(hasta).X1 Step -100
            señal1(hasta).Left = i
            Pausa (0.3)
        Next i
        señal1(hasta).Visible = False
        señal1(hasta).Left = Line1(hasta).X2
        For k = 0 To 7 Step 1
            señal1(k).FillColor = &HFF&
            señal(k).FillColor = &HFFFF&
        Next k
        señal1(hasta).FillColor = &HFF&
        Transmision.Enabled = False
    End If
    ***
    Frame2.top = 4635
    Frame2.Left = 3690
    Frame1(0).Visible = False
    Me.Refresh
    ***
    Label2(45).Caption = "3:"
    Label2(46).Caption = "4:"
    BarEstado.Panels(1).Text = "Inactivo"
    BarEstado.Panels(2).Text = ""
End Sub
```

**HERRAMIENTA DE SIMULACION DE PROTOCOLOS DE ACCESO AL
MEDIO EN TOPOLOGÍAS DE RED DE ÁREA LOCAL**

VICKY XIMENA BONILLA SILVA

ESPERANZA CLARET ROMERO ABAD

**CORPORACION UNIVERSITARIA TECNOLOGICA DE BOLIVAR
FACULTAD DE INGENIERIA DE SISTEMAS**

CARTAGENA DE INDIAS D.T.H. T C.

1999

**HERRAMIENTA DE SIMULACIÓN DE PROTOCOLOS DE ACCESO AL MEDIO
EN TOPOLOGÍAS DE RED DE ÁREA LOCAL**

**VICKY XIMENA BONILLA SILVA
ESPERANZA CLARET ROMERO ABAD**

**Trabajo de Grado presentado como requisito parcial para optar el título de
Ingenieros de Sistemas**

**Director:
CARLOS FERRIOL
Lic. en Ciencias Computacionales**

**Asesor:
MOISES QUINTANA
Lic. en Ciencias Computacionales**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA DE SISTEMAS**

CARTAGENA DE INDIAS D.T.H. Y C.

1999

TITULO

**HERRAMIENTA DE SIMULACION DE PROTOCOLOS DE ACCESO AL MEDIO
EN TOPOLOGÍAS DE RED DE ÁREA LOCAL**

AUTORES

BONILLA SILVA, Vicky Ximena

ROMERO ABAD, Esperanza Claret

Fecha de terminación: Diciembre de 1999

Estudiantes de la Facultad de Sistemas

DESCRIPCION

Desarrollar una herramienta de simulación de protocolos de acceso al medio en topología de red de área local, que brinde al estudiante la posibilidad de contar con una herramienta de software en una red de computadoras de la universidad, en la cual puedan realizar prácticas en las que observarán los pasos o reglas en que se fundamentan los protocolos para asignar un recurso.

METODOLOGIA

Para el desarrollo de este software se llevo a cabo una exhaustiva investigación acerca de todos los temas relacionados con el funcionamiento de los protocolos de acceso al medio; se estudió, analizó y seleccionó los aspectos más

importantes que nos permitieran dar paso a la implementación de la herramienta de simulación.

Después de un estudio de factibilidad se escogió como lenguaje de programación Visual Basic; ya que cuenta con las herramientas necesarias de soporte para nuestro software, además que posee manejo de sockets, lo cual es un aspecto fundamental en la aplicación cliente/servidor que implementamos.

RESULTADOS

Los logros a los que conlleva la utilización de éste programa de simulación para el docente son contar con una herramienta que le facilite el proceso de enseñanza de los protocolos de comunicación, así como de dinamizar las clases. A su vez, permite a los estudiantes ampliar los conocimientos relacionados con este tema y hacer comparaciones del rendimiento de cada protocolo en su respectiva topología; además los motiva a desarrollar nuevos proyectos que tengan una finalidad académica como ésta.

Al culminar nuestro trabajo de grado, se entregará a la Corporación Universitaria Tecnológica de Bolívar, una herramienta en la cual se simula el funcionamiento de los protocolos de acceso al medio, dentro de las topologías de área local Bus, Anillo y Estrella; por medio de la cual tanto docentes como estudiantes facilitarán la enseñanza y aprendizaje en las áreas de Comunicaciones y Redes. Junto con ésta herramienta de apoyo se entregará:

- ❖ Documento final del proyecto.
- ❖ Manual de usuario y del sistema.
- ❖ Librerías y software debidamente documentados.

GLOSARIO

Acceso Método de: Este es un protocolo que determina qué dispositivo en una red de área local tiene acceso al medio de transmisión en cualquier momento. CSMA/CD es un ejemplo de un método de acceso. IBM utiliza el mismo término para tipos específicos de "software" de comunicación que incluyen protocolos para el intercambio de datos, construcción de archivos y otras funciones.

ACK o Acknowledge: Un carácter o secuencia de caracteres enviados por el receptor de un mensaje para notificar al emisor, que el último mensaje se recibió correctamente. ACK también se envía como una respuesta de "adelante" a secuencias de selección.

Ancho de banda (bandwidth): es el rango (las frecuencias comprendidas entre dos límites) de las frecuencias que se pueden pasar a través de un canal de comunicación.

ANSI: American National Standard Institute; Organización no gubernamental donde sus miembros apoyan, diseñan, adoptan y generan estándares en los Estados Unidos, aunque a veces muchos otros países también los adoptan.

ArcNet: Sistema de Red de área local (LAN) desarrollada por Datapoint. Utiliza las técnicas de pasa fichas ("token") pero no es una anillo ("ring") sino que sigue la topología física de estrella y permite un máximo de 256 nodos en la red. Datapoint licenció la tecnología ARCNET para la Tandy Corporation, Davon (fuera del mercado actualmente), Novell y Standard Microsystems. Es una de las redes locales más baratas actualmente. Su velocidad máxima actual es de 2.5 Mbps.

Banda: Se refiere al rango de frecuencias entre dos límites, uno límite bajo y otro alto. Por ejemplo, se dice que la banda de voz es entre 300 y 20,000 hz.

Baseband LAN (LAN de banda base): LAN que usa una sola frecuencia portadora sobre un solo canal. Ethernet, Token Ring y Arcnet usan transmisión de banda base.

Binario: Es el nombre del sistema numérico de base 2.

Bit (bitio): La unidad más pequeña de información para el proceso de datos. Un bit (o dígito binario) asume el valor de 1 o 0.

Bps: Bits por segundo, unidades de velocidad de transmisión.

Bridge (puente): Dispositivo de red que conecta dos LAN's y remite o filtra paquetes de datos entre ellas, según sus direcciones de destino. Los puentes operan al nivel de enlace de datos (o capa MAC) del modelo de referencia OSI, y es transparente a los protocolos y a los dispositivos de niveles más altos como los routers.

Broadband (banda ancha): Técnica de transmisión de datos que permite que múltiples señales de alta velocidad compartan el ancho de banda de un solo cable mediante la multiplexación por división de frecuencias.

Broadband Network (red de banda ancha): Red que usa múltiples frecuencias portadoras para transmitir señales multiplexadas en un solo cable. Varias redes pueden coexistir en un solo cable sin interferir entre ellas.

Broadcast: Se refiere al mensaje que se envía a todas las estaciones en una conexión lógica ("data link") multipunto.

Router: Dispositivo que enruta protocolos específicos, como TCP/IP e IPX, y remite otros protocolos, combinando las funciones de routers y puentes.

Byte: Unidad de datos de ocho bits.

Cabecera: La parte inicial de un paquete de datos o trama conteniendo información de identificación como la fuente de los datos, su destino, y longitud.

Cable: Conducto que conecta dispositivos de la red entre sí. El tipo de cable a utilizar depende del tamaño de la red y la topología de la misma.

Cable Coaxial: Cable eléctrico con conductor de alambre sólido en el centro rodeado por materiales aislantes y un conductor como pantalla de metal exterior con un eje de curvatura que coincide con el conductor interno - de ahí que se denomine "coaxial". Ejemplos son el cable Ethernet normal (grueso) y el Thinwire (el cable de Ethernet fino).

Cable de fibra-óptica: Medio de transmisión compuesto de un cable de vidrio central, rodeado por malla y una funda protectora exterior. Transmite signos digitales en forma de luz modulada por un láser o diodo (light-emitting diode).

Cable de par trenzado: Cable de múltiples conductores, de bajo coste, formado por uno o más pares de 18 a 24 AWG. Los pares se trenzan para mejorar la protección contra interferencias electromagnéticas y de radiofrecuencia. Este cable que puede ser apantallado o sin apantallar, se usa en comunicaciones de

baja velocidad, como cableados telefónicos. Sólo se emplea en redes de banda base debido a su estrecho ancho de banda.

Canal: El camino de los datos entre dos nodos.

Capa de Enlace de Datos: Capa 2 de la siete capas del modelo de referencia OSI para la comunicación entre ordenadores en redes. Esta capa define los protocolos para los paquetes de datos y cómo se transmiten hacia/desde cada dispositivo de la red. Es un nivel de enlace de comunicaciones independiente del medio, situada por encima de la capa Física, y esta dividida en dos subcapas: control de acceso al medio (MAC o Medium Access Control) y control del enlace lógico (LLC o Logical Link Control).

Capa física: La capa 1, la inferior del modelo OSI, implementada por el canal físico. La capa Física aísla la capa 2, la capa de enlace de datos, de las características físicas dependientes del medio como transmisión en banda base, banda ancha o fibra óptica. La capa 1 define los protocolos que gobiernan los medios de transmisión y las señales.

Capas: Se refiere a la organización por estratos jerárquicos de ciertos modelos de comunicación y sus protocolos. Cada capa o estrato se comunica exclusivamente con las capas vecinas. Lo que facilita su organización y estructuración modular. La capa más baja o capa FÍSICA, hace interface con el "hardware" de la computadora. La capa de APLICACIÓN es la séptima (y más alta) capa del modelo OSI, que interactúa con los programas de aplicación o con un usuario.

Carrier (Portadora): Señal que se usa para "acarrear" o transportar señales de banda base sobre un medio de comunicación. La frecuencia de la portadora es usualmente más alta que la frecuencia de la señal de banda base.

Caudal de información: Número total de bits de información transmitidos por unidad de tiempo.

Collision (colisión): El resultado de dos nodos de la red que transmiten al mismo tiempo en el mismo camino. Los datos transmitidos no son utilizables.

Collision Detect (detección de colisión): Señal indicando que una o más estaciones están conteniendo por la transmisión. El signo es enviado por la capa Física a la de Enlace de Datos en un nodo Ethernet/IEEE 802.3.

Concentrador: Dispositivo que acepta varias entradas de datos y provee una salida compuesta. Esta salida compuesta es generalmente una cadena de datos de alta velocidad, multiplexada en el tiempo, pero con identificadores antes de cada mensaje de qué dispositivo provienen.

CRC (Cyclical Redundancy Checking): Verificación de Redundancia Cíclica. Método moderno de calcular un BCS que asegura que muy pocos errores se escapan sin ser detectados. Tiene la enorme ventaja de que detecta errores cuando vienen estos seguidos ("burst").

CSMA/CD: Carrier Sense Multiple Access with Collision Detection. El acceso múltiple por detección de portadora y detección de colisión es el medio de comunicación físico de Ethernet. Todos los dispositivos se conectan a la red y contienden igualmente para transmitir. Si un dispositivo descubre el signo de otro dispositivo que está transmitiendo, aborta la transmisión y lo reintentará después de una breve pausa.

Dirección (Address): Un nombre, etiqueta, número o secuencia de bits que se usa para identificar: al receptor de un mensaje, a un dispositivo en particular en una línea multipunto, la trayectoria de una ruta, etc. Es un lugar único en la memoria, este también sirve para identificar un nodo en una red.

Dirección física: Dirección que identifica a un único nodo.

Enlace de datos: Conexión lógica entre dos nodos en el mismo circuito.

Enlace lógico: Conexión temporal entre los nodos fuente y destino, o entre dos procesos del mismo nodo.

Entramado: División de los datos para su transmisión en grupos de bits, agregándoles una cabecera y un código de verificación para formar una trama.

Estación: **Cualquier máquina que recibe o envía datos en un "data link" o a través de una red.**

Ethernet: La tecnología de LAN más popular actualmente. La norma IEEE 802.3 define las reglas para configurar una red Ethernet. Es una red CSMA/CD de banda base a 10 Mbps., que funciona con cableado coaxial fino y grueso, par trenzado y fibra óptica.

Frame: Enmarcado. Procedimiento mediante el cual un protocolo le añade a los datos originales un encabezado ("header") y una cola ("trailer"). En los protocolos

de bits se refieren a los bloques de datos como "frames" o marcos. Marco de información (también llamado token) que se comparte por dos o más estaciones dividiendo el marco en partes iguales. Esta técnica se usa por algunos LAN's de topología de anillo.

Frecuencia: El número de ciclos de una señal de corriente alterna por unidad de tiempo.

GHZ: Giga Hertz; se refiere a billones de hertz.

Group Address: Es una dirección se usa para designar dos o más estaciones de un grupo predesignado. Por ejemplo, si una estación "host" quiere transmitir un mensaje a todas las terminales en un área de control de inventario. Los terminales de control de inventario comparten la misma dirección grupal.

Half Duplex Transmisión: Se refiere al diálogo entre dos estaciones donde ninguna estación enviará datos mientras la otra está enviando datos.

Hertz: Es la unidad de frecuencia. Es el término preferido para referirse a las frecuencias en lugar de usar el antiguo "ciclos por segundo".

Hub: Concentrador. Dispositivo que se utiliza típicamente en topología en estrella como punto central de una red, donde por ende confluyen todos los enlaces de los diferentes dispositivos de la red.

IEEE (Institute of Electrical and Electronic Engineers): Instituto de Ingenieros Eléctricos y Electrónicos. Es una entidad que ha generado muchísimos estándares en telecomunicaciones.

IEEE802: Estos son los estándares para la conexión física y eléctrica de LAN's desarrollado por IEEE (Institute of Electronic and Electrical Engineers).

IEEE 802.1D: Estándar de la IEEE para el nivel de acceso de control para puentes o "bridges" interLAN , entrelazando redes IEEE 802.3, 802.4 y 802.5.

IEEE 802.2: Estándar para la capa de conexión lógica, para usarse con redes IEEE 802.3, 802.4 y 802.5.

IEEE 802.3: La norma del IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) que define el método de acceso al medio CSMA/CD y las especificaciones de las capas físicas y de datos de una área local. Entre otros, incluye las aplicaciones Ethernet 10BASE-2, 10BASE-5, 10BASE-FL y 10BASE-T.

IEEE 802.3 1Base5: Especificación de la IEEE que iguala el antiguo producto de AT&T StarLAN. Este designa un rate de 1 mbps, técnica de base de banda y un máximo de distancia de cable de 500 metros.

IEEE 802.3 10Base2: Esta especificación de la IEEE iguala el cableado estrecho de Ethernet. Este designa un rate de señal de 10 mbps, técnica de base de banda, y un máximo de distancia de cable de 185 (casi 200) metros.

IEEE 802.3 1Broad36: En esta especificación de la IEEE se describe un tipo de cableado de la Ethernet pero de larga distancia con un rate de 10 megabit por segundo y una distancia de cable de 3,600 metros.

IEEE 802.4: Aquí se describe un LAN que usa un rate de señales de 10 megabits por segundo, control de acceso para "token-passing" y una topología de bus física. Este es típicamente usado como parte de redes que siguen a MAP (Manufacturing Automation Protocol) desarrollado por General Motors. Este es a veces confundido con ARCnet pero no es el mismo.

IEEE 802.5: Esta especificación de la IEEE describe un LAN que usa 4 o 16 megabits por segundo, MAC "token-passing" y una topología física de anillo. Es utilizado por los sistemas IBM de Token-Ring.

IEEE 802.6: Este estándar de la IEEE para MAN's describe lo que se llama un "Distributed Queue Dual Bus" (DQDB). La topología DQDB incluye cableado paralelo-típicamente de fibra óptica-entrelazando cada nodo (típicamente un router para un segmento de LAN) utilizando rates de señales de 100 megabits por segundo.

Impedancia: Propiedad eléctrica de un cable, combinando capacidad, instalación y resistencia y se mide en "ohms".

Interface: Un interface provee los medios para la interconexión de equipo (o procesos) localizados en un lugar específico.

Internetworking: Término general empleado para describir a la industria dedicada a productos y tecnologías usados para crear redes.

ISO: "International Standards Organization". Organismo de las Naciones Unidas, con sede en París, cuya misión es el generar y difundir estándares entre las naciones, logrando así la compatibilidad y complementariedad en servicios y productos internacionalmente. Desarrolló el modelo de comunicación abierta OSI.

LAN: Local Area Network o Red de Área local. Sistema de comunicación de datos que consiste en un grupo de ordenadores interconectados, compartiendo aplicaciones, datos y periféricos. El área geográfica normalmente es un edificio o grupo de edificios.

Latencia: El retraso en el que incurre un conmutador o puente entre la recepción de la trama y su remisión.

Local Network Interconnect - LNI (Interconector local de red): Multiplicador de puertos, o concentrador que soporta varios dispositivos activos o controladores de comunicaciones, independientes o conectados mediante el cable Ethernet normalizado.

MAU: Medium Attachment Unit o Unidad de Conexión al Medio. Dispositivo usado para convertir señales de un medio Ethernet a otro.

Mbps.: Megabits por segundo.

Mensaje: Cualquier cadena de caracteres, bytes, etc. que sea enviado de un dispositivo a otro sobre una conexión "data link".

Método de acceso: Este es un método software/hardware de transferencia de información entre programas de aplicación residentes y equipo a larga distancia. Este término se usa para describir los programas de comunicación que residen en una computadora "mainframe".

MNP: Siglas en inglés para Microcom Networking Protocol. Este es un protocolo de microcomputadoras desarrollado por Microcom. MNP usa técnicas modernas similares a las de HDLC.

Modelo de capas ISO: La Organización de Normas Internacionales (ISO) fija las normas para los ordenadores y las comunicaciones. Su modelo de referencia Open Systems Interconnection (OSI - Interconexión de Sistemas Abiertos) especifica cómo dispositivos informáticos diferentes, como Tarjetas de Interfaz de Red (NICs), puentes y encaminadores, intercambian datos en una red. El modelo consiste en siete capas. De la más baja a la más alta, son: Física, Enlace de Datos, Red, Transporte, Sesión, Presentación y Aplicación. Cada capa realiza servicios para la capa situada sobre ella.

Modulación: Es el proceso de modificación de algunas características de la onda portadora de acuerdo con valores puntuales de la información a ser transmitida.

Monitor activo: En una red "token-ring", solamente una máquina es el monitor activo. Esta máquina es la responsable de iniciar una ficha, se encarga de

mantener el tiempo y los retrasos del anillo y de recuperarse de ciertos errores de la red. Cualquier maquina en la red puede llevar a cabo todas estas funciones si el monitor activo llegara a fallar.

Multicast: Mensaje multicast es el que se envía a múltiples dispositivos de la red desde un servidor.

Network (red): Sistema de ordenadores interconectados que pueden comunicarse entre sí y compartir archivos, datos y recursos.

NIC: Network Interface Card o Tarjeta de Interfaz de red. Tarjeta adaptadora que se inserta en un ordenador, y contiene la electrónica y el software necesarios para permitir a la estación comunicarse a través de la red. O sea, los cables de la red se conectan a la computadora a través de las tarjetas de Interface de red.

Nodo: Cualquier dispositivo inteligente conectado a la red. Esto incluye servidores de terminales, servidores, y cualquier otro dispositivo (como impresoras y terminales) que se conectan directamente a la red. Se puede decir que un nodo es cualquier dispositivo que tiene una "dirección de hardware".

Paquete: Serie de bits que contienen datos e información de control, incluyendo la dirección del nodo fuente y destino, estructurados para su transmisión de un nodo a otro.

Par trenzado: Cable similar a los pares telefónicos estándar, que consiste en dos cables aislados "trenzados" entre sí y encapsulados en plástico. Los pares aislados vienen en dos formas: cubiertos y descubiertos.

Portadora controlada: Es una característica de un módem que permite a la señal portadora del módem ser encendida o apagada bajo el mandato del DTE. Esto es necesario en las líneas multipunto.

Protocolo: Un conjunto de reglas formales que describen como se transmiten los datos, especialmente a través de la red. Los protocolos de bajo nivel definen los estándares eléctricos y físicos que deben observarse, mientras que los protocolos de alto nivel definen los formatos de los datos.

Protocolo de acceso: Estas son las reglas de tráfico a las que se sostienen estaciones de trabajo LAN para evitar la colisión de datos cuando se envían señales a través de un medio de red compartido. También conocido como MAC o "Media Acces Control protocol". Ejemplos comunes de esto lo es el CSMA o "carrier sense multiple access".

Puerto: El conector físico de un dispositivo que permite hacer la conexión.

Punto a punto: Circuito que únicamente conecta dos nodos, o una configuración que requiere una conexión física separada entre cada par de nodos.

Red: Disposición de equipos de computación de comunicaciones y líneas de transmisión que permite el enfoque del conjunto como un sistema de procesamiento de datos con características definidas.

Rendimiento: Cantidad de información útil que la red es capaz de transportar en relación con la cantidad de bits transportados realmente.

Repetidor: Un dispositivo que intensifica las señales de la red. Los repetidores se usan cuando el largo total de los cables de la red es mas largo que el máximo permitido por el tipo de cable. No en todos los casos se pueden utilizar.

Retardo: Tiempo medio de transferencia de paquetes entendido como el intervalo de tiempo medio desde la generación de un paquete en la estación origen hasta su total recepción en el destino.

Router – Ruteador: Dispositivo que dirige el tráfico entre redes y que es capaz de determinar los caminos más eficientes, asegurando un alto rendimiento.

Servidor: Ordenador que proporciona recursos para ser compartido en la red, como archivos (servidor de ficheros) o terminales (servidor de terminales).

Servidor de comunicaciones: Sistema independiente dedicado que gestiona actividades de comunicaciones para otros ordenadores.

Sesión: Conexión a un servicio de la red.

Sobrecarga: Esta constituida por el tiempo de espera en la obtención del acceso al medio, la transmisión de preámbulos para sincronización del receptor y el tiempo de transmisión de direcciones, control de errores y otra información de control dependiente del protocolo.

Topología: La "forma" de la red. Predominan tres tipos de tecnologías: Bus, Estrella y Anillo.

Topología Anillo: Topología de red en la que los nodos se conectan en un bucle cerrado. Los datos se transmiten de nodo en nodo alrededor del bucle, siempre en la misma dirección.

Topología Bus: En una topología de Bus cada nodo se conecta a un cable común. No se requiere un hub en una red con topología de bus.

Topología Estrella: En la topología estrella, los nodos radian desde un hub. El hub o concentrador es diferente dependiendo de la tecnología utilizada Ethernet, FDDI, etc. La mayor ventaja de esta topología es que si un nodo falla, la red continúa funcionando.

Token (testigo): La secuencia de caracteres o trama, que se pasa en secuencia de nodo a nodo, para indicar que el nodo que lo controla tiene el derecho de transmitir durante una determinada cantidad de tiempo.

Transceiver (transceptor): El dispositivo real que une la red y el nodo local. El término generalmente se refiere a cualquier conector, como un MAU que activamente convierte señales entre la red y el nodo local.

Transmisión: El traslado de un archivo o información de un nodo de la red a otro. Generalmente se refiere a transferir un archivo de un servidor, como una host, a un "pequeño" nodo.

Utilización del canal: Fracción de tiempo dedicada a la transmisión de los bits de información comparada con el tiempo total de transmisión de los bits de información y los bits de sobrecarga.

UTP: Par trenzado no apantallado, uno o más pares de cable rodeados por un aislamiento. UTP normalmente se usa como cable telefónico.

WAN- Wide Area Network: Red de área amplia: Una red generalmente construida con líneas en serie que se extiende a distancias mayores a un kilómetro.