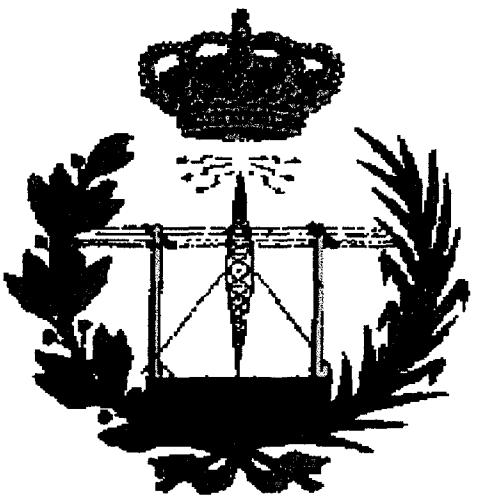
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ESCUELA UNIVERSITARIA DE

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN



PROYECTO FIN DE CARRERA

SIMULADOR X25. Terminal de usuario.

ESPECIALIDAD: Equipos Electrónicos.

AUTOR: Marcos Martín Pérez.

TUTORES: Carlos Ley Bosch, Roberto Domínguez Rodríguez.

FECHA: Septiembre 1997.

CALIFICACIÓN:

Firma Tutores Firma Alumno Firma Tribunal

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Proyecto Fin de carrera

Simulador X25

Terminal de usuario

Memoria

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Indice

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1

Introducción

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.1 Presentación

En la sociedad actual, la importancia de la información y los sistemas de tratamiento de la misma es primordial. El enorme y vertiginoso desarrollo que en los últimos años ha alcanzado la informática la colocan en puestos estratégicos de toda actividad. Paralelamente a ella, ha crecido la tecnología de la transmisión de datos, que permite la difusión y tratamiento de la información desde lugares remotos.

Entre las normas y protocolos que se han ideado para regir las comunicaciones entre dos terminales a través de una Red en un sistema abierto, destaca la recomendación X25, desarrollada por el CCITT para definir los niveles más bajos de la jerarquía OSI. Introducida a finales de los años 70, la X25 ha sufrido varias revisiones que le permiten seguir siendo en la actualidad uno de los protocolos más extendidos y de mayor vigencia.

El presente proyecto pretende ser una herramienta de trabajo para el aprendizaje de la norma X25. Para ello se ha creado y desarrollado el programa **Simulador X25**, a través del cual se pueden seguir las evoluciones de una red virtual de comunicaciones bajo casi todas las situaciones reales de trabajo. En este proyecto se han definido los algoritmos y procedimientos que sigue el programa desde el punto de vista del **Terminal** que se complementa con el proyecto en el que se estudia el fenómeno correspondiente visto desde la **Red.**

La gran complejidad de desarrollo y las enormes prestaciones con que se

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

ha dotado al **Simulador X25** han aconsejado la división del mismo en dos partes diferenciadas: El lado del Terminal de Usuario y el lado de Nodos de Red. Este proyecto estudia la primera parte, donde se han definido los procedimientos que sigue el programa para cumplir con la norma visto siempre desde el terminal. Algunas funciones y unidades son compartidas por los nodos y a la par algunas de éstos son comunes a los terminales, pero lo que es el centro de la estructura de los algoritmos de comunicación están pergectamente diferenciados y las unidades que se han implementado a tales efectos son completamente diferentes y ajenas entre sí.

Desde el lado de terminal de usuario se defienden las acciones y procedimientos que observa un ETD conectado a una Red y que son definidos a voluntad por el usuario. Se tiene la potestad de iniciar y finalizar cuantas comunicaciones se quieran e introducir los parámetros que se deseen. En todo caso corresponde a la Red asegurar que las acciones de l terminal sean lícitas.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.2 Fundamentos

Los próximos capítulos explicarán con cierto detalle los motivos por los que se ha decidido hacer el presente proyecto sobre la norma X25. Se analizará la importancia de la transmisión de datos en la sociedad actual y la importancia de los protocolos desde el punto de vista de su eficiencia, se hará una breve introducción de la norma X25 que ayude al lector a fijar su solidez, funcionamiento y vigencia y se terminará con una alusión al lenguaje de programación Delphi de Borland.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.2.1 Norma X25.

La norma X25 es un protocolo para transmisión de datos estructurada en los tres primeros niveles de la arquitectura ISO normalizada por el CCITT. Define los procedimientos a seguir por los terminales y la Red en el interfaz ETD-ETCD. Los niveles que abarca son el nivel Físico, nivel de Enlace y nivel de Red.

1.2.1.1 Transmisión de datos.

Si tomamos la palabra dato en su sentido literal de colección de hechos, fundamentos y antecedentes a los que puede asignarse un significado, el concepto de transmisión de datos tiene un amplio significado, ya que podeíamos considerar el Tam-tam o las señales de humo como transmisión de datos.

Hoy en día los términos transmisión de datos, teletratamiento o teleproceso se usan normalmente para describir el movimiento de información codificada sobre algún sistema de comunicación y cuyo destino final, de una firma directa o indirecta, es su tratamiento por parte de un ordenador.

Históricamente nos encontramos con que dos tecnologías como la informática y las telecomunicaciones, que tenían muy poco en común. empiezan a tener puntos de contacto cuando, a partir de la década de los sesenta, se consigue romper la barrera del tiempo y lograr que cierta información de la que disponen los grandes ordenadores, que procesaban centralizadamente el conjunto de las aplicaciones administrativas y científicas de una entidad, pueda llegar a puntos

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

periféricos de recogida y utilización de la misma haciendo uso de las líneas de transmisión, con lo que las compañías de telecomunicación se enfrentaron a un problema similar al existente en el comienzo de las comunicaciones telefónicas: la conexión entre dos puntos terminales.

A partir de la década de los ochenta la simbiosis de la informática y las telecomunicaciones ha llegado a un grado tal que podemos hablar de una nueva ciencia llamada telemática mediante la cual el hombre podría hacer el mejor uso posible de la información, concebida ésta como el motor de toda actividad.

Un sistema teleinformático es un conjunto de equipos informáticos para la captura, proceso, envío y visualización de datos, y la red de telecomunicaciones, constituida por circuitos y equipos de transmisión y conmutación de datos, para que el tiempo transcurrido entre el envío de información y la recepción de la misma sea el menor posible.

1.2.1.2 Protocolos orientados a bits.

El rápido crecimiento de las aplicaciones de comunicación de datos, junto con la evolución de la tecnología que proporciona ordenadores y terminales más rápidos, más potentes y más baratos, motivó la búsqueda de protocolos más eficientes que los orientados a carácter. Estos protocolos se diseñaron para cumplir una amplia variedad de requerimientos del enlace de datos, entre los que podemos mencionar

1.- Adaptación a las diversas configuraciones del enlace: punto a punto, multipunto y otras, mediante líneas conmutadas o dedicadas.

2.- Posibilidad de manejar estaciones primarias, secundarias y combinadas.

3.- Operación duplex o semiduplex. Posibilidad de conectar terminales que trabajen en estos dos modos, al mismo tiempo, al mismo enlace.

4.- Posibilidad de funcionar en enlaces de gran o pequeño tiempo de tránsito y por supuesto en enlaces de alta y baja capacidad.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Además los protocolos se diseñaron para satisfacer los siguientes objetivos:

1.- Independencia del código. Los mensajes, a nivel superior pueden estar constituidos por cualquier combinación de bits.

2.- Gran eficiencia. La relación entre bits de datos transmitidos y bits de control de protocolo debe ser alta.

3.- Elevada fiabilidad. Tanto los tramos de datos como los de control deben protegerse con métodos potentes de detección de errores.

La clave para cumplir estos requerimientos y objetivos es la utilización de una estructura de trama monoformato, con guión de apertura y cierre y campos de significación posicional.

1.2.1.3 Estructura de la norma X25

La norma X25 define los tres primeros niveles de una jerarquía OSI para transmisión de datos en modo paquete. La estructura tiene en el nivel inferior al soporte Físico, sobre el cual se incorpora el nvel de Enlace, que a su vez da sustento al nivel deRed. Se definen a continuación estos niveles.

1.2.1.3.1 **Nivel Físico.**

Se encarga de describir las características físicas eléctricas y funcionales de la interfaz entre el ETD y el equipo de transmisión de acceso a la Red. La conexión entre ETD y Red se realiza a través de uno o varios circuitos físicos punto a punto. La transmisión será síncrona y la velocidad para cada uno de los circuitos podrá elegirse en contratación con la Red. La Red proporcionará los circuitos de datos utilizando los equipos de transmisión normalizados para cada una de las velocidades contratadas.

1.2.1.3.2 **Nivel de Enlace.**

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

El Nivel 2 asegura la compatibilidad de los protocolos del nivel del enlace que proporcionan una transmisión libre de errores, así como el acceso al medio de comunicación.

El procedimiento de control de enlace X25 utiliza los principios del protocolo HDLC siendo la transmisión duplex y el modo de trabajo balanceado asíncrono. La norma HDLC tiene su origen en los estudios realizados por la British Standards Institution a finales de los 60, más tarde normalizada por la ISO.

Los objetivos del procedimiento de control de enlace son:

Asegurar la sincronización entre los dos extremos del enlace.

Detectar y recuperar los errores que aparezcan en la transmisión.

En el modo Asíncrono balanceado ambas estaciones, RED y ETD, tienen la misma capacidad de transferencia y control sobre el enlace.

1.2.1.3.3 **Nivel de Paquete o de Red.**

El Nivel 3 es el encargado de mantener las comunicaciones extremo a extremo. Para ello gestiona el establecimiento y liberación de canales, las características lógicas de la comunicación, el control de flujo y el soporte al nivel superior fundamentalmente. Todo paquete de Nivel 3 es una estructura de datos que tiene significado propio.

Cada paquete, al ser transferido a través del interfaz ETD-Red, deberá estar contenido en el campo de información de una trama info del procedimiento de enlace (Nivel 2), el cual delimitará su longitud. Unicamente un paquete podrá estar contenido en el campo de información mencionado.

La Red trabajará con llamadas virtuales y circuitos virtuales permanentes. Para ello se usan canales lógicos. A cada llamada virtual o circuito virtual permanente se le asigna un canal lógico. A los que se usan para las llamadas virtuales se les denomina conmutados, siendo permanentes el nombre que se les da a los utilizados

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

para circuitos virtuales permanentes.

1.2.1.4 ***Comentarios a la Norma X25***

Actualmente, nuevos protocolos de Red están rompiendo la barrera entre las Redes de Area Extensa y las Redes de Area Local. Protocolos como Frame Relay están aumentando considerablemente el ancho de banda disponible para el usuario basándose fundamentalmente en la mejora de los medios de transmisión. Al mejorar la calidad y la velocidad de las líneas de comunicación, en su mayoría de tecnología digital, los protocolos han reducido su control sobre los errores, ventanas y congestiones que debían tener en las capas bajas, llevando la supervisión a las capas superiores y presumiendo que la calidad de los circuitos hacen poco necesario el control exhaustivo que se tenía en X25.

Por ejemplo, si implementamos una Red de Area Extensa uniendo varias Redes de Area Local sobre Protocolo TCP/IP y encapsulamos las tramas TCP/IP en Frame Relay en la línea de unión, deberá ser el procedimiento TCP/IP el encargado de supervisar y controlar la pérdida de paquetes, ya que el protocolo Frame Relay no le indicará nada al respecto.

De alguna manera, podemos decir que el protocolo Frame Relay, realiza un control de datos "al por mayor", para propiciar una alta velocidad. En contra de esto, el protocolo X25, realiza una gestión pormenorizada de cada uno de sus paquetes, permitiendo que sean sus capas más básicas quienes realicen este control. Por ello, el Protocolo X25, permite una implementación más racional en máquinas menos potentes, mientras que los nuevos protocolos, al simplificar el nivel de control que ellos proporcionan, están cediendo éste a niveles superiores implementados en máquinas que cuentan con amplios recursos.

Por último, el protocolo X25, didácticamente, como ejemplo de un protocolo de Red, proporciona al estudiante una idea mucho más rica de los

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

elementos y técnicas que implementan las tres capas iniciales de la definición OSI para interconexión de estaciones que trabajan en modo paquete.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.2.2 Lenguaje de programación Delphi

El lenguaje de programación Delphi de Borland es un lenguaje para desarrollo de aplicaciones que corran bajo Windows de Microsoft. Está orientado a objetos y se basa en los conceptos y filosofía de Turbo Pascal. Utiliza una interfaz gráfica donde se pueden escoger los elementos de programación adecuados a cada sentencia o procedimiento.

Para desarrollar el **Simulador X25** se ha optado por el lenguaje de programación Delphi. Gracias a su sencillez y versatilidad permite al programador desarrollar su tarea siguiendo las pautas de una interfaz gráfica de fácil comprensión. Así mismo, las enormes posibilidades que Delphi aporta a la programación han permitido lograr un excelente acabado a la aplicación.

1.2.2.1 Antecedentes al desarrollo en Delphi

Las interfaces gráficas de usuario han revolucionado la industria de la microcomputadora. Son una prueba de que el refrán 'Más vales una imagen que mil palabras' sigue siendo cierto. En lugar del críptico indicativo C:>, al que están tan acostumbrados los usuarios del DOS, se ofrece a los usuarios un escritorio lleno de iconos y programas que utilizan el ratón y menús. Quizá más omportante que el aspecto de las aplicaciones de Windows de Microsoft, sea la sensación que dan las aplicaciones desarrolladas. Las aplicaciones de Windows generalmente tienen una interfaz de usuario consistente. Esto significa que los usuarios pueden dedicarle más tiempo a perfeccionar la aplicación y menos a preocuparse de las teclas a pulsar dentro de los menús y de los cuadros de diálogo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

A pesar de que los programadores no han sentisdo nunca un gran entusiasmo por las GUIs, a los usuarios neófitos comienzan a agradarles, y esperan que los programas de Windows estén basados en el modelo GUI. (tanto en apariencia como en contenido). Por tnto, si es necesario desarrollar programas para Windows, será indispensable disponer de una herramienta par desarrollar se modo eficiente aplicaciones basadas en GUI.

Durante mucho tiempo no existieron herramientas de este tipo. Antes de que se introdujera Visual Basic en 1991, el desarrollo de aplicaciones era mucho más complicado en Windows que en DOS. Los programadores tenían que estar pendientes de lo que ocurría con el ratón, del menú en el que estaba el usuario y de si él o ella estaban pulsando una o dos veces en un lugar concreto. La realización de una aplicación en Windows requería programadores expertos en C y cientos de líneas de código para llevar a cabo las tareas más sencillas. Incluso los expertos tenían problemas. (El Software Development Kit, Equipo de Desarrollo de Software, de Windows de Microsoft que se necesitaba junto con el compilador de C pesaba alrededor de cuatro kilogramos.

Visual basic comenzó cambiando este proceso en sus tres primeras versiones, Delphi le dió la vuelta. Las aplicaciones complejas de Windows pueden realizarse en un tiempo menor del requerido anterirmente. Los errores de programación no suceden tan a menudo y, si tienen lugar, son mucho más fáciles de detectar y de corregir. Sencillamente, con Delphi programar para Windows no sólo se ha convertido en ina actividad más eficiente, sino también más divertida la mayoría de las veces.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.2.2.2 ***Actualidad de Delphi***

Alguna de las ventajas de Delphi sobre las tres primeras versiones de Visual Basic son:

Las aplicaciones desarrolladas con Delphi son básicamente tan rápidas como las desarrolladas en C o en C++.

Con Delphi pueden construirse programas ejecutables reales que incluyen DLLs, pidra angular de la programación en Windows.

Pueden construirse objetos reutilizables siguiendo los paradigmas de la programación orientada a objetos.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.3 Programa Simulador X25.

El programa **Simulador X25** es una herramienta que permite crear y seguir la mayoría de las situaciones que se producen en el interfaz ETD-ETCD de una comunicación basada en el protocolo X25. La disciplina interactiva de trabajo posibilita al usuario intervenir de forma dinámica en todo momento de la ejecución del programa. Se puede comprobar y seguir el desarrollo del protocolo X25 a nivel de Red y/o de terminal de ususario.

Se ha pretendido en la implementación del programa, enriquecer éste con la experiencia que sus autores tienen en la Red Iberpac de la Compañía Telefonica. De esta manera, se ha intentado plasmar en el programa los sucesos más usuales en la Red. La intención es llevar al usuario lo más cerca de la realidad.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.3.1 Elementos del programa

El programa **Simulador X25** se compone de diferentes objetos que se relacionan entre sí observando rigurosamente las prescripciones de la norma. Sobre el area de trabajo se sitúan los terminales y nodos de Red como elementos principales. Ellos, debido a su complejidad, son los que han aconsejado dividir el **Simulador X25** en dos proyectos. Además se han desarrollado analizadores y otros elementos que intervienen en la gestión de una Red. También se podrán abrir diversas ventanas de control y seguimiento de la aplicación.

Gracias a las posibilidades de **Configuración,** todos los elementos que intervienen en el programa son implementados en cantidad y calidad según los criterios del ususario. Se pueden generar tramas y situaciones que obligarán, tanto a la **Red** como a los **Terminales**, a tomar decisiones en cada momento para cumplir con el Protocolo. Desde las condiciones de trabajo de los **Terminales** se pueden establecer tantas comunicaciones como se quieran y con las características que se definan desde su configuración. Así mismo, las opciones de **Analizador** de los diferentes objetos, hacen un seguimiento en tiempo real de todos los eventos que se suceden en las lineas virtuales que se hayen establecidas. Con el control sobre las **Rutas** podemos encaminar o cortar el tráfico entre nodos. Las opciones disponibles para generar **Informes** permiten generar una traza de desarrollo del programa. Desde el menú de **Ayuda** podemos solicitar de la aplicación todo tipo de explicaciones y aclaraciones sobre los eventos del **Simulador X25**.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.3.2 Estructura del programa

El algoritmo del programa se basa en los objetos terminales, conmutadores (nodos), analizadores y de etapa física que actúan guiados por los procedimientos de control de comunicaciones de nivel dos, tres y rutas así como por las configuraciones que se definan. Todo esto es implementado observando con rigor el cumplimiento de la norma en todos sus pasos. Los medios que se han utilizado para desarrollar el **Simulador X25** son el lenguaje de programación **Delphi** corriendo bajo el entorno **Windows** de **Microsoft**. Esta herramienta ha permitido definir el programa como una aplicación orientada a objetos.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

1.4 Documentación

Acompaña al ejecutable del **Simulador X25** una extensa documentación que ayudará al usuario a la comprensión y uso tanto del programa como de la norma X25.

En el capítulo de introducción se hace una defensa del proyecto, aportando unas breves explicaciones acerca de los fundamentos que lo soportan y la vigencia de sus propuestas. Así mismo se reseñan los capítulos que conforman la documentación aportada.

En el **Indice** se enumeran todos los apartados de la documentación.

El **Manual de usuario** constituye una referencia rápida de todas las funciones disponibles durante la ejecución de la aplicación junto a una breve reseña de las mismas. Recorre a través de los menús desplegables una a una todas las opciones a las que se tiene acceso.

Se incluye además el **Curso X25** que contiene un método rápido y ameno de aprendizaje de la norma X25 basado en el programa. Se estructura en capítulos comenzando con el estudio de los terminales a nivel de usuario. Se explican los pasos para activar y cerrarlos, establecer y liberar canales, etc. En el siguiente capítulo se estudia a través de las posibilidades de los analizadores, los procedimientos de nivel 2 y 3 de la norma con las respectivas tramas y paquetes. El capítulo de Nodos explica la misión de los mismos y la capacidad de

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

enrutamiento de los paquetes a través de la Red. Es posible conocer y practicar con la mayoría de configuraciones específicas de usuario cuando se llega al capítulo de las mismas, a la par que se muestra y define los directorios de trabajo del programa. En el apartado dedicado al control del **Simulador X25** se explican los distintos elementos que intervienen en el desarrollo del mismo. Se podrán seleccionar para seguir la traza en las evoluciones del programa y se tendrá una mejor comprensión de la aplicación. Por último, se presentarán los menús de ayuda que explican en todo momento las posibilidades y significado del **Simulador X25**.

El capítulo de **Algoritmo y estructura** define en detalle los objetos, elementos, procedimientos, funciones, etc en que se basa el algoritmo que rige el desarrollo de los objetos terminales, analizadores y funciones de cálculo. Se estudian las unidades más importantes de la aplicación por separado, con las subestructuras que las conforman. Así se refeleja el coportamiento de la unidad Term1, que especifica a los terminales de usuario, CCTerm, que controla el Nivel 3 del terminal,CCTerm2, que hace lo propio con el Nivel 2, ETFi, asegura que se mantenga ek control del nivel físico en todos los elementos, Analizad, observa el desarrollo de los analizadores y Otros, donde se incluyen el resto de unidades asociadas al presente proyecto que intervienen en el programa.

El capítulo de **Presupuesto** realiza un cálculo en profundidad de los medios empleados y el coste generado en la realización del **Simulador X25** y del presente proyecto. Se hace un análisis de tiempo y materiales empleados con su correspondiente valoración a los cuales se les ha aplicado los coeficientes oportunos.

Así mismo se entrega un anexo que contiene el listado del código referido a las partes del **Simulador X25** mencionadas en el capítulo de algoritmos y ampliamente comentado para mejorar su análisis y comprensión. También se incluye una **bibliogafía** reseña los títulos que se han consultado en la elaboración del proyecto, si bien es necesario hacer mención a la experiencia de los autores en

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

el terreno profesional para completarla de alguna manera. Por último se suma un glosario con los términos técnicos más frecuentes usados en los distintos capítulos de la presente memoria.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2

Manual de Usuario

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.1.Introducción

El presente texto constituye el manual de usuario para el programa **Simulador X25**. Está estructurado en diversos capítulos que explican paso a paso toda la potencia y facilidades que aporta esta herramienta de trabajo.

El **Simulador X25** permite de forma dinámica e interactiva, comprobar y seguir el desarrollo del protocolo X.25 definido por el CCITT. Gracias a las posibilidades de **Configuración,** todos los elementos que intervienen en el programa son implementados en cantidad y calidad según los criterios del usuario. Se pueden generar tramas y situaciones que obligarán, tanto a la **Red** como a los **Terminales**, a tomar decisiones en cada momento para cumplir con el Protocolo. Así mismo, las opciones de **Analizador** de los diferentes objetos, hacen un seguimiento en tiempo real de todos los eventos que se suceden en las líneas virtuales que se hallen establecidas. Con el control sobre las **Rutas** podemos encaminar o cortar el tráfico entre nodos. Las opciones disponibles para generar **Informes** permiten generar una traza de desarrollo de la simulación. Desde el menú de **Ayuda** podemos solicitar de la aplicación todo tipo de explicaciones y aclaraciones sobre los eventos del **Simulador X25**.

Se trata en definitiva, de una poderosa ayuda para probar en el laboratorio cuales serán las respuestas que se producirán en una Red que trabaje bajo la norma X25. El **Simulador X25** se orienta, pues, hacia un fin didáctico con capacidad de reacción ante cualquier situación que se le pueda plantear.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.2.Para empezar

Los próximos capítulos realizan un recorrido a través de las funciones, tanto básicas como complejas, del programa **Simulador X25**. Estos ejercicios pretenden orientarle rápidamente y mostrarle formas típicas de ejecución de operaciones.

Las imágenes que se usan en el curso práctico se tomaron de la aplicación durante la ejecución del programa. Todas ellas son imágenes en color. Si el ordenador dispone de un adaptador de gráficos VGA de 16 colores, o superior podrá apreciar que la visualización de las imágenes en color es óptima. En caso de que desee trabajar con frecuencia con otro tipo de tarjeta, le recomendamos incluir en el sistema un adaptador de gráficos adecuado así como un monitor en color compatible.

Le suponemos familiarizado con ciertos términos como 'arrastrar' y 'hacer clic'. Si no comprende alguno en concreto, consulte el glosario o el índice situado al final de este manual. También encontrará un glosario en el sistema de ayuda en línea de **Simulador X25**, y un método para aprender a usarlo al final de este capítulo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.2.1.Instalar el programa Simulador X25

Inserte el CDRom o diskettes suministrados en la unidad correspondiente de su computadora. Si usa los diskettes introduzca primero el que lleva el número 1 y está rotulado como 'Disco de INSTALACION'. A medida que los vaya necesitando, el programa le irá pidiendo el resto de dikettes.

Desde el DOS o desde Windows ejecute la aplicación 'instalar.exe'.

El programa de instalación se hará cargo de crear los subdirectorios y cargar todos los ficheros que el **Simulador X25** necesita para su funcionamiento. Por defecto creará un subdirectorio propio con nombre 'SimulX25' conteniendo las aplicaciones principales del programa. De este apartado colgarán a su vez, los subdirectorios que almacenarán a los ficheros auxiliares de ayuda, configuración, etc. Durante la instalación del programa se abrirán varios cuadros de diálogo que le permitirán modificar estas opciones.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.2.1.1.Requisitos mínimos

- Computadora Pc compatible con procesador 486 o superior.
- Mermora RAM mínima de 4Mb.
- Espacio disponible en disco duro de 10 Mb.
- Entorno Windows 3.1x o posterior.
- Ratón.

2.2.2.Activar el programa Simulador X25

Una vez instalado el programa **Simulador X25**, puede activarlo como cualquier aplicación Windows.

Para abrir el **Simulador X25**:

.- Lance Windows. Si necesita ayuda, consulte el manual de usuario de Microsoft Windows.

.- En el área de trabajo de Windows, haga doble clic sobre el icono de **Simulador X25**.

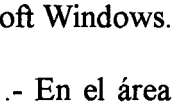


Ilustración 2.1 . Icono Simulador X25

Este icono está situado en la ventana de aplicación **Simulador X25** (a menos que se haya cambiado el nombre del icono del grupo o desplazado el icono) y abre la ventana de presentación de la aplicación **Simulador X25** tal y como se muestra en la ilustración.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

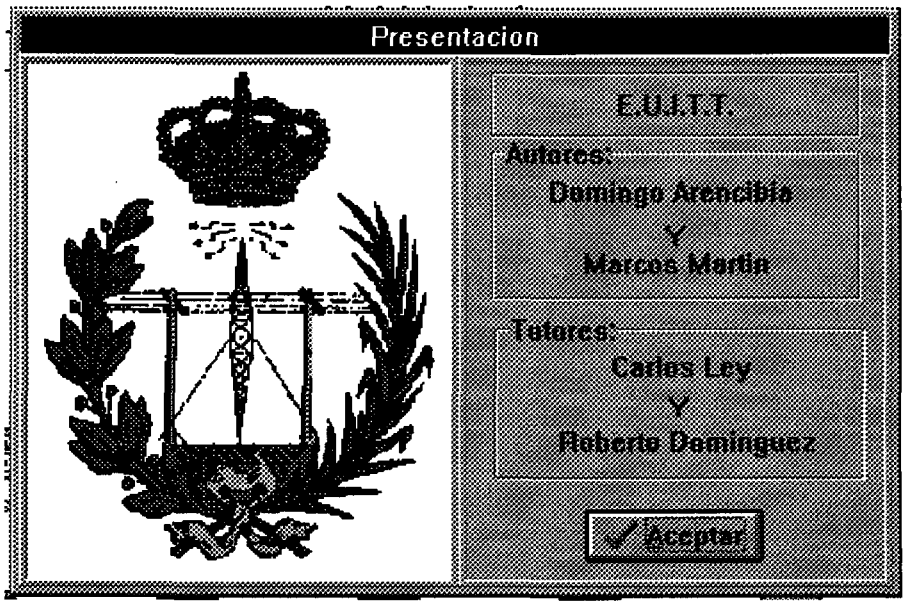


Ilustración 2.2. Ventana de presentación.

Tras salvar esta pantalla haciendo clic en el botón Aceptar, se pasa a la pantalla principal, verdadero corazón del programa.

© Del documento, los autores. Digitalización realizada por ULPGC. Bibliotecas Universitaria, 2006

2.2.3.LA PANTALLA PRINCIPAL

La pantalla principal es el centro de operaciones del programa **Simulador X25**. Desde aquí se tiene acceso a todos los elementos que intervienen en el mismo y se controla el desarrollo de la aplicación.

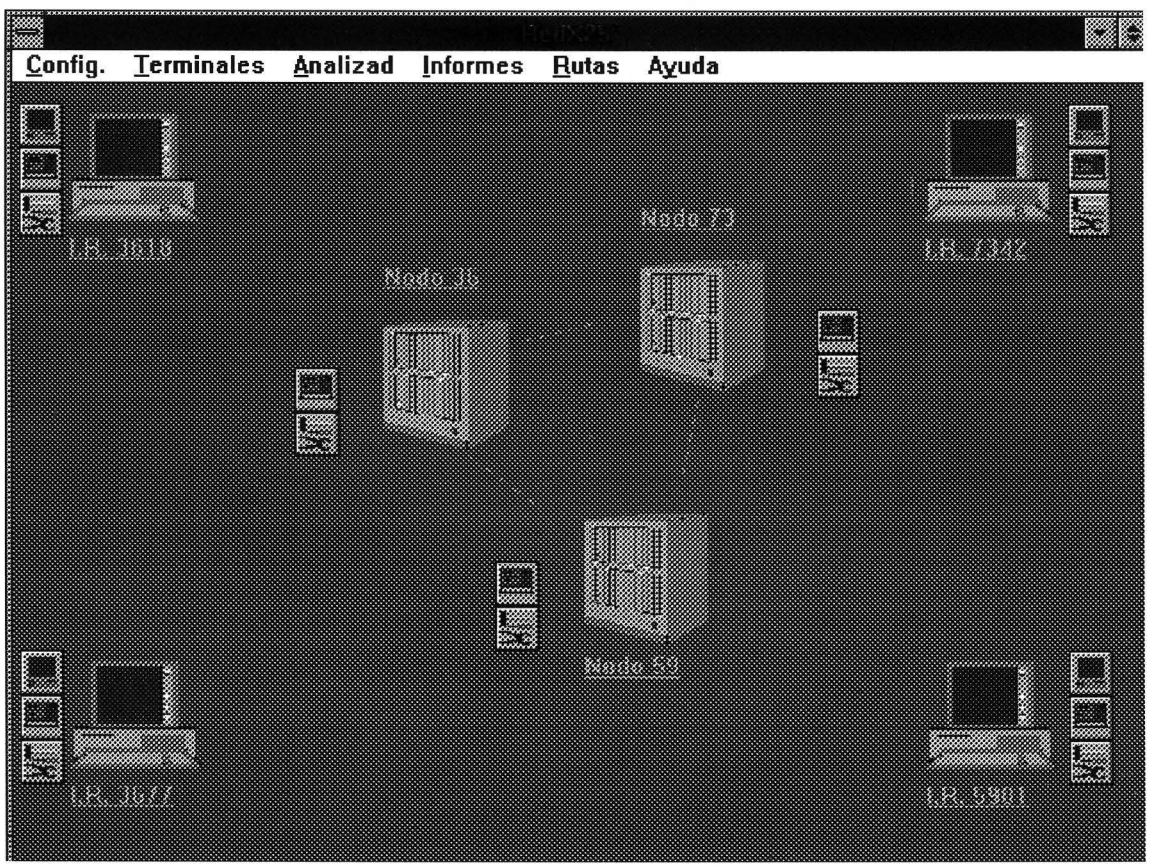


Ilustración 2.3. Pantalla principal.

La pantalla principal del **Simulador X25** dispone de tres zonas diferenciadas con distintos elementos asociados a cada sector. Estas son: La barra de título, la barra de menús y el área de trabajo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.2.3.1.La barra de títulos

Contiene el nombre del programa y los botones propios de control de ventana de Windows.

2.2.3.2.La barra de menús

Cuenta con las opciones de Configuración, Terminales, Analizadores, Informes, Rutas y Ayuda. Desplazando con el ratón el cursor sobre las mismas y haciendo clic se tiene la potestad de abrir los distintos menús desplegables que dan acceso a la definición, activación y utilización de los objetos. Parte de estas mismas opciones se repiten en los iconos que se encuentran en el área de trabajo. Así se accede al funcionamiento del programa de forma interactiva.

2.2.3.3.El área de trabajo

En él es donde se encuentran repartidos los objetos del programa: terminales y nodos y donde se hace el seguimiento de las evoluciones de la aplicación a través de las distintas ventanas..

2.2.3.3.1.**Terminales**

Estos iconos situados cerca de las cuatro esquinas de la pantalla, hacen referencia a los Terminales, elementos del programa que trabajan en modo

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

paquete. Simulan un sistema de tratamiento de datos capaz de trabajar conectado a una Red de transmisión bajo norma X25. En la simulación de la Red es necesario implementar una manera de entregar los datos a transmitir al nivel de Red (ISO), para ello se ha introducido el terminal que actúa como empaquetador/desempaquetador, enviando los datos escritos en un editor o presentes en un fichero, a la línea de transmisión y mostrando los datos recibidos de dicha línea. Este icono no tiene una aplicación específica, aunque sobre él se puede desplegar una ventana de ayuda dinámica a través de la opción Ayuda 'Acerca de ...' de la barra de menú.

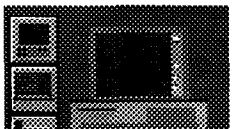


Ilustración 2.4. Terminales.

Tiene asociados sus iconos de Activación, Analizador y Configuración de terminal. La función de estos elementos coincide con algunas opciones de la barra de menús.

2.2.3.3.2. **Nodos**

Bajo este icono se representa la figura de los nodos de Red. Los nodos de Red son los elementos activos que cumplen dos misiones fundamentales: por un lado mantienen el protocolo con los terminales locales que tiene asociados a sus puertas, por otro mantienen el diálogo y las vías de comunicación pertinentes con otros nodos para intercambiar información con estos y lograr la unión extremo a extremo entre terminales. Este icono no tiene ninguna función operativa, pero se asocian a él todas las actividades referentes a este elemento.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

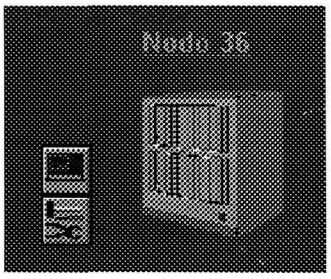


Ilustración 2.5. Nodos.

Tiene asociados los iconos de Analizador de ruta y Configuración de Red. La función de estos elementos coincide con algunas opciones de la barra de menús.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.LA BARRA DE MENÚS

Está situada en la zona superior de la superficie de trabajo. Haciendo clic sobre alguna de las palabras claves reservadas se tiene acceso al control del programa. Algunas opciones se repiten en los iconos que se encuentran en la superficie de trabajo.

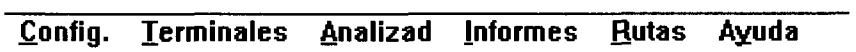


Ilustración 2.6. Barra de menús.

Las palabras clave son: Config (configuración), Terminales, Analizad (analizadores), Informes, Rutas y Ayuda. En los próximo capítulos se hace un estudio detallado de las posibilidades de cada una.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.1.Config.

Esta opción da lugar a los diferentes submenús de configuración de los elementos del programa. Estas configuraciones se guardan en ficheros que la mantienen después de abandonar la operación.

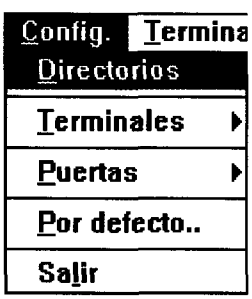


Ilustración 2.7. Configurar.

2.3.1.1.Directorios

Se abre desde aquí un cuadro de diálogo donde se especifica los subdirectorios y vías de acceso donde se ubican los archivos de configuración del programa, así como éste mismo. Si al comenzar la ejecución del programa éste no encuentra los ficheros que necesita o si los subdirectorios especificados no son válidos, el programa **Simulador X25** los creará con su contenido en blanco.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

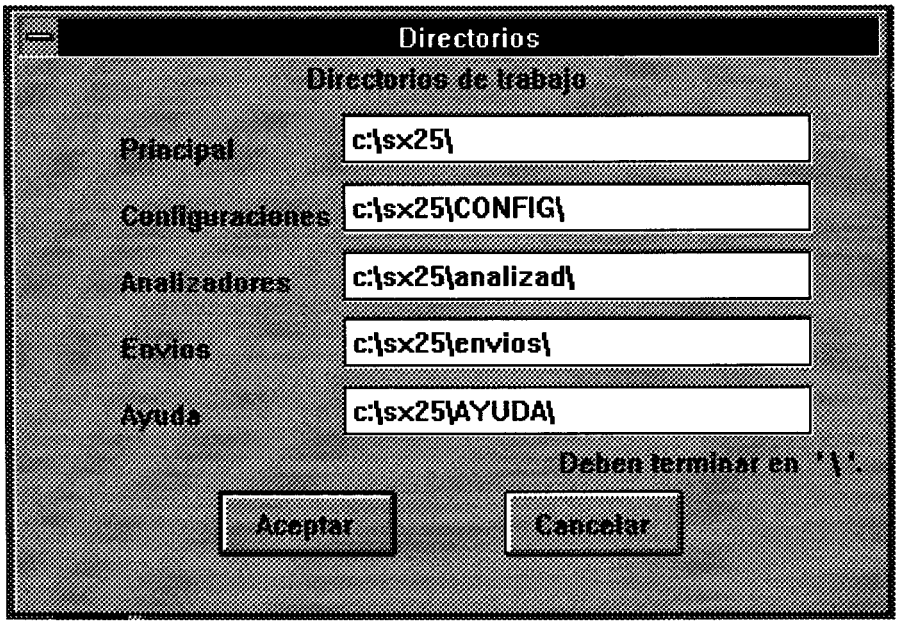


Ilustración 2.8. Cuadro de directorios.

Los subdirectorios a ser direccionados son:

2.3.1.1.1.**Principal**

Se guardará en este subdirectorio el programa **Simulador X25**.

2.3.1.1.2.**Configuraciones**

Se guarda en este subdirectorio los ficheros que almacenan las configuraciones en uso de terminales y nodos de Red del programa.

2.3.1.1.3.**Analizadores**

Se guarda en este subdirectorio los ficheros que se crean para recoger los datos provenientes de las capturas de los analizadores.

2.3.1.1.4.**Envíos**

Se guardan en este subdirectorio los ficheros que se podrán transmitir como tales desde la opción de envíos del menú de terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.1.1.5.**Ayuda**

En este subdirectorio se guardan los diferentes ficheros de ayuda dinámica que ofrece el programa durante la ejecución del mismo.

2.3.1.2.Terminales

Define las distintas opciones de configuración de los terminales. Abre un cuadro de diálogo que coincide con el que se abre en el icono de Configuración de terminal situado en el tablero de trabajo. Cada terminal abre su propio cuadro de diálogo donde se encuentran las diferentes solapas que dan acceso a los puntos de configuración. Estas opciones deben ser idénticas a las que guarda el icono de configuración de Red colateral para asegurar un correcto funcionamiento de estos elementos. Las solapas son:

2.3.1.2.1.**Nivel 2**

Se configuran en esta solapa las variables deNivel 2. Sus opciones son:

Ilustración 2.9. Configuración de terminal. Nivel 2.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.1.2.1.1. *Ventana de nivel 2*

Se define el valor de las **ventanas en Recepción** y **Transmisión**. Debe estar comprendido entre 0 y 7. Por defecto se fija a 7

2.3.1.2.1.2. *Temporizadores y Contadores*

Definimos aquí el valor de los temporizadores **T1** y **T3**, así como del contador **N2** que controlan el funcionamiento de la Red.

T1 define el tiempo que debe pasar antes de que la Red o Terminal inicie una secuencia de envío de tramas para establecer el enlace. Por defecto se asigna el valor de 3 sg.

T2 Define el tiempo que debe pasar entre el envío de dos tramas de establecimiento de enlace dentro de una secuencia. Por defecto se asigna el valor de 5 sg.

N2 Fija el número de tramas que componen cada secuencia para establecimiento del enlace. Por defecto toma el valor 7.

2.3.1.2.2. **CVP**

Configuraremos en esta ventana los parámetros de control que rigen el funcionamiento de los canales virtuales permanentes. Sus opciones son:

2.3.1.2.2.1. *Ventana de nivel 3*

Se define el valor de las **ventanas en Recepción** y **Transmisión** de nivel 3 para los CVPs. El valor de estos parámetros debe estar comprendido entre 1 y 7. Por defecto se le asigna el valor 2.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

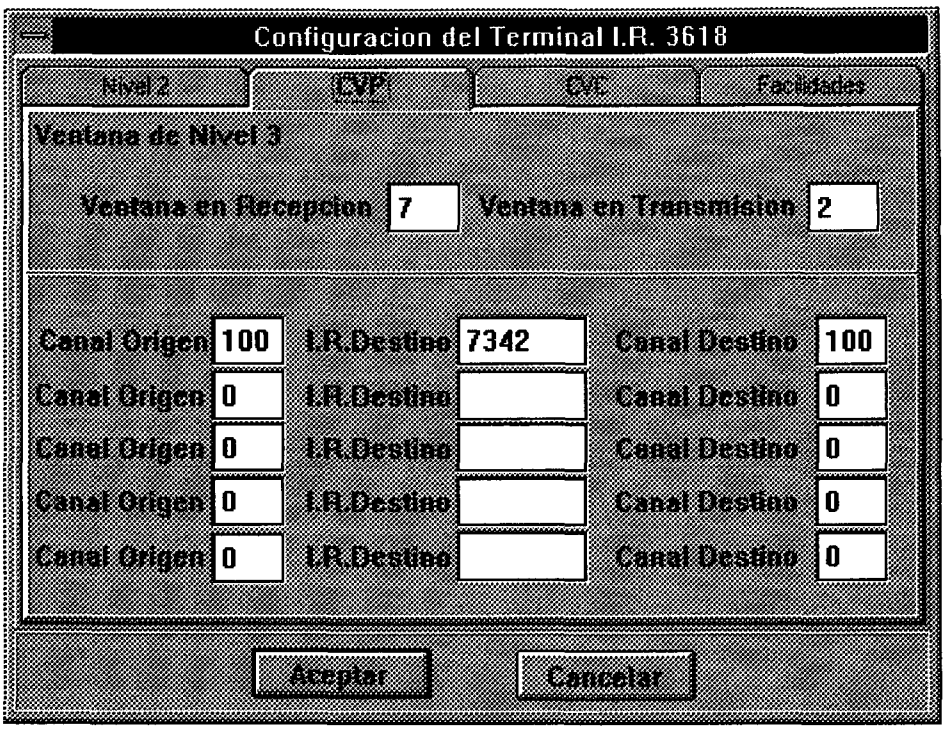


Ilustración 2.10. Configurar terminales. CVP.

2.3.1.2.2.2.Configuración de CVPs

Aquí se define la relación de los CVPs con los terminales destino y el CVP que éstos tienen asociado. Se despliega un casillero configurable de cinco renglones con tres casillas cada uno que hace referencia a los CVPs que se pueden definir. Cada renglón define un CVP y precisa de relacionar: el **Canal Origen**, que especifica el valor en decimal del canal que utilizará el terminal local, el **I.R.Destino**, que especifica el Identificativo de Red del terminal distante y el **Canal Destino**, que especifica en notación decimal el canal que utilizará el terminal destino para el enlace permanente. Si no se desea asignar ningún CVP, el valor de estos campos debe ir a 0.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.1.2.3.**CVC**

Bajo este epígrafe se encuentran las opciones de configuración de los canales virtuales conmutados.

Ilustración 2.11. Configurar terminales. CVC.

*2.3.1.2.3.1.**Ventana de Nivel 3***

Se define el valor de las **ventanas en Transmisión** y **Recepción** de nivel 3 para los canales presentes bajo este epígrafe. El valor de estos parámetros han de estar comprendidos entre 1 y 7. Por defecto se les asigna el valor 2.

*2.3.1.2.3.2.**Numeración de CVCs***

Se especifican aquí los CVCs que se asocien a cada terminal. Estos deben tener numeración rigurosamente ordenada y consecutiva a los CVPs. Esto quiere decir que se debe asignar los canales cuya numeración sea inmediatamente superior al canal más alto asignado como CVP y se debe empezar a asignar, si los hubiera, con los CVCs Entrantes, a continuación los Entrantes/Salientes, y por último, los CVCs Salientes. Los valores que se definen para cada apartado son:

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Límite Inferior, que especifica el valor en notación decimal con el que empieza la numeración de los canales de que se trate y **Límite Superior**, que especifica el valor con el que terminan los canales a los que se hace referencia. Estos parámetros deben estar comprendidos entre los valores decimales 100 y 104. Si no se desea asignar ningún CVC de alguno de los tipos, el valor de estos campos debe ir a 0.

2.3.1.2.4. **Facilidades**

En este apartado se configuran las facilidades de usuario que posibilita la Red.

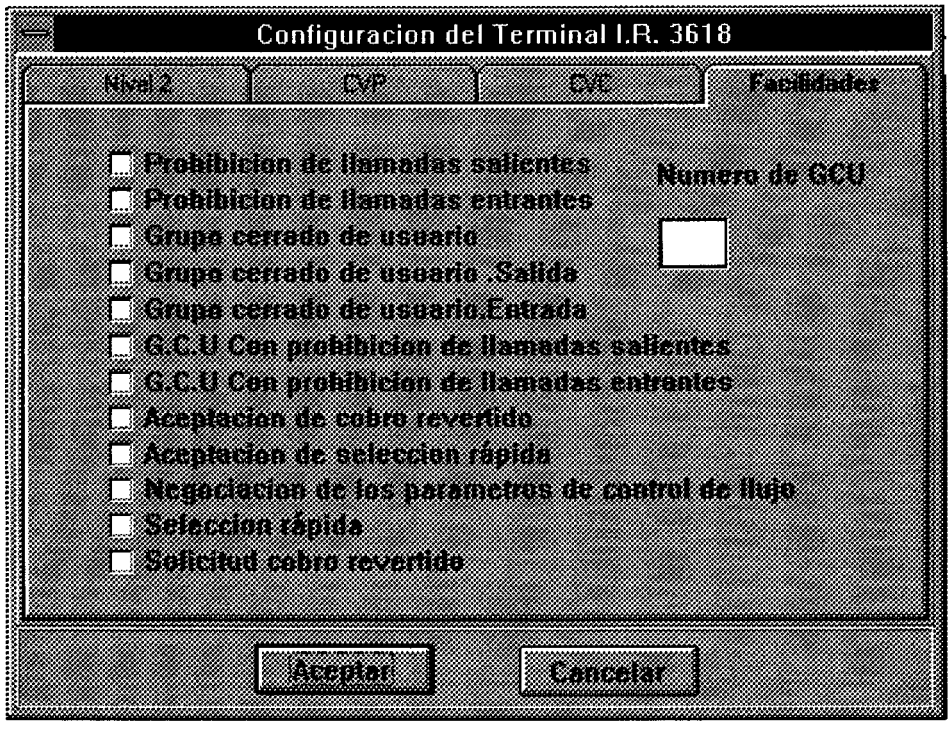


Ilustración 2.12. Configurar terminales. Facilidades.

Sus opciones son:

2.3.1.2.4.1. Prohibición de llamadas salientes

Esta opción prohibe al terminal seleccionado realizar llamadas.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.1.2.4.2.*Prohibición de llamadas entrantes*

Esta opción prohibe al terminal seleccionado recibir llamadas.

2.3.1.2.4.3.*Grupo cerrado de usuario*

Se denomina grupo cerrado de usuario a un conjunto de terminales que funcionan como una Red cerrada a los terminales ajenos. A los elementos de este grupo cerrado se les puede autorizar para tener contacto con equipos externos merced a las facilidades de acceso salida o entrada..

Número GCU.

Cada Grupo cerrado de usuarios lleva asociada una numeración propia que le diferencia de otros que existan.El **Simulador X25** permite una numeración comprendida entre 0 y255 como valor de este parámetro.

2.3.1.2.4.4.*Grupo cerrado de usuario. Salida*

Permite a un terminal que pertenece a un grupo cerrado de usuario el efectuar llamadas a terminales que no pertenezcan a dicho grupo.

2.3.1.2.4.5.*Grupo cerrado de usuario. Entrada*

Permite a un terminal que pertenece a un grupo cerrado de usuario recibir llamadas de terminales que no pertenezcan a dicho grupo.

2.3.1.2.4.6.*GCU con prohibición de llamadas salientes.*

Combina al mismo tiempo las facilidades de pertenencia a un grupo cerrado de usuario y la prohibición de efectuar llamadas.

2.3.1.2.4.7.*GCU con prohibición de llamadas entrantes.*

Combina al mismo tiempo las facilidades de pertenencia a un grupo cerrado de usuario y la prohibición de recibir llamadas.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.1.2.4.8. Aceptación de cobro revertido.

Opción asociada a la de cobro revertido que permite al terminal que tenga definida esta facilidad entrar en comunicación con un distante que haya pedido esta modalidad en su paquete de llamada.

2.3.1.2.4.9. Negociación de los parámetros de control de flujo

Esta opción permite a los terminales que la lleven, asignar dinámicamente en el establecimiento de las llamadas el tamaño de las ventanas de nivel tres tanto en transmisión como en recepción.

2.3.1.2.4.10. Selección rápida

Esta opción permite incluir un campo de información en el paquete de llamada a los terminales que lleven esta configuración.

2.3.1.2.4.11. Aceptación de selección rápida

Opción asociada a la anterior que permite al terminal que tenga definida esta facilidad entrar en comunicación con un distante que haya pedido esta modalidad en su paquete de llamada.

2.3.1.2.4.12. Solicitud de cobro revertido

Permite a un terminal cualquiera solicitar en su paquete de llamada esta modalidad en su comunicación. Para que la operación concluya con éxito es preciso que su colateral tenga definida la opción de aceptación de cobro revertido.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.1.3.Nodos

Abre un cuadro de diálogo donde se accede a las distintas opciones de configuración de las puertas de los nodos de Red. Esta operación coincide con el icono de Configuración de nodo situado en el tablero de trabajo. El cuadro de diálogo que se despliega tiene una barra de título con el nombre del objeto a que hace referencia.

Las opciones de configuración coinciden con las disponibles cuando se configura el terminal y que ya han sido estudiadas. Para un correcto funcionamiento del sistema es imprescindible que ambos elementos tengan idéntica definición.

2.3.1.4.Por defecto

Carga todos los elementos del programa con una configuración predefinida que se encuentra almacenada en los ficheros correspondientes dentro del subdirectorio 'Pred' que cuelga del subdirectorio que se haya especificado para las configuraciones.

2.3.1.5.Salir

Haciendo clic en esta opción se finaliza la ejecución del programa **Simulador X25** y se cede el control al entorno Windows.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.2.Terminales

Activa el terminal seleccionado, provocando en su interfaz de unión con la Red, el establecimiento del Nivel 2. Esta opción coincide con el Icono de Activación de Terminal que se encuentra en el tablero de trabajo.

Ilustración 2.13. Terminales.

La activación del terminal provoca la apertura de la ventana de terminal desde el que se definen y controlan las actuaciones del mismo.

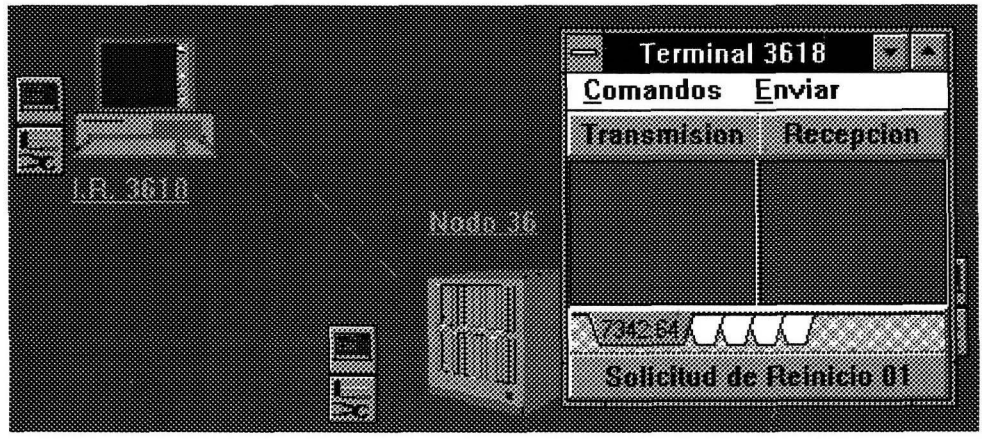


Ilustración 2.14. Activación de terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.2.1. Ventana de terminal

Esta ventana visualiza y ordena los procedimientos de nivel superior ordenados por el usuario. Desde ella se controla el establecimiento, uso y liberación de comunicaciones con el resto de terminales de la Red. Se divide, a su vez, en cuatro zonas bien diferenciadas:

2.3.2.1.1. **Barra de título**

Que contiene el identificativo de Red (I.R.) del terminal en cuestión y los botones de control y dimensionamiento de ventanas propios de Windows.

2.3.2.1.2. **Barra de menú**

Contiene los epígrafes de opciones para la gestión dinámica de las comunicaciones del terminal. Sus opciones son:

*2.3.2.1.2.1. **Comandos***

Permite desplegar el menú con las posibilidades de establecer, liberar o reiniciar canales, así como activar o desactivar el bit D. Sus opciones son:

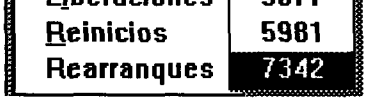


Ilustración 2.15. Comandos.

2.3.2.1.2.1.1. **Llamadas**

Permite generar llamadas a los distintos Terminales definidos en la Red. Abre a su vez un menú desplegable con las opciones de los identificativos de Red de los distintos terminales.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.2.1.2.1.2.**Liberaciones**

Provoca la generación de una liberación del CVC que esté seleccionado en ese momento.

2.3.2.1.2.1.3.**Reinicios**

Cursa un paquete de reinicio al canal que se haya seleccionado en ese momento.

2.3.2.1.2.1.4.**Rearranque**

Cursa un paquete de rearranque con origen en el terminal. Este procedimiento supone una liberación de todos los canales conmutados y el reinicio de todos los permanentes que tenga definido el terminal. El rearranque se envía siempre por el canal 0.

2.3.2.1.2.1.5.**Bit D=0/Bit D=1**

Esta opción funciona como un interruptor biestable que activa/desactiva el estado del bit D para exigir la validación de paquetes extremo a extremo.

2.3.2.1.2.1.6.**GCU / No GCU**

Esta opción funciona como un interruptor biestable que activa/desactiva la codificación del GCU a que pudiera pertenecer un terminal. Debe estar activa para que se trabaje bajo las características que supone esta facilidad.

2.3.2.1.2.2.Enviar

Permite desplegar un menú con las opciones de gestionar la transferencia de ficheros a través de la Red. Sus opciones son:

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 2.16. Enviar.

2.3.2.1.2.2.1.**Archivo**

Esta opción permite enviar algún archivo a través del canal que se halle seleccionado en ese momento. Se abre un cuadro de diálogo donde se define el archivo, sus parámetros y ubicación para ser transferido por el terminal. Este archivo debe ser necesariamente un archivo de texto para el correcto funcionamiento del **Simulador X25**. Si se intenta transmitir cualquier otro tipo de fichero se pondría al programa en una situación inestable con una respuesta impredecible. Esto es debido a que la presentación esporádica de caracteres especiales de edición tal como Saltos de página, Retornos de carro, etc, podrían ser mal interpretados por los editores que muestran los datos en las ventanas del **Simulador X25**,

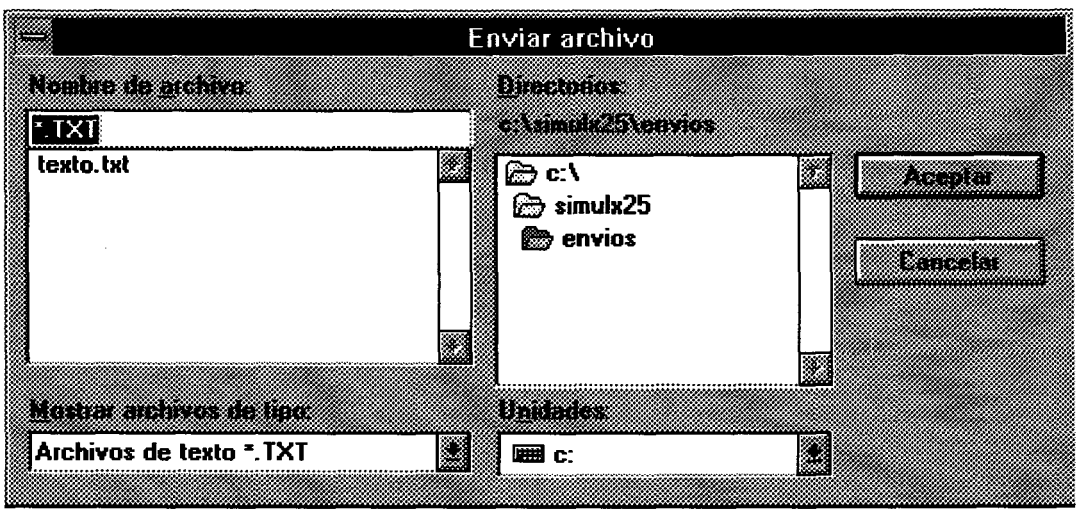


Ilustración 2.17. Cuadro enviar archivo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

El **Simulador X25** se encargará de fraccionar el fichero y añadirle los datos necesarios para convertirlo en paquetes que cumplan la norma X25 para posteriormente enviarlos a línea.

2.3.2.1.2.2.2.**Limpiar pantalla**

Esta opción permite la generación de un paquete de datos con el bit cualificador Q en estado activo. Esto genera una comunicación de nivel superior entre terminales. En este caso en concreto, la acción que se define es la generación de una orden de 'Limpiar pantalla' al extremo colateral.

2.3.2.1.3.**Ventana de canales**

Esta zona de la ventana de terminal es la que permite visualizar, a nivel de usuario, el estado y las conexiones de los canales definidos para cada terminal. Se seleccionan los distintos canales posibles a través de las solapas que se encuentran al pie. Estas solapas muestran el I.R., identificativo de Red, del terminal distante en notación decimal y, separados por un punto, el número del canal lógico empleado en notación hexadecimal.

En cuanto al color de fondo de esta ventana, define el tipo de canal y el estado del mismo: un color verde oscuro define un canal permanente en estado inactivo, mientras que cambia a azul claro cuando ese canal está establecido extremo a extremo. Lo mismo sucede para los canales virtuales conmutados con los fondos de color gris para el estado inactivo y blanco para el estado de establecimiento origen destino en fase de datos.

En estas ventanas se muestran en tiempo real los datos que emite y recibe cada terminal en el canal seleccionado por la solapa activa. Se subdividen en una zona para los datos de transmisión y otra para los datos recibidos.

2.3.2.1.4.**Barra de estado**

Se trata de una barra en la parte inferior de la ventana que informa del tipo de paquetes recibidos por el terminal en cada momento.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.3.Analizad

Esta opción provoca la instalación de un analizador de protocolos sobre el interfaz de línea ETD-ETCD que se haya designado. El enganche se produce en paralelo tipo T. Este analizador es completamente configurable y trabaja en tiempo real con todas las opciones que se hallen activas en ese momento.

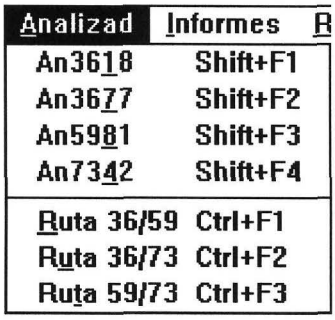


Ilustración 2.18. Analizadores.

Esta opción coincide con el icono de Analizador que se encuentra en el tablero de trabajo. Bajo este icono se esconde la ventana de analizador.

Ilustración 2.19. Ventana de analizador.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

La ventana de Analizador define todos los procedimientos y controles que afectan al analizador. Se subdivide en tres zonas bien diferenciadas:

2.3.3.1.Barra de Título

En ella se encuentra la lectura del elemento o línea que se está tratando en ese momento.

2.3.3.2.Barra de Menú

En ella se encuentra los epígrafes de los diferentes menús desplegables que dan acceso a toda la potencia del analizador. Sus opciones son:

2.3.3.2.1.**Nivel**

Especifica el nivel que se desea visualizar en pantalla. Se pueden escoger:

Ilustración 2.20. Nivel.

*2.3.3.2.1.1.**Nivel Físico***

Muestra las tramas codificadas en notación hexadecimal según aparecen en línea, filtrando únicamente los flags de línea. Estos sólo se muestran al inicio y final de cada trama a pesar de que su envío es constante entre dos estaciones (Terminal y Nodo). La visualización se hace en notación hexadecimal para los

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

campos de control, mientras que los elementos de información de los paquetes de datos de Nivel 3 se representan en caracteres ASCII. esto es debido a las limitaciones del lenguaje Pascal en cuanto a la longitud de las variables tipo String. Siendo el 256 octetos el número máximo de extensión de estas y teniendo en cuenta que los paquetes de datos del **Simulador X25** pueden llegar a un total de 128 caracteres, si se codificaran en notación hexadecimal (dos octetos por caracter), se desbordaría la variable antes indicada. Por esto se emplea para los datos la notación ASCII (un icteti por caracter) mientras que se respeta la notación Hexadecimal (dos octetos por caracter) para la cabecera del Nivel 3 y todos los elementos de Nivel 2.

*2.3.3.2.1.2.**Nivel de Enlace***

Muestra las tramas según van siendo enviadas o recibidas en línea. En este nivel se muestra una cabecera en la pantalla que especifica los diferentes elementos de la trama, a saber:

Ilustración 2.21. Cabecera nivel enlace.

TxRx indica quien es el emisor de la trama en el interfaz.

Dir indica el campo de dirección contenido en la trama.

Tipo Especifica el tipo de trama de nivel 2.

NrPNs Indica los números de secuencia de trama en transmisión y en recepción así como el estado del bit P/F.

*2.3.3.2.1.3.**Nivel de Red***

Muestra los paquetes de nivel 3 que se transmiten por la línea.. En este nivel se enseña una cabecera para englobar los datos que se muestran:

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 2.22. Cabecera nivel de Red.

TxRx Indica el emisor del paquete en el interfaz.

QDMoCan Indica el estado de los bits Q y D, así como el módulo activo y el canal lógico por el que se envía el paquete.

Tipo Indica el tipo de paquete de nivel 3.

PrMPs Indica el valor de los contadores de paquetes en transmisión y recepción así como el estado del bit 'más datos' M.

2.3.3.2.1.4.Todos los niveles

Muestra simultáneamente en pantalla los datos analizados en los tres apartados anteriores.

2.3.3.2.2.Filtro

Se encierra en este epígrafe las posibilidades de ocultar datos a la visualización del usuario. Sus opciones son:

Ilustración 2.23. Filtro.

2.3.3.2.2.1.Parar captura

Detiene la captura de datos del analizador. Esta opción funciona como interruptor biestable, para continuar con la captura de datos debe volver a activarse.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.3.2.2.2.*Modificar filtros*

Esta opción despliega un cuadro de diálogo con todas las alternativas configurables a las que el usuario tiene acceso. Según se encuentren activas las diferentes posibilidades de este cuadro, se visualizarán o no en la ventana del analizador.

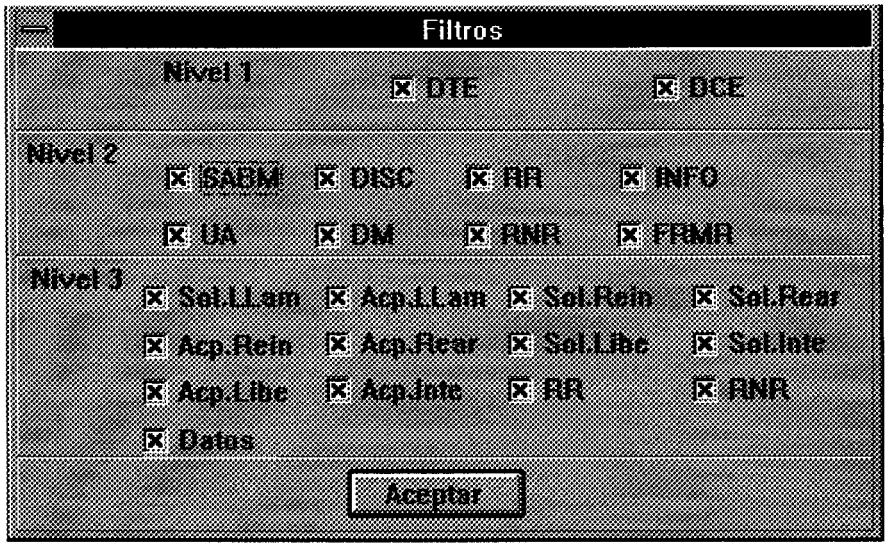


Ilustración 2.24. Cuadro de filtros.

Las condiciones de este apartado funcionan en conjunto según el principio lógico de un operador AND. Las posibilidades para las mismas son:

2.3.3.2.2.2.1.**Nivel Físico**

Hace referencia a las posibilidades configurables en este nivel:

DTE: Visualizará todo lo que envíe a línea el terminal local.

DCE: Visualizará todo lo que se reciba del nodo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.3.2.2.2.2. **Nivel 2**

Hace referencia a las posibilidades configurables a nivel de enlace. Muestra u oculta las siguientes tramas:

SABM

UA

DISC

DM

RR

RNR

INFO

FRMR

2.3.3.2.2.2.3. **Nivel 3**

Hace referencia a las posibilidades configurable a nivel de Red. Visualiza u oculta los siguientes paquetes:

Sol.llam: Solicitud de llamada.

Acp.llam: Aceptación de llamada.

Sol.Rein: Solicitud de reinicio.

Acp.Rein: Aceptación de reinicio.

Sol.Rear: Solicitud de rearranque.

Acp.Rear: Confirmación de rearranque.

Sol.Libe: Solicitud de liberación.

Acp.Libe: Confirmación de liberación.

Sol.Inte: Solicitud de interrupción.

Acp.Inte: Confirmación de interrupción.

RR

RNR

DATOS

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.3.2.2.3.Trampas

Esta opción permite definir las diferentes acciones que ejecutará el programa cuando se encuentre ante situaciones predeterminadas.

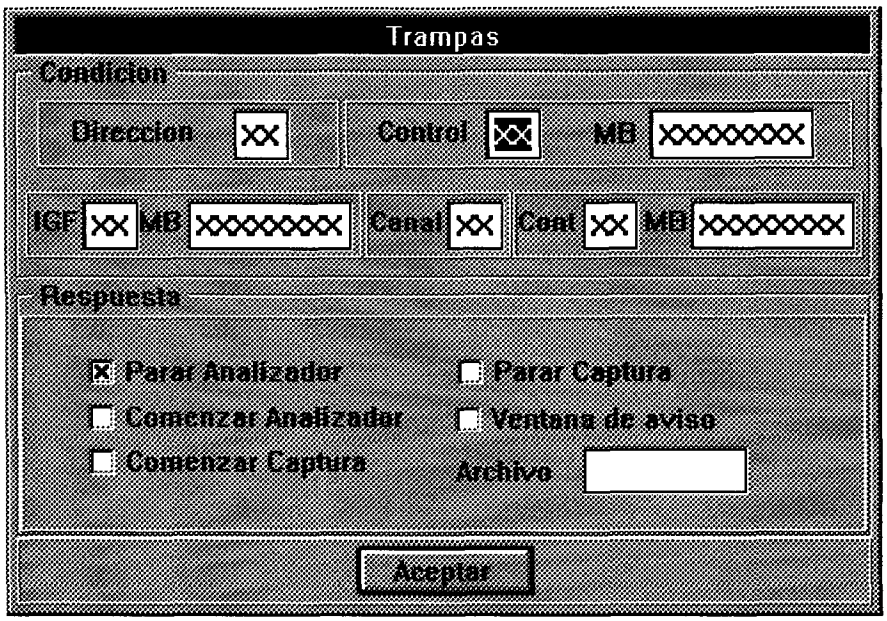


Ilustración 2.25. Cuadro de trampas.

Las condiciones de este apartado funcionan en conjunto según el principio lógico de un operador AND. Las posibilidades para las mismas son:

Dirección: Tiene capacidad para codificar la dirección en notación hexadecimal. Las posibilidades válidas son 01(dir. ETD) ó 03(dir. Red).

Control: Determina el campo de control de la trama. Se puede definir en notación hexadecimal usando el primer registro o bien por máscara de bits usando el segundo.

IGF: Determina el tipo de paquete de nivel tres que activará la trampa. Al igual que en el apartado anterior se puede seleccionar en notación hexadecimal o por máscara de bits.

Canal: Fija el canal que se desea seleccionar. Su notación es hexadecimal.

Cont:Fija el valor del octeto que determina el tipo de paquete de Nivel 3. En caso de ser de datos el valor de éste parámetro ha de ser par para lo que se utizará la máscara de bit que le acompaña.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Las posibilidades para las acciones a emprender son:

Parar Analizador: Detiene el analizador.

Comenzar analizador: Inicializa el analizador.

Comenzar captura: Arranca una captura sobre en fichero que se defina en la ventana **Archivo**.

Parar captura: Detiene la captura que se estuviera produciendo.

Ventana de aviso: Muestra una ventana de aviso cada vez que se produzca la condición especificada.

2.3.3.2.3. Archivos

Esta opción del analizador permite definir los archivos y su correspondiente ubicación donde se almacenen los datos provenientes de la ejecución del programa. Sus opciones son:

Ilustración 2.26. Archivos.

2.3.3.2.3.1. Guardar

Con esta alternativa podemos desviar los datos que se hallan en el buffer del analizador hacia un fichero. Guarda los datos de cada uno de los niveles predefinidos en el analizador.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

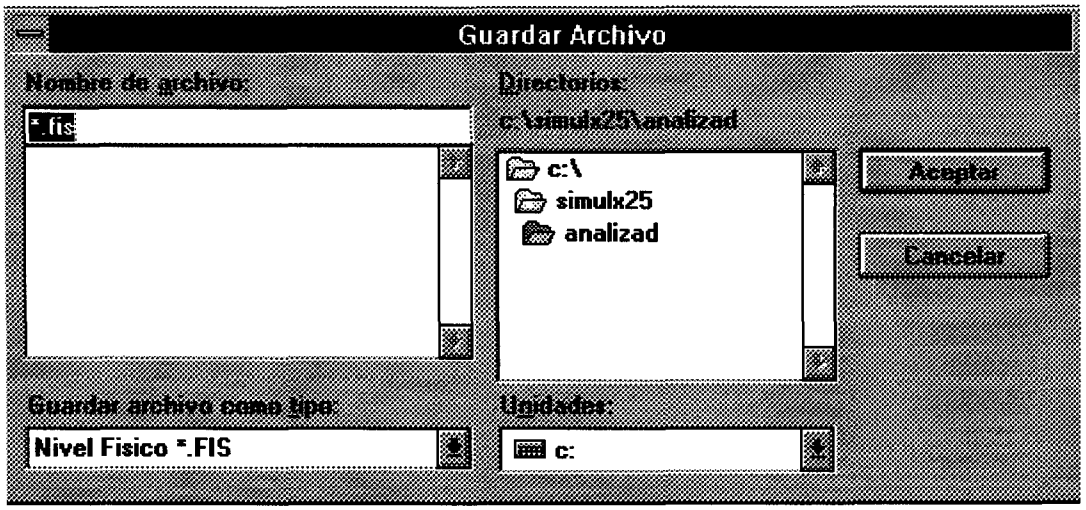


Ilustración 2.27. Cuadro de guardar archivo.

Cada nivel muestra su propio cuadro configurable de Windows que hace referencia a la ubicación de ficheros. Cada una de las posibilidades con las que cuenta se salva individualmente.

2.3.3.2.3.1.1.**Nivel Físico**

2.3.3.2.3.1.2.**Nivel de Enlace**

2.3.3.2.3.1.3.**Nivel de Red**

2.3.3.2.3.1.4.**Todos los niveles**

*2.3.3.2.3.2.**Capturar***

Desvía todos los datos que se recogerán por el analizador hacia un fichero. Despliega un cuadro de diálogo donde se define dicho fichero y su ubicación. La diferencia con la opción anterior consiste en que en Guardar se mandan los datos que ya han sido recogidos y que aún se encuentran en el buffer al fichero, mientras que ésta comenzará a mandar los datos que se vayan recogiendo a medida que lleguen al interfaz.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.3.2.3.3. ***Limpiar Analizador***

Ejecuta una orden directa para borrar los datos que se hallan presentes en la pantalla del analizador. A la par también se borran los datos que se hallen en el buffer de memoria.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.4.INFORMES

Se trata de una opción que permite un seguimiento de la norma X25 a través de los distintos elementos del **Simulador X25**. En este apartado se incluyen las facilidades de programa que permiten controlar a nivel lógico la evolución del mismo. Se trata de un conjunto de herramientas que permiten conocer el estado y desarrollo de los distintos objetos del programa.

Ilustración 2.28. Informes.

Su única opción es el panel de estado.

2.3.4.1.Panel de estados

Bajo este epígrafe se abren las posibilidades de seguir las evoluciones del **Simulador X25**. Se puede filtrar los elementos que se deseen y así obtener un informe selectivo de todas las decisiones y estados por los que pasan los objetos seleccionados. Este panel está pensado para mejorar la comprensión de los procesos que se suceden tanto en los elementos como en los niveles que intervienen en una comunicación bajo protocolo X25.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

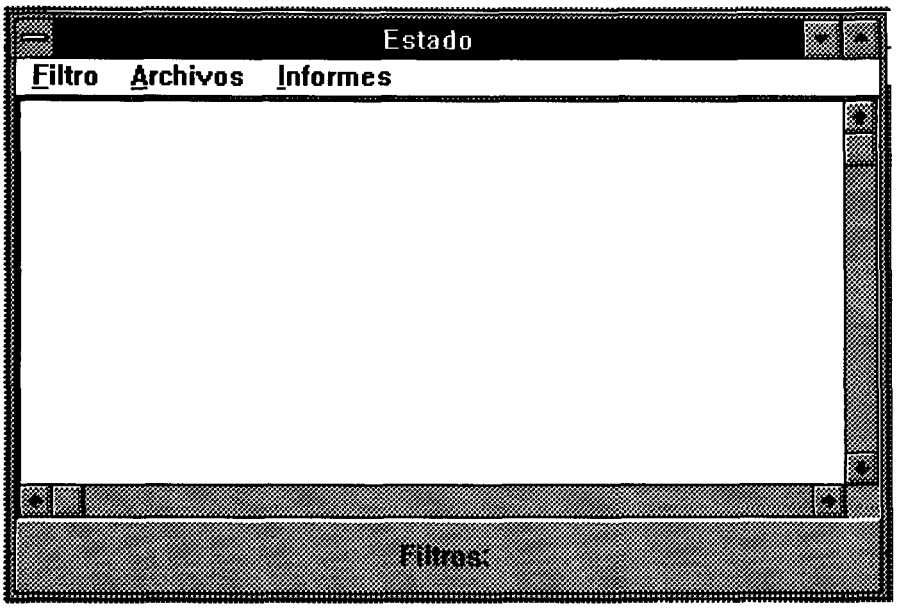


Ilustración 2.29. Panel de estados.

Despliega un cuadro cuyas posibilidades son:

2.3.4.1.1. Filtro

Abre todas las posibilidades de filtraje para los elementos a ser analizados. Sus posibilidades son:

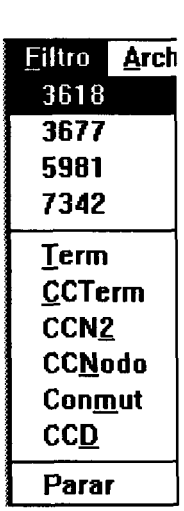


Ilustración 2.30. Filtro.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.4.1.1.1.***Terminales***

Despliega un cuadro con los elementos definidos en la Red. Se pueden seleccionar cuantos terminales se quieran para visualizar las acciones que el **Simulador X25** toma para mantener dentro de protocolo a estos elementos.

2.3.4.1.1.2.***Term***

Filtra las acciones que controla el objeto Terminal. Define las acciones de nivel superior que acontecen en un elementos terminal.

2.3.4.1.1.3.***CCTerm***

Filtra las acciones que controla el objeto Controlador de comunicaciones del Terminal. Define las acciones de nivel 3 que maneja el programa para mantener al terminal dentro del protocolo.

2.3.4.1.1.4.***CCN2***

Filtra las acciones que controla el objeto Controlador de comunicaciones de nivel dos. Define las acciones que se refieren al nivel dos en las comunicaciones de terminales y nodos.

2.3.4.1.1.5.***CCNodo***

Filtra las acciones que controla el objeto Controlador de comunicaciones del nodo. Define las acciones de nivel tres en las comunicaciones vistas desde el nodo.

2.3.4.1.1.6.***Conmut***

Filtra las acciones que controla el objeto Conmutador. Define las acciones de nivel superior que ocurren en los nodos de Red.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.4.1.1.7.**CCD**

Filtra las acciones que controla el objeto Controlador de comunicaciones de datagrama (ruta). Define las acciones de nivel tres que ocurren en la comunicación entre nodos.

2.3.4.1.1.8.**Parar**

Ejecuta la orden de parar las capturas del panel de estados.

2.3.4.1.2.**Archivos**

Especifica los ficheros con sus correspondientes ubicaciones y vías de acceso donde se almacenarán los datos capturados por el panel. Sus opciones son :

Ilustración 2.31. Archivos.

2.3.4.1.2.1.**Capturar**

Con esta alternativa podemos desviar los datos que se filtran en el panel hacia un fichero. Muestra su propio cuadro configurable de Windows que hace referencia a la ubicación de ficheros.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

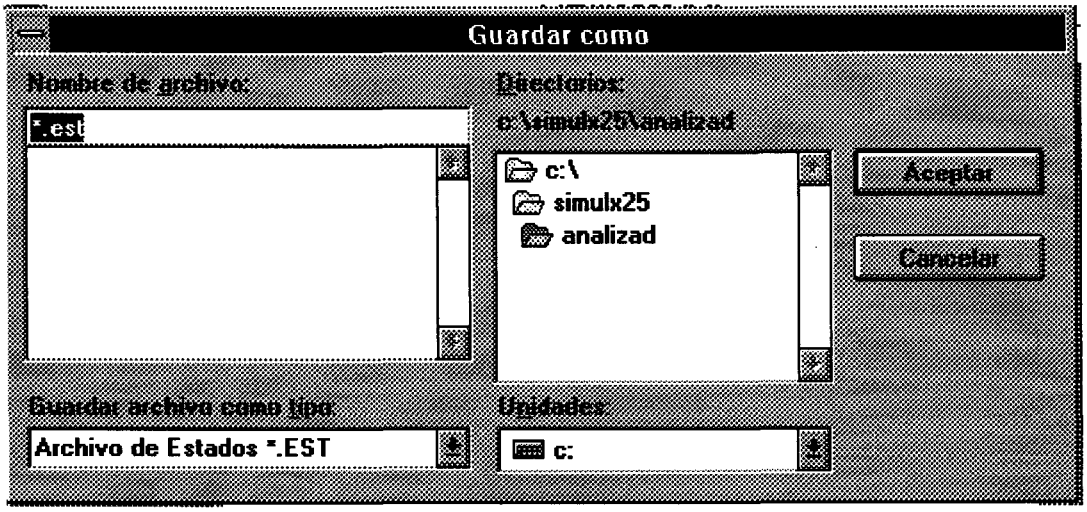


Ilustración 2.32. Capturar.

*2.3.4.1.2.2.**Guardar***

Con esta alternativa podemos desviar los datos que se hallan en el buffer del panel hacia un fichero. Muestra su propio cuadro configurable de Windows que hace referencia a la ubicación de ficheros.

Ilustración 2.33. Guardar.

*2.3.4.1.2.3.**Limpiar panel***

Ejecuta la orden de limpiar de la pantalla todo lo que se halle visualizado.A su vez libera los datos de su memoria temporal (buffer).

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.4.1.3.Informes

Controla las configuraciones y canales que tiene establecidos cada nodo de la Red. Sus opciones son:

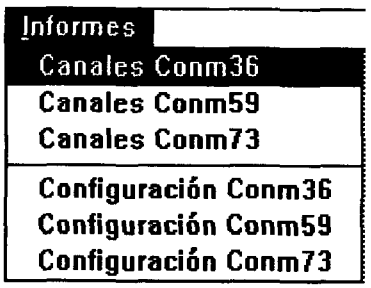


Ilustración 2.34. Informes.

2.3.4.1.3.1.Canales de los Conmutadores

Hacen alusión a los distintos canales que tienen establecidos cada uno de los nodos. La información que se muestra informa acerca del tipo de canal, su numeración, el ETD origen y destino y los parámetros que se han definido para el mismo.

2.3.4.1.3.2.Configuración de los Conmutadores

Explica la configuración de cada puerta en el momento de ejecución del programa.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.5.Rutas

Esta opción da acceso al control sobre las rutas del programa. Funciona como interruptor biestable permitiendo cortar/activar las rutas entre nodos de Red. Ante un corte de línea, el **Simulador X25** reencamina automáticamente el tráfico que era transferido por la misma hacia una vía alternativa. El estado de una ruta se hace visible en el área de trabajo por el color en el que aparece la misma: fijo en rojo para una ruta cortada y oscilante en azul-amarillo para una ruta activa.

Ilustración 2.35. Rutas.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.3.6.Ayuda

Esta opción da acceso a las ayudas dinámicas que ofrece el programa así como a la presentación del mismo. Sus posibilidades son:

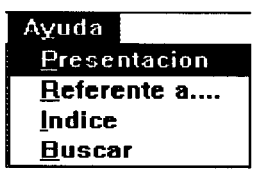


Ilustración 2.36. Ayudas.

2.3.6.1.Presentación

Despliega el cuadro de presentación del programa **Simulador X25**.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

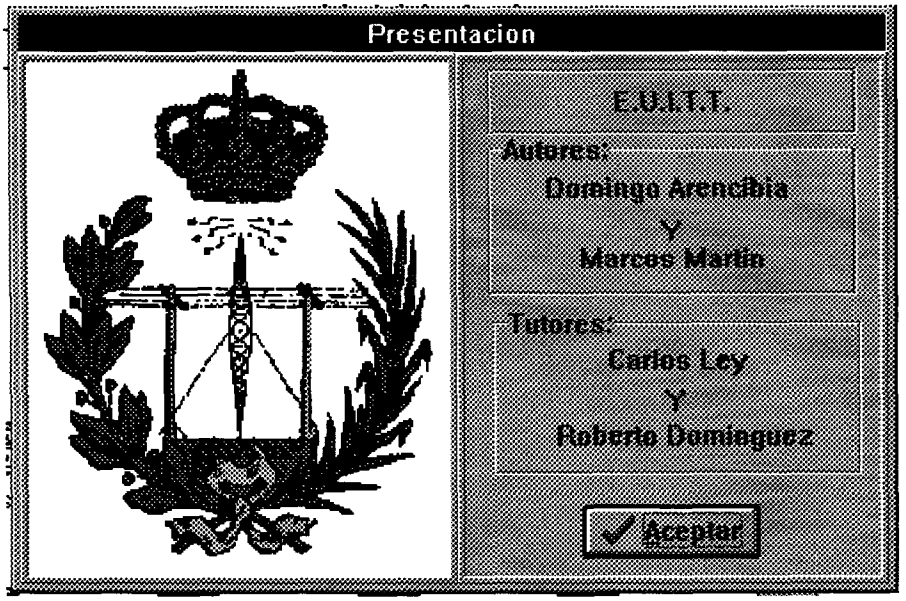


Ilustración 2.37. Presentación.

2.3.6.2. Referente a...

Da acceso al curso interactivo del protocolo X25 a través del **Simulador X25**. Con esta opción se activa el modo de ayuda de tal forma que cuando se haga clic sobre algún evento del programa se desplegará un cuadro de ayuda referente al evento que se cuestiona. Mientras esté activo este menú a la espera de elegir el evento que interese interrogar, el formato del cursor cambia añadiendo a la flechita habitual de Windows un cuaderno de consulta.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

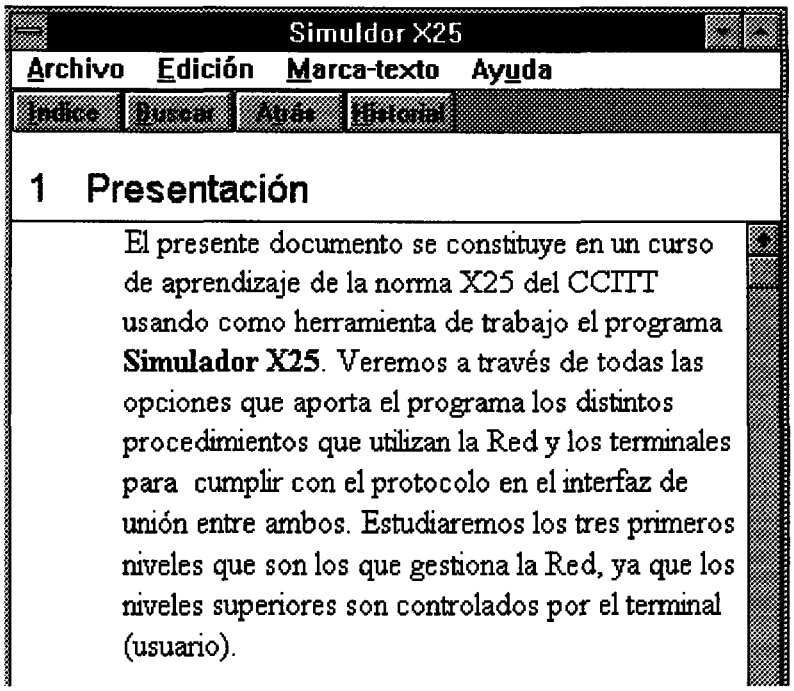


Ilustración 2.38. Ventana de ayuda

2.3.6.3.Indice

Se abre una ventana con la relación de los temas que pueden ser consultados.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.4.Iconos del AREA de TRABAJO

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.4.1.Icono de TERMINAL

Representan a los objetos terminales del programa. Los iconos no cumplen ninguna misión especifica pero aceptan que se despliegue el menu de ayuda sobre él.

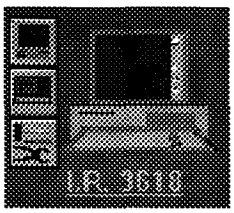


Ilustración 2.39. Icono de terminal.

2.4.1.1.Icono de ACTIVACION de terminal

Estos iconos situados en la periferia del icono principal de terminal, dan acceso a las posibilidades de ejecución de los objetos terminales del **Simulador X25**. Al hacer clic sobre este icono se abre la ventana de comunicaciones del terminal. Las posibilidades coinciden con las que se explicaron bajo el título de terminal de la barra de menús.

Ilustración 2.40. Icono de activación de terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.4.1.2.Icono de ANALIZADOR de terminal

Este icono despliega un analizador sobre el terminal que se está tratando. En su funcionamiento se asocia el concepto ETD al terminal, mientras que se llama DCE al nodo. Las opciones que se negocian en este apartado son rigurosamente idénticas a las ofertadas en la opción de Analizad del menú principal por lo que se remite a aquella explicación para describir sus posibilidades.

Ilustración 2.41. Icono de Analizador de terminal.

2.4.1.3.Icono de CONFIGURAR TERMINAL

Este icono situado en la parte inferior de la columna que se encuentra próxima a los iconos principales de terminales, abre un cuadro de diálogo donde se accede a las distintas opciones de configuración de los terminales. Tiene el mismo significado que la opción terminal del menú de configuraciones.

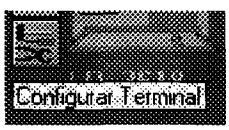


Ilustración 2.42. Icono de configuración de terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.4.2.Icono de NODO

Bajo este icono se representa la figura de los nodos de Red. Este icono no tiene ninguna función operativa, pero se asocian a él todas las actividades referentes a este elemento.

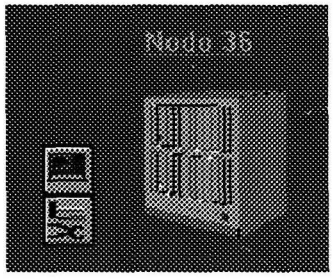


Ilustración 2.43. Icono de nodo.

2.4.2.1.Icono de ANALIZADOR de ruta

Este icono despliega un analizador sobre la ruta que se está tratando. En su funcionamiento se asocia el concepto ETD al nodo de menor numeración, mientras que se llama DCE al nodo de mayor Las opciones que se negocian en este apartado son rigurosamente idénticas a las ofertadas en el icono de Analizador de terminal por lo que se remite a aquella explicación para describir sus posibilidades.

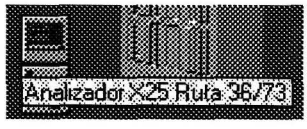


Ilustración 2.44. Icono de Analizador de ruta.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

2.4.2.2.Icono de CONFIGURACION de Red

Este icono guarda las opciones de configuración que guarda la Red de los terminales que cuelgan de ella. Esta configuración debe ser idéntica a la que guarda el icono de configuración de terminal para asegurar un correcto funcionamiento de estos elementos. Las opciones que se negocian en este apartado son rigurosamente idénticas a las ofertadas en el icono de Configuración de terminal por lo que se remite a aquella explicación para describir sus posibilidades.

Ilustración 2.45. Icono de configuración de Red.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3

Curso X25

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.1.Para empezar

El presente documento constituye en un curso de aprendizaje de la norma X25 del CCITT usando como herramienta de trabajo el programa **Simulador X25**. Veremos a través de todas las opciones que aporta el programa los distintos procedimientos que utilizan la Red y los terminales para cumplir con el protocolo en el interfaz de unión entre ambos. Estudiaremos los tres primeros niveles que son los que gestiona la Red, ya que los niveles superiores son controlados por el terminal (usuario).

El programa **Simulador X25** es una herramienta que permite crear y seguir la mayoría de las situaciones que se producen en el interfaz ETD-ETCD de una comunicación basada en el protocolo X25. La disciplina interactiva de trabajo posibilita al usuario intervenir de forma dinámica en todo momento de la ejecución del programa.

Los próximos capítulos realizan un recorrido a través de las funciones, tanto básicas como complejas, del programa **Simulador X25**. Estos ejercicios pretenden proporcionarle experiencia práctica y, al mismo tiempo, orientarle rápidamente y mostrarle formas típicas de ejecución de operaciones con el programa.

Para mejorar la comprensión del texto se ha optado por utilizar el tipo de letra 'Times New Roman' y el formato justificado en las explicaciones de carácter

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

la alineación a la derecha en los párrafos que hacen referencia al **Simulador X25** en condiciones de uso. Estos párrafos deben ser seguidos con el programa en marcha y observando los ejercicios que se sugieren.

Las imágenes que se emplean en el curso práctico han sido tomadas de la aplicación durante la ejecución del programa. Todas ellas son imágenes en color. Si el ordenador dispone de un adaptador de gráficos VGA de 16 colores o superior, podrá apreciar que la visualización de las imágenes en color es óptima. En caso de que desee trabajar con frecuencia con otro tipo de tarjeta, le recomendamos incluir en el sistema un adaptador de gráficos adecuado así como un monitor en color compatible.

Le suponemos familiarizado con ciertos términos como 'arrastrar' y 'hacer clic'. Si no comprende uno en concreto, consulte el glosario o el índice situado al final de este manual. También encontrará un glosario en el sistema de ayuda en línea del **Simulador X25** y un método para aprender a usarlo al final de este capítulo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.1.1.Generalidades sobre X25

La norma X25 es una recomendación elaborada por el CCITT que cumple con las condiciones de la arquitectura ISO para transmisión de datos en modo paquete. Especifica las características y funcionalidades del interfaz ETD-ETCD. Cubre los tres primeros niveles de la jerarquía. Para que se establezca un nivel es imprescindible que el inmediatamente inferior ya esté establecido. La información del nivel superior está contenida en los campos correspondientes de las unidades del nivel inferior. Así un paquete de nivel 3 va contenido íntegramente en una trama Info de nivel 2 y sólo podrá transmitirse cuando éste nivel esté establecido y en fase de datos.

El Nivel Físico describe las características físicas, eléctricas y funcionales del interfaz. El nivel de Enlace especifica los procedimientos para establecer la sincronización entre el terminal y la Red. Este nivel fija el formato de unas unidades de información llamadas tramas cuya misión es llevar al interfaz a las condiciones de transferencia de información para las capas superiores. Por decirlo de una forma burda, este nivel es el encargado de que la puerta de entrada de información a la Red esté abierta. El Nivel de Paquete es el que procura la unión extremo a extremo de los terminales a través de la Red, a la par que sirve de apoyo a los niveles superiores. Es el encargado de establecer las comunicaciones entre extremos y definir y supervisar unas condiciones de trabajo coherentes entre los terminales y la Red. Se encarga de que los paquetes que se transmiten por el interfaz sean los debidos y que se manden a quién corresponde.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

La norma X25 se convierte pues, en un protocolo estructurado que permite a terminales y Red fijar unas condiciones mínimas de trabajo basándose en un sistema de transferencia jerárquico de información.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.1.2.Activar el programa Simulador X25

Comenzaremos el curso con una breve reseña a los elementos que componen el **Simulador X25** para después estudiar detenidamente cada uno de ellos y las opciones que tiene asociadas. Se supone al lector de este documento con unas nociones mínimas de comunicaciones de datos, protocolo X25 y manejo del programa Windows de Microsoft.

Una vez instalado el programa **Simulador X25**, puede activarlo como cualquier aplicación Windows.

Para abrir el **Simulador X25**:

.- Lance Windows. Si necesita ayuda, consulte el manual de usuario de Microsoft Windows.

.- En el área de trabajo de Windows, haga doble clic sobre el icono de **Simulador X25**.

Ilustración 3.1. Icono Simulador X25.

Este icono está situado en la ventana de aplicación Simulador X25 (a menos que se haya cambiado el nombre del grupo o desplazado el icono) y abre la ventana de presentación de la aplicación **Simulador X25** tal y como se muestra en la ilustración.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.2. Cuadro de presentación.

Haga clic sobre el botón aceptar o simplemente pulse intro para pasar a la pantalla principal de la aplicación.

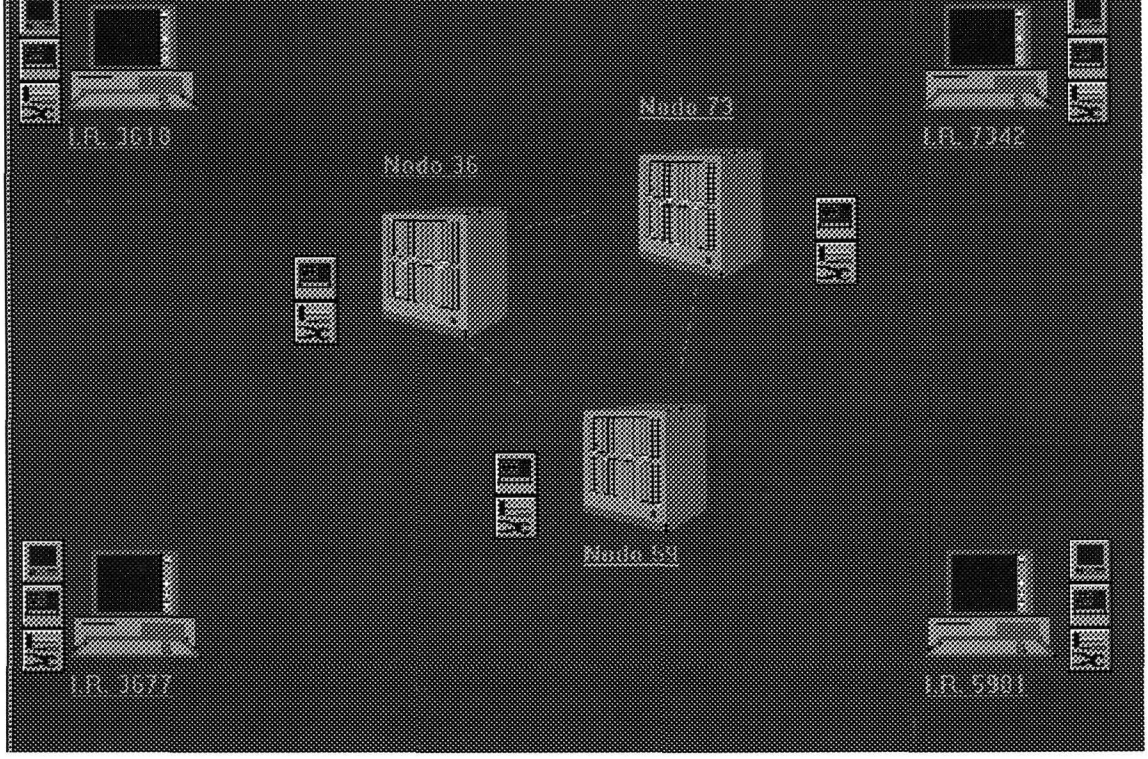


Ilustración 3.3. Pantalla principal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Los elementos que se muestran en la misma son:

3.1.2.1.La barra de título.

Denota el título de la aplicación y tiene asociados los botones de cualquier aplicación Windows.

3.1.2.2.La barra de menús desplegables.

En ella se encuentran todos los elementos y posibilidad de configuración del programa.

3.1.2.3.El área de trabajo

Ocupa la mayor parte de la superficie de la pantalla. En ella se encuentran los iconos de los elementos que conforman la aplicación y se despliegan los cuadros de diálogo que gestionan el programa. Cada uno de estos elementos generan posibilidad de diálogo y decisión por parte del usuario durante el desarrollo de la aplicación.

El programa **Simulador X25** se encarga en todo momento de que se cumpla estrictamente el protocolo X25 para las capas más bajas de la arquitectura ISO para transmisión de datos en modo paquete. Así mismo la aplicación asume las funciones de la Red y deja al usuario la potestad de control sobre los terminales.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Antes de comenzar es preciso tener definida una configuración válida en todos los elementos que conforman el **Simulador X25** para que la comunicación entre estos elementos sea ajena de vicios o dificultades impropias de una situación de trabajo. Se usará la **configuración por defecto** que brinda el programa en la opción rotulada como tal en este mismo menú (**Configuración**).

*Haga clic en este apartado (**Config.- Por defecto**), tanto los terminales como los nodos de Red del **Simulador X25** se cargan con una configuración coherente predeterminada que se encuentra almacenada en los archivos auxiliares que acompañan al programa. Esta configuración especifica para los terminales una serie de limitaciones para el establecimiento de canales o vías de comunicación pero no les configura ninguna atribución especial en cuanto a características de los mismos. En el capítulo de configuraciones se tratará en más profundidad este asunto.*

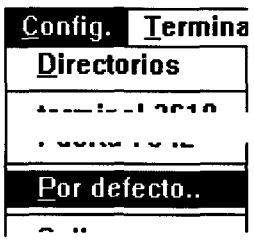


Ilustración 3.4. Configuración por defecto.

Una vez que se arranca el programa, el seguimiento del mismo se realiza a través de la superficie de trabajo mediante la interacción con las diferentes ventanas configurables a las que se va teniendo acceso.

Para abandonar la aplicación puede hacerse cerrando la ventana del **Simulador X25** como cualquier aplicación de Windows o bien siguiendo la secuencia **Config-Salir** de la barra del menú principal.

No haga uso de esta opción en este momento si desea continuar con el tutorial. Simplemente se ha inclído esta información en este punto para dejar constancia de la misma y permitir al lector abandonar la aplicación en el momento que lo desee.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

constancia de la misma y permitir al lector abandonar la aplicación en el momento que lo desee.

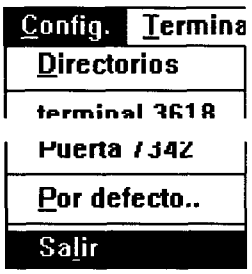


Ilustración 3.5. Salir.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.2. TERMINALES

En este capítulo del tutorial sobre la norma X25 se estudiarán los procedimientos básicos para el establecimiento, uso y gestión por parte del ususario de los canales de comunicación entre dos terminales.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.2.1.Concepto de terminal

Una Red de comunicaciones se define como el conjunto de elementos conectados a un sistema de transmisión que permite el flujo de información entre ellos. Bajo el protocolo X25 se define el concepto de terminales como los elementos que se conectan a la Red, capaces de soportar los niveles superiores de la arquitectura ISO para transmisión de datos en modo paquete y al mismo tiempo pueden gestionar el protocolo en el interfaz (niveles inferiores). Los canales son aquellas vías virtuales de comunicación que se establecen entre dos terminales. Estos pueden tener tantos canales activos como sean capaces de gestionar y son independientes entre sí.

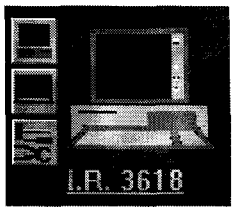


Ilustración 3.1. Terminales.

Bajo las opciones que se ocultan tras la palabra de clave de **terminal** se definen y controlan las acciones que, a nivel de usuario, realizan los mismos.

*Con el programa en condiciones de configuración por defecto, haga clic en la palabra clave **Terminales** de la barra de menús. Se desplegará un menú con los terminales que se hayan disponibles. En este momento lo están todos.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.2. Activación de terminales.

Abra el terminal 3618. Observe como la línea de unión entre el terminal y su nodo cambia de color indicando la activación de aquél. La barra de estado al pie de la pantalla visualiza los estados a los que accede el terminal a medida que se establece el Nivel de Enlace. Cuando se abre un terminal el ***Simulador X25*** *envía de forma automática las tramas de Nivel 2 para completar el establecimiento del enlace. Estas se estudiarán en el próximo capítulo de* ***Analizadores****.*

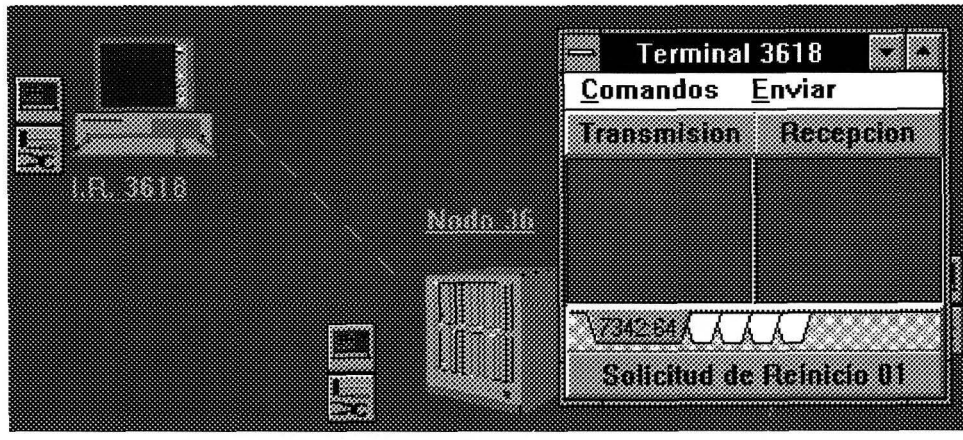


Ilustración 3.3. Ventana de terminal.

Una vez establecido el Nivel de enlace, el terminal ya puede iniciar los procedimientos adecuados para establecer el nivel superior. Ello lo puede llevar a cabo por alguno de los canales que tiene disponibles.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.2.2.Concepto de canal

Un canal lógico es una vía virtual de comunicación entre dos estaciones que se sustenta sobre diferentes medios de transmisión físicos. Estos canales se definen a nivel local en contratación con la Red. Al ser asignado un identificativo de Red a un terminal, lo que equivale a ser dado de alta en la misma, se contratan tantos canales como el terminal requiera para sus comunicaciones. El número de canales que tiene asignado un terminal permanecerá invariable hasta que se renegocie el contrato. El **Simulador X25** contempla esta posibilidad en las opciones de configuración. Los canales se dividen en dos categorías: permanentes y conmutados.

3.2.2.1.Canales virtuales permanentes

Se define así a aquellos en los que el establecimiento del mismo corre a cargo de la Red y se les supone operativos desde que los terminales establecen el Nivel de Enlace. Cuando esto sucede, tanto el terminal como la Red se intercambian sendos paquetes de rearranque para confirmar la disposición de este canal y no es necesario enviar ningún otro paquete de Nivel 3 para conectar con el colateral.

3.2.2.2.Canales virtuales conmutados

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.2.2.2. Canales virtuales conmutados

Se define así a aquellos que el terminal establece, mantiene y libera libremente haciendo uso de los diferentes paquetes de Nivel 3. Los canales conmutados se dividen en tres tipos:

3.2.2.2.1. **Entrantes**

Son aquellos en los que el terminal no puede generar paquetes de llamada para conectarse con otros, aunque si puede recibir llamadas externas.

3.2.2.2.2. **Entrantes/Salientes**

Son aquellos en los que el terminal puede generar y recibir las llamadas.

3.2.2.2.3. **Salientes**

Son aquellos por los que el terminal no puede recibir ninguna llamada, aunque si las puede generar.

Para averiguar los canales que un terminal tiene establecidos en un momento dado, no tiene más que mirar en la parte inferior de la ventana de terminal. Cada una de las solapas presentes representan los canales disponibles. La lectura en su interior denota el identificativo de Red del terminal distante y, separado por dos puntos, el número de canal, en notación hexadecimal, que se emplea en la unión con aquél. El tipo de canal a que hace referencia se distingue por el color de fondo de la ventana correspondiente. En el caso de los conmutados es necesario comprobar la configuración del terminal para averiguar si es entrante, saliente o mixto.

Observe que ya existe un canal establecido con el terminal 7342 a través del canal lógico 100 (64H). Este es un canal permanente. Pique sobre esta solapa. Observe que el fondo de la pantalla es de color verde oscuro. Esto es

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.4. Ventana de terminal.

Active el terminal 7342. Esta vez utilice el icono de ***activación de terminal*** *situado en la periferia del citado terminal.*

Ilustración 3.5. Icono de activación de terminal.

El terminal 7342 tiene contratados dos canales virtuales permanentes, uno con el terminal 3618 y otro con el terminal 5981. Observe como el fondo de la pantalla en la solapa del canal que le une con el terminal 3618 toma el color azul claro. Lo propio pasa con su colateral. Esto es debido a que dicho canal se haya establecido y está operativo. Haga clic en la zona de transmisión y escriba un texto, por ejemplo la palabra 'tururu'. Pulse la tecla enter. El programa pasa esta información de nivel superior a las rutinas de transmisión para convertir los datos a nivel de usuario en paquetes de Nivel 3 para ser enviados. La información aparece inmediatamente en la ventana de recepción del colateral.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

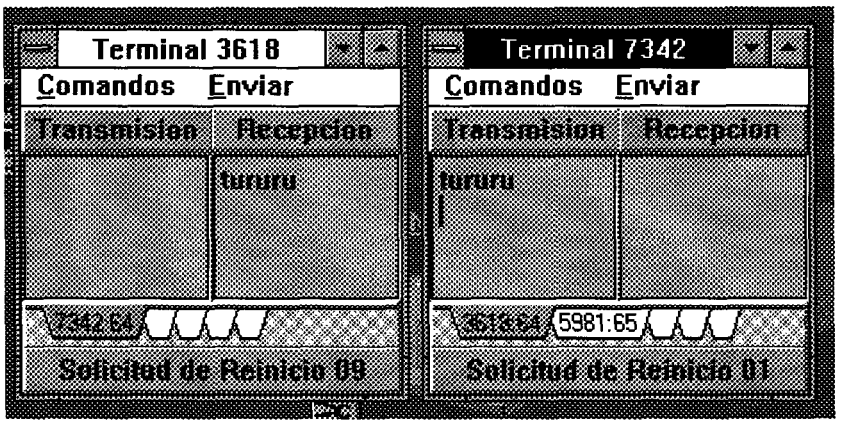


Ilustración 3.6. Transmisión de datos.

Como puede ver en la parte inferior de la ventana de transmisión del terminal 3618, existen cuatro solapas más de canales posibles. Estas no tienen asociadas ninguna numeración en este momento. Esto es debido a que se trata de canales virtuales conmutados que no están establecidos todavía.

Para establecer, gestionar o liberar canales conmutados es imprescindible usar los comandos que especifica la norma.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.2.3.Comandos

Son unidades organizadas de información que se intercambian entre el terminal y la Red y que afectan al control del interfaz. Existen varios tipos que se estudiarán con detenimiento en los próximos capítulos y sólo tienen significado en el nivel ISO en que sean tratados. Se señalan a continuación los comandos que se emplean para gestionar los canales virtuales.

3.2.3.1.Llamada.

Es el comando que se usa para establecer un canal virtual conmutado entre dos terminales. El terminal que inicia el procedimiento lanza un paquete de llamada de Nivel 3 en el que indica a la Red, entre otras cosas, el canal que utiliza, las longitudes y los números de Red llamante y llamado, etc. Un paquete de solicitud de llamada debe usar el canal lógico que, estando disponible, tenga el menor número dentro de los márgenes contratados con la Red. La llamada entrante usará el canal lógico disponible con mayor número del terminal destino.

Active ahora el terminal 3677. Observe que éste no tiene canales permanentes definidos. Haga clic sobre el menú de ***Comandos*** *y elija la opción de* ***Llamada****. Llame al terminal 3618.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

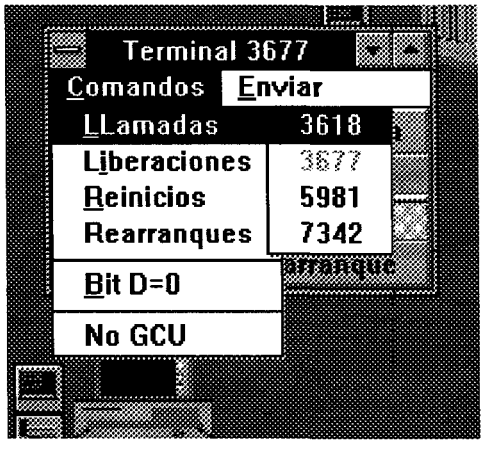


Ilustración 3.7. Comandos.

La llamada progresa por la Red y se consigue la conexión entre los dos terminales a través del canal lógico 100 (64H, el de numeración más baja disponible) del terminal origen y canal 104 (68H, el de numeración más alta disponible) del terminal destino. Puede enviar el texto que desee entre ambas estaciones insertándolo en las correspondientes ventanas de transmisión de dichos terminales.

Pruebe a hacer una llamada desde el 3618 al terminal 5981. La Red devuelve un paquete de liberación con causa 09 (terminal distante fuera de servicio). Se tiene notificación de ello en la barra de estado de la ventana de terminal.

Dos terminales pueden tener cuantos canales y del tipo que sean conectados entre sí siempre y cuando los puedan establecer. Desde el terminal 7342 lance una llamada al terminal 3618. Observe que se establece entre ambas estaciones un nuevo canal, éste de carácter conmutado.

Ilustración 3.8. Aceptación de Llamada.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.2.3.2.Liberación.

Es el comando que se usa para finalizar una comunicación y liberar un canal conmutado. Esta fase puede ser iniciada por un terminal o por la Red en caso de estar congestionada.

Sitúese en el terminal 3618 y pique sobre la solapa del CVC que lo comunica con el 7342. Elija la opción ***Liberaciones*** *del menú de comandos. Observe como termina la comunicación y los correspondientes mensajes de confirmación que se reciben en los terminales con causa de terminal.*

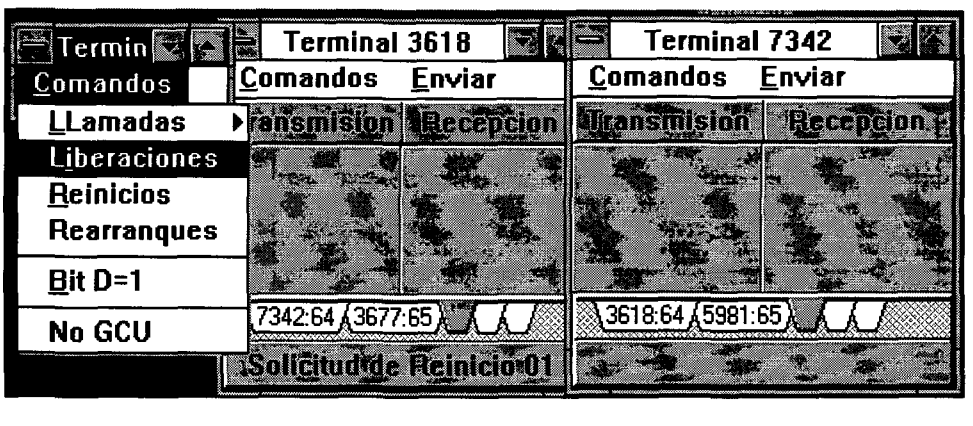


Ilustración 3.9. Liberaciones.

Desactive el terminal 7342. Para ello basta con cerrar la ventana del mismo. Observe que la Red informa de este evento a los terminales distantes con los que está conectado enviando los correspondientes paquetes de liberación o rearranque según sea el tipo de canal con que estuvieran conectados. El fondo de los canales permanentes vuelve a su tono verde oscuro.

3.2.3.3.Reinicio.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.2.3.3.Reinicio.

El objetivo de este paquete es poner en estado inicial un circuito virtual. El control de las comunicaciones dentro de un canal se realiza usando unos contadores dentro de un sistema llamado de ventana deslizante que se explicará más adelante. Hacer un reinicio a un canal equivale a fijar los parámetros de la ventana a 0. A nivel de usuario el reinicio es una acción transparente. El procedimiento para llevar a cabo la reiniciación puede tener origen en un terminal o en la Red.

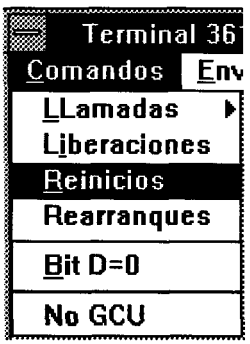


Ilustración 3.10. Reinicios.

Sitúese en el terminal 3677 y seleccione el canal conmutado que tiene establecido. Envíe de nuevo nuestro texto favorito 'tururu'. Haga lo propio con origen en el terminal 3618. Envíe un paquete de reinicio a la Red. Observe que a pesar de que dicho paquete tiene efecto sobre los parámetros de control de flujo del enlace, no se aprecia acción alguna (es transparente) a nivel de usuario.

3.2.3.4.Rearranque.

Este paquete se utiliza para iniciar todo el interfaz. El efecto que produce es el mismo que si hiciéramos un reinicio sobre todos los circuitos virtuales permanentes y una liberación sobre los conmutados que hay establecidos. Los rearranques son siempre enviados por el canal 0. Este canal lo tienen reservado todos los terminales

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

terminal distante (o a los terminales) y reiniciar los canales virtuales permanentes del terminal remoto (o los terminales remotos).

La Red puede mandar también rearranques por un error de procedimiento, un fallo en ruta o un fallo en un centro de Red. En tal caso la Red mandará rearranques a las dos estaciones del canal que mantienen comunicación, esperando confirmaciones de rearranque de ambas estaciones.

*Si envía un paquete de **Rearranque** desde cualquier terminal a la Red en este momento, obtendrá la misma respuesta que en el apartado anterior respecto a los canales permanentes: una acción invisible al nivel en que lo estamos tratando y cancelará todas las comunicaciones por los canales conmutados. En el próximo capítulo se verán con detalle los procedimientos que sigue la Red ante la recepción de estos paquetes.*

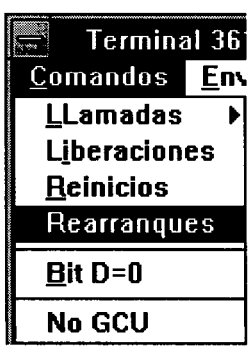


Ilustración 3.11. Rearranque.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.3.RUTAS

El presente capítulo trata de explicar los procedimientos internos de Red y la gestión de transferencia de paquetes que realiza para llevar al extremo distante la información originada en un terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.3.1.Concepto de Red

Se define una Red de comunicaciones como el conjunto de oredenadores, medios de transmisión, e interfaces que posibiliten el acceso de los terminales de usuario a la misma y, a través de ella, a otros terminales.

La mayoría de los países avanzados han acometido en los últimos años la tarea de desarrollar una Red pública conmutada específicamente orientada para la transmisión de datos digitales, teniendo en cuenta los inconvenientes fundamentales de tipo social y político indicados en las redes dedicadas, así como las limitaciones del uso de las redes públicas conmutadas existentes actualmente para constituir el soporte de telecomunicaciones requerido por la telemática, al haber sido diseñadas para otras aplicaciones.

Dichas redes se apoyan en la planta telefónica de transmisión, por lo que se requieren en la actualidad los equipos de conversión de señales digital/ analógico, pero utilizan centrales de conmutación especialmente desarrolladas para la conmutación de señales digitales y que, la casi totalidad de las mismas, se basan en técnicas de conmutación de paquetes. En estas redes públicas de conmutación de paquetes se transmite por la Red la información generada mediante un terminal o un ordenador en forma de paquete, realizándose en los centros de conmutación un encaminamiento de dichos paquetes hasta el terminal u ordenador de destino a través de los circuitos que enlazan dichos centros de conmutación formando una Red mallada.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.3.2.Concepto de ruta

Se llama ruta a la vía de comunicación que une a dos nodos de Red. Estas rutas pueden soportar el tráfico de información que estos centros de Red tengan establecidos y pueden estar constituídos por uno o más soportes físicos de transmisión. Siempre que exista posibilidad directa de diálogo entre los nodos implicados, se considerará que la ruta se halla activa.

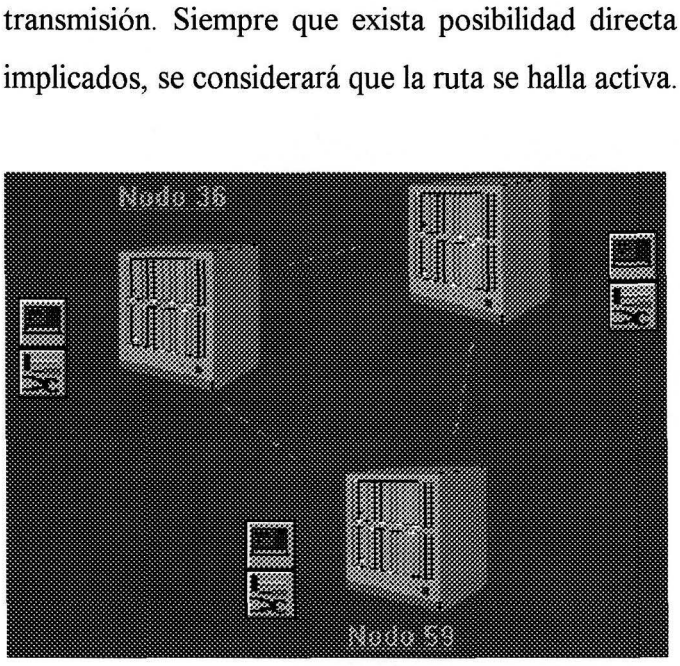


Ilustración 3.1. Rutas.

El tráfico de información que se produce en una ruta es dirigido por el gestor de la Red libremente, no existiendo ninguna recomendación o norma formulada a tal efecto. A pesar de ello, muchas redes siguen con ciertas modificaciones los principios básicos de la norma X25 para su gestión interna. Para la interconexión de Redes se ha fijado la norma X75, que tiene significado a nivel internacional.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.3.3.Concepto de encaminamiento

Los procedimientos de encaminamiento gestionan las colas de salida de los nodos de la Red, decidiendo cuándo y dónde debería ser transmitido un determinado mensaje, tratando al hacerlo de minimizar el retardo del mensaje en cuestión y de optimizar el caudal o flujo efectivo de acuerdo con las condiciones del tráfico en las rutas. Existen diversas técnicas de encaminamiento según el algoritmo empleado para determinar la ruta y que pueden encajarse dentro de las siguientes categorías:

Tipo determinístico. En este caso los paquetes entre nodos circulan siguiendo unas rutas predeterminadas que se definen en unas tablas que gestionan la Red.

Tipo Adaptativo o dinámico. Se basan en un control lógico del tráfico entre nodos guiado por programas de explotación de Red que definen en cada momento la ruta idónea para los paquetes según las circunstancias temporales de las líneas de unión.

El programa ***Simulador X25*** *utiliza un método particular de datagrama (tipo dinámico) según el cual los paquetes a ser transferidos de un nodo a otro disponen como vía principal de una ruta directa entre ellos y en caso de caída de la misma pasarían a buscar una ruta alternativa.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.3.4.Control de rutas

Esta opción da acceso al control sobre las rutas del programa. Funciona como interruptor biestable permitiendo cortar/activar las rutas entre nodos de Red. Ante un corte de línea, el **Simulador X25** reencamina automáticamente el tráfico que era transferido por la misma hacia una vía alternativa. El estado de una ruta se hace visible en el área de trabajo por el color en el que aparece la misma: fijo en rojo para una ruta cortada y oscilante en azul-amarillo para una ruta activa.

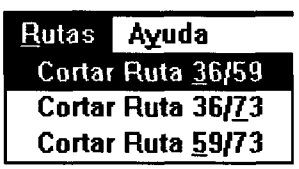


Ilustración 3.2. Control de rutas.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.ANALIZADORES

Bajo este epígrafe se estudian las opciones que se encuentran en esta herramienta. Podremos analizar los datos que corren por las líneas y actuar en consecuencia. Se estudiará así mismo en este capítulo el formato de tramas y paquetes y el protocolo para el establecimiento de los Niveles 2 y 3 de la norma X25.

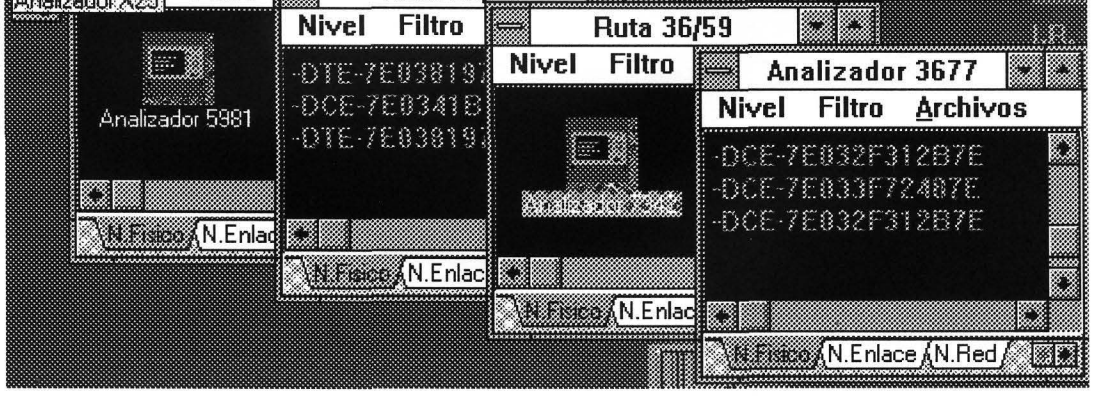


Ilustración 3.1. Analizadores.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.1.Concepto de Analizador.

Los analizadores son elementos que permiten visualizar el tráfico de la línea sobre la que se insertan. Su punto de conexión es el propio interfaz de unión entre equipo terminal de circuito de datos y el terminal o nodo. La conexión se realiza en derivación tipo T. Las funciones que incluyen los analizadores permiten además un cierto tratamiento de análisis sobre los datos recogidos. Así se pueden grabar los mismos sobre ficheros prefijados siempre que se den unas condiciones predeterminadas que se especifiquen, activar/desactivar las capturas, filtrar eventos, etc.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.2.Concepto de nivel

El tráfico de información bajo la norma X25 se estructura en conjuntos ordenados de bits que tienen significado propio según sea el nivel en que se traten. Así pues los bits se agrupan formando unidades de información denominadas tramas en el Nivel 2 y dentro de ellas, si la hubiera, iría incluida la información de Nivel 3, ordenada en unidades llamadas Paquetes.

Ilustración 3.2. Niveles.

3.4.2.1.Nivel de Red

Estudia los paquetes de Nivel 3 que se transmiten por la línea. El Nivel 3 es el encargado de mantener las comunicaciones extremo a extremo. Para ello gestiona el establecimiento y liberación de canales, las características lógicas de la comunicación, el control de flujo y el soporte al nivel superior fundamentalmente. Todo paquete de Nivel 3 es una estructura de datos que tiene significado propio.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.2.1.1.Formato de los paquetes.

Cualquiera que sea el paquete de Nivel 3, va íntegro en el campo de datos de la trama Info de Nivel 2.

Estructura de los paquetes

|Q|D|Mod|GRUPO|CANAL|TIPO, C|DATOS O PARAMETROS|

Q D Mod GRUPO

CANAL

TIPO

DATOS o PARAMETROS

....

El bit **Q** permite distinguir los paquetes enviados por niveles superiores para advertir al Nivel 3 que los datos que aparecen en este paquete deben ser analizados por el nivel superior.

El bit **D** indica al extremo colateral la exigencia de una respuesta inmediata.

Mod es un grupo de dos bits que indica el módulo del paquete, o sea el margen de numeración de paquetes sin que se retorne al valor inicial. Si Mod=01, el módulo es igual a 8 y el margen de numeración de los paquetes puede tomar valores entre 0 y 7 y el módulo será 128 si Mod=10. El resto de los valores de MOD no están permitidos.

El campo **GRUPO** (4 bits) concatenado con el **CANAL** (8 bits) forman un número binario de 12 bits encargado de designar el número de canal virtual. Por tanto se tienen de 0 a 4095 canales, como el canal 0 esta reservado para señalización se tienen 4095 canales posibles para transferencia de datos.

Los campos **TIPO** (7bits)+ **C** (1bit) designan el tipo de paquete. Cuando el bit de C=1 (número de Canal impar) es un paquete de control. Por otro lado, cuando el bit de C=0 (número de canal par) es un paquete de datos. Cuando C es par, el campo TIPO cambia en el significado de sus bits. Los bits del campo TIPO pasan a denominarse:

[Pr][M][Ps][0]

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

[Pr][M][Ps][0]

Donde Pr (3bits) y Ps (3bits) están asociados al control de flujo.

Ps es el número de secuencia del paquete enviado.

Pr menos uno es el número de secuencia del último paquete que se ha reconocido en recepción.

M indica, en los paquetes que lo tengan activo, que van a formar un grupo en secuencia o mensaje.

Cierre todos los terminales si estuvieran activos. En condiciones de configuración por defecto abra el ***Analizador*** *asociado al terminal 3618. pique sobre la solapa rotulada como* ***N.Red*** *al pie de la ventana o haga la secuencia:* ***Nivel-Red*** *sobre los menús desplegables.*

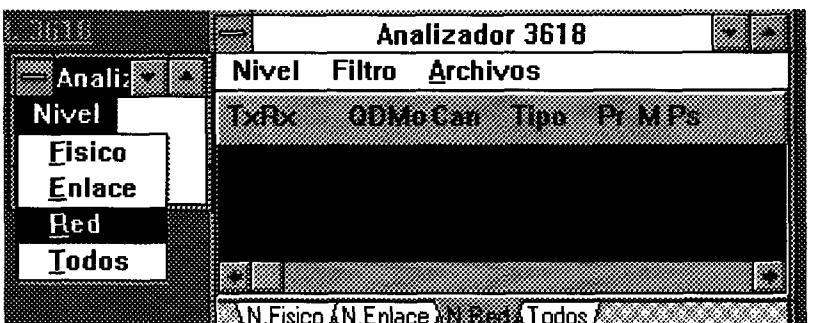


Ilustración 3.3. Nivel de Red.

En este Nivel se enseña una cabecera para distinguir los datos que se muestran:

TxRx Indica el emisor y receptor del paquete en el interfaz.

QDMoCan Indica el estado de los bits Q y D, así como el módulo activo y el canal lógico por el que se envía el paquete.

Tipo Indica el tipo de paquete de Nivel 3.

PrMPs Indica el valor de los contadores de paquetes en transmisión y recepción así como el estado del bit 'más datos' M.

En este momento sólo se podrán ver los paquetes de Nivel 3 que se envíen a línea. Estos paquetes sólo están presentes durante el establecimiento y

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Abra el terminal 3618. Inmediatamente se cruzan en línea dos ***Solicitudes de Rearranque*** *que son los paquetes que se envían terminal y Red para establecer el Nivel 3. Como se ve, los rearranques se envían única y exclusivamente por el canal 0. El estado de los bits Q y D permanece inactivo y el valor del módulo de la ventana de transmisión es 01 (el diseño del* ***Simulador X25*** *no permite trabajar con ningún valor diferente). En cuanto a los contadores Pr y Ps no se muestran ya que estos paquetes no atienden a este control por ser no numerados.*

Ilustración 3.4. Rearranque.

*3.4.2.1.1.1.**Rearranque.***

Se utiliza para iniciar todo el interfaz. Cuando un terminal establece el Nivel de Enlace, lo normal es que cruce con la Red sendos paquetes de rearranque para advertir al Nivel 3 de este hecho. Sin embargo este procedimiento no es obligatorio según la norma. Los rearranques son siempre enviados por el canal 0. Si un terminal manda una solicitud de rearranque a la Red, ésta se encarga de liberar todos los canales virtuales conmutados transmitiendo al terminal distante (o a los terminales) paquetes de liberación y reiniciar los canales virtuales permanentes del terminal remoto (o los terminales remotos) mandando paquetes de reinicio.

3.4.2.1.1.1.1.**Formato del paquete de solicitud de rearranque.**

bin: [0001][0000][00000000][11111011][CAUSA][CODIGO]

hex: 10 00 FB causa, código

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.2.1.1.1.1.Formato del paquete de solicitud de rearranque.

bin: [0001][0000][00000000][11111011][CAUSA][CODIGO]

hex: 10 00 FB causa, código

Q D Mod Grupo	0001 0000	10
Canal	0000 0000	00
Tipo	1111 1011	FB
Causa	xxxx xxxx	XX
Código	xxxx xxxx	XX

3.4.2.1.1.1.2.Causas de rearranque:

01 Error de procedimiento local.

03 Congestión de Red.

07 Red en operación.

>7F Códigos de ETD.

3.4.2.1.1.1.3. Formato del paquete de confirmación de rearranque.

bin: [0001][0000][00000000][11111111]

hex: 10 00 FF

Q D Mod Grupo	0001 0000	10
Canal	0000 0000	00
Tipo	1111 1111	FF

Los siguientes paquetes que se intercambian terminal y red son los de ***Solicitud de reinicio*** *y* ***Aceptación de reinicio*** *motivados por la existencia del CVP que lo une con el terminal 7342. Como se ve el envío de estos paquetes no tiene efecto sobre la ventana de comunicaciones.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.5. Reinicio.

*3.4.2.1.1.2.**Reinicios.***

El efecto de este paquete es poner en estado inicial un circuito virtual (parámetros del control de la comunicación y control de la ventana de transmisión a 0). Al igual que en otros procedimientos, se utiliza un paquete de **solicitud de reinicio** enviado por la estación emisora y otro de **confirmación de reinicio** enviado por el colateral. El procedimiento para llevar a cabo la reiniciación puede tener origen en un terminal o en la Red. Cuando tiene origen en un terminal, este envía a la Red un paquete de **solicitud de reinicio**, ésta lo progresa al terminal destino como una **indicación de reinicio**, que, a su vez, responde con una **confirmación de reinicio**. La Red la progresa hacia el terminal emisor de la solicitud. En caso de que sea la Red la que origina el reinicio, ésta envía sendos paquetes de solicitud a las dos estaciones y estas contestarán con sendos paquetes de **confirmación de reinicio**.

3.4.2.1.1.2.1.**Formato del paquete de solicitud (indicación) de reinicio.**

bin: [0001][GRUPO][CANAL][00011011][CAUSA][CODIGO]

hex: 1X XX 1B causa código

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	0001 1011	1B

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Tipo	0001 1011	1B
Causa	xxxx xxxx	XX
Código	xxxx xxxx	XX

Las posibles causas para el reinicio son:

00 Reiniciación por ETD

01 ETD fuera de servicio. (Sólo en CVP)

03 Error de procedimiento remoto.

05 Error de procedimiento local.

07 Congestión de Red.

09 ETD remoto en operación. (Sólo en CVP)

0F Red en operación.

11 Destino incompatible.

1D Red fuera de servicio. (Sólo en CVP)

>7F Códigos de ETD.

3.4.2.1.1.2.2. Formato del paquete de confirmación de reinicio.

bin: [0001][GRUPO][CANAL][00011111]

hex: 1X XX 1F

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	0001 1111	1F

3.4.2.1.2. Control de canales virtuales conmutados.

Los canales virtuales conmutados son aquellos que el terminal establece gestiona y libera mediante los mecanismos y controles que le aporta la norma para ello. Estos canales son temporales por definición y corresponde a los terminales la iniciativa para su establecimiento, mantenimiento o liberación. Pueden ser liberados por los mismos de forma natural o por la Red de forma

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

*3.4.2.1.2.1.**Establecimiento de la llamada.***

Define el procedimiento a seguir para establecer un CVC con un terminal distante. Tiene origen siempre en un terminal y nunca en la Red. Esta fase se realiza utilizando los paquetes de control **Solicitud de llamada** y **Llamada aceptada**. El proceso seguido es el siguiente. El ETD que quiere establecer el CVC envía a la Red un paquete de solicitud de llamada pidiendo el establecimiento de la misma e indicando la dirección de destino. La Red utiliza esta dirección para encaminar el paquete hacia el ETD destino, pasándoselo como una llamada entrante. Entonces el ETD destino envía un paquete de llamada aceptada que es transmitido a través de la Red al ETD origen de la llamada.

En el paquete de solicitud de llamada pueden aparecer los campos de Longitud de facilidades y facilidades, en los que se expresan las características que el ETD origen propone para la llamada virtual y que pueden ser cobro revertido, prioridad, longitud máxima del paquete, tamaño de ventana, etc. Si el ETD destino no está de acuerdo con alguno de estos parámetros, puede modificarlos indicándolo en el campo F de la "llamada aceptada". En cualquier caso los nuevos valores propuestos sólo podrán variar de los originales para acercarse más a los valores por defecto, no para diferir más de éstos.

3.4.2.1.2.1.1.**Formato del paquete de llamada.**

bin:[QDMod][Grupo][Canal][00001011][LD1][LD2][DIR][00LF][F][dat

hex: 1X XX 0B ...

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	0000 1011	0B
LD1 LD2	xxxx xxxx	XX
NR destino		
NR origen		
Control de red		
proc. interno de red		
Long. Facilidades	00xx xxxx	
Facilidades		
Datos de usuario	<0128	

LD1 Indica la longitud en bits del NR (número de red) del terminal llamado.

LD2 Indica la longitud en bits del NR (número de red) del terminal llamante. Este campo es opcional.

NR destino. En este campo se codifica el NR (número de Red) del terminal llamado

NR origen. En este campo se codifica el NR (número de Red) del terminal llamante. Este último es opcional.

Control de red: La Red tiene la facultad de incluir en el paquete de solicitud de llamada unos octetos para uso propio entre centros de retransmisión. Al terminal no se le entrega este trozo. Por tanto esta información sólo es posible verla en los analizadores que visualizan el tráfico en las rutas y que se estudiará en próximos capítulos.

LF. Longitud del campo de facilidades. Indica el número de facilidades de usuario de que va a hacer uso el terminal en la llamada en curso.

F. Campo de facilidades de usuario, previamente negociadas en la contratación de las facilidades de usuario.

Dat. El paquete de llamada puede incluir información de usuario si así lo contrata con la Red. Es una facilidad opcional.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

00,0F	Separador facilidades extremo a extremo
01,01	Cobro revertido.
01,80	Selección rápida sin restricción.
01,C0	Selección rápida con restricción.
03,GG	Grupo cerrado se usuarios (abreviado).
08,XX	Notificación modificación dirección del ETD llamado.
	01 Redirección por ETD llamado ocupado.
	07 Grupo de captura.
	09 Redirección por ETD llamado fuera de servicio.
	0F Redirección sistemática.
	>7F Originados en ETD.
09,XX	G.C.U. con acceso salida.

3.4.2.1.2.1.2.Formato del paquete de llamada aceptada.

bin: [IGF][GRUP][CANAL][00001111][LD1][LD2][DIR][00 LF][F]

hex: 1X XX 0B ....

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	0000 1011	0B
LD1 LD2	xxxx xxxx	XX
NR destino		
NR origen		
Loong. Facilidades	00xx xxxx	
Facilidades		
Datos de usuario	<0128	

Los campos **LD1, LD2, DIR, LF y F** se codifican de igual forma que en el paquete de solicitud de llamada.

Un paquete de solicitud de llamada debe usar el canal lógico que, estando disponible, tenga el mayor número dentro de los márgenes acordados (contratados)

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Un paquete de solicitud de llamada debe usar el canal lógico que, estando disponible, tenga el mayor número dentro de los márgenes acordados (contratados) con la Red. Este procedimiento de elección de canales desde el ETD hacia la Red, así como el procedimiento contrario desde la Red hacia el ETD, minimizan el riesgo de colisiones en las llamadas.

El paquete de llamada aceptada es la respuesta a una solicitud de llamada. En él se especifica el mismo canal por el que se ha solicitado la llamada. Una vez recibido este paquete, se está en disposición de transferencia de datos.

Para ver estos paquetes a Nivel 3, sitúese de nuevo en el terminal 3618 y lance una llamada al 7342. Vea como se establece con éxito la comunicación. Las facilidades especificadas por defecto en el ***Simulador X25*** *son nulas, por lo que este campo en los paquetes de llamada permanece a cero como ya se verá cuando se estudie este paquete a nivel físico.*

Ilustración 3.6. Llamada.

3.4.2.1.2.2. ***Liberación de la llamada.***

En este proceso se utilizan los paquetes de control **Solicitud de liberación** y **Confirmación de liberación**. En el paquete de solicitud de liberación se indica la causa que motiva la liberación de la llamada y una información adicional de diagnóstico.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

liberación. El terminal remoto responde con una confirmación de liberación que será progresada por la Red hasta el terminal origen de la solicitud.

3.4.2.1.2.2.1. **Formato del paquete de solicitud de liberación.**

bin; [QDMod][GRUPO][CANAL][00011011][CAUSA][CODIGO] [DIAGNOSTICO]

hex: 1X XX 1B ....

Q D Mod Grupo	0001 xxxx	1x
Canal	xxxx xxxx	xx
Tipo	0001 1011	1B
Causa		
Código		

Las posibles causas para la liberación son:

00	Liberación por ETD
01	Número ocupado.
03	Solicitud de facilidad inválida.
05	Congestión de Red.
09	Fuera de servicio.
0B	Acceso prohibido.
0D	Número desconocido.
11	Error de procedimiento remoto.
13	Error de procedimiento local.
19	No aceptación de cobro revertido.
21	Destino incompatible.
29	No aceptación de selección rápida.
>7F	Códigos de ETD.

3.4.2.1.2.2.2. **Formato del paquete de Confirmación de liberación.**

bin: [IGF][GRUPO][CANAL][00011111]

hex: 1X XX 1F

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.2.1.2.2.2. Formato del paquete de Confirmación de liberación.

bin: [IGF][GRUPO][CANAL][00011111]

hex: 1X XX 1F

Q D Mod Grupo	0001 xxxx	1x
Canal	xxxx xxxx	xx
Tipo	0001 1111	1F

*Repita los pasos dados cuando se explicaron las opciones del menú de **Terminal**. Active ahora el terminal 3677. Haga clic sobre el menú de **Comandos** y elija la opción de **Llamada**. Llame al terminal 3618. Se consigue la conexión entre los dos terminales a través del canal lógico 100 (64H) del terminal destino y canal 104(68H) del colateral. Pruebe a hacer una llamada desde el 3618 al terminal 5981. La Red devuelve un paquete de liberación con causa 09 (terminal distante fuera de servicio). Cierre ahora el terminal 3677 y observe como el 3618 recibe una liberación por parte de la Red con la misma causa. En cambio, si libera el canal que tiene establecido con el terminal 7342, la causa será 00, liberación por terminal.*

Ilustración 3.7. Liberaciones.

*Sitúese en el terminal 7342. Este terminal posee un CVP con el terminal 3618. Seleccione este canal y envíe el mensaje 'tururu'. Observe que la Red transforma esta información en un paquete de tipo **DATOS** y lo progresa hacia*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.8. Datos.

*3.4.2.1.2.3.**Paquetes de datos.***

Los paquetes de datos se caracterizan por tener el bit de control a cero y se utilizan tanto en circuitos virtuales permanentes como en circuitos virtuales conmutados. Su formato es el siguiente:

Q D MOD | GRUP | CANAL | Pr | M | Ps | 0 | ....datos.....|

bin: [0001][GRUPO][CANAL][xxxxxxx0]

hex: 1X XX XXpar

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	Pr M Ps 0	XX par
Datos	....	....

Cuando Mod=01, esto es, cuando estamos en módulo 8, los campos Pr y Ps tienen longitud de tres bits y es suficiente con tres octetos de cabecera. A este módulo se le llama formato normal. Cuando estamos en módulo 128, son necesarios 7 bits para Ps y Pr, con lo que son necesarios 4 octetos para la cabecera.

3.4.2.1.3. **Control de flujo.**

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

se le llama formato normal. Cuando estamos en módulo 128, son necesarios 7 bits para Ps y Pr, con lo que son necesarios 4 octetos para la cabecera.

3.4.2.1.3. **Control de flujo.**

Durante la fase de transferencia de datos hay un flujo continuo de información desde y hacia cada ETD. La transmisión de paquetes se controla para cada canal lógico y en ambos sentidos por el mecanismo de ventana deslizante, utilizando los contadores Pr y Ps. Las secuencias Pr y Ps son los mecanismos que se introducen para numerar los paquetes. Con Ps se numeran los paquetes que se envían desde el terminal local mientras que Pr indica el valor del próximo paquete que se espera recibir de la estación distante. El terminal local podrá enviar paquetes con Ps comprendido entre 0 y el valor del módulo-1, siempre que este valor sea inferior al del contador Pr del extremo distante más el valor de la ventana de transmisión. Se llama ancho de ventana de transmisión al número máximo de paquetes que puede enviar un terminal sin recibir acuse de recibo por el colateral. Se define la ventana de transmisión como el intervalo de valores de posibles de Pr que tiene como valor inferior el del último paquete que ha sido confirmado en recepción y como valor máximo el valor anterior más el ancho de ventana. Una vez que el contador Ps del terminal local alcance el valor superior de ventana, debe esperar la confirmación de entrega de la estación distante para actualizar el límite inferior de la ventana y seguir la transmisión. La actualización de los límites inferiores de las ventanas de transmisión del terminal distante se consigue enviándole un paquete en el que se especifique el valor del contador Pr. Estos se incluyen en los paquetes de datos o los de control RR y RNR.

En X25 el tamaño que se asigna por defecto a la ventana de Nivel 3 es dos, es decir, como máximo puede haber dos paquetes que hayan atravesado el interfaz sin que se haya recibido reconocimiento de ninguno de ellos. Sin embargo, en la fase de establecimiento se pueden especificar otros tamaños de ventana (máximo 7 en el módulo 8) o contratarlos de manera fija. El **Simulador X25** contempla la variación de este parámetro en las opciones de configuración.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

*3.4.2.1.3.1.**RR y RNR.***

Otros paquetes de datos que se utilizan en el control de flujo son los de receptor preparado RR y receptor no preparado RNR. Ambos se utilizan por las estaciones receptoras para controlar el ritmo de entrega de paquetes de datos .RNR se utiliza para indicar a la estación origen de los datos que debe parar la emisión hasta que el receptor esté dispuesto. RR se utiliza para mandar reconocimientos cuando no hay tráfico (paquetes) en sentido inverso. Esto es que no hay paquetes de datos cuyos indicadores de la ventana deslizante reconocerían los paquetes recibidos por su estación emisora. Como el ETD emisor tiene un número máximo de paquetes de datos que puede enviar sin haber recibido reconocimiento, el envío de esta validación por la estación receptora controlará el envío de paquetes por el ETD emisor.

3.4.2.1.3.1.1.**Formato de los Paquetes RR.**

bin: [QDMod][GRUPO][CANAL][P][00001]

hex: 1X XX X1

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	Pr 0 0001	X1

3.4.2.1.3.1.2.**Formato de los paquetes RNR.**

bin: [QDMod][GRUPO][CANAL][P][00101]

hex: 1X XX X5

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	Pr 0 0101	X5

3.4.2.1.4.**Procedimiento de control extremo a extremo: bit D.**

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Q D Mod Grupo	0001 xxxx	1X
Canal	xxxx xxxx	XX
Tipo	Pr 0 0101	X5

3.4.2.1.4. Procedimiento de control extremo a extremo: bit D.

Cuando se activa el bit confirmación de entrega (D) por el ETD en un paquete de datos, se indica que el ETD desea recibir del ETD destino la aceptación a la recepción de dicho paquete, esto es, la validación extremo a extremo.

Un ETD que vaya a utilizar los procedimientos del bit de confirmación de entrega durante la transferencia de datos, puede activar este bit en el paquete de solicitud de llamada para informar al ETD remoto de esta circunstancia. La Red pasa este bit transparentemente al destino en el paquete de llamada entrante. También puede ser activado este bit por el ETD llamado en el paquete de llamada aceptada, ocurriendo lo mismo que en el caso anterior.

*El **Simulador X25** permite la activación del bit D en cualquier momento de la ejecución del programa. Para activar/desactivar el bit D pique en la opción que lleva el mismo nombre dentro del menú desplegable de terminales.*

Ilustración 3.9. Bit D.

Compruebe como al enviar un mensaje el terminal colateral procede a su inmediata validación con un paquete RR sin esperar a agotar el ancho de la ventana disponible.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.10. Validación extremo a extremo.

3.4.2.1.5. Paquetes en secuencia. Bit M

El bit M se usa para indicar a los niveles superiores del terminal distante que se están enviando paquetes en secuencia. Este procedimiento se hace necesario cuando se envíe un archivo cuya longitud exceda la longitud máxima permitida por la Red y deba ser reconstruido en el extremo colateral. El terminal local procederá a fraccionarlo en tantos paquetes como sea necesario incluyendo en ellos la indicación de Más datos activa.

La longitud máxima permitida por la Red en ambos sentidos de la transmisión es de 128 bytes para un formato normal y de 256 octetos para paquetes de usuario con facilidad de longitud ampliada. El campo de datos de un paquete ha de contener un número de bits múltiplo de 8 menor o igual al máximo permitido.

El bit M es la indicación de más datos. Cuando la indicación de más datos va en un paquete de datos completo indica al nivel superior que siguen más paquetes con datos.del mismo mensaje La Red podrá combinar un paquete con los sucesivos sólo cuando éste sea un paquete completo con marca de más datos y sin confirmación de entrega.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Desde el terminal 3618 seleccione la opción ***enviar archivo*** *del menú* ***enviar****. Se despliega un cuadro de diálogo donde podrá seleccionar cualquier archivo que desee. Elija el que se suministra: 'texto.txt'.*

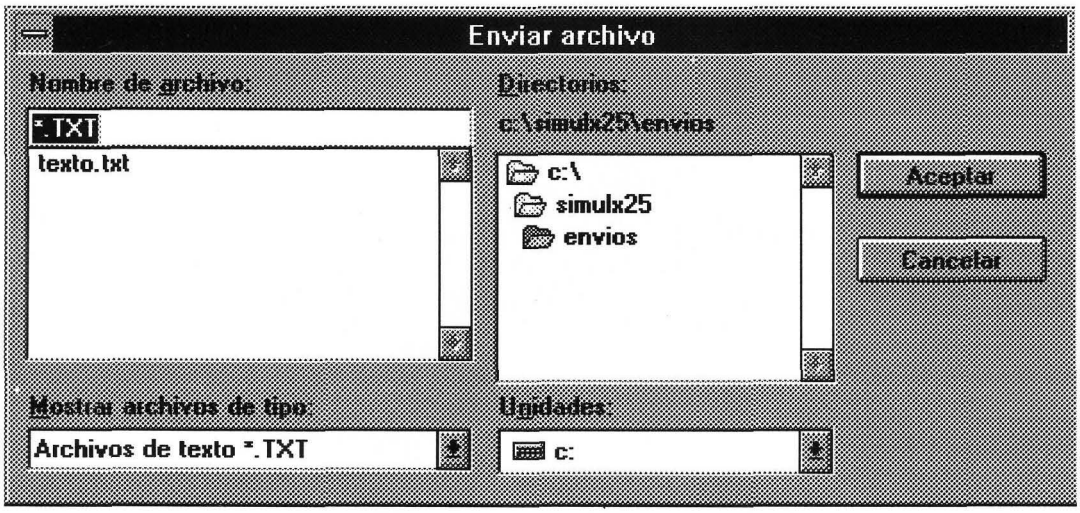


Ilustración 3.11. Enviar archivo.

Observe como el ***Simulador X25*** *fracciona adecuadamente el archivo y lo envía a línea con el bit M activo.*

Ilustración 3.12Bit M activo.

3.4.2.1.6. Procedimientos en varios niveles. Bit Q.

Existe la posibilidad de emitir en dos niveles para una secuencia completa de paquetes. Si un ETD quiere transmitir datos en más de un nivel, utiliza el indicador llamado bit calificador (Q).Con esto consigue interrumpir puntualmente el desarrollo

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

el desarrollo de una secuencia con un mensaje al nivel superior. No se pueden mezclar niveles dentro de una secuencia completa (paquetes con el bit M activo). El bit calificador no es verificado por la Red.

Desde el terminal 3618 y seleccionando el CVP que tiene definido con el 7342, elija la opción ***Limpiar pantalla*** *del menú* ***Enviar****. Observe como la acción se produce en la pantalla del terminal 7342, ya que el paquete generado lleva el bit calificador Q activo, lo que indica al colateral que este paquete de datos debe ser analizado por un nivel superior, y en el caso concreto del* ***Simulador X25****, provoca la orden de borrar la pantalla.*

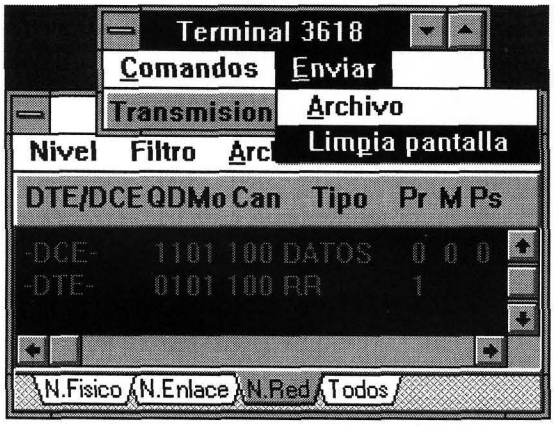


Ilustración 3.13. Limpiar pantalla.

3.4.2.2.Nivel de ENLACE

Los objetivos del procedimiento de control de enlace son:

Asegurar la sincronización entre los dos extremos del enlace.

Detectar y recuperar los errores que aparezcan en la transmisión.

En el modo Asíncrono balanceado ambas estaciones, Red y ETD, tienen la misma capacidad de transferencia y control sobre el enlace.

3.4.2.2.1.**Tramas**

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

La información intercambiada entre el ETD y la Red se estructura en tramas, o grupos de bits, cuyo formato se describe a continuación.

Indicador	8 bits
Dirección	8 bits
Control	8 bits
Información	
SVT	16 bits
Indicador	8 bits

*3.4.2.2.1.1.**Indicadores.***

Los indicadores o 'flags'(banderas) se encargan de señalar el principio y final de una trama. Constan de ocho bits de la forma 01111110 ó 7E en hexadecimal. Su función principal es la de asegurar la sincronización de la señal eléctrica entre los extremos del enlace, siendo por ésto por lo que se continúan enviando indicadores entre ETD y Red aún cuando no hay información que intercambiar.

El problema se plantearía si en el interior de la trama apareciera esa secuencia de bits sin que se quisiera indicar con ello que se ha acabado la trama. La estación receptora daría por finalizada la trama y la analizaría dando por hecho que su recepción estaba completa y como consecuencia que la recepción había sido errónea por SVT incorrecto. Para evitar esto y proporcionar al sistema una total transparencia e independencia entre los bits de información y los de sincronización en la comunicación se establece el procedimiento de inserción de ceros. Si la estación emisora de la trama advierte que en el interior de la trama hay un grupo de 6 bits uno consecutivos que será confundido en recepción con un indicador, antes de la emisión introducirá un cero detrás del quinto bit uno. La estación receptora extraerá de la trama cualquier cero que esté detrás de cinco bits uno seguidos con lo que se asegura la transparencia.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

01111110 1100000010001000**11111110**01010001010010011001001111110

indicador información original indicador

01111110 1100000010001000**11111101**0010100101001001100101111110

indicador cero insertado indicador

3.4.2.2.1.2.Campo de dirección. Comandos y respuestas.

El campo de dirección contiene codificada la dirección del ETD o la de la Red. La dirección de la Red para un monoenlace será la 01 en hexadecimal, ó 00000001 en binario. La dirección del ETD será 03 en hexadecimal, ó 00000011 en binario. Las tramas pueden ser comandos o respuestas. Si se emite una trama comando, ésta debe llevar la dirección de la estación receptora del comando. Si la trama emitida es una trama respuesta, ésta debe llevar codificada la dirección de la estación emisora de la respuesta.

[ETD] [Red]

comando: dir. 01 ----> respuesta: dir. 01

respuesta: dir. 03 <---- comando: dir. 03

3.4.2.2.1.3.Campo de control.

En este campo se codifica el tipo de trama. Estas se qgrupan en tres categorías: las de información, las de supervisión y las no numeradas. En las primeras el campo de control se utiliza para numerar las tramas de manera que sea correcta la recepción ordenada de éstas. En las de supervisión, RR, RNR y REJ el campo de control, a la vez que codifica estas tramas, lleva asociado un valor numérico de las variables de estado de transmisión o recepción de las tramas de información. Estos contadores llamados Nr y Ns tiene un funcionamiento análogo

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

En las tramas no numeradas el campo de control codifica el tipo de que se trata: SABM, UA, DISC, DM y FRMR. En este campo el quinto bit es el denominado 'P/F' o bit de petición de respuesta. Si una trama lleva este bit activo (P=1) indica al receptor que es necesaria una trama respuesta a ésta con el bit activo (F=1). Esta respuesta debe establecerse en el menor tiempo posible.

Tabla de tramas. Codificación del campo de control.

tipo	**Comando**	**Respuesta**	1 2 3 4 5 6 7 8	hex.
Info	INFO		N(r) P N(s) 0	XXpar
Supervisión	RR		N(r) P/F 0001	X1
	RNR		N(r) P/F 0101	X5
	REJ		N(r) P/F 1001	X9
No numeradas	SABM	SABM	0 0 1 P 1 1 1 1	2F 3F
		UA	0 1 1 F 0 0 1 1	63 73
	DISC		0 1 0 P 0 0 1 1	43 53
		DM	0 0 0 F 1 1 1 1	0F 1F
	FRMR		1 0 0 P 0 1 1 1	87 97

INFO: Tramas que transportan información a los niveles superiores.

RR: Tramas que controlan el flujo de información a nivel 2. Cumplen un cometido análogo a las tramas homónimas de nivel 3.

RNR: Tramas que controlan el flujo de información a nivel 2. Cumplen un cometido análogo a las tramas homónimas de nivel 3.

REJ: Tramas que determinan un rechazo selectivo de una trama recibida previamente.

SABM: Tramas encargadas de la petición del levantamiento del nivel 2.

UA: Tramas encargadas de la aceptación del levantamiento del nivel 2.

DISC: Trama que indica la desconexión del enlace.

DM: Trama que indica el modo de desconexión en una estación.

FRMR: Trama que indica el rechazo total de tramas.

En caso de que sea codificación extendida, no se haría sobre ocho bits con lo que la codificación sería distinta.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

En caso de que sea codificación extendida, no se haría sobre ocho bits con lo que la codificación sería distinta.

*3.4.2.2.1.4.**Campo de información.***

Es el campo propio de las tramas de información. Es el encargado de transportar la información hacia niveles superiores. No tiene ninguna limitación propia del Nivel 2 que no sea el de su tamaño. En él se incluyen íntegramente los paquetes de Nivel 3. Las tramas FRMR tienen también un campo de información de tres octetos donde se codifican las causas del rechazo.

*3.4.2.2.1.5.**Secuencia de verificación de trama.***

Todas las tramas incluyen una secuencia de verificación de trama de 16 bits, con el fin de detectar posibles errores en la transmisión. Esta secuencia afecta a los campos de dirección, control e información, no incluyendo los ceros insertados para lograr la transparencia. La secuencia de verificación de trama es el resto de un proceso de división, modulo dos, utilizando como divisor un polinomio generador. El polinomio generador es el aconsejado por la norma V.41 del CCITT.

*Pique con el ratón en la solapa de Nivel enlace, o si lo prefiere siga la secuencia **nivel, enlace** desde el menú desplegable de **Analizador**. En este nivel se muestran las tramas tal y como aparecen en línea, siendo filtrados únicamente los flags de sincronismo (7E). Estos sólo se visualizan al principio y final de cada trama. En este nivel se muestra una cabecera en la pantalla que especifica los diferentes elementos de la trama, a saber:*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

TxRx indica quien es el emisor de la trama en el interfaz.

Dir indica el campo de dirección contenido en la trama.

Tipo Especifica el tipo de trama de Nivel 2.

NrPNs Indica los números de secuencia de trama en transmisión y en recepción así como el estado del bit P/F.

Cierre todos los terminales si es que están abiertos. mantenga abierto el analizador del terminal 3618. Observe como la Red envía constantemente tramas SABM para iniciar el enlace. Cada cinco tramas la Red procede a activar el bit P/F para forzar al terminal a que envíe una respuesta. Ahora abra el terminal 3618. Observe como éste devuelve la trama SABM a lo que la Red contesta con un UA. A continuación la Red insiste con un nuevo SABM a la espera de la confirmación del terminal. Una vez iniciado el enlace, es la Red la que inicia el procedimiento para el establecimiento del Nivel 3, para ello codifica un paquete de rearranque dentro de una trama INFO de Nivel 2. Observe como esta trama se encuentra perfectamente numerada con su secuencia Ns. el terminal confirma la recepción de la misma con una trama RR de Nivel 2. A partir de aquí el proceso se repite con los paquetes de Nivel 3 que son enviados tanto por el terminal como por la Red.

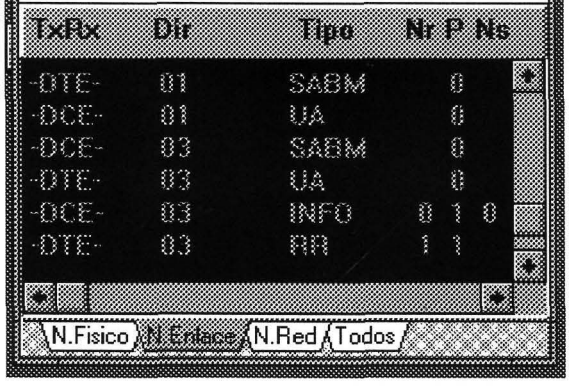


Ilustración 3.15. Establecimiento de enlace.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Observe que en estado de ausencia de paquetes de Nivel 3, el ***Simulador X25*** *asegura la sincronización de las estaciones enviando constantemente tramas RR de Nivel 2. A esta situación se le conoce con el nombre de cruce de RR y no es obligatorio según la norma.*

Ilustración 3.16. Cruce de RR.

Abra ahora el terminal 7342. Recuerde que este tiene definido un canal permanente con el 3618. Observe como llegan nuevas tramas Info al terminal 3618. Estas contienen los paquetes de Reinicio que envía la Red para indicar que el terminal distante se encuentra en operación. Seleccione la opción ***Archivo*** *del menú* ***Enviar****. Escoja el archivo texto.txt del cuadro desplegable. Observe como a pesar de que los datos viajan en un sólo sentido, las tramas info de Nivel 2 se cruzan hacia ambas estaciones. Esto es debido al envío de RR de Nivel 3 por parte del terminal receptor que se codifican dentro de tramas info de Nivel 2.*

3.4.2.3.Nivel FISICO

En este nivel se estudia la información que corre por el interfaz a nivel de bits. Debido a lo engorroso para el usuario y a lo poco práctico del tema, la mayoría de los analizadores muestran esta información agrupada según las

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

de Nivel 2 y codificada en hexadecimal. De nuevo se filtran los flags de sincronismo.

3.4.2.3.1. Formato de Tramas.

Se repite a continuación el formato de las tramas de Nivel 2 que ya se explicaron en el pasado capítulo correspondiente al nivel de enlace a fin de que el lector pueda disponer de la información con más comodidad.

Indicador	8 bits
Dirección	8 bits
Control	8 bits
Información	
PAQUETE de NIVEL 3	
(si lo hubiera)	
.....	
SVT	16 bits
Indicador	8 bits

Tabla de tramas. Codificación del campo de control.

tipo	**Comando**	**Respuesta**	1 2 3 4 5 6 7 8	hex.
Info	INFO		N(r) P N(s) 0	XXpar
Supervisión	RR		N(r) P/F 0001	X1
	RNR		N(r) P/F 0101	X5
	REJ		N(r) P/F 1001	X9
No numeradas	SABM	SABM	0 0 1 P 1 1 1 1	2F 3F
		UA	0 1 1 F 0 0 1 1	63 73
	DISC		0 1 0 P 0 0 1 1	43 53
		DM	0 0 0 F 1 1 1 1	0F 1F
	FRMR		1 0 0 P 0 1 1 1	87 97

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Los octetos se transmiten de forma que se envía en primer lugar el bit de menor orden. La secuencia de verificación de trama SVT se transmite de forma que se envía en primer lugar el coeficiente del término de mayor grado.

3.4.2.3.2.**Estructura de los paquetes**

Al igual que se hiciera en el apartado anterior, se repite en éste la estructura de los paquetes ya reseñada con anterioridad para facilitar la comodidad en el acceso a la información por parte del lector.

|Q |D |MOD |GRUPO |CANAL |TIPO,C |DATOS O PARAMETROS |

Q D Mod GRUPO
CANAL
TIPO
DATOS o PARAMETROS
....

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Los tipos de paquetes se codifican en el campo Tipo de la manera siguiente:

PAQUETE	cod. binario	hex.
Solicitud de llamada	0000 1011	0B
Llamada aceptada	0000 1111	0F
Solicitud de liberación	0001 0011	13
Confirmación de liberación	0001 0111	17
Interrupción	0010 0011	23
Confirmación de interrupción	0010 0111	27
Receptor preparado	ppp0 0001	P1
Receptor no preparado	ppp0 0101	P5
REJ	ppp0 1001	P9
Reinicio	0001 1011	1B
Confirmación de reinicio	0001 1111	1F
Rearranque	1111 1011	FB
Confirmación de rearranque	1111 1111	FF

Pique con el ratón en la solapa de Nivel físico, o si lo prefiere siga la secuencia ***nivel, físico*** *desde el menú desplegable de* ***Analizador****. En este nivel se muestran las tramas tal y como aparecen en línea, siendo filtrados únicamente los flags de sincronismo. Estos sólo se visualizan al principio y final de cada trama. La visualización se hace en notación hexadecimal para los campos de control, mientras que los elementos de información se representan en caracteres ASCII.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.17. Nivel físico.

En los apartados anteriores se ha visto los formatos de las distintas tramas, por lo que no tendrá ningún problema en entender el significado de los caracteres representados. Pruebe a establecer un canal entre dos terminales y enviar de uno a otro el archivo texto.txt suministrado. Trate de descifrar las distintas tramas que aparecen en Analizador.

3.4.2.4.TODOS los niveles

Con esta opción el Analizador mostrará simultáneamente los tres niveles anteriormente descritos. Es una posibilidad muy útil a la hora de comprender el funcionamiento del protocolo X25. Al seleccionar esta visualización el Analizador muestra las cabeceras de los Niveles 2 y 3 y la información es precedida de las etiqueta N2 y N3 para diferenciar los niveles a que pertenecen.

Si tuvo problemas en el apartado anterior, vuelva a intentar la tarea en éste. Observará como la labor se torna mucho más accesible y grata. Envíe de un terminal a otro el archivo texto.txt y observe la diferencia de notaciones que el ***Simulador X25*** *hace para diferenciar los datos que se visualizan (Hexadecimal para campos de control y SVT frente a decimal en los campos de datos). Esta diferenciación no viene dada por la norma sino que es una particularidad del programa para facilitar la comprensión de los datos. Pruebe a repetir el envío de paquetes de llamada, reinicios y rearranques.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

particularidad del programa para facilitar la comprensión de los datos. Pruebe a repetir el envío de paquetes de llamada, reinicios y rearranques.

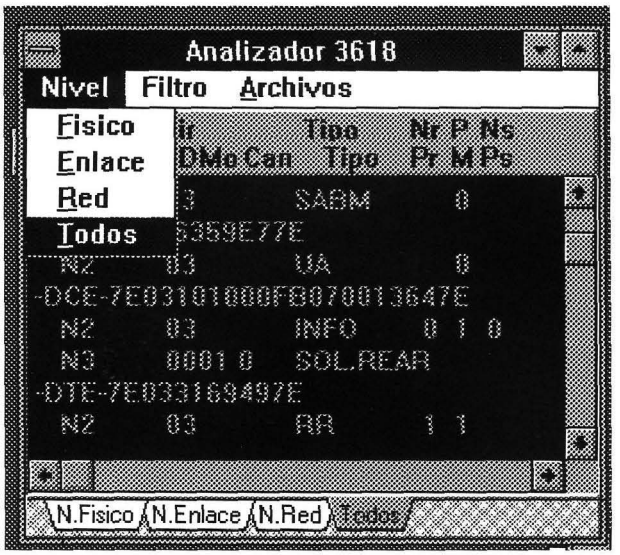


Ilustración 3.18. Todos los niveles.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.3.FILTROS

Bajo este apartado se encuentran las opciones que permiten configurar el funcionamiento del Analizador. Se podrán indicar situaciones determinadas ante las cuales el Analizador deberá dar una respuesta activa.

Ilustración 3.19. Filtro.

3.4.3.1.Concepto de filtro y trampa

Se define filtro a un conjunto de valores para unas variables determinadas. Los filtros son pues, unos esquemas o plantillas configurables de condiciones de trabajo del interfaz. La definición del filtro viene asociada a la tarea del analizador. El filtro especifica que tramas o paquetes se deben visualizar o enviar a los archivos de captura. Esto no significa que el resto se pierda o no se transmita por el interfaz, solamente se les hace transparentes a la acción del analizador.

La trampa se define como una respuesta controlada ante una situación predefinida en el interfaz. En este apartado, las condiciones de disparo se pueden fijar a nivel de octetos o de máscaras de bits. Si se hace usando la primera

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

fijar a nivel de octetos o de máscaras de bits. Si se hace usando la primera alternativa se pueden codificar tipos de tramas o paquetes, canales, etc. Combinando estas restricciones con las máscaras de bits se puede delimitar aún más la situación que se desee observar. Las acciones a tomar en caso de satisfacer las condiciones de la trampa en el interfaz también son definibles por el usuario.

3.4.3.2.Parar Captura

Esta opción le permite parar la captura de datos del Analizador, bien sea hacia el buffer o hacia algún fichero según la opción **Capturar**.

Envíe el archivo texto.txt de un terminal a otro. Antes de finalizar la transferencia, pique con el ratón sobre parar captura. Observe en la ventana de los terminales que la transmisión de datos continua a pesar de no visualizarse en la ventana del Analizador.

3.4.3.3.Modificar Filtros

Esta opción permite definir las tramas, paquetes y sentidos de transmisión que se quieren visualizar. Cualquier evento que no se especifique no se mostrará en la ventana del Analizador.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Ilustración 3.20. Modificar filtros.

Pruebe a activar solamente las tramas RR de Nivel 2 en el sentido de recepción. Observe que al enviar el archivo texto.txt de un terminal a otro, a pesar del tráfico variado de la línea sólo se visualizan estas tramas.

3.4.3.4.Trampas

Esta opción permite definir las diferentes acciones que ejecutará el programa cuando los datos que se recogen en el analizador coinciden con situaciones predeterminadas. Su uso es muy común para estudiar problemas esporádicos que sobrevienen a las condiciones normales de trabajo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

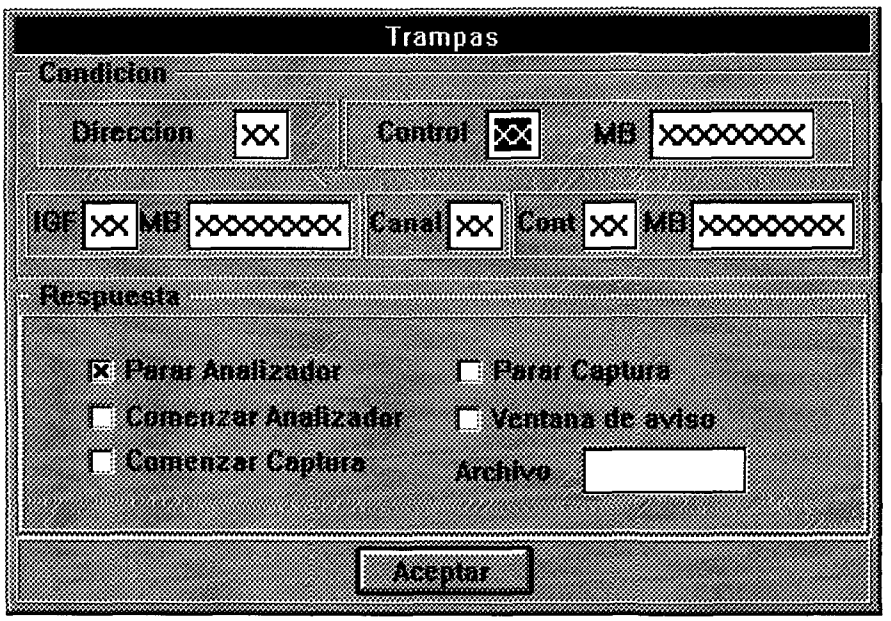
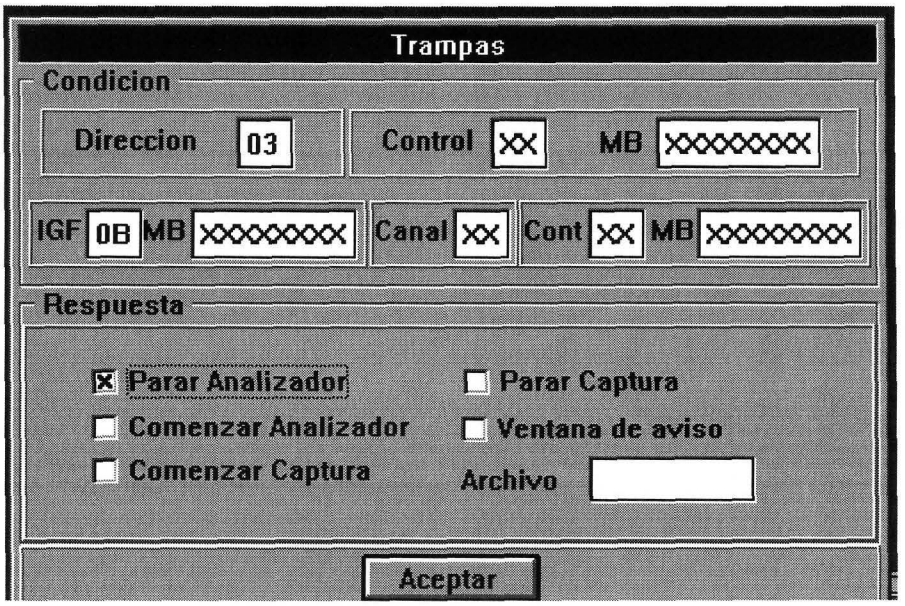


Ilustración 3.21. Cuadro de trampas.

Seleccione esta opción dentro del menú correspondiente. Se despliega un cuadro configurable donde se especifican los eventos que se pueden detectar y las acciones previstas cuando esto suceda. Por ejemplo se va a parar la captura del Analizador cuando se reciba de la Red un paquete de llamada en el terminal 3618. Para esto seleccione del menú de ***Trampas*** *las opciones 03 (dirección), en la parte de Nivel físico y 0B* ***(Sol.LLam.)*** *en la zona de Nivel 3. Con esto dejamos seleccionado el evento de recibir un paquete de solicitud de llamada proveniente de la Red en el terminal predefinido. Como acción a tomar seleccione la opción de parar captura.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006



Ahora haga clic sobre aceptar, abra el terminal 3618 y el 7342. Observe que estos terminales se hallan en comunicación por el canal virtual permanente que tienen definido. Uselo y envíe datos a través de él. No se detendrá el Analizador. Sobre el terminal 3618 haga una ***Llamada*** *al 7342. Observe que el Analizador sigue sin detenerse debido a que el paquete se ha enviado desde el terminal y no desde la Red. Ahora desde el terminal 7342 repita esta misma acción en sentido contrario. Una vez llega el paquete de* ***Solicitud de llamada*** *al terminal 3618 se detiene el Analizador.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.4.Archivos

Esta opción del analizador permite definir los archivos y su correspondiente ubicación donde se almacenen los datos provenientes de la captura de datos del mismo. Sus opciones son:

Ilustración 3.22. Archivos.

3.4.4.1.Guardar

Con esta alternativa podemos desviar los datos que se hallan en el buffer del analizador hacia un fichero. Guarda los datos de cada uno de los niveles predefinidos en el analizador. Cada nivel muestra su propio cuadro configurable de Windows que hace referencia a la ubicación de ficheros.

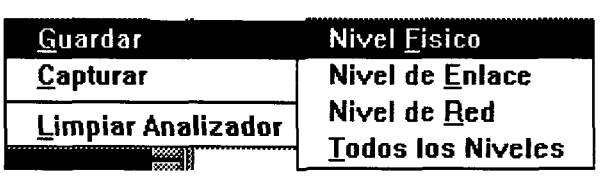


Ilustración 3.23. Guardar archivo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.4.1.1.Nivel Físico

3.4.4.1.2.Nivel de Enlace

3.4.4.1.3.Nivel de Red

3.4.4.1.4.Todos los niveles

Ilustración 3.24. Cuadro de guardar archivo.

Se aprovecharán los datos que tiene almacenados en el buffer del Analizador del Terminal 3618 en este momento para guardarlos en el archivo Prueba1.txt. Abra el cuadro de diálogo que se encuentra en ***Archivo, Guardar*** *y direccione un subdirectorio donde ubicar este archivo. Observe que este archivo tiene formato de texto y puede analizarlo con cualquier procesador que acepte este modo de trabajo.*

3.4.4.2.Capturar

Esta opción es similar a la anterior con la diferencia de que a partir de su activación los datos que se capturan en el analizador no se almacenan en el buffer del mismo, sino que pasan en tiempo real a ser guardados dentro de un

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

del mismo, sino que pasan en tiempo real a ser guardados dentro de un fichero. Despliega un cuadro de diálogo donde se define dicho fichero y su ubicación.

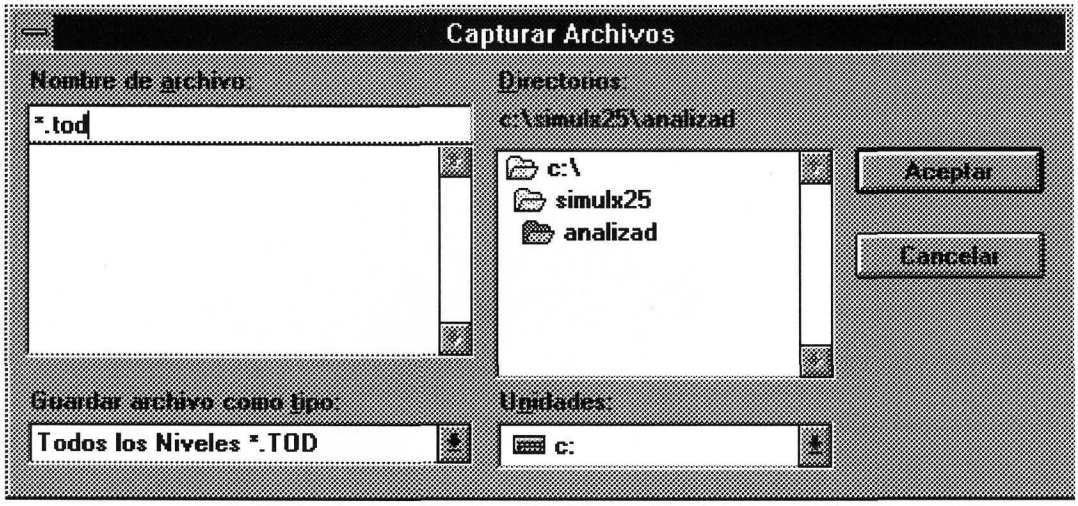


Ilustración 3.25. Cuadro de captura.

Abra el cuadro de diálogo que se encuentra en ***Archivo, Capturar*** *y direccione un subdirectorio donde guardar el archivo Prueba2.txt. Observe que, como ocurriera en el caso anterior, este archivo tiene formato de texto y puede analizarlo con cualquier procesador que acepte este modo de trabajo.*

3.4.4.3. Limpiar Analizador

Ejecuta una orden directa para borrar los datos que se hallan presentes en la pantalla del analizador.

Haga clic en esta opción para inicializar la pantalla.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.4.5.Analizadores de RUTAS

Su configuración y funcionamiento es exactamente igual a la de los analizadores de terminales que ya se han estudiado. Tienen la particularidad de que a través de ellos se visualiza todo el tráfico que soporta la ruta. Compete a las opciones de filtraje la potestad de separar las comunicaciones de un terminal determinado o bien de un canal en concreto. La notación que sigue el **Simulador X25** designa con el término DTE al nodo de menor numeración y DCE al de mayor.

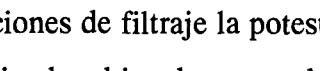


Ilustración 3.26. Icono de analizador de ruta.

Abra el Analizador de ruta 36/73. Abra el Analizador del terminal 3618. Abra todos los terminales. Mantenga diálogo entre todos con el 3618. Observe que en el Analizador de ruta sólo se visualiza los datos que se cruzan entre el terminal 3618 y el 7342 ya que los otros terminales no usan esta ruta para su comunicación. Observe de que a pesar de no ser obligatorio, el ***Simulador X25*** *tiene implementado un protocolo de sincronismo entre nodos basado en el cruce de RR de Nivel 2 explicado con anterioridad.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.CONFIGURAR TERMINAL Y NODO

Estas opciones se abren con los iconos situados en la parte inferior de la columna que se encuentra próxima a los iconos principales de terminales y nodos o en los títulos ubicados en el menú de configuración de la barra principal. Dan acceso a un cuadro de diálogo donde se accede a las distintas opciones de configuración de los elementos de Red. El cuadro de diálogo que se despliega tiene una barra de título con el nombre del objeto a que hace referencia. Bajo él se encuentran las diferentes solapas que dan acceso a los puntos de configuración.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.1.Concepto de configuración

Las posibilidades con las que cuentan los terminales y las puertas de las que cuelgan en los nodos de Red son las que definen completamente las comunicacniones de todos los elementos. Por ello, en la configuración que se fija, se especifica en gran medida las limitaciones que se tendrán a la hora de establecer y gestionar los canales. Las opciones que se decidan para un terminal se contratan con la Red y deben ser rigurosamente idénticas a las que guarda el elemento colateral (nodo y terminal distante en su caso) para asegurar un correcto funcionamiento de estos elementos. Se estudiarán a continuación.

Ilustración 3.1. Configuraciones.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.2.Nivel 2

Configuraremos en esta solapa las posibilidades de actuación sobre el Nivel 2. Sus opciones se dividen entre los propios parámetros que afectan al control de la ventana y los contadores y temporizadores que gestionan el establecimiento del enlace.

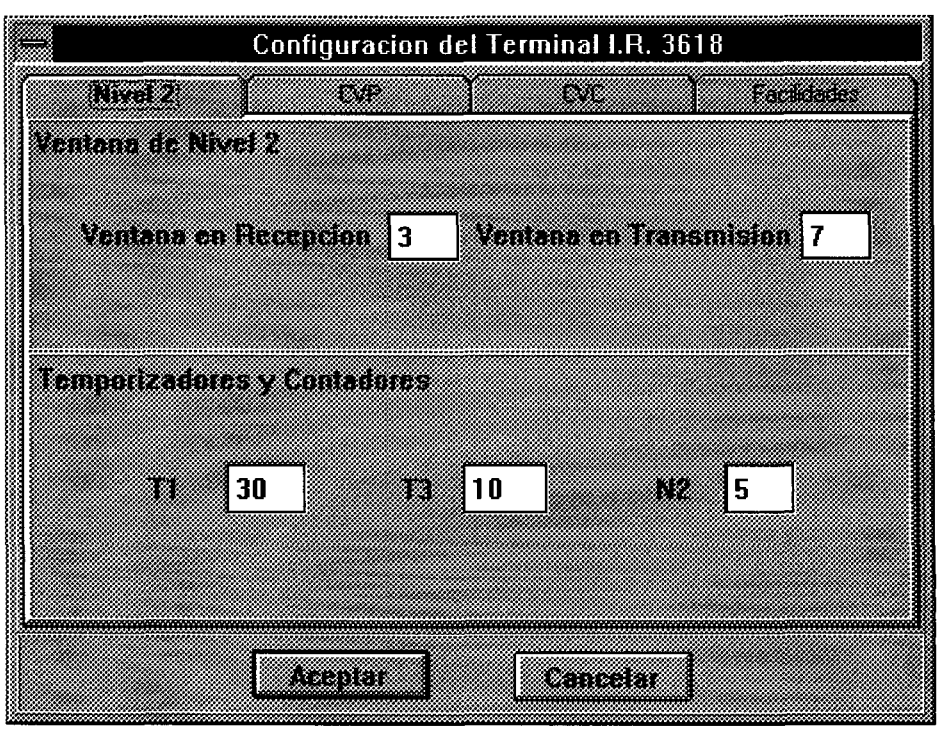


Ilustración 3.2. Ventana de configuración del nivel 2.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.2.1.Ventana de Nivel 2

Se define el valor de las **ventanas en Recepción** y **Transmisión**. El valor de este parámetro debe estar comprendido entre 0 y 7 debido a que el **Simulador X25** trabaja en módulo 8.

3.5.2.2.Temporizadores y Contadores

Se define aquí el valor de los temporizadores **T1** y **T3**, así como del contador **N2** que controlan el funcionamiento de la Red. La fase de establecimiento del Nivel 2 se realiza por sucesivos ciclos de envío de tramas SABM. Estos ciclos se caracterizan por la existencia de una de estas tramas con el bit P/F activo mientras que en el resto va fijo a 0. El temporizador T1 fija el intervalo que existe entre tramas SABM de un mismo ciclo. El temporizador T3 fija el tiempo entre la última trama de un ciclo y la promera del siguiente. El contador N2 indica el número de tramas que compone cada ciclo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.3.CVP

Se configuran en esta ventana los parámetros de control que rigen el funcionamiento de los canales virtuales permanentes. Sus opciones se dividen entre los parámetros de configuración de la ventana de transmisión y la definición de los CVP que se le asignen al terminal.

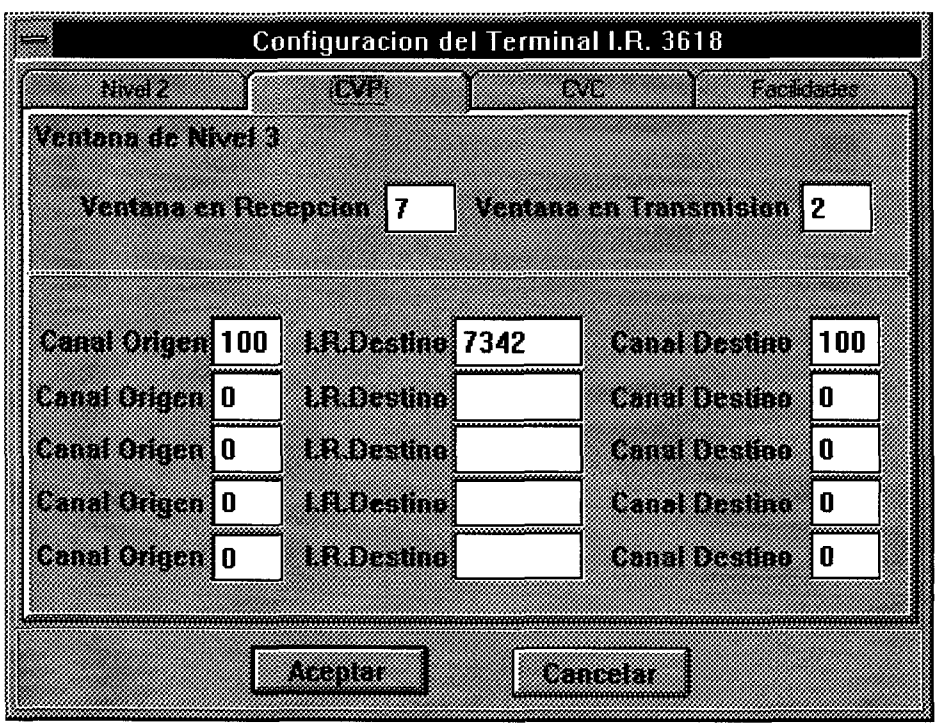


Ilustración 3.3. Ventana de configuración de CVP.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.3.1.Ventana de Nivel 3

Se definire el valor de las **ventanas en Recepción** y **Transmisión** de Nivel 3 para los CVPs. El valor de estos parámetros debe estar comprendido entre 1 y 7 debido a que el **Simulador X25** trabaja en módulo 8.

3.5.3.2.Configuración de CVPs

Aquí se define la relación de los CVPs propios con los terminales destino y el CVP que estos tienen asociado. Se despliega un casillero configurable de cinco niveles con tres casillas cada uno que hace referencia a los CVPs que se pueden definir. Cada nivel define un CVP y precisa de relacionar: el **Canal Origen**, que especifica el valor en decimal del canal que utilizará el terminal local, el **I.R.Destino**, que especifica el Identificativo de Red del terminal distante y el **Canal Destino**, que especifica en notación decimal el canal que utilizará el terminal destino para el enlace permanente.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.4.CVC

Bajo este epígrafe se encuentran las opciones de configuración de los canales virtuales conmutados. Sus opciones abarcan el control de la ventana de nivel 3 para los CVC y la definición de los mismos.

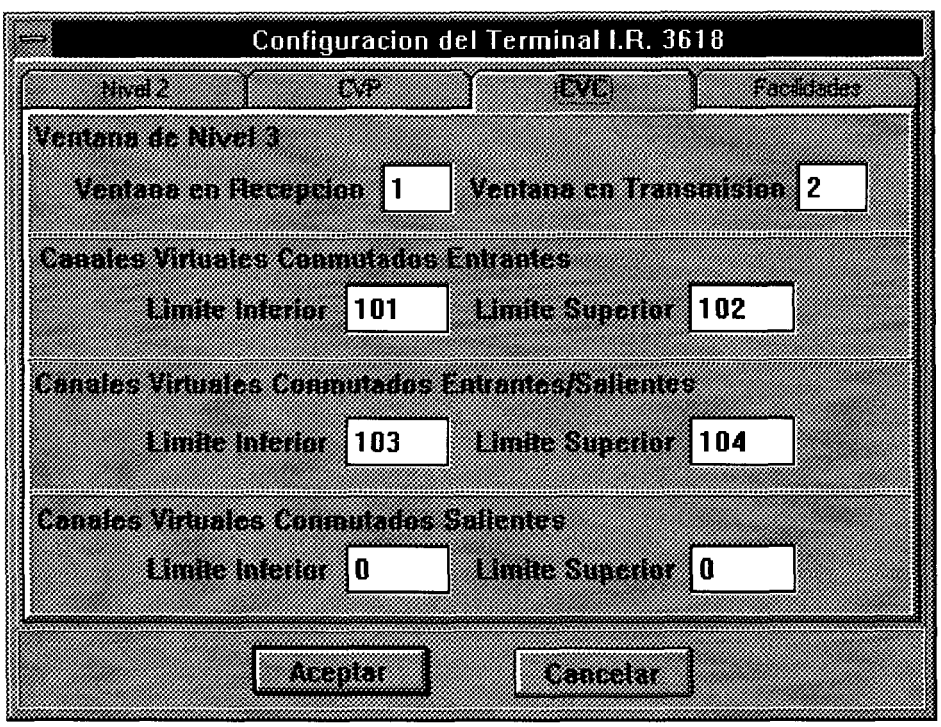


Ilustración 3.4. Ventana de configuración de CVC.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.4.1.Ventana de Nivel 3

Se define el valor de las **ventanas en Transmisión** y **Recepción** de Nivel 3 para los canales definidos bajo este epígrafe. Los valores de estos parámetros han de estar comprendidos entre 1 y 7.

3.5.4.2.Numeración de CVCs

Se especifican aquí los CVCs que se asocien a cada terminal. Estos deben tener numeración rigurosamente ordenada y consecutiva a los CVPs. Esto quiere decir que se debe asignar los canales cuya numeración sea inmediatamente superior al canal más alto asignado como CVP y se debe empezar a asignar, si los hubiera, con los CVCs **Entrantes**, a continuación los **Entrantes/Salientes**, y, por último, los CVCs **Salientes**. Los valores que se definen para cada apartado son: **Límite Inferior**, que especifica el valor en notación decimal con el que empieza la numeración de los canales de que se trate y **Límite Superior**, que especifica el valor con el que terminan los canales a los que se hace referencia. Estos parámetros deben estar comprendidos entre los valores decimales 100 y 104.

Proceda en este momento a definir los canales conmutados del terminal 5981. Abra la ventana correspondiente en el icono de configuración de este terminal. En el espacio de tamaño de ventana cambie las opciones de transmisión y recepción a 4. en la zona de canales definiremos uno entrante, 2 entrantes/salientes y uno saliente. Para ello comentaremos por el primero al que daremos el número 100 (64H) como inicio y 100 como límite superior (un sólo canal). Para fijar la numeración de los entrantes/salientes ponga el número 101 como límite inferior y 102 como límite superior (2 canales). Para finalizar fije 103 para ambos valores en el espacio de canales salientes. Acepte los valores y fijados y abra el cuadro homónimo en el nodo del que cuelga este terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Repita los pasos. Con el terminal y el nodo de Red perfectamente configurados compruebe el significado de la decisión tomada. Abra el terminal 3618 y lance una llamada al 5981. Observe que esta entrará por el canal 100 (canal entrante). Lance una segunda y tercera llamada y vea como ahora entran por los canales entrantes salientes del terminal destino. Inténtelo nuevamente. La Red devuelve una liberación debido a la indisponibilidad de más canales entrantes en el terminal 5981. Ahora sitúese en este terminal y lance la llamada al colateral. La operación culmina con éxito. Libere todos los canales mediante un rearranque.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.5.Facilidades

En este apartado se configuran las facilidades de usuario que posibilita la Red. Sus opciones son:

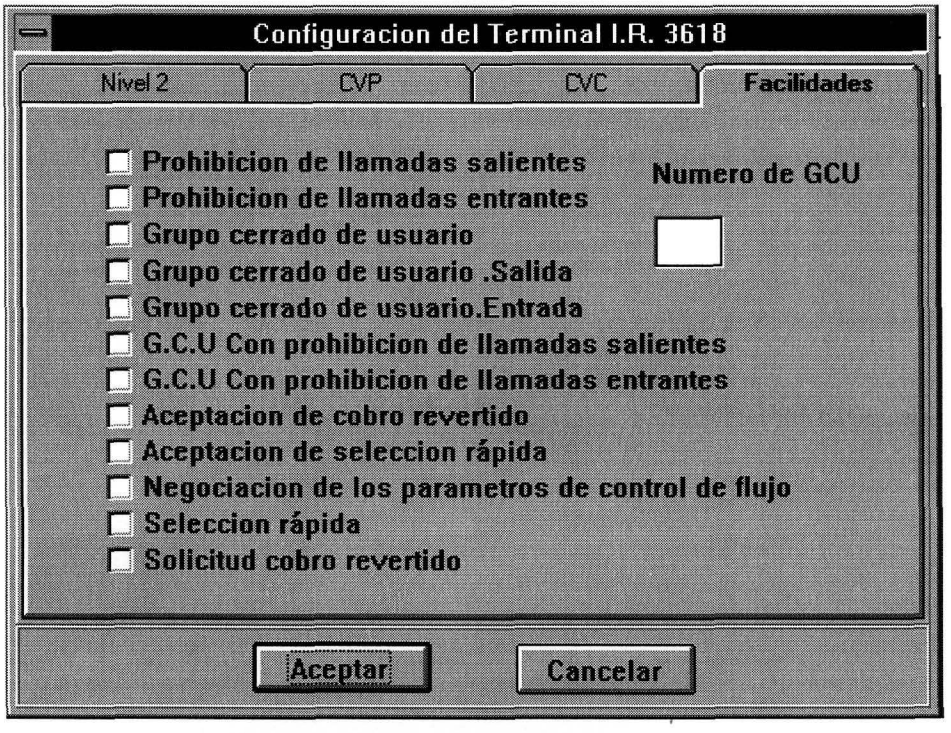


Ilustración 3.5. Ventana de configuración de facilidades.

3.5.5.1.Prohibición de llamadas salientes

Esta opción prohíbe al terminal seleccionado realizar llamadas. Se aplica a todos los canales lógicos para llamadas virtuales de un interfaz entre un ETD y la

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Red. Cuando un ETD está abonado a esta facilidad, la Red no acepta llamadas del ETD, si bien éste puede recibir llamadas entrantes de ETDs remotos. Si la Red recibe una llamada de un ETD con esta facilidad, emitirá una liberación con causa "error de procedimiento local" y diagnóstico "Paquete no permitido".

Escoja el terminal 3618 y configúrelo con esta opción. Copie esta configuración en el lado de Red. Intente hacer una llamada a cualquier otro. Vea como la Red devuelve la liberación de la llamada. Para los sucesivos ejercicios de esta qpartado, devuelva la configuración a su estado inicial en cada uno de los puntos.

3.5.5.2.Prohibición de llamadas entrantes

Esta opción prohibe al terminal seleccionado recibir llamadas. Esta facilidad que se aplica a todos los canales lógicos para llamadas virtuales en un interfaz ETD-Red, evita a un ETD la recepción de llamadas de ETDs remotos, si bien mantiene la posibilidad de generar llamadas salientes. Si la Red recibe una llamada con destino a un ETD suscrito a esta facilidad, la liberará con causa "Acceso prohibido".

Escoja el terminal 3618 y configúrelo con esta opción. Copie esta configuración en el lado de Red. Intente hacer una llamada desde cualquier otro. Vea como la Red devuelve la liberación de la llamada.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.5.3.Grupo cerrado de usuario

Se denomina grupo cerrado de usuario a un conjunto de terminales que funcionan como una Red cerrada a los terminales ajenos a la misma. A los elementos de este grupo cerrado se les puede autorizar para tener contacto con equipos externos merced a las facilidades de acceso salida o entrada.. Se permite que un ETD que pertenezca a un grupo cerrado de usuarios establezca llamadas virtuales con los otros ETDs del grupo cerrado pero se impide la comunicación con otros ETDs. Un ETD puede pertenecer a uno o más grupos cerrados de usuarios. El ETD que origina la llamada especificará el grupo cerrado de usuarios seleccionado en el campo de facilidades opcionales de usuario en el paquete de solicitud de llamada. La Red indicará al ETD destino el grupo cerrado en el campo de facilidades de usuario del paquete de llamada entrante. Si un ETD intenta una comunicación con un ETD al que no tiene permitido el acceso, la Red liberará la llamada con causa "acceso prohibido".

Cuando un ETD pertenece únicamente a un grupo cerrado de usuarios o realiza una llamada dentro del grupo cerrado de usuarios preferente, puede no efectuar ninguna indicación en el paquete de solicitud de llamada.

Escoja los terminales 3618 y 7342 y active la opción de grupo cerrado de usuario en sus menús de configuración. No olvide trasladar esta acción en los correspondientes nodos de Red. Abra ambos terminales. Compruebe como en los paquetes de llamada se progresa la información de pertenencia a GCU. Ahora abra el terminal 3677. Intente hacer una llamada desde ese terminal a cualquiera de los otros dos y viceversa. Observe la imposibilidad de tal evento en este momento.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

*El **Simulador X25** tiene restringida la existencia de grupos cerrados de usuario a una única posibilidad. Es decir sólo se puede especificar uno aunque en e'l se pueden incluir todos los terminales si se desea.*

3.5.5.4. Número GCU.

Se define en este apartado el número identificativo del GCU que se esté tratando.

3.5.5.5. Grupo cerrado de usuario. Salida

Permite a un terminal que pertenece a un grupo cerrado de usuario el efectuar llamadas a terminales que no pertenezcan a dicho grupo. Cuando un ETD hace uso de esta facilidad no tiene que indicar nada referente al grupo cerrado en el paquete de solicitud de llamada.

Seleccione el terminal 3618 y confiérale esta opción en su configuración. Haga lo propio en el nodo de Red. Haga la llamada al 3677. Observe que el canal se establece con normalidad.

3.5.5.6. Grupo cerrado de usuario. Entrada

Esta facilidad permite a un ETD que pertenezca a uno o varios grupos cerrados de usuarios recibir llamadas desde terminales pertenecientes a clase abierta o a un grupo ajeno. La Red no incluirá indicación de facilidad de grupo cerrado de

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

que no pertenezca al grupo cerrado de usuarios con destino a un ETD suscrito a esta facilidad.

Seleccione el terminal 3618 y confiérale esta opción en su configuración. Haga lo propio en el nodo de Red. Haga la llamada desde el 3677. Observe que el canal se establece con normalidad.

3.5.5.7.GCU. Prohibición de llamadas salientes

Esta facilidad permite recibir llamadas virtuales a un ETD perteneciente a un grupo cerrado de usuarios desde otro ETD perteneciente a dicho grupo cerrado e impide generar llamadas salientes hacia los terminales del grupo cerrado de usuarios referido.

Seleccione el terminal 3618 y confiérale esta opción en su configuración. Haga lo propio en el nodo de Red. Intente hacer una llamada a cualquier terminal, dentro o fuera del GCU. Observe la imposibilidad de establecer ningún canal con normalidad.

3.5.5.8.GCU. Prohibición de llamadas entrantes

Esta facilidad permite a un ETD que pertenezca a un grupo cerrado de usuarios, generar llamadas a otro ETD perteneciente al mismo grupo cerrado de usuarios, pero impidiendo la recepción de llamadas virtuales procedentes de los ETDs del grupo cerrado.

Seleccione el terminal 3618 y confiérale esta opción en su configuración. Haga lo propio en el nodo de Red. Intente hacer una llamada desde cualquier terminal, dentro o fuera del GCU. Observe la imposibilidad de establecer ningún canal con normalidad.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

terminal, dentro o fuera del GCU. Observe la imposibilidad de establecer ningún canal con normalidad.

3.5.5.9.Solicitud de cobro revertido

Permite a un terminal cualquiera solicitar en su paquete de llamada esta modalidad en su comunicación. Para que la operación concluya con éxito es preciso que su colateral tenga definida la opción de aceptación de cobro revertido.

3.5.5.10.Aceptación de cobro revertido.

Opción asociada a la anterior que permite al terminal que tenga definida esta solicitud entrar en comunicación con un distante que haya pedido esta modalidad en su paquete de llamada. Esta facilidad opcional autoriza a la Red a transmitir al ETD una solicitud de llamada con destino al mismo solicitando la facilidad de cobro revertido. Si el ETD no está abonado a esta facilidad, la Red no transmitirá al mismo la llamada entrante que solicita la facilidad de cobro revertido y se enviará al ETD origen de esta llamada un liberación con causa "Destino no suscrito a aceptación de cobro revertido".

*El **Simulador X25** observa en la fase de establecimiento de llamada el que el terminal llamante y llamado tengan configurados las opciones de cobro revertido, pero no leva control sobre posibles tarificaciones. Configure dos terminales cualesquiera con estas posibilidades y establezca un CVC entre ellos. Inténtelo después hacia un terminal que no tenga la facilidad de aceptación del cobro revertido y vea la imposibilidad de establecer el canal.*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.5.11.Selección rápida

Esta opción permite incluir un campo de información en el paquete de llamada a los terminales que lleven esta configuración. El ETD puede solicitar esta facilidad con o sin restricción de respuesta. La Red no permitirá a un ETD no abonado a esta facilidad solicitar una llamada de selección rápida. Si un ETD solicita esta facilidad puede incluir en el paquete de solicitud de llamada un campo de datos de hasta 128 octetos. La Red sólo progresará el paquete de solicitud de llamada si el ETD destino está abonado a la facilidad de aceptación de selección rápida. La Red indicará además en el campo de facilidades del paquete de indicación de llamada si el ETD origen ha solicitado restricción de respuesta o no. Si no se solicita restricción en la respuesta, el ETD llamado podrá contestar al paquete de llamada entrante, con un paquete de llamada aceptada o solicitud de liberación con campos de datos de usuario de hasta 128 octetos que serán procesados hacia el ETD origen. Si contesta con el paquete de llamada aceptada, la llamada entra en fase de datos normal, aplicándose todos los procedimientos establecidos. Una vez establecida la fase de datos normal, la Red no admitirá ya liberaciones con datos. Si se solicita restricción en respuesta, el ETD llamado sólo podrá responder al paquete de llamada entrante con un paquete de solicitud de liberación con campo de datos de usuario de hasta 128 octetos. Si el ETD destino responde con llamada aceptada, la Red liberará la llamada con causa "Error de procedimiento" y diagnóstico "Paquete no permitido". Si el ETD destino no está abonado a la facilidad de aceptación de selección rápida, la Red no genera comunicación y manda una liberación hacia el llamante con causa "Destino no suscrito a selección rápida".

*El **Simulador X25** envía la señal horaria cuando se selecciona esta opción y no contempla la posibilidad de respuesta. Puede comprobar este hecho analizando el paquete de llamada desde los analizadores..*

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.5.12.Aceptación de selección rápida

Opción asociada a la anterior. Esta facilidad opcional de usuario autoriza a la Red a transmitir al ETD abonado a la misma llamadas entrantes que solicitan la facilidad de selección rápida. Si el ETD no está suscrito a esta facilidad la Red no progresará la llamada al ETD destino e informará al ETD origen mediante una liberación con causa "Destino no suscrito a aceptación de selección rápida".

3.5.5.13.Negociación de los parámetros de control de flujo

Esta opción permite a los terminales que la lleven activa asignar dinámicamente en el establecimiento de las llamadas el tamaño de las ventanas de Nivel 3 tanto en transmisión como en recepción. Permite al usuario la negociación de los parámetros de control de flujo en cada llamada particular. Los parámetros de control de flujo que se consideran son los tamaños del paquete y de las ventanas de nivel de paquete. Si el ETD que genera la llamada está suscrito a esta facilidad, puede solicitar separadamente para cada dirección de transmisión el tamaño de paquete y de la ventana a aplicar en dicho interfaz. Si no se especifica nada en la llamada, se toman los valores por defecto normalizados o contratados con la Red.

Cuando un ETD está suscrito a esta facilidad, la Red indicará en el paquete de llamada entrante los valores de ventana y tamaño de paquete a partir de los cuales puede comenzar la negociación con el ETD. La Red decidirá en cada momento, dependiendo de su estado, la relación entre los valores de los parámetros solicitados en el paquete de solicitud de llamada y los indicados en el paquete de llamada

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

en todos los paquetes de llamada conectada los valores para los tamaños de paquete y ventana aplicables a ambos sentidos de la transmisión en dicho interfaz y para la llamada virtual correspondiente. Las indicaciones que incluye la Red en el paquete de llamada conectada son función de las solicitudes de valores incluidas por el ETD en el paquete de solicitud de llamada.

Configure esta opción sobre los terminales 3618 y 7342. Fije así mismo los valores de ventana en transmisión y recepción en 6 y 5 para el primero y 5 y 3 para el segundo. Con los nodos fijaremos así mismo los valores de 5 y 5 para la puerta del primero y 4 y 4 para la del segundo. Lance una llamada desde el 3618 al colateral. Mantenga activos los distintos analizadores de terminales y nodos para observar el proceso. Al progresar el paquete de llamada, el primer nodo restringe estos valores a 5 en transmisión y 5 en recepción. El nodo73 los baja a 4 y 4. Por último el terminal 7342 fija su restricción en el valor de 3 para la ventana de recepción, lo cual es indicado en el paquete de confirmación de llamada. Al completar el proceso los valores de ventana quedan fijados en 4 para la transmisión y 3 para la recepción.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.6.Configuración Por Defecto

En esta sección se verá qué configuraciones carga el **Simulador X25** sobre los terminales y nodos cuando se hace clic sobre esta opción.

3.5.6.1. Ventana de nivel 2

Se fijan el valor 7 tanto para la ventana de transmisión como para la de recepción y se definen los valores de 30 y 10 para los temporizadores T1 y T3 respectivamente y 5 para el contador N2.

3.5.6.2. Ventana de CVP

Se fija el valor 2 para las ventanas en recepción y transmisión. Se asignan los siguientes CVPs:

Canal 100 del terminal 3618 que le une con el canal 100 del terminal 7342.

Canal 101 del terminal 7342 que lo une con el canal 100 del terminal 5981.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.5.6.3.Ventana de CVCs

Se fija el valor 2 para las ventanas en recepción y transmisión. Se asignan los siguientes CVCs:

Terminal 3618: 4 CVCs Entrantes Salientes en los canaes 101, 102, 103 y 104.

Terminal 3677: !CVC Entrante. canal 100. 2 CVCs Entrantes/Salientes, canales 101y 102. 1 CVC Saliente, canal103.

Terminal 5981: 4 CVCs Entrantes Salientes en los canaes 101, 102, 103 y 104.

Terminal 7342: 3 CVCs Entrantes Salientes en los canaes 102, 103 y 104.

3.5.6.4.Facilidades

No se fija ninguna facilidad para ningún terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.6.SUPERVISION DEL SIMULADOR X25

Se verá en este apartado las opciones que tiene el usuario para seguir a nivel de algoritmo el desarrollo de la ejecución del **Simulador X25**. Se podrán seleccionar objetos y procedimientos para seguir la traza del programa. También se podrá capturar datos en archivos direccionados por el usuario a fin de facilitar el estudio de los mismos.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.6.1.INFORMES

Se trata de una opción que permite un seguimiento de la norma X25 a través de los distintos elementos del **Simulador X25**. En este apartado se incluyen las facilidades de programa que permiten controlar a nivel lógico la evolución del mismo. Se trata de un conjunto de herramientas que permiten conocer el estado y desarrollo de los distintos objetos del programa. Su única opción es el panel de estado.

Ilustración 3.1. Informes.

3.6.1.1.PANEL DE ESTADOS

Bajo este epígrafe se abren las posibilidades de seguir las evoluciones del **Simulador X25**. Se puede filtrar los elementos que se deseen y así obtener un informe selectivo de todas las decisiones y estados por los que pasan los objetos seleccionados. Este panel está pensado para mejorar la comprensión de los procesos que se suceden tanto en los elementos como en los niveles que intervienen en una comunicación bajo protocolo X25.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

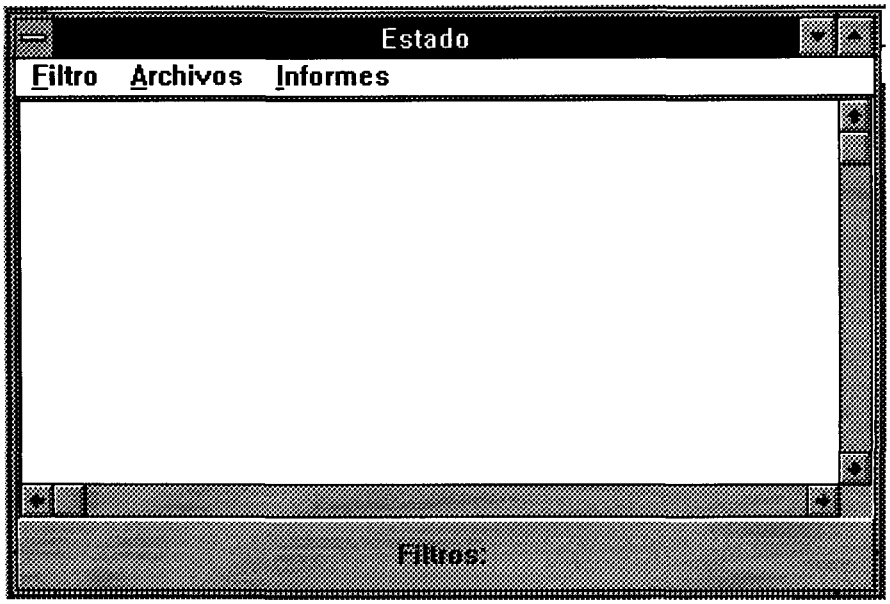


Ilustración 3.2. Panel de estados.

Despliega un cuadro cuyas posibilidades son:

3.6.1.1.1.Filtro

Abre todas las posibilidades de filtraje para los elementos a ser analizados. Seleccionando uno u otro y combinándolos entre sí se podrá estudiar los eventos que suceden a los objetos del programa con significación propia. Sus posibilidades son:

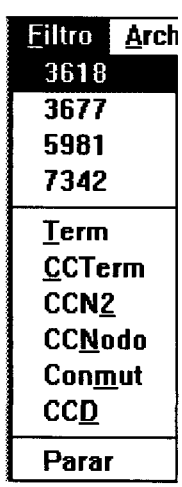


Ilustración 3.3. Filtros.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Despliega un cuadro con los objetos terminales definidos en la Red. Se pueden seleccionar cuantos terminales se quieran para visualizar las acciones que el **Simulador X25** toma para mantener dentro de protocolo a estos elementos.

*3.6.1.1.1.2.**Term***

Filtra las acciones que controla el objeto Terminal. Define las acciones de nivel superior que acontecen en un elemento terminal.

*3.6.1.1.1.3.**CCTerm***

Filtra las acciones que controla el objeto Controlador de comunicaciones del Terminal. Define las acciones de Nivel 3 que maneja el programa para mantener al terminal dentro del protocolo.

*3.6.1.1.1.4.**CCN2***

Filtra las acciones que controla el objeto Controlador de comunicaciones de Nivel 2. Define las acciones que se refieren al Nivel 2 en las comunicaciones de terminales y nodos.

*3.6.1.1.1.5.**CCNodo***

Filtra las acciones que controla el objeto Controlador de comunicaciones del nodo. Define las acciones de Nivel 3 en las comunicaciones vistas desde el nodo.

*3.6.1.1.1.6.**Conmut***

Filtra las acciones que controla el objeto Conmutador. Define las acciones de nivel superior que ocurren en los nodos de Red.

*3.6.1.1.1.7.**CCD***

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

*3.6.1.1.1.7.**CCD***

Filtra las acciones que controla el objeto Controlador de comunicaciones de datagrama (ruta). Define las acciones de Nivel 3 que ocurren en la comunicación entre nodos.

*3.6.1.1.1.8.**Parar***

Ejecuta la orden de parar las capturas del panel de estados.

Para comprender la potencia y utilidad de esta herramienta es preciso que escoja selecciones y combinaciones de todas ellas. Empieze por escoger el terminal 3618 y Term. pruebe a establecer canales con otros terminales y mandar mensajes de uno hacia otro. Ejecute rearranques en los terminales distantes y visualice el resultado en el panel de estados. Limpie el panel con la opción correspondiente. Cambie la opción Term por CCN2 y repita el proceso.

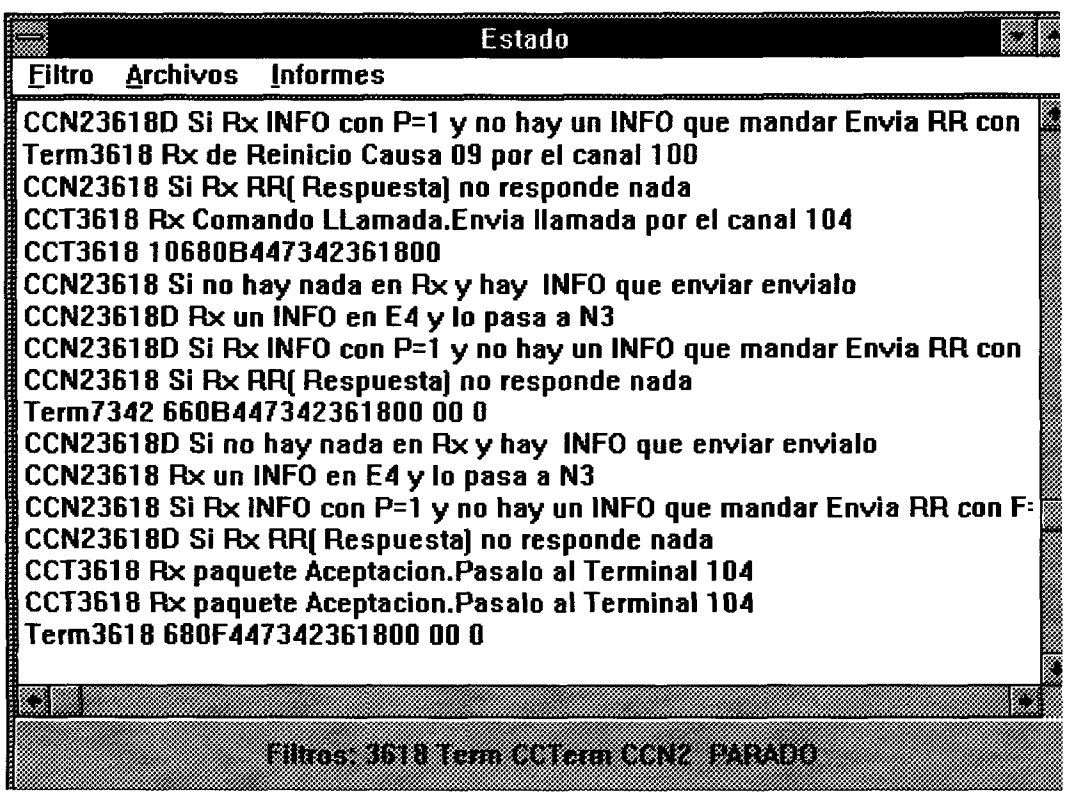


Ilustración 3.4. Panel de estado.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Especifica los ficheros con sus correspondientes ubicaciones y vías de acceso donde se almacenarán los datos capturados por el panel. Sus opciones son :

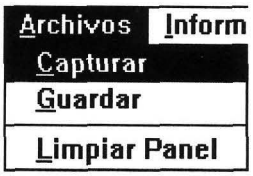


Ilustración 3.5. Archivos.

*3.6.1.1.2.1.**Capturar***

Con esta alternativa podemos desviar los datos que se filtran en el panel hacia un fichero. Muestra su propio cuadro configurable de Windows que hace referencia a la ubicación de ficheros.

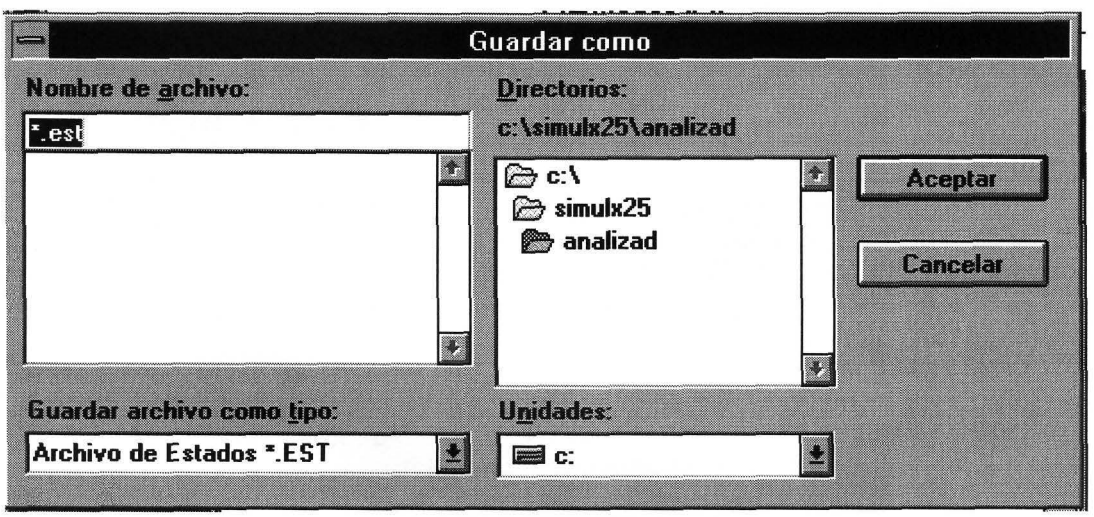


Ilustración 3.6. Cuadro de captura.

*Se desviarán los datos que recoja el Panel de estados a partir de este momento para grabarlos en el archivo Prueba1.txt. Abra el cuadro de diálogo que se encuentra en **Archivo**, **Capturar** y direccione un subdirectorio donde ubicar este archivo. Observe que este archivo tiene formato de texto y puede analizarlo con cualquier procesador que acepte este modo de trabajo.*

*3.6.1.1.2.2.**Guardar***

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.6.1.1.2.2. **Guardar**

Con esta alternativa se puede desviar los datos que se hallan en el buffer del panel hacia un fichero. Muestra su propio cuadro configurable de Windows que hace referencia a la ubicación de ficheros.

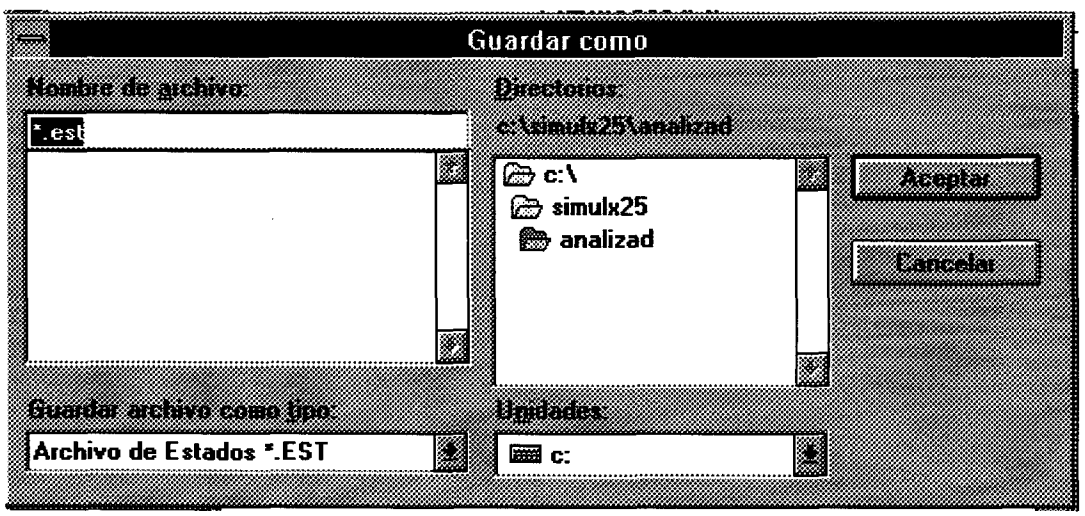


Ilustración 3.7. Guardar.

Se aprovecharán los datos que tiene almacenados en el buffer del Panel de estados en este momento para guardarlos en el archivo Prueba1.txt. Abra el cuadro de diálogo que se encuentra en ***Archivo***, ***Guardar*** *y direccione un subdirectorio donde ubicar este archivo. Observe que este archivo tiene formato de texto y puede analizarlo con cualquier procesador que acepte este modo de trabajo.*

3.6.1.1.2.3. **Limpiar panel**

Ejecuta la orden de limpiar de la pantalla todo lo que se halle visualizado.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.6.1.1.3.Informes

Controla las configuraciones y canales que tiene establecidos cada nodo de la Red. Sus opciones son:

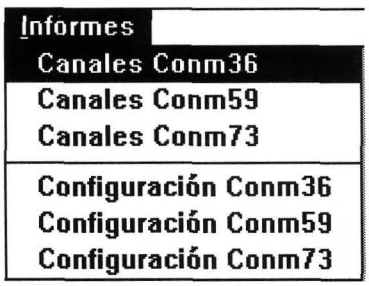


Ilustración 3.8. Informes.

3.6.1.1.3.1.Canales de los Conmutadores

Hacen alusión a los distintos canales que tienen establecidos cada uno de los nodos. La información que se muestra informa acerca del tipo de canal, su numeración, el ETD origen y destino y los parámetros que se han definido para el mismo.

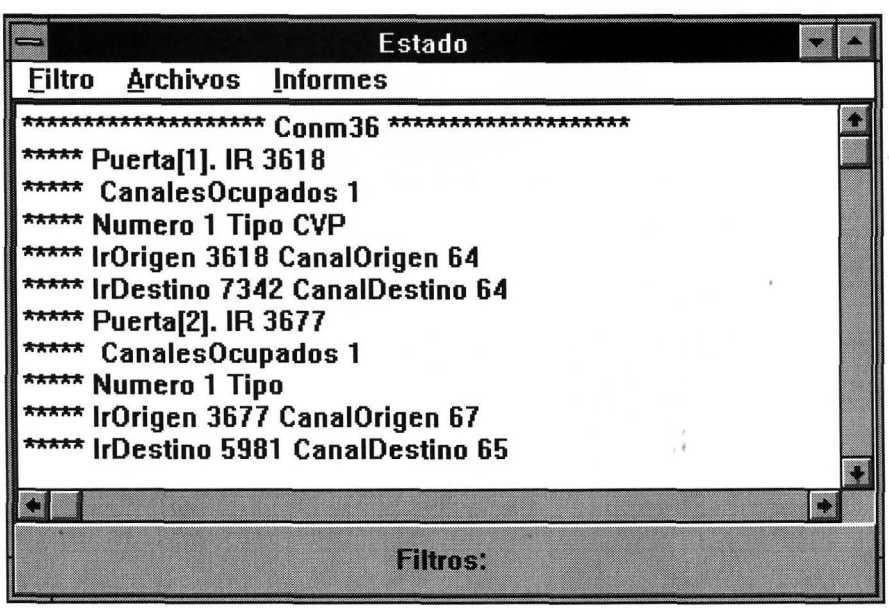


Ilustración 3.9. Canales de los conmutadores.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.6.1.1.3.2. *Configuración de los Conmutadores*

Explica la configuración de cada puerta en el momento de ejecución del programa.

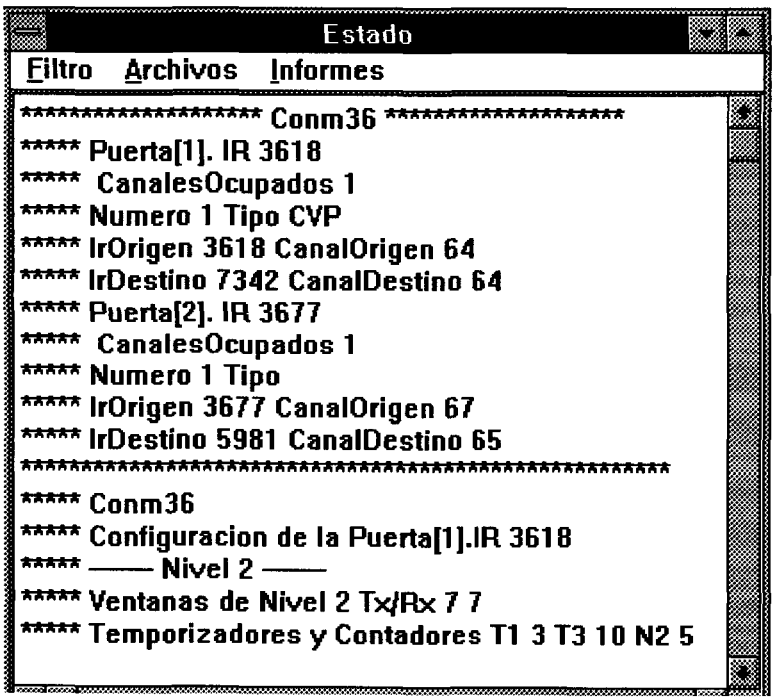


Ilustración 3.10. Configuración de conmutadores.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.7.CONTROL DEL SIMULADOR X25

En este apartado se definen todas las posibilidades de configuración de que disponen los distintos elementos del **Simulador X25**. Algunas ya se han estudiado a través de las condiciones de uso de la aplicación en los elementos de terminales y nodos. Otros se verán a continuación.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.7.1.Config.

Esta opción da lugar a los diferentes submenús de configuración de los elementos del programa. Estas configuraciones se guardan en ficheros que la mantienen después de abandonar la operación.

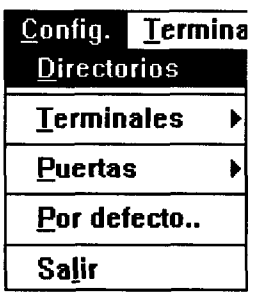


Ilustración 3.1. Configuraciones.

3.7.1.1.Directorios

Se abre desde aquí un cuadro de diálogo donde se especifica los subdirectorios y vías de acceso donde se ubican los archivos de configuración del programa, así como este mismo. Los ficheros a ser direccionados son:

3.7.1.1.1.**Principal**

Se guardará en este subdirectorio el programa **Simulador X25**.

3.7.1.1.2.**Configuraciones**

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.7.1.1.2.**Configuraciones**

Se guardan en este subdirectorio los ficheros que almacenan las configuraciones actuales de terminales y nodos de Red del programa.

3.7.1.1.3.**Analizadores**

Se guarda en este subdirectorio los ficheros que se crean para recoger los datos provenientes de las capturas de los analizadores.

3.7.1.1.4.**Envíos**

Se guardan en este subdirectorio los ficheros que se podrán transmitir como tales desde la opción de envíos del menú de terminal. Estos ficheros han de ser exclusivamente de texto para asegurar el correcto funcionamiento del programa.

3.7.1.1.5.**Ayuda**

En este subdirectorio se guardan los diferentes ficheros de ayuda dinámica que ofrece el programa durante la ejecución del mismo.

3.7.1.2.Terminales

Esta opción coincide con el icono de Configuración de terminal situado en el tablero de trabajo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.7.1.3.Puertas

Esta opción coincide con el icono de Configuración de nodo presente en el tablero de trabajo.

3.7.1.4.Por defecto

Esta opción es la que permite cargar todos los elementos del **Simulador X25** con una configuración predeterminada que suministra el programa.

3.7.1.5.Salir

Haciendo clic en esta opción se finaliza la ejecución del programa **Simulador X25** y se cede el control al entorno Windows.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.8.AYUDAS

En este apartado se encuentran las opciones de ayuda dinámica y tutorial que ofrece el **Simulador X25**.

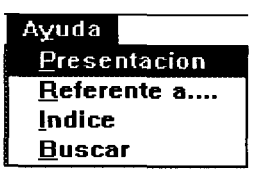
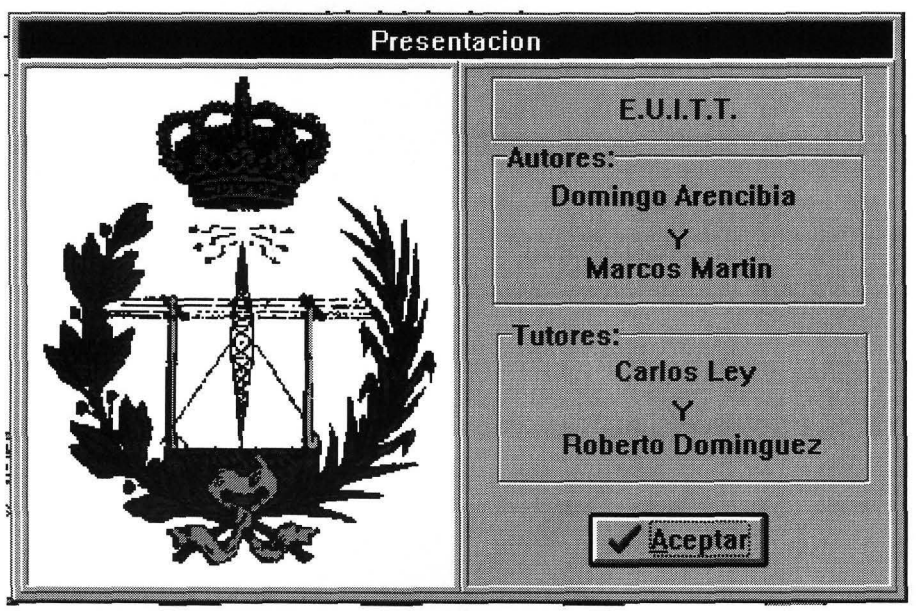


Ilustración 3.1. Ayuda.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.8.1.Presentación

Esta elección despliega el cuadro de presentación del programa.



© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.8.2.Referente a...

Da acceso al curso interactivo del protocolo X25 a través del **Simulador X25**. Con esta opción se activa el modo de ayuda de tal forma que cuando se haga clic sobre algún evento del programa se desplegará un cuadro de ayuda referente al evento que se cuestiona. Mientras esté activo esté menú a la espera de elegir el evento que se desea interrogar, el formato del cursor cambia añadiendo a la flechita habitual de Windows un cuaderno de consulta.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

3.8.3.Indice

Se abre una ventana con la relación de los temas que pueden ser consultados.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4

Estructura y Algoritmos

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.1 Programa Simulador X25.

El programa SimuladorX25 está realizado con Delphi 1.0. Este programa permite trabajar en el desarrollo de aplicaciones basadas en MSWindows sobre el lenguaje de programación Pascal.

Delphi distribuye el código entre unidades y archivo de proyecto. El archivo de proyecto es el archivo principal que organiza el resto del programa. El archivo de proyecto se llama en nuestro caso SX25.DPR. Este archivo da nombre al archivo ejecutable y registra todas las unidades que van a formar el programa una vez compilado. El resto del código del programa se organiza en unidades. En este proyecto se han creado las siguientes unidades.

1. SIMX25.PAS

2. TERM1. PAS

3. CCTERM.PAS

4. CCTERM2.PAS

5. CCNODO.PAS

6. CONMUT.PAS

7. CCDAT.PAS

8. ETFI.PAS

9. CFGNODO.PAS

10.ANALIZAD.PAS

11.ANFIL.PAS

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

12.TRAM.PAS

13.CALCULO.PAS

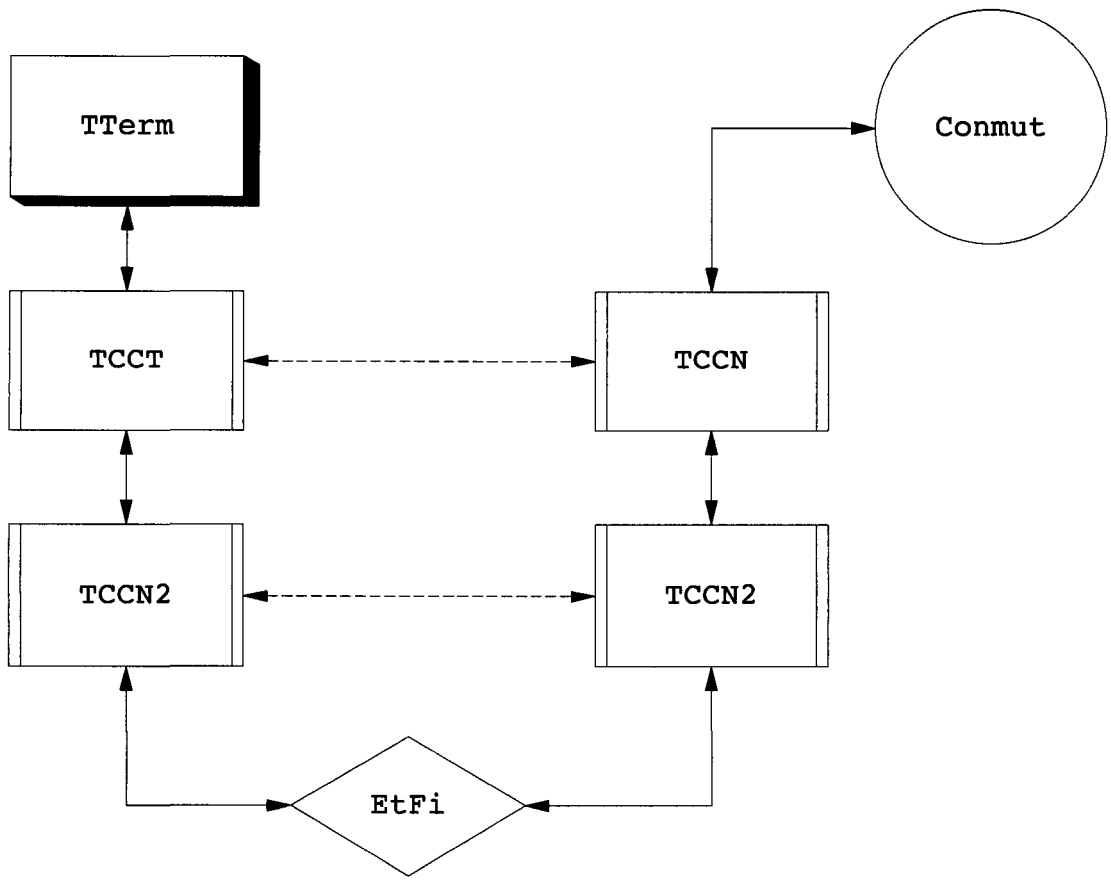


Ilustración 1.Objetos implicados en laconexión

En los pròximos capítulos se explicarán las funciones que tienen cada una de las unidades implicadas en el presente proyecto: CCTerm, CCTerm2, Term1, EtFi, Analizd, AnFil, Tram y Cálculo, así como el algoritmo que se implementa en el código.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2 TERM1.Terminal de usuario.

Esta unidad contiene el código del terminal de usuario conectado a la Red X25. Su función principal es la de ofrecer al usuario del programa una simulación de las posibilidades que tiene un terminal conectado a una Red X25. Por lo tanto se implementa la capacidad de enviar datos para construir paquetes de datos X25 así como las órdenes necesarias para que la capa inferior genere los paquetes de llamadas, rearranques, reinicios etc.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.1 Objetos

Se crea el objeto TTerminal que visualiza una ventana que consta de un menú, unas lengüetas de selección y un editor de Tx y otro de Rx por cada uno de los canales configurados.

4.2.1.1 TMemo

Se crean objetos editores Delphi TMemo, dos por cada canal que sea configurado, uno para la transmisión y otro para la recepción. En la transmisión el usuario puede escribir el texto a transmitir y pulsa "Intro", con lo que el texto es enviado. El objeto TMemo siguiente es el editor de recepción, que se visualiza conjuntamente con el anterior. Es en el editor donde aparece el contenido de los paquetes de datos enviados a este terminal por el terminal distante.

Dado que se ha decidido limitar el número máximo de canales configurables a cinco, se han creado los siguientes objetos para dar forma a esta ventana: Tmemo1, Tmemo2, TMemo3, Tmemo4, TMemo5, TMemo6, TMemo7 ,TMemo8, TMemo9, y TMemo10. Los pares están destinados a recibir y los impares a la transmisión.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.1.2 TTabSet

El objeto capaz de crear las lengüetas es el TabSet1. Mediante el procedimiento TabSet1Click, que se ejecuta cada vez que el usuario selecciona una lengüeta con el ratón, se puede seleccionar cualquiera de los canales configurados.

Cada vez que se produzca el establecimiento de un canal, en su lengüeta se escribirá el número de canal utilizado en notación hexadecimal y el Identificativo de Red del terminal remoto.

4.2.1.3 TPanel

El objeto TPanel aparece en la parte inferior de la ventana del terminal y es el encargado de mostrar el estado del terminal. Cada vez que se produzca un evento reseñable en la comunicación se escribirá en este objeto una indicación.

4.2.1.4 TMainMenu

Es el objeto de la librería de Delphi que crea el menú de la ventana. De él cuelgan los objetos TMenuItem que son las órdenes seleccionables del menú. Pulsado una de las opciones del menú se llama a un procedimiento que atiende este suceso.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2 Procedimientos y funciones principales

4.2.2.1 FormCreate

Procedimiento al que llama el programa cuando se le ordena crear la ventana del terminal. Se encarga de dar el valor de inicio a las variables fundamentales. Pone a cero las variables de transmisión, limpia los editores, configura los editores de recepción para que sean de solo lectura, da el valor máximo de líneas por editor igual a 200, etc.

4.2.2.2 TrasmiteDatos

Es la función encargada de transmitir los paquetes de datos a el objeto TCCT de la unidad CCTerm. La función transmite datos es del tipo "String",- Cadena de caracteres -,. Toma el valor de la variable TxDatos y pone la variable HayDatosTx como falsa y vacía la variable TxDatos. Esta función es llamada por el procedimiento IDLE del objeto TRedX25 en la unidad SimX25.Pas así como las cuatro siguientes,

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2.3 RecibeDatos(Datos : String)

Procedimiento de recepción de paquetes de datos procedentes del objeto TCCT correspondiente. Pone la cadena recibida en la variable RxDatos y activa la bandera HayDatosRx. Sólo se encarga de los paquetes de datos. Para transmitir las órdenes al nivel tres inferior se utilizarán otros procedimientos.

4.2.2.4 TransmiteComandos

Función idéntica a TransmiteDatos. Se encarga de transmitir cadenas de caracteres, -Paquetes-, tomados de la variable TxComandos, y desactiva la bandera HayComandosTx.

4.2.2.5 RecibeComandos

Procedimiento encargado de recibir las indicaciones de nivel tres que modifican el comportamiento de la comunicación. Los paquetes recibidos pasan a la variable RxComandos y se activa la bandera HayComandosRx.

4.2.2.6 Control

Es el procedimiento fundamental de análisis y actuación a partir de los datos recibidos. Este procedimiento se activa desde el Idle del objeto principal TRedX25 cada vez que hay Datos o Comandos en recepción,es decir, cada vez que las variables HayDatosRx o HayComandosRx están activas.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Por tanto podemos dividir el procedimiento en dos condiciones

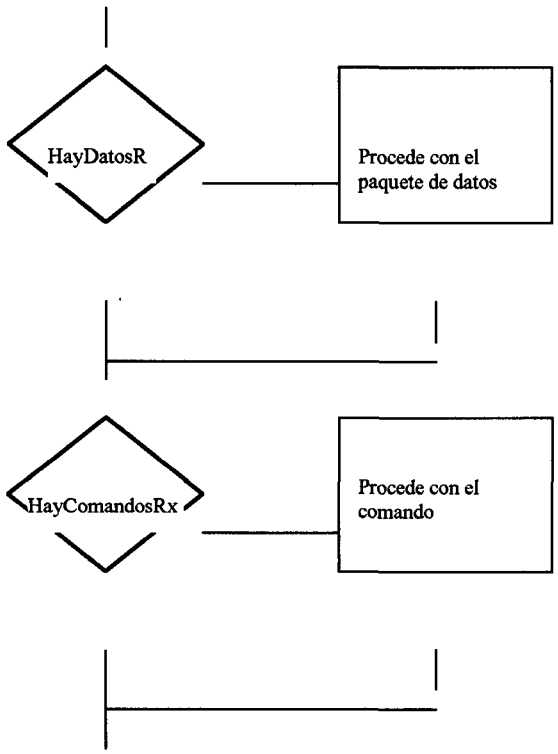


Ilustración 2.Procedimiento Control.

4.2.2.6.1 Si hay datos en recepción

En la primera condición se mira si se tiene un paquete de datos. En tal caso se extrae su canal, su IGF y el campo de datos de usuario. Posteriormente se comprueba por cada uno de los canales configurados , si este canal es coincidente con alguno de ellos. Si lo es, se estudia el bit Q, si este bit esta activo, se mira el contenido de el paquete de datos. Si el contenido del paquete de datos es "CLEAR" se procede a limpiar la pantalla del editor. Si el bit Q no está activo, se inserta el campo de datos de usuario como línea en el editor correspondiente de recepción mediante el procedimiento MiraLineaN.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

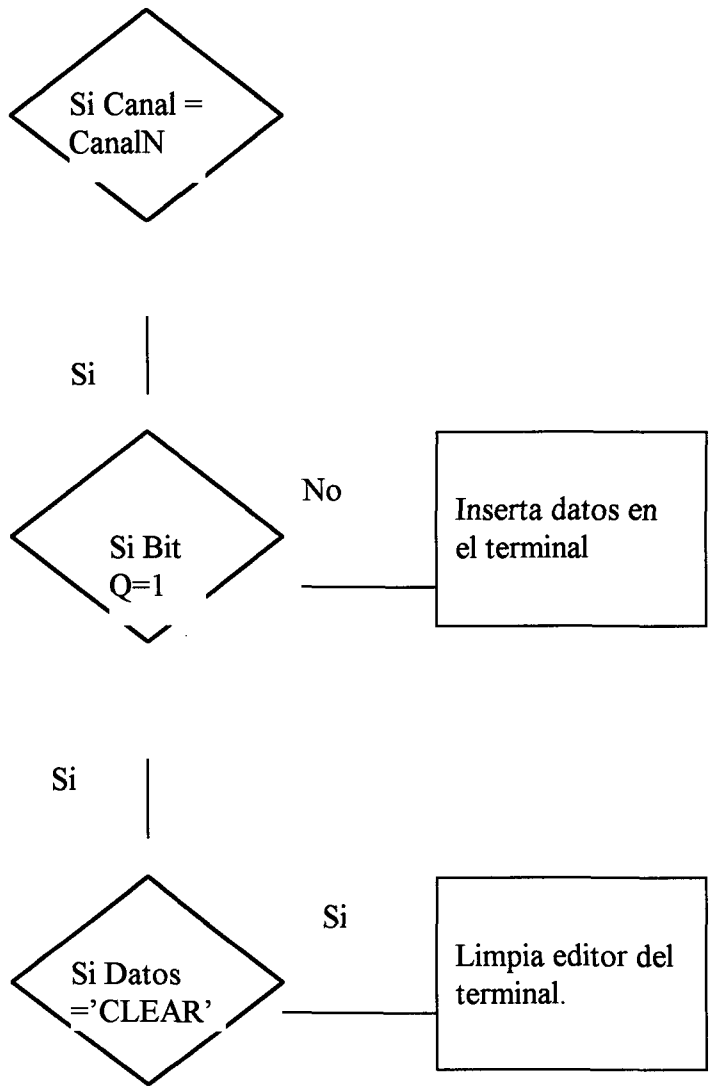


Ilustración 3.Recepción de paquetes de datos.

4.2.2.6.2 Si hay comandos en recepción.

En este caso se realiza una comparación "Case of" con el octeto de tipo del paquete recibido, creándose una subrutina para cada uno de los valores de los paquetes posibles.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2.6.2.1 Solicitud de llamada

Primero se avisa del evento en el Panel del terminal, extrayendo el Ir origen de la llamada y el canal del mismo paquete.

Mediante el canal y una comparación "Case of" determina a que editor de recepción y a que lengüeta corresponde el nuevo canal.

Si el canal no corresponde en la configuración a un canal permanente se activan los editores de transmisión y recepción. Se comprueba si hay selección rápida y si la hubiera se presenta su contenido en pantalla.

Si el canal por el que se recibe la llamada perteneciera a un canal permanente, se enviaría un mensaje de error al objeto TPanel.

4.2.2.6.2.2 Llamada aceptada

Se avisa de la recepción de un paquete de aceptación de llamada en el Panel del Terminal. Con la comparacion "Case of " se asigna el canal recibido en el paquete a una lengüeta y a un editor de recepcion y otro de transmisión.

Por último se hacen visibles estos editores y si hay selección rápida se presenta el contenido de datos de usuario del paquete en el editor de recepción

4.2.2.6.2.3 Solicitud de liberación

Al igual que en los casos anteriores ,presenta en el panel el evento, mira que canal trae el paquete y después encuentra a que editores de transmisión y de recepción pertenece.

A continuación comprueba que no sea un canal permanente. Si es conmutado libera los editores y los oculta .Si es permanente no hace nada.

4.2.2.6.2.4 Confirmación de liberación

Presenta el evento en el panel de estados de la ventana del terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2.6.2.5 Reinicio

Primero presenta el panel de estados.Extrae el canal del paquete.Si la causa de la reiniciación fuera "ETD fuera de servicio" o "Red fuera de servicio" miraría a que editores corresponde y cambiaría el color de fondo de estos para dar a entender que es un canal permanente no activo.

Si la causa fuera "ETD remoto en operacion",esto querría decir que el ETD remoto se ha activado. Por lo tanto miraría a que editores corresponde el canal y cambiaría el color de fondo y permitiría la recepcion ya que supondría desde este momento que el canal permanente esta activo.

En cualquiera de los dos sucesos anteriores se envía información del suceso al objeto TPanel.

Si el canal no fuera permanente se limitaría a presentar el evento en el panel de estados.

4.2.2.6.2.6 Confirmación de reinicio

Se limita a presentarla en la barra de estados.

4.2.2.6.2.7 Rearranque

En el caso del rearranque ,antes que nada se habilitan las opciones de menú que a continuación se enumeran: Llamada, Liberación, Reinicio y Rearranque. Esto es debido a que para iniciar el nivel 3 primero se recibe un Rearranque a la vez que se envía. Antes de establecerse el nivel 3 es imposible ordenar cualquiera de las opciones mencionadas anteriormente.

En el caso de una recepción de rearranque que no fuera por inicio de nivel 3, también se activarían los menús, pero al estar ya activos la acción no tiene ningún efecto. A continuación se presenta el evento en el panel de estados del terminal. Por último se comprueba si hay canales conmutados activos. Para ello se comprueba si algún editor tiene asignado un canal conmutado, si así fuera, se liberan los editores y se ocultan. Las lengüetas se borran de contenido

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2.6.2.8 Confirmación de Rearranque

Actualiza el evento en el panel de estados. Comprueba si alguno de los editores activos tiene asociado un canal conmutado. Si fuera así lo libera de contenido y lo oculta, a la vez que limpia la lengüeta.

4.2.2.6.3 Si está activo el envío de archivos.

Para esta opción del menú del terminal se útilizan las siguientes variables: EnvActivo, Permiso, ContPermiso.

EnvActivo es del tipo booleana e indica que está activo el envío de archivos.

Permiso. Variable booleana .Cada vez que se cumple N veces que el programa pasa la supervisión al procedimiento control, la variable Permiso se activa para permitir el envío de un paquete de datos.

ContPermiso.Variabl de tipo entero. Si Permiso no está activa, se incrementa cada vez que se cede el control al terminal. Una vez que la variable llega a N, Permiso se activa.

Cuando EnvActivo y Permiso están activas y no hay datos pendientes de ser transmitidos, se leen 128 caracteres del archivo seleccionado y se pasan por el canal indicado al procedimiento TxDatos. Si el archivo no contiene 128 caracteres esto quiere decir que éste es el último paquete a transmitir, con lo que se pasa el resto de caracteres al procedimiento TxDatos, se cierra el archivo y se desactivan las variables EnvActivo y Permiso.

En todos los paquetes enviados menos en el último se activa el bit M, antes de pasarlo al procedimiento TxDatos. En el último este bit se pone a cero. Una vez enviado el último paquete se activa la opción de envío de archivos del menú del terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2.7 MemoNKeyPress

Este procedimiento se activa cada vez que se pulsa una tecla en el editor de transmisión. Una vez pulsada, se comprueba que la cadena que se está formando en el editor no sobrepasa los 128 caracteres. Si llega a esta longitud, la trama es automáticamente enviada. Si la longitud es menor de 128, se comprueba que el caracter pulsado no es un retorno de carro, si lo fuera se enviaría el paquete a el nivel 3 para su transmisión

4.2.2.8 InslineaN y MiraLineaN

Procedimientos encargados de manejar el desplazamiento de las líneas en la pantalla. Dado que la memoria disponible es finita, estos procedimientos permiten que se utilicen un número determinado de líneas. Una vez rellenas todas las posiciones, la línea siguiente a presentarse en el objeto TMemo ocuparía la primera posición, eliminando el texto que anteriormente la ocupaba. Para marcar el próximo espacio a llenar, se utilizan los caracteres " >>".

4.2.2.9 N3618Click, N3677Clic, N5981Clic y N7342.

Procedimientos encargados de generar una llamada a los terminales indicados en sus nombres. Se activan desde el menú del terminal. Su función es la de lanzar una llamada al terminal cuyo IR (Identificativo de Red) esté indicado en el propio nombre del procedimiento. Para ello manda una cadena a la función TxComandos. Comprueba uno a uno los campos del registro Parámetros y añade en la zona de facilidades las que se van a seleccionar en esta llamada.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2.10 LiberaClick

Primero mira la lengüeta que está seleccionada actualmente. Por tanto, conoce el canal. Comprueba que el canal seleccionado no es permanente. A continuación limpia los editores de transmisión y recepción del canal y los oculta. Por último pasa la cadena correspondiente a un paquete de liberación con causa ETD distante a la función TxComandos.

4.2.2.11 ReiniClick

Introduce el paquete de reinicio con causa ETD remoto en la Función TXComandos.

4.2.2.12 RearranClick

Hace lo mismo que la función anterior con el paquete de Rearranque.

4.2.2.13 Archivo1Click

Presenta una ventana de diálogo para la elección del archivo a enviar. Utiliza el subdirectorio de envíos del objeto RedX25. Una vez que tiene el nombre del archivo, mira por que canal ha de transmitirlo mirando la lengüeta activa. A continuación abre el archivo, deshabilita la opción de envío de archivos del menú y activa la variable EnvActivo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.2.2.14 ResumenConfig

Este procedimiento envía al Panel de Estados la configuración del terminal. Con esto se pretende resumir la información de la configuración de terminales y nodos en un mismo editor, de manera que pueda ser salvado en un archivo, imprimirlo, etc.

Este procedimiento se ejecuta por orden del TEstado y ejecuta con él el procedimiento del objeto Tconm que hace la función igual a ésta desde la configuración del Nodo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3 CCTERM. Nivel 3 del Terminal

4.3.1 Objetos

En la unidad CCTERM se describen dos objeto:s el objeto Tcanal y el objeto TCCT.

4.3.1.1 Objeto Tcanal

Soporta el canal virtual de nivel 3.

4.3.1.1.1 Variables

1. IR. Indica el Identificativo de Red del terminal distante con el que se asocia el canal.
2. Número. El número que lo denota
3. Tipo. Si es permanente o conmutado.
4. Pr, Ps, M. Variables de control de ventana.
5. Canal. Número de canal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

6. VT, VR. Variables que apuntan el estado actual de las ventanas de transmisión y de recepción.

7. VentN3Tx, VentN3Rx.. Valor de configuración de las ventanas del canal.

8. EstadoN3. Indica el estado en que se encuentra el canal.

4.3.1.1.2 Procedimientos

Procedimiento Create. Este procedimiento es el que inicia el canal con todas sus variables. Inicializa las variables de control de ventana a cero. Al igual que las variables Vt y VR. El estado lo inicia a 'E1'.

El IR no está asignado todavía y se deja en blanco.

El tipo de canal se pasa al procedimiento como parámetro y con la comparación de si es permanente o no realiza la siguiente asignación:. Si es permanente asigna a las ventanas los valores de ventana de un canal permanente. Si es conmutado le asignará las correspondiente a los canales conmutados configurados.

4.3.1.2 Objeto TCCT

4.3.1.2.1 Variables principales.

4.3.1.2.1.1 Nombre

En esta variable se define el IR del ETD dentro del que trabaja el objeto TCCT.

4.3.1.2.1.2 Canal

Primero se crea una matriz de objetos Tcanal numerados del 100 al 104 y se asignan a la variable Canal definida dentro del objeto TCCT.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.1.3 CanalActivo

Es una matriz de variables booleanas númeradas del 100 al 104. Se utilizará para saber que canales asignados han sido creados y cuales no.

4.3.1.2.1.4 DatosRx

Variable de cadena de caracteres en la que se almacenan los datos recibidos del terminal

4.3.1.2.1.5 DatosTx

Variable del tipo String (Cadena de caracteres) en la que se almacena el paquete de datos de nivel 3 que se transmite al terminal.

4.3.1.2.1.6 ComandosRx

Variable del mismo tipo que las anteriores, en la que se almacenan los comandos recibidos del terminal.

4.3.1.2.1.7 ComandosTx

Cadena de caracteres que representa los comandos a transmitir hacia el terminal

4.3.1.2.1.8 PaqueteTx

Cadena de caracteres que representa los paquetes de nivel 3 a transmitir al objeto encargado de soportar el nivel 2.

4.3.1.2.1.9 PaqueteRx

Variable de tipo String que guarda los paquetes de nivel 3 recibidos del nivel inferior.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.1.10 HayDatosRx

Bandera que se activa cuando hay Datos de usuario en recepción. Se activa al recibir del terminal una cadena de datos de usuario.

4.3.1.2.1.11 HayDatosTx

Lo mismo que el anterior pero indicando al terminal de usuario que hay pendiente un paquete de datos de usuario a transmitir.

4.3.1.2.1.12 HayComandosRx

Variable del tipo booleana que se activa cuando hay comandos en recepción pendientes de ser tratados por el procedimiento control.

4.3.1.2.1.13 HayComandosTx

Bandera que indica que hay comandos pendientes de ser transmitidos al terminal de usuario.

4.3.1.2.1.14 HayPaquetesTx

Indica al tomar valor verdadero que existe un paquete pendiente de ser enviado al nivel 2 del ETD.

4.3.1.2.1.15 HayPaquetesRx

Bandera que indica, una vez activa, que ha llegado un paquete de nivel 3 al objeto TCCT desde el nivel 2 o nivel de enlace.

4.3.1.2.1.16 N2Activo

Puntero asociado a la variable que indica, dentro del objeto que controla el nivel 2, que éste está activo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.1.17 N2ActivoAnterior.

Variable booleana que toma el valor al que apunta N2Activo una vez que se sale del procedimiento control. De esta manera, si se compara N2ActivoAnterior con N2Activo se sabe si el nivel dos del terminal ha pasado de inactivo a activo o de activo a inactivo o no ha sufrido variación.

4.3.1.2.1.18 IGF

Almacena el IGF recibido en los paquetes de nivel 3 así como el utilizado en la construcción de las tramas que se envían.

4.3.1.2.1.19 Parámetros.

Variable de tipo registro definida en la unidad CFGNodo en el que se almacena la configuración del termianl de usuarios.

4.3.1.2.1.20 ArchCfg.

Variable archivo a la que se asigna el archivo en el que se encuentran los parámetros de configuración de nivel tres utilizados por el nivel 3.

4.3.1.2.2 Procedimientos principales

4.3.1.2.2.1 Create

Su función principal es iniciar las variables utilizadas por el programa. Pone todas las banderas o variables booleanas a su estado inactivo o falso. Inicia la variable IGF con el valor '10', con lo que indica que es un paquete de 128 caracteres y que no se activan ni el bit Q ni el bit D.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.2.2 TxDatos

Función que retorna una cadena que representa un paquete de datos de usuario, para que lo recoja el Terminal. Toma el valor de la variable DatosTx. Tambien cambia a falso el valor de HayDatosTx.

4.3.1.2.2.3 RxDatos

Procedimiento que toma como parámetro una variable 'string'. Esta cadena se pasa a la variable DatosRx y se activa la bandera HayDatosRx.

4.3.1.2.2.4 TxComandos

Función identica a TxDatos que actúa sobre los comandos transmitidos hacia el terminal

4.3.1.2.2.5 RxComandos

Procedimiento que repite la misma utilidad que RxDatos y que actúa sobre los comandos recibidos del terminal.

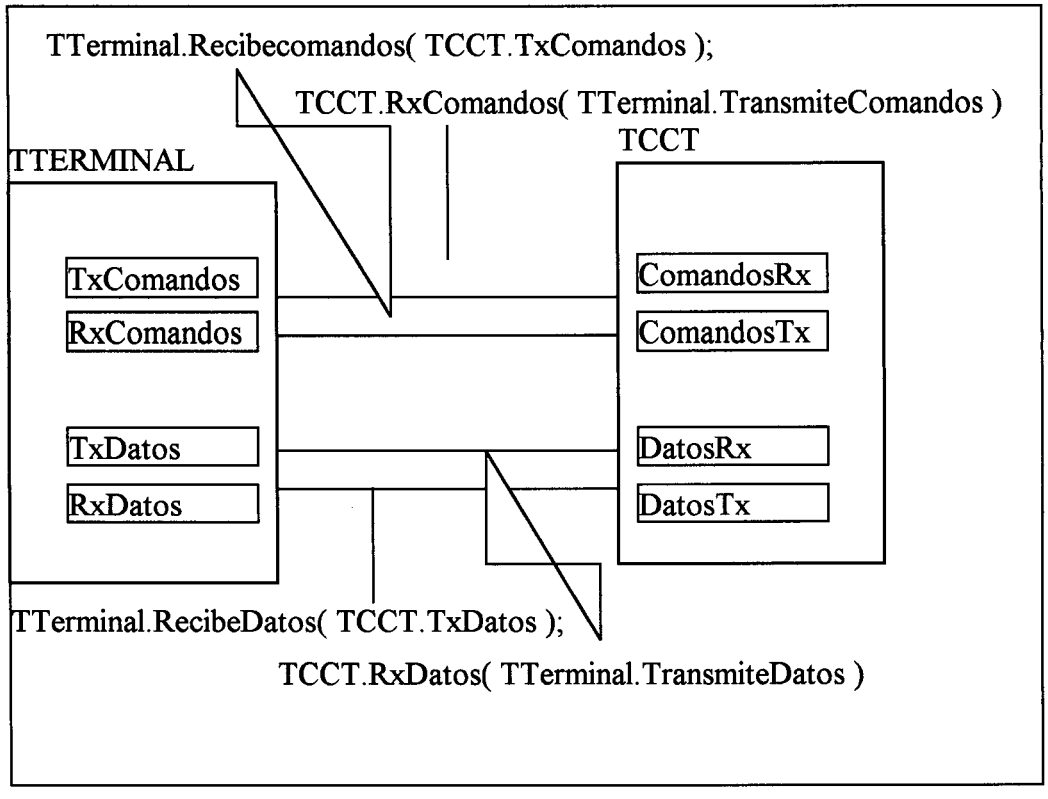
4.3.1.2.2.6 TxPaquetes

Función que se encarga de pasar el valor de PaqueteTx al objeto encargado de mantener el nivel 2. Anula la bandera HayPaqueteTx.

4.3.1.2.2.7 Rxpaquetes

Procedimiento que pasa el valor de la cadena que se le pasa como parámetro a la variable PaqueteRx. Activa la variable HayPaqueteRx.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006



Interconexión entre objetos.

Ilustración 4.Conexión entre Tterminal y TCCT

4.3.1.2.2.8 Control

Procedimiento encargado de implementar el protocolo de nivel tres. Se le ejecuta desde el procedimiento Idle del objeto RedX25 si hubiera Datos, Comandos o Paquetes en recepción.

4.3.1.2.2.8.1 Si hay Comandos en Recepción

Se estudia la variable HayComandosRx. Si está activa, se extrae, del paquete recibido, el canal, que se guarda en la variable CanalAct. Posteriormente se realiza una comparación 'Case of' con el octeto de tipo del paquete. Con cada uno de los valores posibles de este octeto se implementa una subrutina para tratarlos.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.2.8.1.1 Solicitud de llamada

El paquete de llamada recibido desde el terminal no tiene aún asignado canal .Por mantener la estructura de trama se decididió en el diseño del programa mantener este octeto a cero hasta que el objeto TCCT eligiera un canal válido para el paquete. Se comprueba que hay canales disponibles mediante el procedimiento HayCVCS. Si los hay, decide, mediante la función DecideCVCS, el canal que le asigna o, en su caso, la causa de liberación. Esta función toma el valor de la causa de liberación si ha encontrado dificultades en asignar un canal y si lo hubiera asignado correctamente toma el valor '00'. También toma en consideración si hay algo pendiente de transmitir. Si lo hubiera no haría nada a la espera de que se liberara el registro de transmisión.

Si se cumplen todas las condiciones anteriores se crea el objeto Tcanal asignado a este canal. En él se da valor a las variables IR destino, estado y se activa la variable CanalActivo. A continuación se progresa el paquete de llamada con el canal asignado a la variable de transmisión de paquetes y se activa la bandera HayPaqueteTx.

Despues de enviar el paquete se manda un mensaje al objeto Tpanel indicando el suceso acontecido.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

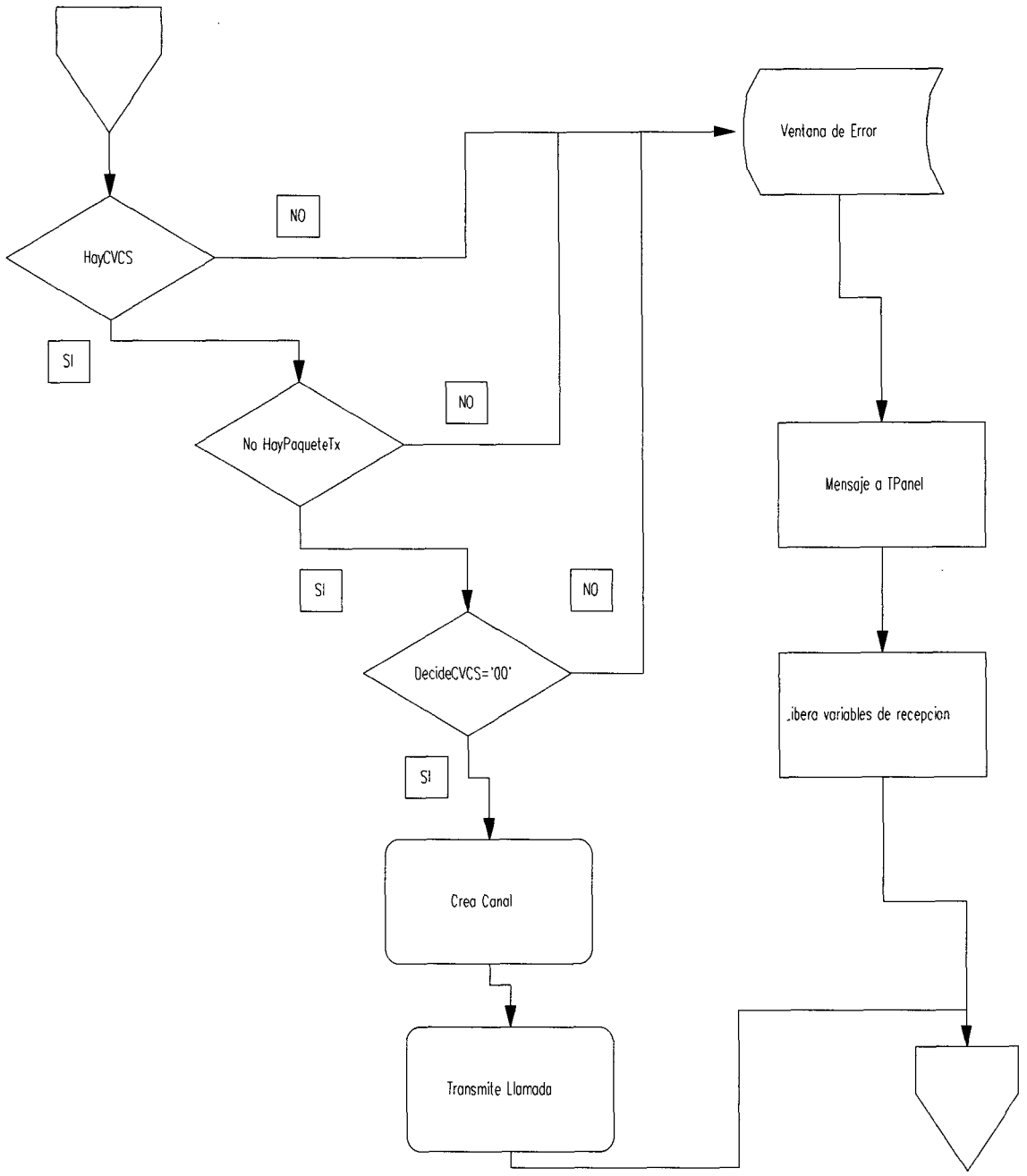


Ilustración 5.Solicitud de llamada

Si la función DecideCVC no pudiera asignar un canal, el programa abre una ventana de aviso indicando el motivo por el cual no ha podido progresar la llamada. A la vez que envía un mensaje a Tpanel. Después de esto se anulan las variables de entrada HayComandosRx y ComandosRx

Si HayCVCS indicara que no hay canales disponibles pasaría lo mismo que en el caso anterior.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.2.8.1.2 *Solicitud de liberación*

Primero comprueba que el canal es válido y está activo, si no hay paquetes pendientes de transmitir y el canal no es un canal permanente, si se dan todas las condiciones anteriormente mencionadas se progresa la solicitud de liberación hacia el terminal remoto. También se pasa el canal a estado P6 y se manda un mensaje al Tpanel. Por último, se liberan las variables de recepción HayComandosRx y ComandosRx.

4.3.1.2.2.8.1.3 *Solicitud de reinicio*

Si el canal es válido, está activo y no hay nada en transmisión, progresa el paquete, pasa el canal a estado D2, envía un mensaje a el Tpanel y pone a cero todas las variables de control de ventana.

4.3.1.2.2.8.1.4 *Solicitud de rearranque*

Se comprueba que viene por el canal 0 y que en el interfaz haya al menos un canal activo. También se comprueba que no hay ningún paquete pendiente de ser transmitido y entonces se progresa el paquete. Se liberan las variables de recepción y se envía un mensaje al Tpanel.

4.3.1.2.2.8.2 Si hay Datos en recepción

El objeto TTerminal pasa el paquete de datos con las variables Ps y Pr a cero. Es el TCCT quien se encarga de controlar las ventanas. Primero se comprueba que el canal que tiene configurado está activo. Si no lo está, se ignora el paquete. Se mira si hay paquetes en Tx, si los hubiera. no se haría nada. De esta manera en el siguiente paso del programa por esta rutina, encontraría de nuevo este paquete y continuaría su analisis. Si el canal está dentro de ventana, continúa en el análisis. Si no esta dentro de ventana, no hace nada. En la próxima pasada por esta rutina puede haberse abierto la ventana al recibir de la estación remota un Datos o un RR. Extrae el IGF del paquete recibido. Mira si viene con el bit M activo o no para actualizar esta variable en los parámetros del canal. A

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

continuación se construye el paquete y se envía. Se incrementan las variables de la ventana y se estudia el bit D del IGF. Si éste estuviera activo, se incrementa la ventana de transmisión hasta cerrarla. De esta manera se obliga a que sea un paquete enviado por el distante quien acepte este paque y que hasta que esto no suceda no se manden más. Si el bit D no está activo, se incrementa en una el contador de ventana de transmisión .

Una vez enviada una trama se pone a cero el contador de ventana de recepcion. La variable Ps que es la que se incluye en la trama, se incrementa en una unidad si su valor es menor que 7 y si no se le da valor cero. Por último se liberan las variables de recepción HayDatosRx y DatosRx y se manda un mensaje al Tpanel.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

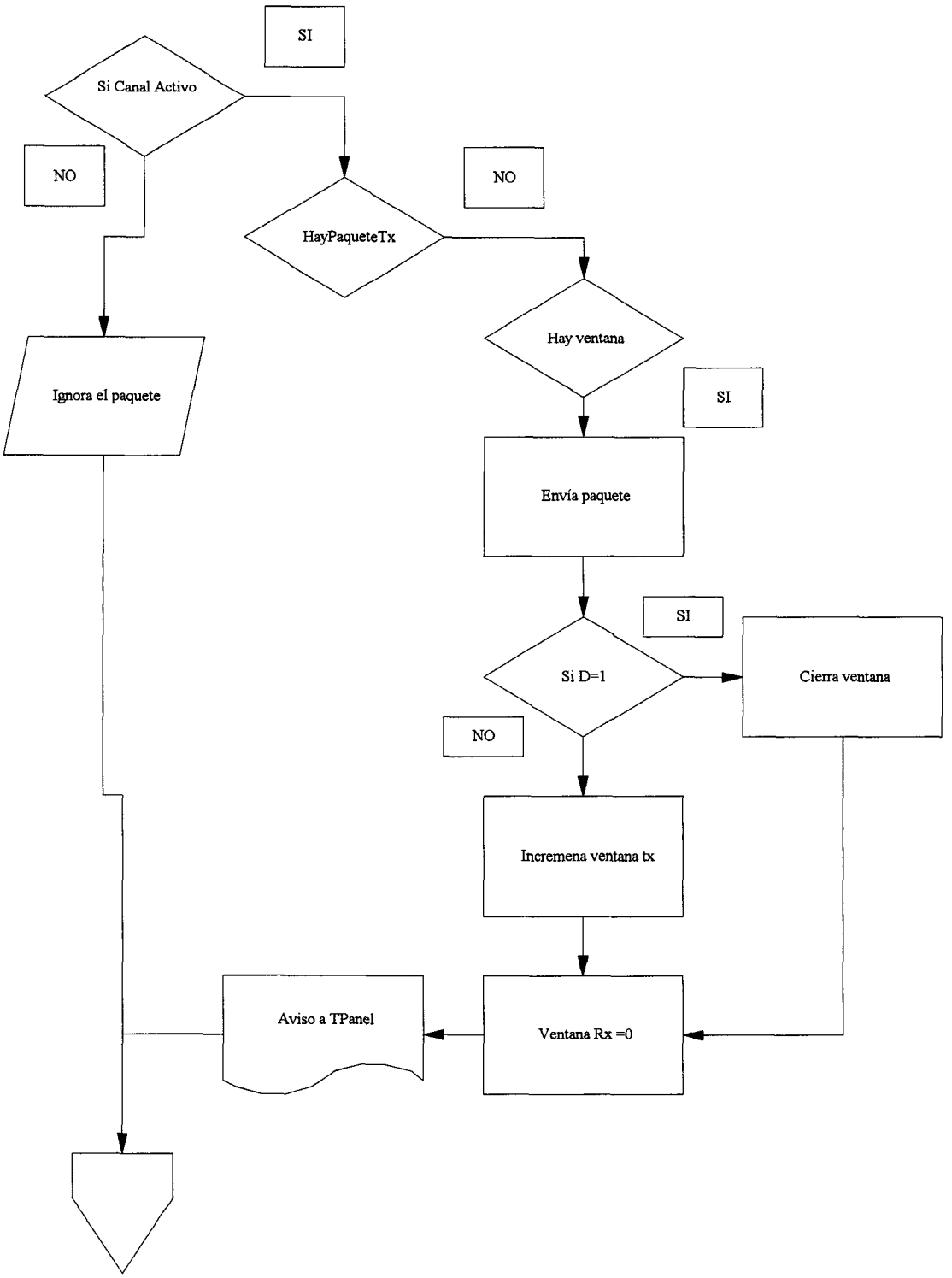


Ilustración 6.Recibido paquete de datos del Tterminal

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.2.8.3 Si hay Paquetes en recepción.

Si está activo HayPaqueteRx, es que se ha recibido un paquete de datos de la Red. Primero se extrae el canal del paquete y se asigna a la variable CanalAct.

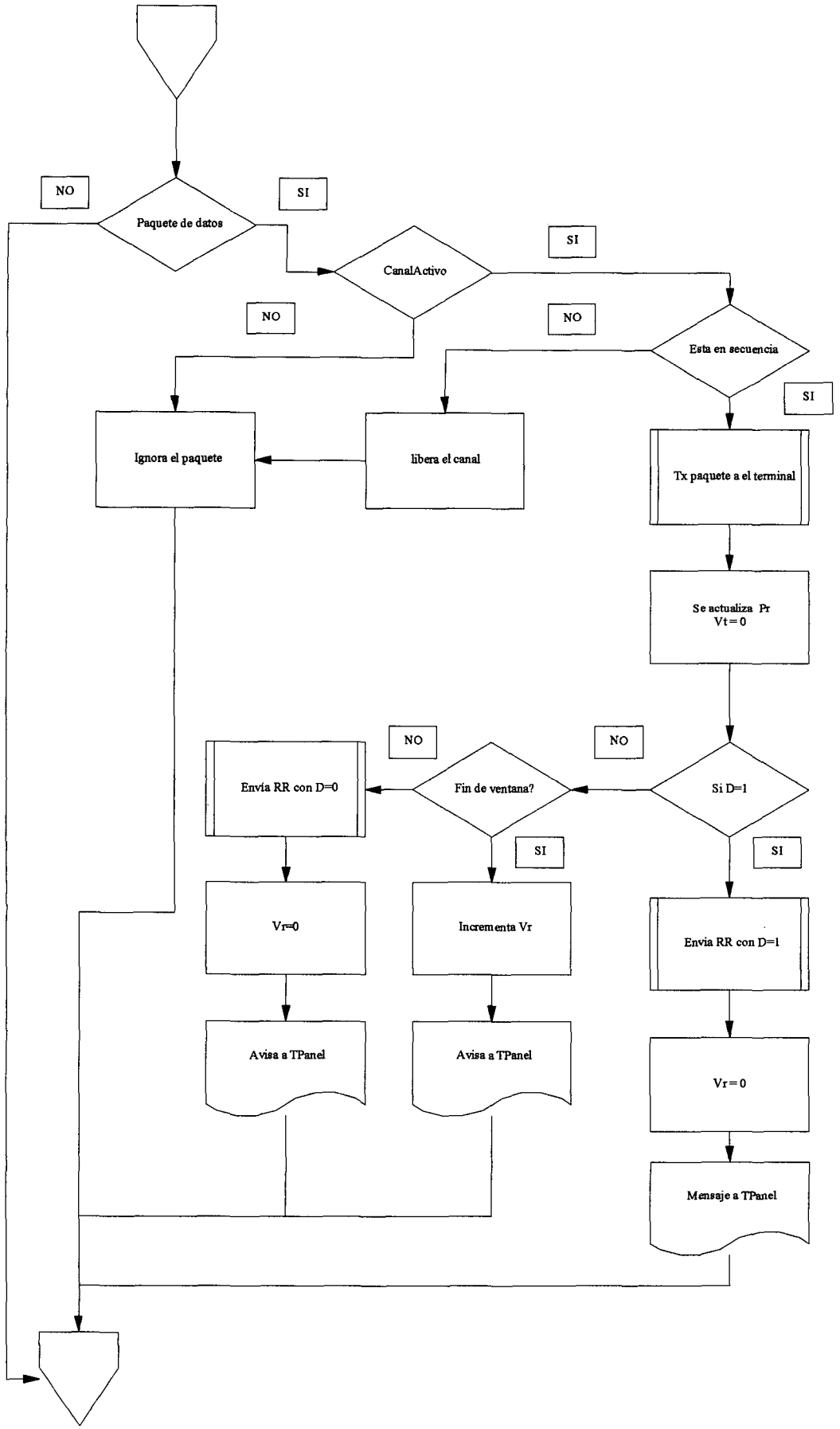
4.3.1.2.2.8.3.1 Paquete de datos

Si el byte de control es par, es un paquete de datos. Se comprueba que el canal corresponde a un canal activo. A continuación se mira si la secuencia de la ventanade recepción del paquete es la esperada. Si es la esperada se pasa el paquete hacia el terminal, se actualiza la variable Pr y se pone la ventana en transmisión a cero. A continuación se estudia el bit D

Si el bit D está a uno, envía un RR con el bit D activo, pone a cero la ventana de recepción y pasa un mensaje a Tpanel. Si el bit D está a cero mira si se ha llegado al límite de la ventana. En el caso de que no se haya llegado se incrementa la ventana de recepción y se avisa al Tpanel. Cuando se llega al límite de la ventana de recepción se envía un RR con el bit D a cero y se pone a cero la ventana de recepción. También se avisa al Tpanel.

En caso de que el paquete no sea el esperado en secuencia se envía un aviso de error al Tpanel y En caso de que llegue un paquete de datos por un canal no activo se ignora.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006



© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.2.8.3.2 RR

Si el octeto de control tiene el último semiocteto igual a uno es un RR. Si llega un RR pone a cero la ventana de transmisión de este canal (Vt=0), y avisa al Tpanel de la recepción de un RR.

4.3.1.2.2.8.3.3 Solicitud de llamada

Si el octeto de control de nivel tres de la trama recibida es igual a '0B' es un paquete de llamada. Inicia la aceptación mirando si el canal por el que se recibe la llamada es válido y está libre. A continuación mira por el procedimiento AceptaCVCE si este canal está configurado correctamente. Si es así, esta función devuelve '00', si no, entregaría la causa de la liberación a realizar. A continuación el procedimiento Aceptafacilidades mira si las facilidades indicadas en el paquete de llamada son posibles. Si lo son y está libre el registro de transmisión, se procede de la siguiente manera:

Primero se crea el objeto Tcanal adecuado, asignándole valor a las variables. A continuación estudia si hay negociación de los parámetros de control de flujo y si la hubiera se modifican los valores de ventana del canal con respecto a lo solicitado en el paquete de llamada. Una vez hecho esto se traslada el paquete al terminal y se envía una aceptación de llamada hacia la estación remota y un informe al objeto Tpanel.

Si no se aceptan las facilidades pedidas en el paquete de llamada, se libera. De igual manera, si no se acepta el canal solicitado en el paquete de llamada, se libera con la causa que entrega el procedimiento AceptaCVCE.

Si el canal no está entre los válidos, el paquete de llamada es ignorado.

4.3.1.2.2.8.3.4 Aceptación de llamada

Si en el octeto de control se encuentra la cedena '0F' es una aceptación de llamada.

Se activa la variable CanalActivo. Se mira si en la aceptación hay negociación de parámetros de control de flujo. Si lo hubiera se actualizarían los

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

valores de ventana para este canal. Se pasa el paquete como un comando al terminal y se avisa al Tpanel.

4.3.1.2.2.8.3.5 Solicitud de liberación

Si el octeto de control es igual a '13', es una solicitud de liberación. Si el canal está activo y no es un canal permanente, se procede a desactivarlo y a liberar el objeto Tcanal. Se envía el comando al Terminal, se envía hacia la estación remota una aceptación de liberación y se informa al Tpanel.

4.3.1.2.2.8.3.6 Confirmación de liberación

Si el octeto de control es '17' es una confirmación de liberación. Si el canal no es permanente, envía el comando al terminal, libera el objeto Tcanal y avisa al Tpanel.

4.3.1.2.2.8.3.7 Solicitud de reinicio

'1B' en el octeto de control de nivel 3 indica que es una solicitud de reinicio. Primero se ponen las variables de ventana a cero y se envía el paquete al terminal como un comando. Hacia el terminal distante se envía una aceptación de reinicio y se pasa una información de estado al Tpanel.

4.3.1.2.2.8.3.8 Confirmación de reinicio

Se ponen las variables de ventana a cero, se pasa el comando al Terminal y se avisa al Tpanel.

4.3.1.2.2.8.3.9 Solicitud de rearranque

Pueden suceder dos cosas: que hayan canales activos o que no los haya.

Si no hay ningún canal activo es un rearranque por inicio del nivel tres. En este caso, el terminal responderá con una solicitud de rearranque, ya que un cruce

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

de rearranques supone confirmación de ambos. En este caso, se avisa del evento al Tpanel.

Cuando hay canales activos, primero se envía una aceptación de rearranque hacia el distante. Posteriormente se pasa el comando al terminal y a continuación se libera uno a uno todos los canales activos que no sean permanentes.

4.3.1.2.2.8.3.10 **Confirmación de rearranque**

Si hay uno o más canales activos y estos no son permanentes se liberan. Se pasa el comando al Terminal y se avisa al Tpanel.

4.3.1.2.2.9 RRN3

Procedimiento que costruye un RR de nivel tres.

4.3.1.2.2.10 RNRN3

Procedimiento que construye un RNR de nivel 3

4.3.1.2.2.11 LeerConfig

Procedimiento que lee el archivo asignado al terminal y lo guarda en la variable Parámetros.

4.3.1.2.2.12 HayCVCS

Procedimiento que decide si hay canal para progresar una llamada originada en el terminal.

4.3.1.2.2.13 DecideCVCS

Procedimiento que decide el canal disponible por el que se progresa la llamada del terminal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.3.1.2.2.14 AceptaCVCE

Procedimiento que analiza si la configuración del canal por el que se recibe la llamada permite aceptar esta.

4.3.1.2.2.15 HayNPCF

Función que estudia si el paquete de llamada tiene negociación de parámetros de control de flujo y extrae los valores que en él se proponen.

4.3.1.2.2.16 PonVentanaRx

Procedimiento que añade el valor indicxado de ventana de recepción en un paquete de llamada que va a ser enviado con negociación de parámetros de control de flujo.

4.3.1.2.2.17 PonVentanaTx

Lo mismo que el anterior pero para la ventana de transmisión.

4.3.1.2.2.18 AceptaFacilidades.

Función que compara las facilidades solicitadas en el paquete de llamada con las que le otorga al ETD su configuración.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4 CCTERM2. Nivel 2 del terminal, Nodo y Ruta

Esta unidad es la encargada de crear y mantener el nivel de enlace entre dos estaciones. Estas estaciones pueden ser el terminal con el nodo o un nodo con otro nodo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1 Objetos

En esta unidad sólo se crea el objeto TCCN2 encargado de establecer y mantener el nivel de enlace del protocolo X25.

4.4.1.1 Variables principales.

Dentro del objeto se definen las siguientes variables

4.4.1.1.1 Nombre

Una vez que se crea cada uno de los objetos TCCN2 en cada uno de los interfaces, se le da a cada uno de ellos un nombre que identifica al terminal o nodo al que pertenece. De esta manera los mensajes que envía cada uno de los objetos al objeto Tpanel pueden ser identificados al incluir el nombre del objeto que lo envía en el mensaje.

4.4.1.1.2 Estado

En la norma X25 se establece una serie de estados del interfaz. La menera de llegar a cada uno de estos estados y salir de ellos está considerado en el procedimiento X25 y esta variable indica en que estado se encuentra el interfaz en cada momento.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.1.3 DirLocal, DirRemota, DirActual

Variables encargadas de almacenar la dirección local y la remota. La dirección actual es una variable en la que se almacena la dirección del paquete comando de nivel 2 para implementar una trama respuesta con la misma dirección.

4.4.1.1.4 T1Activo, T1Inicial, T1

Variables encargadas de controlar el temporizador T1

4.4.1.1.5 T3Activo, T3Inicial, T3

Variables encargadas de controlar el temporizador T3 de la norma X25 en el nivel de Enlace

4.4.1.1.6 N2, ContN2

Variables encargadas de implementar el contador N2 propio del protocolo X25 en su nivel de enlace.

4.4.1.1.7 Nr, P, Ns

Variables utilizadas para implementar, en las tramas, la secuencia de control de ventana y el bit P/F.

4.4.1.1.8 Vt, Vr, VentN2Tx, VentN2Rx

Variables que controlan la ventana de transmisión y de recepción. Las variables Vt y Vr almacenan el estado actual de las ventanas de recepción y de transmisión mientras que las variables VentN2Tx y VentN2Rx nombran el valor nominal que tiene definida la ventana por configuración.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.1.9 TramasRx, TramsTx

Varibles donde se almacenan las cadenas que forman la trama. En la variable TramasRx se guardan las tramas recibidas de la etapa física mientras que en TramasTx se guardan las cadenas que estan a la espera de ser transmitidas hacia la etapa física.

4.4.1.1.10 PaqueteTx,PaqueteRx.

De igual manera que las tramas estas variables almacenan los paquetes recibidos y a transmitir hacia el nivel tres.

4.4.1.1.11 HayTramasTx, HayTramasRx

Variables booleanas encargadas de indicar si hay alguna trama en recepción o en transmisión.

4.4.1.1.12 HayPaqueteTx, HayPaqueteRx

Variables booleanas encargadas de señalar si hay algún paquete pendiente de ser transmitido o si se ha recibido un paquete del nivel superior.

4.4.1.1.13 N2Activo

Variable booleana que se activa cuando el nivel dos está en el estado de intercambio de tramas Info. En el objeto TCCT existe un puntero que señala esta variable de manera que se tiene información inmediata en el nivel de Red que el nivel de Enlace está activo.

4.4.1.1.14 Parámetros, ArchCfg.

En la unidad CFGNodo se define el tipo CFG como un registro. En este registro se guardan todos los parámetros de configuración. La variable

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Parámetros es del tipo CFG y la variable ArchCfg es un archivo de registros tipo CFG. Es de este archivo del que este objeto leerá su configuración.

4.4.1.2 Procedimientos principales

4.4.1.2.1 Create.

Procedimiento al que se llama cuando se crea el objeto. Inicia la variable estado como 'E1' con todos los temporizadores y contadores desactivados. La variable N2Activo se inicia como falsa.

4.4.1.2.2 Txtramas, RxTramas.

La función TxTrama y el procedimiento RxTrama se encargan de comunicarse con la etapa física y, a través de ella, con el procedimiento de nivel dos enfrentado a éste.

4.4.1.2.3 TxPaquete, RxPaquete.

La función TxPaquete y el procedimiento RxPaquete se comunican con el nivel 3 que comanda el objeto TCCN2.

4.4.1.2.4 DisparaT1.

Procedimiento que inicia la cuenta del temporizador T1. Realmente lo que hace este procedimiento es tomar la hora actual y almacenarla en segundos.

4.4.1.2.5 FinalT1

Cuando el programa llama a la función FinalT1, ésta toma la hora del sistema, la pasa a segundos y le resta a ésta la obtenida por DisparaT1. Si la resta

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

en segundos es mayor que el valor de T1, la función FinalT1 es cierta, si el valor es menor la función es falsa.

4.4.1.2.6 DisparaT3 y FinalT3

Actúan de igual manera que la función FinalT1 y el procedimiento DisparaT1

4.4.1.2.7 SABM

Función que entrega como valor una cadena de caracteres que constituye una trama SABM. Como parámetro se le indica la dirección y el bit P/F

4.4.1.2.8 UA

Función que entrega como valor una cadena de caracteres que constituye una trama UA. Como parámetro se le indica la dirección y el bit P/F

4.4.1.2.9 DISC

Función que entrega como valor una cadena de caracteres que constituye una trama DSC. Como parámetro se le indica la dirección y el bit P/F

4.4.1.2.10 DM

Función que entrega como valor una cadena de caracteres que constituye una trama DM. Como parámetro se le indica la dirección y el bit P/F

4.4.1.2.11 FRMR

Función que entrega como valor una cadena de caracteres que constituye una trama FRMR .Como parámetro se le indica la dirección y el bit P/F

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.12 RR

Función que entrega como valor una cadena de caracteres que constituye una trama RR.Como parámetro se le indica la dirección, el bit P/F y el valor de Nr

4.4.1.2.13 Info

Función que entrega como valor una cadena de caracteres que constituye una trama Info. Como parámetro se le indica la dirección, el bit P/F, los valores de Nr y Ns y el paquete de nivel tres que va a transportar.

4.4.1.2.14 TipoTrama

Función que entrega una cadena en la que se indica el nombre del tipo de la trama por la que se pregunta.

4.4.1.2.15 BitP

Dada una trama la función indica si el bit P/F está activo o no.

4.4.1.2.16 ExtraePaq

Dada una trama del tipo Info esta función entrega el paquete de nivel tres contenida en ella.

4.4.1.2.17 Rearranque

Función que entrega un paquete de rearranque.

4.4.1.2.18 LeerConfig

Procedimiento que se encarga de abrir el archivo, cuyo nombre se le entrega como parámetro y asignar el valor del registro leido a la variable parámetro.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19 Control

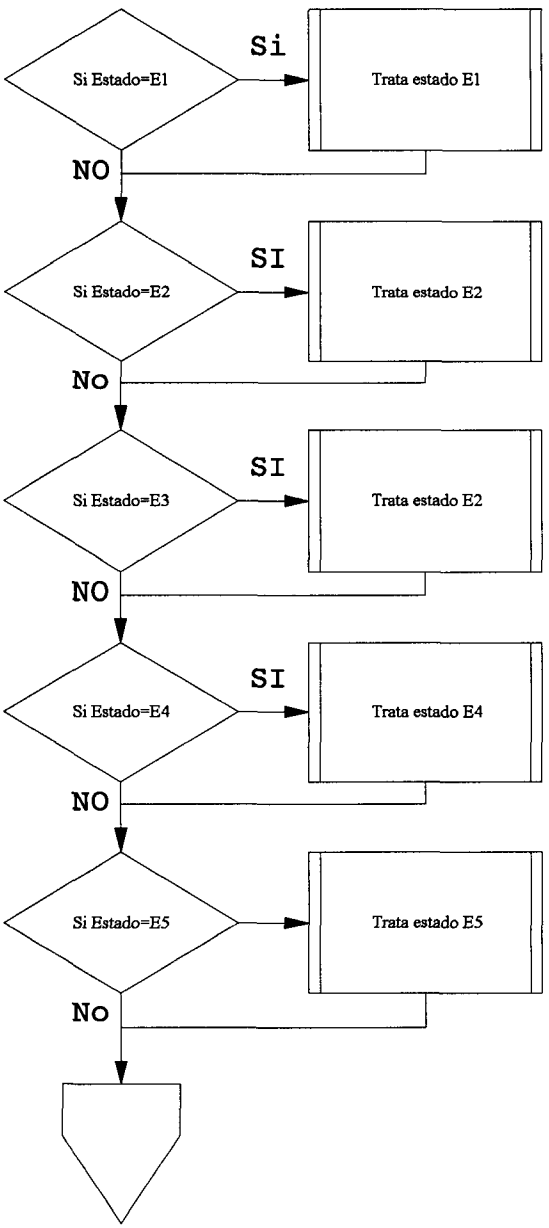


Ilustración 7. Procedimiento control.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19.1 E1 Desconexión

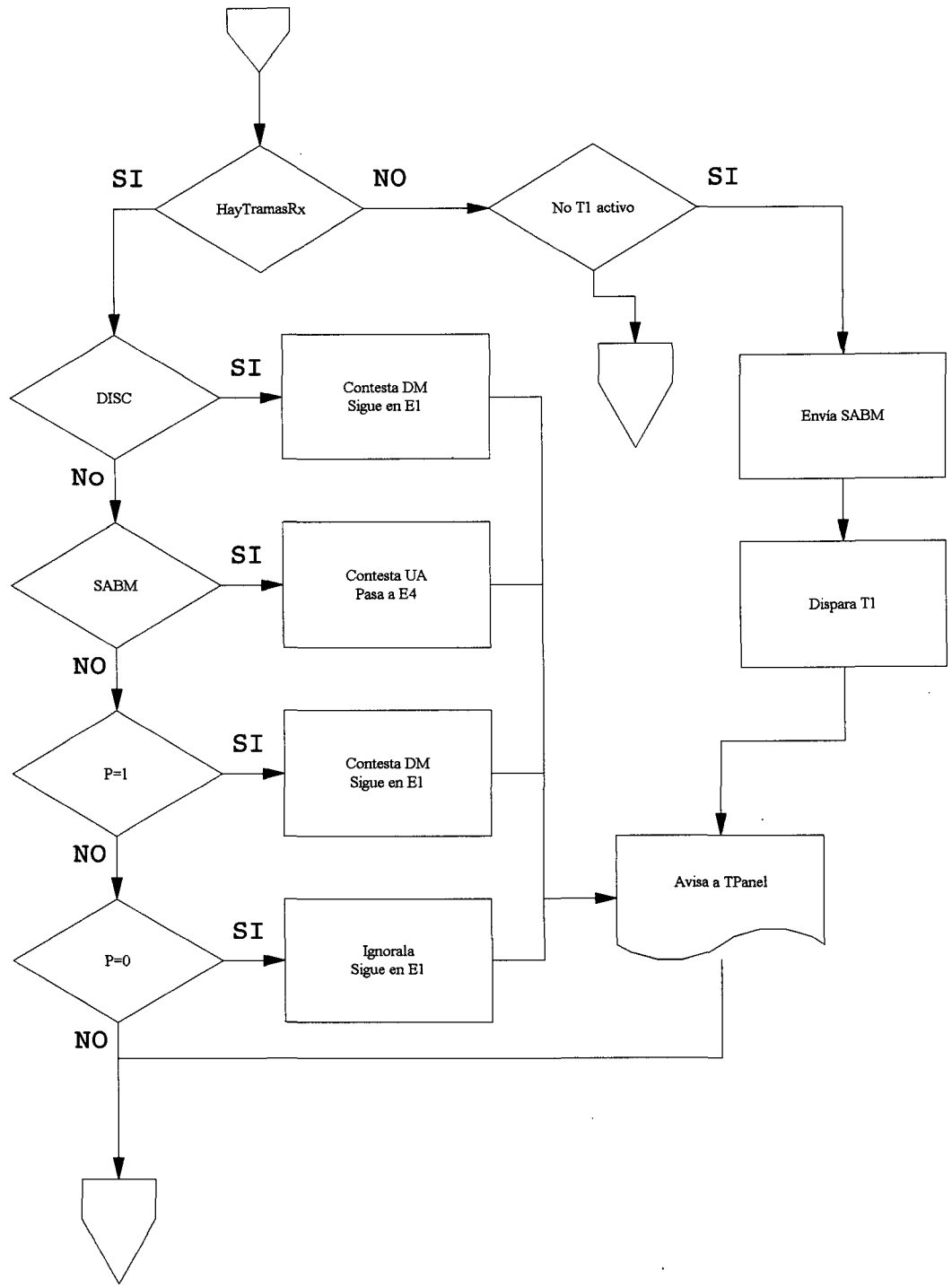


Ilustración 8.Estado E1

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19.1.1 Si hay tramas en recepción

4.4.1.2.19.1.1.1 Si recibe un DISC

Si se recibe una trama DISC de la estación enfrentada a esta y no hay tramas pendientes de ser transmitidas, se sigue en estado E1 y se envía una trama DM con el bit F igual al bit P de la trama recibida. Por último se envía informe de lo acontecido a el Panel de estados. Esto lo hará en cada una de las recepciones de cualquier tipo de trama.

4.4.1.2.19.1.1.2 Si recibe SABM

Si se recibe un SABM y está libre el canal de transmisión, se pasa a estado E4, la variable N2Activo se hace cierta y se transmite un UA.

4.4.1.2.19.1.1.3 Si recibe trama con P=1

Si se recibe cualquier otra trama que no sean las anteriores, sigue en Estado E1 y contesta con un DM con el bit F=1.

4.4.1.2.19.1.1.4 Si recibe trama con P=0

Si se recibe cualquier otra trama con el bit P=0, se ignora y se sigue en estado E1.

4.4.1.2.19.1.2 Si no hay nada en recepción.

Si no hay nada en recepción, se mira si no está activo el temporizador T1. Si T1 no está activo, se envía un SABM con P=0, se dispara T1 y se pasa a estado E2 a la vez que se avisa al Panel de Estados.(Tpanel)

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19.2 E2 Indicación de iniciación

4.4.1.2.19.2.1 **Si hay tramas en recepción**

4.4.1.2.19.2.1.1 Si recibe DISC

Si recibe DISC, contesta con DM y pasa a estado E1, inicia a cero el contador ContN2. y avisa al TPanel.

4.4.1.2.19.2.1.2 Si recibe DM

Si recibe un DM pasa a estado E1 sin contestar nada.

4.4.1.2.19.2.1.3 Si recibe SABM

Contesta UA y sigue en estado E2 .

4.4.1.2.19.2.1.4 Si recibe UA

Si se recibe un UA en estado E2 se pasa a estado E4, se activa la variable N2Activo e inicia a cero el contador ContN2. Si no hay tramas que transmitir y la dirección local es la de la red, se envía un rearranque de nivel tres por Red en operación como indicación de que el nivel tres puede ser iniciado. A continuación se inician las variables de control de ventana. y se avisa al TPanel.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19.2.2 Si no hay nada en recepción

4.4.1.2.19.2.2.1 Si T3 no está activo

Si ha terminado T1, envía un SABM e incrementa ContN2 e informa al Tpanel. Si el contador ha llegado a N2, dispara T3 y hace ContN2=0. Si ContN2 no ha llegado a N2 dispara T1

4.4.1.2.19.2.2.2 Si T3 está activo.

Si está activo y ha finalizado, envía un SABM, avisa al Tpanel, dispara T1 e incrementa ConTN2.

4.4.1.2.19.3 E3 Indicación de desconexión

4.4.1.2.19.3.1 Si hay tramas en recepción

4.4.1.2.19.3.1.1 Si recibe SABM

Si se recibe SABM contesta con DM y sigue en estado E3.

4.4.1.2.19.3.1.2 Si recibe un DISC

Si se recibe un DISC contesta con DM y pasa a estado E1

4.4.1.2.19.3.1.3 Si recibe un UA

Si se recibe un UA, no contesta nada y pasa a estado E1.

4.4.1.2.19.3.1.4 Si recibe un DM

Si recibe un DM no contesta nada y pasa a estado E1.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19.3.2 Si no hay tramas en recepoción

4.4.1.2.19.3.2.1 Si T1 activo

Si T1 ha concluido mira si el contador ContN2 ha sobrepasado N2. Si es así, pasa a estado E1. Si no ha sobrepasado N2, contesta con DM, dispara T1 e incrementa ContN2.

4.4.1.2.19.4 E4 Transferencia de información

4.4.1.2.19.4.1 Si hay tramas en recepción

4.4.1.2.19.4.1.1 Si recibe un SABM

Si recibe una trama SABM en estado E4, contesta UA y sigue en este estado.

4.4.1.2.19.4.1.2 Si recibe DM

Si recibe un DM pasa a estado E1 sin contestar nada.

4.4.1.2.19.4.1.3 Si recibe FRMR

Si recibe un FRMR contesta SABM y pasa a estado E2.

4.4.1.2.19.4.1.4 Si recibe un INFO

Primero extrae el paquete del Info recibido y lo pasa al nivel superior.Toma la dirección de la trama recibida y la almacena en la variable DirActual. A continuación informa al Tpanel. Si T1 está activo lo desactiva, actualiza los contadores Nr y Vr y pone a cero el contador de ventana de transmisión Vt.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Si hay un paquete en recepción enviado por el nivel superior primero mira si el bit P de la trama recibida está activo. Si lo está, mira si está dentro de la ventana de transmisión y si es así mira si la última trama recibida era comando o respuesta mirando la dirección de ésta y comparándola con la variable DirRemota. Si la última trama recibida tenía dirección remota, envía un info con esta dirección, con P/F=1 y el paquete recibido del nivel superior.

Si la dirección de la trama recibida es la local, contesta con un info conteniendo el paquete recibido, con P/F=1 y la dirección local. A continuación actualiza la secuencia de ventana de transmisión Ns, pone Vr igual a cero e incrementa Vt.

Si el bit P no está activo en la trama recibida y hay un paquete pendiente de enviar, repite el procedimiento anterior, pero enviando las tramas con el bit P/F igual a cero.

Si no hay paquetes pendientes de mandar y la trama recibida tiene el bit P activo, responde con un RR con el bit F activo, actualiza DirActual como DirLocal y pone la ventana de recepción Vr igual a cero.

Si no hay paquetes pendientes de mandar y la trama recibida no tiene el bit P activo, se mira si la trama recibida ha cerrado la ventana. Si no la ha cerrado, no contesta nada, pero si la cierra se transmite un RR con dirección remota y bit F a cero. A continuación se pone a cero la variable Vr.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

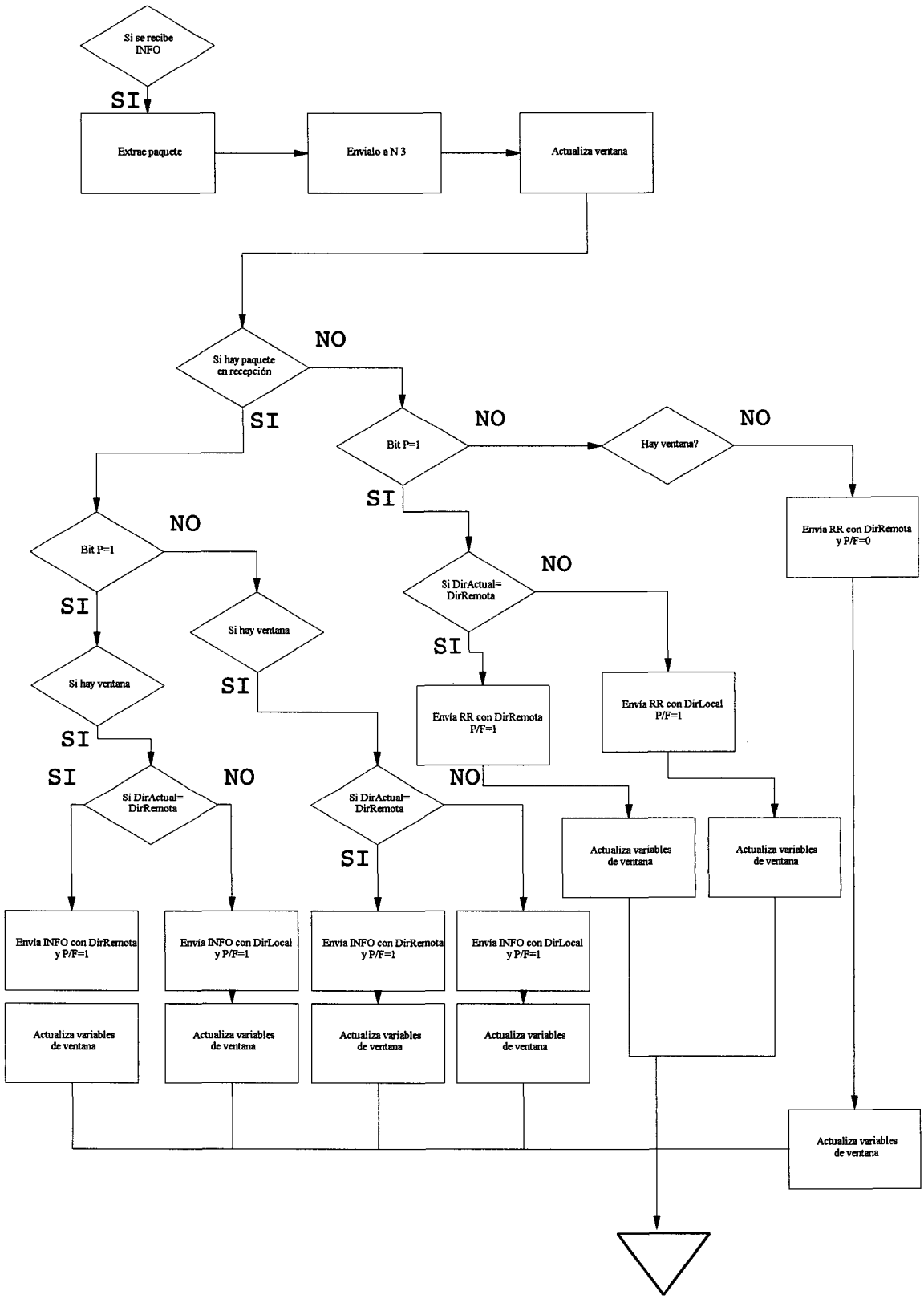


Ilustración 9.Recepción de INFO.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19.4.1.5 Si recibe RR

Si está activo T1 se desactiva y pone el contador de ventana de transmisión a cero. Si hay paquetes que enviar mira primero si hay ventana, si la hay mira la dirección con la que llegó el RR. Si es DirRemota envía un Info con el paquete a enviar, el bit P a cero y dirección remota. Si la dirección es la local, lo envía con la dirección local. A continuación actualiza la secuencia Ns, hace Vr igual a cero e incrementa Vt.

Si no hay paquetes que enviar y el RR recibido es un comando (tiene DirRemota), envía un RR y hace Vr igual a cero. Si el RR recibido es una respuesta no responde nada.

4.4.1.2.19.4.2 Si no hay tramas en recepción

4.4.1.2.19.4.2.1 Si hay paquete en recepción

Si hay ventana envía un Info con el paquete. Actualiza la variable DirActual con el valor de DirLocal, actualiza la secuencia de ventana Ns, incrementa Vt y da valor cero a la variable Vr

4.4.1.2.19.4.2.2 Si no hay paquete en recepción

Si ha finalizado T3 y la dirección local coincide con la de la Red, manda un RR cada T3 segundos.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.4.1.2.19.5 E5 Indicación de rechazo.

4.4.1.2.19.5.1 Si recibe trama

4.4.1.2.19.5.1.1 Si recibe SABM

Contesta UA y pasa al estado E4

4.4.1.2.19.5.1.2 Si recibe DISC

Si recibe DISC en estado E5 contesta DM y pasa a estado E1

4.4.1.2.19.5.1.3 Si recibe FRMR

Pasa a estado E2 y contesta con un SABM

4.4.1.2.19.5.1.4 Si recibe DM

Si recibe un DM no contesta nada y pasa a estado E1. En todos los casos de recepción de trama se avisa al Tpanel de lo sucedido.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.5 ETFI. Etapa Física

4.5.1 Objeto TEF.

Es el objeto encargado de unir los niveles de enlace de dos estaciones. Simula la union física entre la Red y el terminal. En la realidad el terminal o ETD estaría unido a un ETCD a través del interfaz V21bis (V24/V28), el ETCD o modem se uniría a una línea dedicada, que terminaría en el modem de la central y éste se conectaría al nodo o conmutador de la Red. Todo este tramo entre los dos interfaces es el representado por el objeto EtFi.

4.5.1.1 Variables principales

4.5.1.1.1 TramasNT

Variable de tipo string (Cadena de caracteres), que se utiliza para almacenar las tramas recibidas del nodo, que serán enviadas al terminal.

4.5.1.1.2 TramasTN

Variable de tipo string (Cadena de caracteres) que se utiliza para almacenar las tramas recibidas del terminal, que serán enviadas hacia el Nodo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.5.1.1.3 IntNT.

Variable de tipo booleana que se activa cuando hay una trama almacenada en la Etapa Física que está pendiente de ser enviada al terminal.

4.5.1.1.4 IntTN

Variable de tipo booleana que se activa cuando hay una trama almacenada en la Etapa Física que está pendiente de ser enviada al Nodo.

4.5.1.2 Procedimientos y funciones principales.

4.5.1.2.1 Función TxEFT

Función encargada de transmitir la trama recibida del Nodo al terminal. También desactiva la variable IntNT.

4.5.1.2.2 Procedimiento RxTEF (Trama :String)

Procedimiento encargado de recibir la trama enviada por el terminal y almacenarla en la variable TramasTN. También activa la variable IntTN.

4.5.1.2.3 Función TxEFN

Procedimiento encargado de transmitir la trama recibida del terminal al nodo. También desactiva la bandera IntTN.

4.5.1.2.4 Procedimiento RxNEF(Trama: String)

Procedimiento encargado de recibir la trama enviada por el Nodo y almacenarla en la variable TramaNT. También activa la variable IntNT

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

El objeto TEF (Etapa física) es totalmente pasivo. Se limita a ser utilizado por otras unidades para la transferencia entre las dos estaciones responsables del protocolo. No modifica para nada el contenido de las tramas.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.6 ANALIZAD. Analizadores.

El objeto Tanalizador que se crea en esta unidad pretende asemejarse a un analizador de protocolo conectado entre el ETD y el DCE. Mostrará los distintos niveles del protocolo X25 para seguir de forma directa su comportamiento. Añadirá algunas funciones que permitan facilitar algunas tareas de análisis.

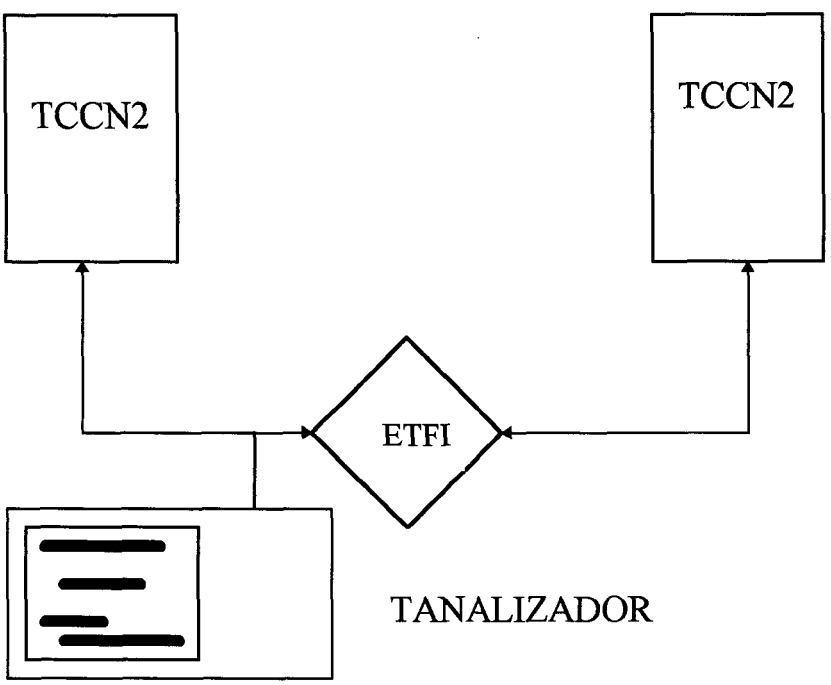


Ilustración 10.Conexión lógica del objeto Tanalizador.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.6.1 ObjetoTAnalizador

Objeto descendiente del Objeto Tform. Este objeto, perteneciente a la librería de Delphi implementa una ventana que se abre al crear el objeto. Dentro de la ventana se implementarán un menú, unas lengüetas y varios editores.

4.6.1.1 Objetos Principales

4.6.1.1.1 Tmemo

El objeto Tmemo se encarga de crear un editor de texto. En el programa se han creado los objetos Memo1, Memo2, Memo3 y Memo4, herederos del objeto Tmemo, encargados de presentar el nivel Físico, el nivel de Enlace, el nivel de Red y una muestra conjunta de todos los niveles.

4.6.1.1.2 TabSet

El objeto TabSet crea las lengüetas que permiten elegir el editor a visualizar. Existe una lengüeta por editor y el evento que se dispara una vez seleccionada una de ellas con el ratón, permite visualizar un editor determinado y ocultar el resto.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.6.1.1.3 Menú

El objeto Tmenú es el encargado de crear un menú desplegable donde se encuentran todas las opciones de este objeto. Cada una de las opciones implementadas en el menú tendrá su propio procedimiento implementado que será llamado por el manejador de eventos cuando se seleccione, bien desde el Ratón o bien desde el teclado.

4.6.1.2 Procedimientos Principales.

4.6.1.2.1 FormCreate

Procedimiento de iniciación. Pone todas las variables en el estado en el que deben estar inicialmente. Limpia los objetos Tmemos y les da la característica de ser de solo lectura, para que el usuario no pueda escribir dentro de estos editores.

4.6.1.2.2 11.1.2.2FormClose

Procedimiento que se activa antes de cerrar la ventana del Analizador por indicación del usuario. Se encarga de activar de nuevo los botones y menús que permiten activar de nuevo el analizador.

4.6.1.2.3 Captura

Es el procedimiento que comunica al objeto Tanalizador con el resto del programa. Este procedimiento es llamado desde el procedimiento Idle en la unidad principal SimX25. Cuando hay una trama que intercambiar, bien en transmisión o en recepción, entre el objeto TCCN2 del terminal y el objeto ETFI; se llama al procedimiento Captura del Tanalizador. Se le pasa como parámetro la trama transmitida o recibida, con una cadena previa que indica si ha sido el DTE o el DCE el origen de la trama. Al llegar una trama al procedimiento captura primero mira si el origen de la trama es el DTE o el DCE para comparar el sentido con los filtros establecidos. Si uno de los filtros estuviera activo sólo se mostrarían las tramas con ese sentido. A continuación se mira si no ha sido

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

activado la opción 'Parar' del menú. Si ha sido activada no se muestra nada de lo acontecido en línea. Si se cumplen las condiciones anteriores se comienza a distribuir la información recibida en los editores correspondientes a los distintos niveles:

1. En el nivel Físico se presenta la trama tal como llega al procedimiento captura. El procedimiento encargado de llevar la trama a su editor es el InsLinea1.

2. Para presentar la trama de nivel de Enlace se llama a la función PresentaN2. Esta función entrega como resultado una cadena de caracteres que ha de presentarse en el editor de nivel de Enlace. Si la función no devuelve una cadena vacía se llama al procedimiento Inslinea2 para que presente la cadena formada por PresentaN2 en el editor correspondiente.

3. La función PresentaN3 cumple el mismo cometido para el nivel de Red que PresentaN2 para el nivel de Enlace. Si entrega una cadena no nula es que la trama era una trama Info y ésta portaba en su interior un paquete de nivel tres. Si es así, la muestra utilizando el procedimiento InsLinea3 en el editor asignado.

4. A continuación se presentan todos los niveles en el editor Memo4 mediante el procedimiento InsLinea4. Para esto se mira primero si la trama es a nivel de enlace un Info. Si es así, se muestra la trama tal como llega en la primera línea, en el formato de nivel de Enlace en la segunda línea y en el formato de nivel de Red en la tercera. Si la trama no fuera un Info de nivel dos, sólo se presentarían las dos primeras líneas antes mencionadas.

Por último se estudia si se ha disparado algún evento asociado a las trampas mediante el procedimiento EventoTrampa. Si es así, el procedimiento captura responde con el suceso programado en la ventana de trampas.

4.6.1.2.4 PresentaN2

Este procedimiento entrega una cadena de caracteres que muestra los distintos datos de control que forman la trama de nivel dos. Muestra, por ejemplo, la dirección, el tipo de paquete el bit P/F, y los contadores Ns y Nr. Se le pasa a la función, como parámetro, la trama recibida por el TCCN2 o enviada por éste. La función va estudiando si el octeto de control de la trama corresponde

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

con un tipo determinado de paquete y, si es así, incluye el nombre de este paquete en la cadena de caracteres a presentar. También considera si la variable de filtro coreerespondiente a este tipo de paquete permite mostrarlo o no. Al considerar el tipo de paquete considera también si el bit P/F esta activo o no. Una vez hallado el tipo de paquete se termina de implementar la cadena de nivel 2 del analizador correspondiente a esta trama con la dirección y las secuencias Nr y Ns si procediera.

4.6.1.2.5 PresentaN3

Función a la que se entrega una cadena de caracteres que representa la trama y que entrega una cadena de caracteres que representan los datos del paquete de nivel de Red. Para ello estudia primeramente que la cadena contiene una trama de información de nivel de Enlace. Si es así continua el análisis, si no muestra una cadena nula.

Inicialmente implementa en el comienzo de la línea el sentido que lleva el paquete, a continuación presenta el bit Q y el bit D del paquete y prosigue con el modulo. Después mira el canal por el que llega el paquete. Por último recorre todos los tipos de paquetes de nivel tres posibles comparándolos con el octeto de control de nivel tres. Si localiza el tipo de paquete que es, mira también si el filtro para este tipo de paquete permite que este sea mostrado o no. Si el filtro está activo, añade a la cadena el nombre del paquete recibido. Si fuera un paquete de datos de usuario RR o RNR también añadiría a la cadena a presentar los valores de secuencia de recepción y de transmisión si procediera. El bit M también es mostrado entre los dos parámetros anteriores.

4.6.1.2.6 EventoTrampa

Función que estudia la trama recibida por el analizador y la compara con los datos introducidos en el menú de trampas. Si se produce el evento especificado en la trampa, la función retorna valor cierto, en caso contrario entrega valor falso.

Primero comprueba que en los valores de condición introducidos hay al menos uno que no posea el valor 'Indistinto'. Esto es, que se le pide una

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

condición a la trama, si todos los valores son indistintos 'XX' no se le está pidiendo una condición a la trama. A continuación se van comprobando cada una de las variables de condición que contiene el registro Vtrampas. Cada una de ellas se compara con la trama y activa o no una bandera asociada a la condición. Si la condición fuera indistinta, asumiría la bandera como cierta. Sólo sería falsa la bandera si la condición no se cumpliera, siendo la condición no indistinta. Si todas estas banderas son ciertas se da como sierta la función Eventotrampa.

4.6.1.2.7 ComparaMascara

Una de las opciones que se ofrecen en la ventana de trampas es la de crear una condición que no afecte a todo un octeto de la trama. Asi se podría poner como condición que el bit de menor peso disparara una trampa cuando tomara valor cero. La comparación entre la máscara creada en la ventana de introducción de condiciones para la trampa y el octeto recibido en la trama, lo realiza la función ComparaMascara. Se le pasa como parámetros la máscara y el octeto con el que se compara y entrega valor cierto si la comparación es válida y falso si no lo es. Para la comparación se pasa el octeto a una representación binaria sobre una cadena de caracteres y se comparan los bit uno a uno con la máscara de bit. Si el bit de la máscara se representa por 'X' la comparación se da como cierta.

4.6.1.2.8 InsLinea

Procedimiento de controlar la inserción de líneas en los editores. Se encargan de cambiar los caracteres de retorno de carro y fin de linea por $ y @ respectivamente y de controlar el número de líneas en el editor. Para limitar el número de líneas que pueden presentar los editores se ha utilizado la variable Maxlineas. En esta variable se indica el máximo de lineas que se pueden mostrar. Al llegar a este número, el procedimiento InsLinea sobreescribe la primera e indica la posición donde va a escribir la siguiente que llegue con los caracteres '>>' y dejando la linea en blanco. Llegada una nueva línea se mostraría sobre la segunda línea del editor, donde antes se mostraban los caracteres '>>' y estos

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

caracteres pasan a mostrarse en la tercera línea. Cuando de nuevo se llega al valor MaxLineas se comienza de nuevo en la primera línea.

4.6.1.2.9 Para1Click

Procedimiento enlazado con la opción Parar del Menú. Cuando se activa, éste queda una bandera que utiliza el procedimiento captura como condición para mostrar las tramas o no.

4.6.1.2.10 CambioClick

Procedimiento al que se llama cuando se pulsa con el ratón sobre alguna lengüeta. Se encarga de mostrar el editor del nivel seleccionado. En este procedimiento se eliminan las etiquetas mostradas sobre el panel superior del editor y se oculta el editor anterior. A continuación se muestran las nuevas etiquetas correspondientes a las partes del paquete o a la trama y se muestra el editor del nivel seleccionado a la vez que se destaca la nueva lengüeta en uso.

4.6.1.2.11 Fisico1Click

Procedimiento activado desde la opción Nivel del menú del analizador que realiza la misma función que si seleccionaramos la lengüeta correspondiente al nivel físico.

4.6.1.2.12 Enlace1Click

Procedimiento activado desde la opción Nivel del menú del analizador que realiza la misma función que si seleccionaramos la lengüeta correspondiente al nivel de Enlace.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.6.1.2.13 Red1Click

Procedimiento activado desde la opción Nivel del menú del analizador que realiza la misma función que si seleccionaramos la lengüeta correspondiente al nivel de Red.

4.6.1.2.14 Todo1Click

Procedimiento activado desde la opción Nivel del menú del analizador que realiza la misma función que si seleccionaramos la lengüeta correspondiente a todos los niveles.

4.6.1.2.15 NivelFísicoClick

Procedimiento asociado a la opción de menú Archivos/Guardar/Nivel Físico. Presenta una ventana de diálogo en la que se elige el nombre del archivo de texto en el que se va a guardar el contenido del editor del nivel físico. Se muestra como extensión del archivo la cadena 'fis' para ditinguirlo posteriormente del resto. Asume por defecto el subdirectorio entregado por la unidad simX25 para el Analizador.

4.6.1.2.16 NivelEnlace1Click

Procedimiento asociado a la opción de menú Archivos/Guardar/Nivel de Enlace. Presenta una ventana de diálogo en la que se elige el nombre del archivo de texto en el que se va a guardar el contenido del editor del nivel de Enlace. Se muestra como extensión del archivo la cadena 'enl' para ditinguirlo posteriormente del resto. Asume por defecto el subdirectorio entregado por la unidad simX25 para el Analizador.

4.6.1.2.17 NivelRed1Click

Procedimiento asociado a la opción de menú Archivos/Guardar/Nivel de Red. Presenta una ventana de diálogo en la que se elige el nombre del archivo de texto en el que se va a guardar el contenido del editor del nivel de Red. Se muestra como extensión del archivo la cadena 'red' para ditinguirlo

© Del documento, los autores. Digitalización realizada por ULPGC. Bibliotecas Universitaria, 2006

posteriormente del resto. Asume por defecto el subdirectorio entregado por la unidad simX25 para el Analizador.

4.6.1.2.18 TodosLosNiveles1Click

Procedimiento asociado a la opción de menú Archivos/Guardar/Todos los Niveles. Presenta una ventana de diálogo en la que se elige el nombre del archivo de texto en el que se va a guardar el contenido del editor de todos los niveles. Se muestra como extensión del archivo la cadena 'tod' para ditinguirlo posteriormente del resto. Asume por defecto el subdirectorio entregado por la unidad simX25 para el Analizador.

4.6.1.2.19 Capturar1Click

Procedimiento que permite activar el envío de todas las cadenas que se van a presentar en el editor de texto de Todos los Niveles a un archivo de texto. Para ello se presenta una ventana de elección de nombre de archivo en el mismo subdirectorio que los anteriores con la extensión 'Tod'. Posteriormente se abre el archivo y se activa una bandera que es estudiada por el procedimiento InsLinea4. Este es el procedimiento que inserta las líneas en el editor de 'Todos los Niveles'. Si la bandera de captura está activa también se guarda la cadena mostrada en el archivo abierto.

Cuando se vuelve a seleccionar la opción de menú Archivos\Capturar, se desactiva la bandera y se cierra el archivo en uso.

4.6.1.2.20 LimpiarAnalizador

Esta opción envía una orden a todos los editores para que limpien su contenido.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.6.1.2.21 Trampas1Click

Opción de menú que llama a la ventana de diálogo que permite cambiar las opciones de las trampas. Esta ventana está contenida en la unidad Tram. Al aceptar los valores de la ventana se actualiza la variable Registro Vtrampa que es la que se utiliza en la funcion EventoTrampa.

4.6.1.2.22 Procedimiento N21Click

Dentro del objeto Tanalizador se definen una serie de variables booleanas. Cada una de ellas corresponde a una condición de filtrado en la presentación de datos. Si suponemos como activa la variable FDTE se presentarán las tramas y paquetes que tengan como origen el DTE. Así las siguientes variables determinan que se permite aparecer en pantalla y que no.

1. FDTE.Filtro DTE.
2. FDCE.Filtro de DCE.
3. FSABM.Filtro de SABM.
4. FUA.Filtro de UA.
5. FDISC.Filtro de DISC.
6. FDM.Filtro de DM.
7. FRR.Filtro de RR.
8. FRNR.Filtro de RNR.
9. FINFO.Filtro de INFO.
10. FFRMR.Filtro de FRMR.
11. FSLLAM.Filtro de Solicitud de llamada.
12. FALLAM.Filtro de Aceptación de llamada.
13. FSLIB.Filtro de solicitud de liberación.
14. FALIB.Filtro de aceptación de liberación.
15. FSREIN.Filtro de solicitud de reinicio.
16. FAREIN.Filtro de aceptación de reinicio.
17. FSINTE.Filtro de solicitud de interupción.
18. FAINTE.Filtro de aceptación de interrupcion.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

19.FSREAR.Filtro de solicitud de rearranque.

20.FAREAR.Filtro de aceptación de rearranque.

21.FRRN3Filtro de RR de nivel de Red.

22.FRNRN3.Filtro de RNR de nivel de Red.

23.FDATA.Filtro de paquete de Datos.

Todas estas banderas se inician en activo, permitiendo, por tanto , al comienzo del programa mostrar todos los tipos de paquetes y tramas en los dos sentidos de la comunicación. En el procedimiento que describimos se hace una llamada al objeto Filtros que implementa una ventana de diálogo en la cual se puede modificar los valores de estas variables. Este objeto visual Tfiltros se encuentra implementado en la unidad 'ANFIL.PAS'.

Las banderas de filtro van a ser consultadas en los procedimientos PresentaN2 y PresentaN3 así como en el procedimiento Captura para estudiar el sentido.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.6.2 ANFIL. Pantalla de opciones del filtro.

Unidad encargada de implementar el objeto visual Tfiltros. Este objeto será utilizado por los Analizadores. Implementa una ventana de diálogo que permite seleccionar las tramas ,paquetes y sentido que se quiere que sea o no presentado. El único procedimiento que implementa esta unidad es el ChINFOClick. Este procedimiento se limita a indicar a las variables de filtro que no muestren nada de nivel de Red si el filtro de tramas de información de nivel dos no permite que estas se muestren. Si se permite mostrar las tramas de Información de nivel de Enlace, automaticamente se permitirá que se muestren todos los paquetes de nivel de Red.

Este objeto Filtro no se crea ni se destruye durante la ejecución del programa. sino que permanece en memoria desde el comienzo de la ejecución del programa hasta su finalización.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.6.3 TRAM. Pantalla de opciones de las trampas.

Unidad que implementa el objeto Trampas. Este objeto visual es el encargado de modificar los valores de la variable registro Vtrampas del objeto Analizador.

Esta variable almacena los valores que deberá cumplir la trama recibida para que se produzca el evento también indicado en esta variable.

Los únicos procedimientos que se implementan en la definición del objeto Trampa estudian que no se produzca una incongruencia en la entrada de datos. Así sería incongruente para el funcionamiento del programa que solicitaramos como eventos a ejecutar en el disparo de una trampa la apertura de un archivo de captura y a la vez el cierre del archivo de captura.

Este objeto se llama desde el procedimiento Trampas1Click del objeto Analizador. Se utilizan las variables por él modificadas en los procedimientos Captura y EventoTrampa así como en ComparaMascara.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.7 OTROS

4.7.1 CALCULO.

Unidad de funciones de conversión

4.7.2 Funciones

4.7.2.1 Función DH (D: Byte): String.

Función encargada de la conversion de números en base digital a cadenas de texto que representan el mismo número en base hexadecimal.

4.7.2.2 Función HD(H: String): Byte.

Función encargada de tomar una cadena de texto que representa un número en notación hexadecimal y convertirlo en una variabe numérica en base decimal.

4.7.2.3 Función PrMPsH(A, B, C: Byte):String.

Convierte las variables A, B, C en dos caracteres de una cadena de texto que representan en base hexadecimal el octeto de control de un paquete de datos de nivel 3. A es el valor de Pr, B el valor del bit M y C el valor de la variable Ps.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.7.2.4 Función PrHD (H: String): Byte.

Extrae el valor de Pr de la cadena de dos caracteres que representa en hexadecimal el octeto de controlde un paquetede datos de nivel 3.

4.7.2.5 Función PrRRDH (A, B, C: Byte) String.

Función que genera una cadena de texto representando a un número en base hexadecimal para ser utilizada como octeto de tipo en un paquete RR de nivel tres. Se introducen como parámetros: El Pr, el bit M y 1 como indicación del tipo de paquete.

4.7.2.6 Función MHD (H: String) Byte.

Función que extrae el bit M del byte de tipo de nivel 3.

4.7.2.7 Función PsHD (H: String) Byte.

Función que extrae el valor Ps del byte de tipo de nivel 3.

4.7.2.8 Función VerBitD (H: String) Boolean.

Toma valor cierto si en el byte del IGF introducido como parámetro el bit D está activo. Toma valor falso si el bit D está a cero.

4.7.2.9 Función PonerBitD (H: String): String.

Entrega una cadena de texto representando un octeto en base hexadecimal. Esta difiere de la entregada como parámetro en que la posición correspondiente al bit D está a uno

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.7.2.10 Función QuitarBitD (H: String):String.

Entrega una cadena de texto representando un octeto en base hexadecimal .Esta difiere de la entregada como parámetro en que la posición correspondiente al bit D está a cero

4.7.2.11 Función VerBitQ (H: String):Boolean.

Muestra el valor del bit Q. Se introduce como parámetro el octeto IGF

4.7.2.12 Funcion NrPRR (D, E, F: Boolean):String.

Forma el bit de control de nivel 2 para un RR.oRNR. Se introducen, por estre orden, el Nr, el bit P y el tipo de paquete.

4.7.2.13 Función NrHD (K: String):Byte.

Da el valor numérico del Nr del octeto de control introducido como parámetro.

4.7.2.14 Función NsHD (K: String):Byte.

Da el valor numérico de Ns del octeto de control de nivel 2 introducido como parámetro.

4.7.2.15 Función PHD (K: String):Byte.

Da el valor del bit P del octeto de control introducido como parámetro.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.7.2.16 Función NrPNsDH (D, E, F Byte):String.

Forma el octeto de control de nivel 2 a partir de los valores de Nr, P y Ns, introducidos en este orden, como parámetros.

4.7.2.17 Función CRC(Trama: String):String.

Función que realiza el cálculo del CRC con el polinomio generador normalizado por la IUT. El parámetro introducido es la trama de nivel 2 representada en una cadena de caracteres en la que cada dos caracteres representan un octeto en hexadecimal. Como resultado produce una cadena de cuatro caracteres que equivalen a dos octetos expresados en hexadecimal.

4.7.2.18 Función ConvCadAH (Cad: String):String.

Convierte una cadena de texto ASCII en sus valores numéricos expresados en hexadecimal.

4.7.2.19 Función ConvCadHA (Cad: String):String.

Realiza la conversión inversa de la función anterior.

4.7.2.20 Función VerMod (Cad: String):String.

Muestra el módulo de el IGF introducido como parámetro.

4.7.2.21 Función InfoCRC (Trama: String):String.

Calcula el CRC de una trama tipo INFO.-en la que, por limitaciones de la variable String de Delphi, se representan los datos de usuario en formato ASCII en vez de hexadecimal.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

4.7.2.22 Función HexBin (H: String): String.

Pasa una valor numérico de base hexadecimal a binario.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5

Presupuesto

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.1 Gastos de Personal

5.1.1 Definición de objetivos

Comprende el total de horas que se han invertido en plantear los objetivos básicos y requisitos imprescindibles que debe cumplir el proyecto Fin de carrera **Simulador X25.**

Definición de Objetivos	30h

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.1.2 Especificación de características

Este apartado abarca el tiempo empleado en la pormenorización y detalle de las diferentes características técnicas que se implementan en el proyecto

5.1.2.1 Diseño del formato gráfico

Este apartado especifica las características gráficas de la presentación y rotulación de títulos de los elementos del programa así como de la composición final.

5.1.2.2 Definición de características de Terminal

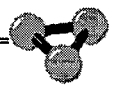
Comprende el estudio de todas las funciones, limitaciones y operativas de los elementos Terminal de usuario.

5.1.2.3 Definición de características del Nodo de Red.

Comprende el estudio de todas las funciones, limitaciones y operativas de los elementos Nodos de Red

5.1.2.4 Definición de la estrustura de Red

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006



Comprende el estudio de todas las funciones, limitaciones y operativas del conjunto de la Red, relacionando a los Terminales con los Nodos, las rutas y los elementos de control y configuración del programa.

Diseño del formato gráfico	21h
Definicion de características del Terminal de usuario	11h
Definicion de características del Nodo de Red	16h
Definición de la estructura de Red	22h
Total horas de planificacion	100h

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.1.3 Implementación del código

En este apartado se hace un balance del tiempo y coste invertido en la formulación,corrección y optimización del código de programa

5.1.3.1 Estructura principal

5.1.3.1.1 **Estructura gráfica**

Define el interfaz gráfico del programa con el usuario a través de su presentación en pantalla.

5.1.3.1.2 **Algoritmo de repartición de tiempos.**

Define las prioridades y colas de espera a las que han de someterse los distintos objetos implementados en el programa. Este apartado está íntimemente relacionado con el procedimiento Idle del lenguaje de programción.

5.1.3.1.3 **Panel de control**

Define el desarrollo de la estructura del objeto Tpanel, que constituye una especie pizarra de control en la que cada objeto del programa notifica las acciones que emprende en todo momento.

Estructura Gráfica 55h

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Algoritmo de repartición de tiempos	60h
Panel de Control	63h

5.1.3.2 Terminal de usuario

5.1.3.2.1 Código del terminal

Comprende las horas dedicados al estudio e implementación del código que rige las acciones del terminal de usuario en sus niveles superiores.

5.1.3.2.2 Nivel de Enlace

Comprende las horas dedicados al estudio e implementación del código que rige las acciones del terminal de usuario en sus niveles superiores.

5.1.3.2.3 Nivel de Red para el Terminal

Comprende las horas dedicados al estudio e implementación del código que rige las acciones del terminal de usuario en su nivel de Red

5.1.3.2.4 Nivel Físico

Comprende las horas dedicados al estudio e implementación del código que rige las acciones del terminal de usuario en su nivel físico

Código del terminal	87h
Nivel de Enlace	41h
Nivel de Red para el terminal	82h
Nivel Físico	15h

5.1.3.3 Nodo de Red

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.1.3.3.1 **Conmutador de Red**

Define el tiempo empleado en la implmentación del código que rige las acciones tomadas por el conmutador de Red, elemento encargado de gestionar las comunicaciones entre nodos.

5.1.3.3.2 **Nivel 3 del conmutador**

Comprende las horas dedicados al estudio e implementación del código que rige las acciones del conmutador o nodo en su nivel de Red.

5.1.3.3.3 **Procedimiento de Rutas**

Define el tiempo empleado en el desarrollo del código que rige el contol de rutas.

Conmutador de Red	81h
Nivel 3 del Conmutador	93h
Procedimiento Rutas	28h

5.1.3.4 Analizadores

Analizadores	135h

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.1.4 Coste total de las horas trabajadas

Total horas de planificación	100h
Horas de escritura de código	740h
Total Horas	840h

Una vez que se han determinado las horas dedicadas a la realización del proyecto calculamos el coste de la hora trabajada de la siguiente manera:

1. Coste horario según el documento "Propuesta de baremos orientativos para el cálculo de honorarios", editado por el Colegio de Ingenieros Técnicos de Telecomunicaciones. Este coste es de 9.700pts/hora.
2. Coeficientes de reducción pormenorizados en el mismo documeto
   Exceso de 72 horas hasta 1080 horas: Coeficiente C1= 0,8
   Exceso de 1080 horas Coeficiente C2= 0,4.

Para el cálculo del total se ha realizado el siguiente cálculo:

Total Coste =72h*9,700pts+(840h-72h)*0,8*9.700pts = 6.658.080 pts

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.2 Gasto en Equipos

Se han utilizado para el desarrollo del proyecto dos ordenadores compatibles PC. Se ha estimado su valor por unidad en 150.000.ptas. Debido a que el tiempo de desarrollo del proyecto ha sido de un año y se considera el tiempo de vida de las unidades de tres años, estimamos el coste de amortización por unidad en 50.000 pts.

Por lo tanto dos ordenadores conllevan un costo de 100.000. pts en amortización.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.3 Presupuesto Total

Gastos de Personal	6.658.080 pts
Amortización de Equipos	100.000 pts
Total	6.758.080 pts

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5.3.1 Presupuesto parcial del proyecto: Simulador X25. Terminal de usuario

Se considera el 50% de las horas trabajadas no especificamente dedicadas al desarrollo del código de terminal de usuario. esto supone:

Total horas de trabajo común	100h
Horas de escritura de código terminal	360h
Total Horas	410h

Cálculo de gasto de personal atendiendo a los coeficientes anteriormente expuestos:

72/2*9700+(410-72/2)*0.8*9700 = 3.251.440 pts

Gastos de Personal	3.251.440 pts
Amortización de Equipos	50.000 pts
Total	3.301.440 pts

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Anexo 1

Listados

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad Term1

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
unit Term1;
```

Unidad Term1 encargada de implementar el Terminal de usuario.

```
interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Menus,
  Ccterm,
  StdCtrls
 ,Buttons,
  Calculo,
  Tabs,
  ExtCtrls,
  CfgNodo;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

A continuación se definen las variables y métodos que conforman el objeto Terminal

```
type
 TTerminal = class(TForm)
   MainMenu1: TMainMenu;
   Com1: TMenuItem;
   Liber: TMenuItem;
   Reini: TMenuItem;
   Rearran: TMenuItem;
   Enviar1: TMenuItem;
   Archivo1: TMenuItem;
   N3618: TMenuItem;
   N3677: TMenuItem;
   N5981: TMenuItem;
   N7342: TMenuItem;
   Llam: TMenuItem;
   Panel1: TPanel;
   Panel2: TPanel;
   Panel3: TPanel;
   Panel4: TPanel;
   Panel5: Tpanel;
   Panel6: TPanel;
   Memo1: TMemo;
   Memo2: TMemo;
   Memo3: TMemo;
   Memo4: TMemo;
   Memo5: TMemo;
   Memo6: TMemo;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Memo7: TMemo;

Memo8: TMemo;

Memo9: TMemo;

Memo10: TMemo;

Panel7: TPanel;

Panel8: TPanel;

TabSet1: TTabSet;

OpenDialog1: TOpenDialog;

N1: TMenuItem;

BitD01: TMenuItem;

N2: TMenuItem;

Limpiapantalla1: TMenuItem;

GCU1: TMenuItem;

© Del documento, los autores. Digitalización realizada por ULPGC. Bibliotecas Universitaria, 2006

Aqui se enumeran todos los métodos de la Clase Terminal.

```
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Control;
Function TRansmiteDatos:String;
Procedure RecibeDatos( Datos : String );
Function TRansmiteComandos:String;
Procedure RecibeComandos( Comandos : String );
procedure FormCreate(Sender: TObject);
procedure Memo1KeyPress(Sender: TObject; var Key: Char);
procedure FormResize(Sender: TObject);
procedure N3618Click(Sender: TObject);
procedure N3677Click(Sender: TObject);
procedure N5981Click(Sender: TObject);
procedure N7342Click(Sender: TObject);
procedure LiberClick(Sender: TObject);
procedure Memo3KeyPress(Sender: TObject; var Key: Char);
procedure Memo5KeyPress(Sender: TObject; var Key: Char);
procedure Memo7KeyPress(Sender: TObject; var Key: Char);
procedure Memo9KeyPress(Sender: TObject; var Key: Char);
procedure TabSet1Click(Sender: TObject);
procedure ReiniClick(Sender: TObject);
procedure RearranClick(Sender: TObject);
procedure Archivo1Click(Sender: TObject);
Procedure InsLinea2( Linea:String );
Procedure InsLinea4( Linea:String );
Procedure InsLinea6( Linea:String );
Procedure InsLinea8( Linea:String );
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
Procedure InsLinea10( Linea:String );
Procedure Miralinea2( Cadena : String );
Procedure Miralinea4( Cadena : String );
Procedure Miralinea6( Cadena : String );
Procedure Miralinea8( Cadena : String );
Procedure Miralinea10( Cadena : String );
procedure BitD01Click(Sender: TObject);
procedure Limpiapantalla1Click(Sender: TObject);
Procedure LeerConfig( Archivo : String );
Function CuentaCanales( Par : CFG ): Integer;
procedure GCU1Click(Sender: TObject);
Function PonHora : String;
Function RxSeleccionRapida( Comando:String ; Var Hora : String ):Boolean;
Procedure ResumenConfig;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Declaración de las variables públicas del Objeto.

```
 private
  { Private declarations }
public
  { Public declarations }
  Nombre : String;
  TxDatos : String;
  RxDatos : String;
  TxComandos : String;
  RxComandos : String;
  HayComandosTx : Boolean;
  HayComandosRx : Boolean;
  HayDatosTx : Boolean;
  HayDatosRx : Boolean;
  Boton      : TSpeedButton;
  Activado : ^Boolean;
  IR       :String;
  Canal0,Canal1,Canal2,Canal3,Canal4:Byte;
  TipoC0,TipoC1,TipoC2,TipoC3,TipoC4:String;
  Canal :Byte;
  TabAnterior :Integer;
  EnvArch : Text;
  EnvActivo : Boolean;
  EnvCanal : Byte;
  Puntero2   : Integer;
  Puntero4   : Integer;
  Puntero6   : Integer;
  Puntero8   : Integer;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  Puntero10  : Integer;
  Maxlineas  : Integer;
  IGF        : String;
  M          : String;
  Permiso    : Boolean;
  ContPermiso: Integer;
  Direct     : String;
  Parametros : CFG;
  ArchCfg    : File of CFG;
  SrHora     : String;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Comienza la implementación del código.

```
var
  Terminal: TTerminal;

implementation

{$R *.DFM}

Uses SimX25;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método llamado cuando se cierra el Terminal de Usuario.

```
procedure TTerminal.FormClose(Sender: TObject; var Action: TCloseAction);
begin
{ Cierra el fichero de envio si esta abierto }
 If EnvActivo then CloseFile( EnvArch );
 Boton.Enabled := True;
 Boton.Visible := True;
 Activado^ := False;
 RedX25.FormPaint( Sender );
 RedX25.CierraTerm( Nombre );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento encargado de implementar las decisiones referentes al protocolo. Se le llama desde el procedimiento Idle de la unidad SimX25.

```
Procedure TTerminal.Control;
Var EnvCad : String;
   Caracter : Char;
   IGFRx    : String;
begin
If HayDatosRx then
 begin
 if RxDatos<>'' then
  begin
  IGFRx:=Copy( RxDatos,1,2 );
  Canal:=HD( Copy( RxDatos ,3,2 ));
  RxDatos:=Copy( RxDatos,7,length( RxDatos )-4 );
  If Canal = Canal0 then
   begin
   If VerBitQ( IGFRx ) then
   begin
   If RxDatos='CLEAR' then Memo2.Clear;
   end
   Else
   MiraLinea2(  RxDatos );
   end;
  If Canal = Canal1 Then
   begin
   If VerBitQ( IGFRx ) then
   begin
   If RxDatos='CLEAR' then Memo4.Clear;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
end
Else
MiraLinea4( RxDatos );
end;
If Canal = Canal2 Then
begin
If VerBitQ( IGFRx ) then
begin
If RxDatos='CLEAR' then Memo6.Clear;
end
Else
MiraLinea6( RxDatos );
end;
If Canal = Canal3 Then
begin
If VerBitQ( IGFRx ) then
begin
If RxDatos='CLEAR' then Memo8.Clear;
end
Else
MiraLinea8( RxDatos );
end;
If Canal = Canal4 Then
begin
If VerBitQ( IGFRx ) then
begin
If RxDatos='CLEAR' then Memo10.Clear;
end
Else
MiraLinea10( RxDatos );
end;
RxDatos:='';
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    HayDatosRx:=False;
  end;
 end;
If HayComandosRx then
 begin
 Case HD(Copy(RxComandos,3,2))of
  11:{ Solicitud de llamada }
   begin
   { Si hay seleccion rapida en la llamada pon la hora en la pantalla }
   HayComandosRx:=False;
   Panel8.Caption:='Recibida Llamada '{+RxComandos};
   Canal:=HD(Copy( RxComandos,1,2));
   TabSet1.Tabs[Canal-100]:= Copy( RxComandos,11,4 )+':'+Copy( RxComandos,1,2);
   TabSet1.TabIndex:=Canal-100;
    Case Tabset1.TabIndex of
   0:
    begin
    If TipoC0<>'CVP' then
     begin
     Memo1.Visible:=true;
     Memo2.Visible:=True;
     Canal0:=Canal;
     If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea2( SrHora );
     end
    Else Redx25.Visual( Nombre+' ERROR. Se recibe una llamada por un canal
Permanente ');
    end;
   1:
    begin
    If TipoC1<>'CVP' then
     begin
     Memo3.Visible:=true;
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    Memo4.Visible:=True;
    Canal1:=Canal;
    If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea4( SrHora );
    end
   Else Redx25.Visual( Nombre+' ERROR. Se recibe una llamada por un canal
Permanente ');
    end;
  2:
    begin
    If TipoC2<>'CVP' then
     begin
     Memo5.Visible:=true;
     Memo6.Visible:=True;
     Canal2:=Canal;
     If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea6( SrHora );
     end
    Else Redx25.Visual( Nombre+' ERROR. Se recibe una llamada por un canal
 Permanente ');
    end;
  3:
    begin
    If TipoC3<>'CVP' then
     begin
     Memo7.Visible:=true;
     Memo8.Visible:=True;
     Canal3:=Canal;
     If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea8( SrHora );
     end
    Else Redx25.Visual( Nombre+' ERROR. Se recibe una llamada por un canal
 Permanente ');
    end;
  4:
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        begin
      If TipoC4<>'CVP' then
       begin
       Memo9.Visible:=True;
       Memo10.Visible:=True;
       Canal4:=Canal;
       If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea10( SrHora );
       end
      Else Redx25.Visual( Nombre+' ERROR. Se recibe una llamada por un canal
Permanente ');
     end;
    end;{ Case tabset }
  RxComandos:='';
    end;
  15:{ Llamada aceptada }
    begin
    { Mira si la aceptacion tiene sr , si es asi pon la hora en pantalla }
    HayComandosRx:=False;
    Panel8.Caption:='Llamada Aceptada'{+RxComandos};
    Canal:=HD( Copy( RxComandos,1,2) );
    {Si es otro canal con el mismo nri añadele un digito }
    TabSet1.Tabs[Canal-100]:= Copy( RxComandos,7,4 )+':'+Copy( RxComandos,1,2);
    TabSet1.TabIndex:=Canal-100;
    Case Tabset1.TabIndex of
    0:
      begin
      Memo1.Visible:=true;
      Memo2.Visible:=True;
      Canal0:=Canal;
      If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea2( SrHora );
      end;
    1:
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 begin
 Memo3.Visible:=true;
 Memo4.Visible:=True;
 Canal1:=Canal;
 If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea4( SrHora );
 end;
2:
 begin
 Memo5.Visible:=true;
 Memo6.Visible:=True;
 Canal2:=Canal;
 If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea6( SrHora );
 end;
3:
 begin
 Memo7.Visible:=true;
 Memo8.Visible:=True;
 Canal3:=Canal;
 If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea8( SrHora );
 end;
4:
 begin
 Memo9.Visible:=True;
 Memo10.Visible:=True;
 Canal4:=Canal;
 If RxSeleccionRapida( RxComandos, SrHora ) then Miralinea10( SrHora );
 end;
end;{ Case tabset }
RxComandos:='';
end;
19:{ Solicitud de liberacion }
 begin
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 HayComandosRx:=False;
Canal:= HD( Copy( RxComandos ,1,2 ));
TabSet1.Tabs[ Canal-100 ]:='  ';
If ( Canal=Canal0 )and( TipoC0<>'CVP' ) then
 begin
 Memo1.Clear;
 Memo2.Clear;
 Memo1.Visible:=False;
 Memo2.Visible:=False;
 Canal0:=0;
 end;
If ( Canal=Canal1 )and( TipoC1<>'CVP' ) then
 begin
 Memo3.Clear;
 Memo4.Clear;
 Memo3.Visible:=False;
 Memo4.Visible:=False;
 Canal1:=0;
 end;
If ( Canal=Canal2 )and( TipoC2<>'CVP' ) then
 begin
 Memo5.Clear;
 Memo6.Clear;
 Memo5.Visible:=False;
 Memo6.Visible:=False;
 Canal2:=0;
 end;
If ( Canal=Canal3 )and( TipoC3<>'CVP' ) then
 begin
 Memo7.Clear;
 Memo8.Clear;
 Memo7.Visible:=False;
```

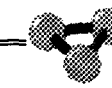
© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  Memo8.Visible:=False;
  Canal3:=0
  end;
 If ( Canal=Canal4 ) and( TipoC4<>'CVP' )then
  begin
  Memo9.Clear;
  Memo10.Clear;
  Memo9.Visible:=False;
  Memo10.Visible:=False;
  Canal4:=0;
  end;
 Panel8.Caption:='Solicitud de liberacion '+Copy(RxComandos,5,2 );
 RxComandos:='';
 end;
23:{Confirmacion de liberacion }
 begin
 HayComandosRx:=False;
 Panel8.Caption:='Confirmacion de liberacion'{+RxComandos};
 RxComandos:='';
 end;
35:{Interrupcion}
        begin
 end;
27:{ Reinicio }
 begin
 HayComandosRx:=False;
 Panel8.Caption:='Solicitud de Reinicio '+Copy(RxComandos,5,2 );
 Canal:= HD( Copy( RxComandos ,1,2 ));
 If ( Copy( RxComandos,5,2 )='01' )or( Copy( RxComandos,5,2 )='1D' )then
  begin
  If ( Canal=Canal0 )and( TipoC0='CVP' ) then
   begin
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  Memo1.Enabled:=False;
 Memo2.Enabled:=False;
 Memo1.Color:=ClTeal;
 Memo2.Color:=ClTeal;
 TabSet1.TabIndex:=0;
 RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
              +' por el canal '+IntToStr( Canal ));
 end;
If ( Canal=Canal1 )and( TipoC1='CVP' ) then
 begin
 Memo3.Enabled:=False;
 Memo4.Enabled:=False;
 Memo3.Color:=ClTeal;
 Memo4.Color:=ClTeal;
 TabSet1.TabIndex:=1;
 RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
              +' por el canal '+IntToStr( Canal ));
 end;
If ( Canal=Canal2 )and( TipoC2='CVP' ) then
 begin
 Memo5.Enabled:=False;
 Memo6.Enabled:=False;
 Memo5.Color:=ClTeal;
 Memo6.Color:=ClTeal;
 TabSet1.TabIndex:=2;
 RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
              +' por el canal '+IntToStr( Canal ));
 end;
If ( Canal=Canal3 )and( TipoC3='CVP' ) then
 begin
 Memo7.Enabled:=False;
 Memo8.Enabled:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006



```
 Memo7.Color:=ClTeal;
 Memo8.Color:=ClTeal;
 TabSet1.TabIndex:=3;
 RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
              +' por el canal '+IntToStr( Canal ));
 end;
If ( Canal=Canal4 ) and( TipoC4='CVP' )then
 begin
 Memo9.Enabled:=False;
 Memo10.Enabled:=False;
 Memo9.Color:=ClTeal;
 Memo10.Color:=ClTeal;
 TabSet1.TabIndex:=4;
 RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
              +' por el canal '+IntToStr( Canal ));
 end;
end;
If Copy( RxComandos, 5,2)='09' then
 begin
 If ( Canal=Canal0 )and( TipoC0='CVP' ) then
  begin
  Memo1.Enabled:=True;
  Memo2.Enabled:=True;
  Memo1.Color:=ClAqua;
  Memo2.Color:=ClAqua;
  TabSet1.TabIndex:=0;
  RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
              +' por el canal '+IntToStr( Canal ));
 end;
 If ( Canal=Canal1 )and( TipoC1='CVP' ) then
  begin
  Memo3.Enabled:=True;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    Memo4.Enabled:=True;
   Memo3.Color:=ClAqua;
   Memo4.Color:=ClAqua;
   TabSet1.TabIndex:=1;
   RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
                +' por el canal '+IntToStr( Canal ));
  end;
 If ( Canal=Canal2 )and( TipoC2='CVP' ) then
  begin
   Memo5.Enabled:=True;
   Memo6.Enabled:=True;
   Memo5.Color:=ClAqua;
   Memo6.Color:=ClAqua;
   TabSet1.TabIndex:=2;
   RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
                +' por el canal '+IntToStr( Canal ));
  end;
 If ( Canal=Canal3 )and( TipoC3='CVP' ) then
  begin
   Memo7.Enabled:=True;
   Memo8.Enabled:=True;
   Memo7.Color:=ClAqua;
   Memo8.Color:=ClAqua;
   TabSet1.TabIndex:=3;
   RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
                +' por el canal '+IntToStr( Canal ));
  end;
 If ( Canal=Canal4 ) and( TipoC4='CVP' )then
  begin
   Memo9.Enabled:=True;
   Memo10.Enabled:=True;
   Memo9.Color:=ClAqua;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    Memo10.Color:=ClAqua;
    TabSet1.TabIndex:=4;
    RedX25.Visual( Nombre+' Rx de Reinicio Causa '+Copy( RxComandos,5,2)
                  +' por el canal '+IntToStr( Canal ));
    end;
   end;
   RxComandos:='';
   end;
31:{ Confirmacion de reinicio }
   begin
   HayComandosRx:=False;
   Panel8.Caption:='Confirmacion de reinicio'{+RxComandos};
   RxComandos:='';
   end;
 251:{ Rearranque }
   begin
    { Rearranque por red en operacion .Inicio de nivel 3 }
    llam.Enabled:=True;
    Liber.Enabled:=True;
    Reini.Enabled:=True;
    Rearran.Enabled:=True;
   HayComandosRx:=False;
   Panel8.Caption:='Solicitud de Rearranque'{+RxComandos};
   RxComandos:='';
   If TipoC0<>'CVP' then
    begin
    Canal0:=0;
    Memo1.Clear;
    Memo1.Visible:=False;
    Memo2.Clear;
    Memo2.Visible:=False;
    If TabSet1.Tabs.Count>=1 then TabSet1.Tabs[ 0 ]:=' ';
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  end;
If TipoC1<>'CVP'then
 begin
 Canal1:=0;
 Memo3.Clear;
 Memo3.Visible:=False;
 Memo4.Clear;
 Memo4.Visible:=False;
 If TabSet1.Tabs.Count>=2 then TabSet1.Tabs[ 1 ]:=' ';
 end;
If TipoC2<>'CVP'then
 begin
 Canal2:=0;
 Memo5.Clear;
 Memo5.Visible:=False;
 Memo6.Clear;
 Memo6.Visible:=False;
 If TabSet1.Tabs.Count>=3 then TabSet1.Tabs[ 2 ]:=' ';
 end;
If TipoC3<>'CVP' then
 begin
 Canal3:=0;
 Memo7.Clear;
 Memo7.Visible:=False;
 Memo8.Clear;
 Memo8.Visible:=False;
 If TabSet1.Tabs.Count>=4 then TabSet1.Tabs[ 3 ]:=' ';
 end;
If TipoC4<>'CVP' then
 begin
 Canal4:=0;
 Memo9.Clear;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
Memo9.Visible:=False;
Memo10.Clear;
Memo10.Visible:=False;
If TabSet1.Tabs.Count>=5 then TabSet1.Tabs[ 4 ]:=' ';
end;
end;
255:{ Confirmacion de rearranque }
begin
HayComandosRx:=False;
Panel8.Caption:='Confirmacion de Rearranque'{+RxComandos};
RxComandos:='';
   If TipoC0<>'CVP' then
begin
Canal0:=0;
Memo1.Clear;
Memo1.Visible:=False;
Memo2.Clear;
Memo2.Visible:=False;
TabSet1.Tabs[ 0 ]:=' ';
end;
If TipoC1<>'CVP'then
begin
Canal1:=0;
Memo3.Clear;
Memo3.Visible:=False;
Memo4.Clear;
Memo4.Visible:=False;
TabSet1.Tabs[ 1 ]:=' ';
end;
If TipoC2<>'CVP'then
begin
Canal2:=0;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    Memo5.Clear;
   Memo5.Visible:=False;
   Memo6.Clear;
   Memo6.Visible:=False;
   TabSet1.Tabs[ 2 ]:=' ';
   end;
  If TipoC3<>'CVP' then
   begin
   Canal3:=0;
   Memo7.Clear;
   Memo7.Visible:=False;
   Memo8.Clear;
   Memo8.Visible:=False;
   TabSet1.Tabs[ 3 ]:=' ';
   end;
  If TipoC4<>'CVP' then
   begin
   Canal4:=0;
   Memo9.Clear;
   Memo9.Visible:=False;
   Memo10.Clear;
   Memo10.Visible:=False;
   TabSet1.Tabs[ 4 ]:=' ';
   end;
  end;

 end;{ End case }
 end;
If Not( Permiso ) then
 begin
 Inc( ContPermiso );
 If ContPermiso>50 then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 begin
 Permiso:=True;
 end;
end;
 { Si esta activa la transmision de Archivo manda una linea con M=1}
 { Si es la ultima linea del archivo pon M=0 }
If ( EnvActivo )and( Not( HayDatosTx ))and( Permiso )then
 begin
 EnvCad:='';
 RedX25.Visual( Nombre+' Entra en la opcion Env Archivo' );
 { Haz M:=1 }
 { Lee 128 caracteres del archivo o hasta el final del Archivo }
 While ( not( Eof( EnvArch )))and(Length( EnvCad )<128) do
  begin
  Read( EnvArch,Caracter );
  EnvCad:=EnvCad+Caracter;
  end;
 RedX25.Visual( Nombre+' Lee del Archivo la cadena '+EnvCad );
 { Si el archivo se termino cierralo }
 If Eof( EnvArch ) then
  begin
  CloseFile( EnvArch );
  Archivo1.Enabled:=True;
  EnvActivo:=False;
  M:='00';
  RedX25.Visual( Nombre+' Termina el archivo y lo cierra ' );
  Panel8.Caption:='Envio Finalizado';
  { Haz M=0 }
  end;
 { Envia como datos lo leido del archivo}
  If EnvCad<>'' then
   begin
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    TxDatos:=IGF+DH( EnvCanal )+M+ EnvCad ;
   HayDatosTx:=True;
   RedX25.Visual( Nombre+' Envia el paquete de datos ' );
   end;
  Permiso:=False;
  ContPermiso:=0;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Métodos encargados de intercambiar datos con la capa de Nivel 3 del protocolo.

```
FuncTion TTerminal.TransmiteDatos : String ;
begin
TransmiteDatos:=TxDatos;
HayDatosTx:=False;
TxDatos:='';
end;
Procedure TTerminal.RecibeDatos( Datos : String );
begin
HayDatosRx:=true;
RxDatos:=Datos;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Métodos encargados de intercambiar comandos con la capa de Nivel 3 del protocolo. Entendemos por comandos todo aquel paquete de Nivel 3 que no es un paquete de Datos ni un RR.

```
FuncTion TTerminal.TransmiteComandos : String ;
begin
TransmiteComandos:=TxComandos;
HayComandosTx:=False;
TxComandos:='';
end;
Procedure TTerminal.RecibeComandos( Comandos : String );
begin
HayComandosRx:=true;
RxComandos:=Comandos;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento de creación del objeto Terminal. Se crea a sí mismo e inicia las variables que utiliza.

```
procedure TTerminal.FormCreate(Sender: TObject);
Var N :Integer;
begin
   TxDatos :=" ;
   RxDatos :=";
   TxComandos:=";
   RxComandos:=";
   HayComandosTx:=False;
   HayComandosRx:=False;
   HayDatosTx :=False;
   HayDatosRx :=False;
   Memo1.Clear;
   Memo2.Clear;
   Memo2.ReadOnly:=true;
   Memo3.Clear;
   Memo4.Clear;
   Memo4.ReadOnly:=True;
   Memo5.Clear;
   Memo6.Clear;
   Memo6.ReadOnly:=True;
   Memo7.Clear;
   Memo8.Clear;
   Memo8.ReadOnly:=True;
   Memo9.Clear;
   Memo10.Clear;
   Memo10.ReadOnly:=True;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    Canal:=0;
   Canal0:=0;
   Canal1:=0;
   Canal2:=0;
   Canal3:=0;
   Canal4:=0;
   TabAnterior:=4;
   llam.Enabled:=False;
   Liber.Enabled:=False;
   Reini.Enabled:=False;
   Rearran.Enabled:=False;
   EnvActivo:=False;
   Puntero2  :=0;
   Puntero4  :=0;
   Puntero6  :=0;
   Puntero8  :=0;
   Puntero10 :=0;
   Maxlineas :=200;
   IGF:='10';
   M:='00';
   Permiso:=False;
   ContPermiso:=0;
   For N:=0 to 4 do TabSet1.Tabs[ N ]:=' ';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que se activa cada vez que se pulsa una tecla desde el editor. Estudia si en la línea hay 128 caracteres o si la última tecla pulsada es un retorno de carro. Si es así envía el contenido de la línea como un paquete de datos al Nivel 3.

```
procedure TTerminal.Memo1KeyPress(Sender: TObject; var Key: Char);
begin
If Length( Memo1.Lines[ Memo1.Lines.Count-1])=128 then
 begin
 TxDatos:=IGF+DH( Canal0 )+M+Memo1.Lines[ Memo1.Lines.Count-1];
 Memo1.Lines.Add('');
 HayDatosTx:=True;
 end;
If Key = Char( 13 ) then
 begin
  TxDatos:=IGF+DH( Canal0 )+M+Memo1.Lines[ Memo1.Lines.Count-1 ];
  HayDatosTx:=True;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que se activa cada vez que se pulsa una tecla desde el editor. Estudia si en la línea hay 128 caracteres o si la última tecla pulsada es un retorno de carro. Si es así, envía el contenido de la línea como un paquete de datos al Nivel 3.

```
procedure TTerminal.Memo3KeyPress(Sender: TObject; var Key: Char);
begin
If Length( Memo3.Lines[ Memo3.Lines.Count-1])=128 then
 begin
 TxDatos:=IGF+DH( Canal1 )+M+Memo3.Lines[ Memo3.Lines.Count-1];
 Memo3.Lines.Add('');
 HayDatosTx:=True;
 end;
If Key = Char( 13 ) then
 begin
  TxDatos:=IGF+DH( Canal1 )+M+Memo3.Lines[ Memo3.Lines.Count-1 ];
  HayDatosTx:=True;
 end;

end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que se activa cada vez que se pulsa una tecla desde el editor. Estudia si en la línea hay 128 caracteres o si la última tecla pulsada es un retorno de carro. Si es así, envía el contenido de la línea como un paquete de datos al Nivel 3.

```
procedure TTerminal.Memo5KeyPress(Sender: TObject; var Key: Char);
begin
If Length( Memo5.Lines[ Memo5.Lines.Count-1])=128 then
 begin
 TxDatos:=IGF+DH( Canal2 )+M+Memo5.Lines[ Memo5.Lines.Count-1];
 Memo5.Lines.Add('');
 HayDatosTx:=True;
 end;
If Key = Char( 13 ) then
 begin
  TxDatos:=IGF+DH( Canal2 )+M+Memo5.Lines[ Memo5.Lines.Count-1 ];
  HayDatosTx:=True;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que se activa cada vez que se pulsa una tecla desde el editor. Estudia si en la línea hay 128 caracteres o si la última tecla pulsada es un retorno de carro. Si es así, envía el contenido de la línea como un paquete de datos al Nivel 3.

```
procedure TTerminal.Memo7KeyPress(Sender: TObject; var Key: Char);
begin
If Length( Memo7.Lines[ Memo7.Lines.Count-1])=128 then
 begin
 TxDatos:=IGF+DH( Canal3 )+M+Memo7.Lines[ Memo7.Lines.Count-1];
 Memo7.Lines.Add('');
 HayDatosTx:=True;
 end;
If Key = Char( 13 ) then
 begin
  TxDatos:=IGF+DH( Canal3 )+M+Memo7.Lines[ Memo7.Lines.Count-1 ];
  HayDatosTx:=True;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que se activa cada vez que se pulsa una tecla desde el editor. Estudia si en la línea hay 128 caracteres o si la última tecla pulsada es un retorno de carro. Si es así, envía el contenido de la línea como un paquete de datos al Nivel 3.

```
procedure TTerminal.Memo9KeyPress(Sender: TObject; var Key: Char);
begin
If Length( Memo9.Lines[ Memo9.Lines.Count-1])=128 then
 begin
 TxDatos:=IGF+DH( Canal4 )+M+Memo9.Lines[ Memo9.Lines.Count-1];
 Memo9.Lines.Add('');
 HayDatosTx:=True;
 end;
If Key = Char( 13 ) then
 begin
  TxDatos:=IGF+DH( Canal4 )+M+Memo9.Lines[ Memo9.Lines.Count-1 ];
  HayDatosTx:=True;
 end;

end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de cambiar los tamaños de los editores, donde se recibe o envía texto a la línea.

```
procedure TTerminal.FormResize(Sender: TObject);
begin
Panel3.Width := ClientWidth div 2;
Panel5.Width := ClientWidth div 2;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de obtener la hora del sistema para utilizarla como ejemplo en la facilidad de selección rápida.

```
Function TTerminal.PonHora : String;
Var N : Integer;
   Hora : String;
begin
Hora:=TimetoStr( Time );
for n:=1 to length( Hora ) do
 if Hora[ N ]=':'then Hora[ N ]:='A';
If length( Hora ) mod 2 <> 0 then Hora :='0'+Hora;
 PonHora:=Hora;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que envía un comando de llamada a la capa de nivel inferior para el terminal remoto con IR 3618.

```
procedure TTerminal.N3618Click(Sender: TObject);
begin
{ Envia llamada al 3618 }
If Cursor=crDefault then
 begin
 TxComandos:=DH(0)+DH(11)+'4'+'4'+'3618'+IR+'00';
  { Longitud de facilidades a cero }
 If GCU1.checked then
  begin
  { tiene grupo cerrado de usuarios }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2)+
          copy(txComandos,17,length( txComandos ))+'0300';
  end;

 If Not( GCU1.Checked )and( Parametros.GCUAS )then
  begin
  { Tiene grupo cerrado de usuarios pero esta llamada es de acceso de salida }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
          copy(txComandos,17,length( txComandos ))+'0900';
  end;
 if ( Parametros.sr )and( not( Parametros.scr )) then
  begin
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        copy(txComandos,17,length( txComandos ))+'0180';
 end;
 if (not( Parametros.sr ))and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0101';
 end;
if ( Parametros.sr )and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0181';
 end;
 if Parametros.NPCF then
 begin
 TxComandos:=Copy( TxComandos, 1, 14 )+
        DH( HD( Copy( TxComandos, 15,2 ) )+3 )+
        copy( TxComandos, 17,length( TxComandos ))+'43'+
        '0'+IntToStr(Parametros.VCVCRx)+
        '0'+IntToStr(Parametros.VCVCTx);
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end
 Else
 begin
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end;

HayComandosTx:=True;
 end;
If Cursor=crDrag then  RedX25.LlamaAyuda( 28 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que envía un comando de llamada a la capa de nivel inferior para el terminal remoto con IR 3677.

```
procedure TTerminal.N3677Click(Sender: TObject);
begin
{ Envia llamada al 3677 }
If Cursor=crDefault then
 begin
 TxComandos:=DH(0)+DH(11)+'4'+'4'+'3677'+IR+'00';
 { Longitud de facilidades a cero }
 If GCU1.checked then
  begin
  { tiene grupo cerrado de usuarios }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2)+
          copy(txComandos,17,length( txComandos ))+'0300';
  end;

 If Not( GCU1.Checked )and( Parametros.GCUAS )then
  begin
  { Tiene grupo cerrado de usuarios pero esta llamada es de acceso de salida }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
          copy(txComandos,17,length( txComandos ))+'0900';
  end;
 if ( Parametros.sr )and( not( Parametros.scr )) then
  begin
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        copy(txComandos,17,length( txComandos ))+'0180';
 end;
if (not( Parametros.sr ))and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0101';
 end;
if ( Parametros.sr )and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0181';
 end;
if Parametros.NPCF then
 begin
 TxComandos:=Copy( TxComandos, 1, 14 )+
        DH( HD( Copy( TxComandos, 15,2 ) )+3 )+
        copy( TxComandos, 17,length( TxComandos ))+'43'+
        '0'+IntToStr(Parametros.VCVCRx)+
        '0'+IntToStr(Parametros.VCVCTx);
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end
Else
 begin
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end;

HayComandosTx:=True;
 end;
If Cursor = crDrag then RedX25.LlamaAyuda( 29 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que envía un comando de llamada a la capa de nivel inferior para el terminal remoto con IR 5981.

```
procedure TTerminal.N5981Click(Sender: TObject);
begin
{ Envia llamada al 5981 }
If Cursor=crDefault then
 begin
 TxComandos:=DH(0)+DH(11)+'4'+'4'+'5981'+IR+'00';
 { Longitud de facilidades a cero }
 If GCU1.checked then
  begin
  { tiene grupo cerrado de usuarios }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2)+
          copy(txComandos,17,length( txComandos ))+'0300';
  end;

 If Not( GCU1.Checked )and( Parametros.GCUAS )then
  begin
  { Tiene grupo cerrado de usuarios pero esta llamada es de acceso de salida }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
          copy(txComandos,17,length( txComandos ))+'0900';
  end;
 if ( Parametros.sr )and( not( Parametros.scr )) then
  begin
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        copy(txComandos,17,length( txComandos ))+'0180';
 TxComandos:=TxComandos+ponHora;
 end;
if (not( Parametros.sr ))and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0101';
 end;
if ( Parametros.sr )and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0181';
 end;
if Parametros.NPCF then
 begin
 TxComandos:=Copy( TxComandos, 1, 14 )+
        DH( HD( Copy( TxComandos, 15,2 ) )+3 )+
        copy( TxComandos, 17,length( TxComandos ))+'43'+
        '0'+IntToStr(Parametros.VCVCRx)+
        '0'+IntToStr(Parametros.VCVCTx);
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end
Else
 begin
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end;
HayComandosTx:=True;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 30 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que envía un comando de llamada a la capa de Nivel inferior para el terminal remoto con IR 7342.

```
procedure TTerminal.N7342Click(Sender: TObject);
begin
{ Envia llamada al 7342 }
If Cursor = crDefault then
 begin
 TxComandos:=DH(0)+DH(11)+'4'+'4'+'7342'+IR+'00';
 { Longitud de facilidades a cero }
 If GCU1.checked then
  begin
  { tiene grupo cerrado de usuarios }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2)+
          copy(txComandos,17,length( txComandos ))+'0300';
  end;

 If Not( GCU1.Checked )and( Parametros.GCUAS )then
  begin
  { Tiene grupo cerrado de usuarios pero esta llamada es de acceso de salida }
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
          copy(txComandos,17,length( txComandos ))+'0900';
  end;
 if ( Parametros.sr )and( not( Parametros.scr )) then
  begin
  TxComandos:=Copy( txComandos,1,14 )+
          DH(HD( Copy( TxComandos,15,2 ) )+2 )+
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        copy(txComandos,17,length( txComandos ))+'0180';
 end;
if (not( Parametros.sr ))and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0101';
 end;
if ( Parametros.sr )and(  Parametros.scr ) then
 begin
 TxComandos:=Copy( txComandos,1,14 )+
        DH(HD( Copy( TxComandos,15,2 ) )+2 )+
        copy(txComandos,17,length( txComandos ))+'0181';
 end;
if Parametros.NPCF then
 begin
 TxComandos:=Copy( TxComandos, 1, 14 )+
        DH( HD( Copy( TxComandos, 15,2 ) )+3 )+
        copy( TxComandos, 17,length( TxComandos ))+'43'+
        '0'+IntToStr(Parametros.VCVCRx)+
        '0'+IntToStr(Parametros.VCVCTx);
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end
Else
 begin
 If Parametros.sr then txComandos:=TxComandos+ponHora;
 end;

HayComandosTx:=True;
end;
If Cursor =crDrag then RedX25.LlamaAyuda( 31 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que libera el canal seleccionado en la lengüeta del editor

```
procedure TTerminal.LiberClick(Sender: TObject);
begin
 { enviar liberacion }
If Cursor = crDefault then
 begin
 Case TabSet1.TabIndex of
 0:
  begin
  If TipoC0<>'CVP' then
   begin
   Memo1.Clear;
   Memo2.Clear;
   Memo1.Visible:=False;
   Memo2.Visible:=False;
   Canal:=Canal0;
   Canal0:=0;
   If Canal<>0 then
    begin
    TabSet1.Tabs[Canal-100]:=' ';
    TxComandos:=DH( Canal )+DH(19)+'0000';
    HayComandosTx:=True;
    end;
   end
  Else ShowMessage(' Se intenta liberar un Canal Permanente ');
  end;
 1:
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 begin
 If TipoC1<>'CVP' then
  begin
  Memo3.Clear;
  Memo4.Clear;
  Memo3.Visible:=False;
  Memo4.Visible:=False;
  Canal:=Canal1;
  Canal1:=0;
  If Canal<>0 then
    begin
    TabSet1.Tabs[Canal-100]:=' ';
    TxComandos:=DH( Canal )+DH(19)+'0000';
    HayComandosTx:=True;
    end;
  end
 Else ShowMessage(' Se intenta liberar un Canal Permanente ');
 end;
2:
 begin
 If TipoC2<>'CVP' then
  begin
  Memo5.Clear;
  Memo6.Clear;
  Memo5.Visible:=False;
  Memo6.Visible:=False;
  Canal:=Canal2;
  Canal2:=0;
  If Canal<>0 then
    begin
    TabSet1.Tabs[Canal-100]:=' ';
    TxComandos:=DH( Canal )+DH(19)+'0000';
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      HayComandosTx:=True;
     end;
    end
   Else ShowMessage(' Se intenta liberar un Canal Permanente ');
   end;
  3:
   begin
   If TipoC3<>'CVP' then
    begin
    Memo7.Clear;
    Memo8.Clear;
    Memo7.Visible:=False;
    Memo8.Visible:=False;
    Canal:=Canal3;
    Canal3:=0;
    If Canal<>0 then
     begin
     TabSet1.Tabs[Canal-100]:='  ';
     TxComandos:=DH( Canal )+DH(19)+'0000';
     HayComandosTx:=True;
     end;
    end
   Else ShowMessage(' Se intenta liberar un Canal Permanente ');
   end;
  4:
   begin
   If TipoC4<>'CVP' then
    begin
    Memo9.Clear;
    Memo10.Clear;
    Memo9.Visible:=False;
    Memo10.Visible:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  Canal:=Canal4;
  Canal4:=0;
  If Canal<>0 then
    begin
    TabSet1.Tabs[Canal-100]:=' ';
    TxComandos:=DH( Canal )+DH(19)+'0000';
    HayComandosTx:=True;
    end;
   end
  Else ShowMessage(' Se intenta liberar un Canal Permanente ');
  end;
 end;{ Case }
 {If Canal<>0 then
  begin
  TabSet1.Tabs[Canal-100]:=' ';
  TxComandos:=DH( Canal )+DH(19)+'0000';
  HayComandosTx:=True;
  end;}
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 32 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite seleccionar las lengüetas que muestran cada uno de los editores asociados con cada uno de los canales virtuales.

```
procedure TTerminal.TabSet1Click(Sender: TObject);
begin
If Cursor= crDefault then
 begin
 Case TabAnterior of
 0:
  begin
   Memo1.Visible:=False;
   Memo2.Visible:=False;
  end;
 1:
  begin
   Memo3.Visible:=False;
   Memo4.Visible:=False;
  end;
 2:
  begin
   Memo5.Visible:=False;
   Memo6.Visible:=False;
  end;
 3:
  begin
   Memo7.Visible:=False;
   Memo8.Visible:=False;
  end;
 4:
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 begin
  Memo9.Visible:=False;
  Memo10.Visible:=False;
 end;
end;{End Case }

Case TabSet1.TabIndex of
0:
 begin
 If canal0<>0 then
  begin
  Memo1.Visible:=True;
  Memo2.Visible:=True;
  Canal:=Canal0;
  end;
 end;
1:
 begin
 If canal1<>0 then
  begin
  Memo3.Visible:=True;
  Memo4.Visible:=True;
  Canal:=Canal1;
  end;
 end;
2:
 begin
 If canal2<>0 then
  begin
  Memo5.Visible:=True;
  Memo6.Visible:=True;
  Canal:=Canal2;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      end;
    end;
  3:
    begin
    If canal3<>0 then
     begin
     Memo7.Visible:=True;
     Memo8.Visible:=True;
     Canal:=Canal3;
     end;
    end;
  4:
    begin
    If canal4<>0 then
     begin
     Memo9.Visible:=True;
     Memo10.Visible:=True;
     Canal:=Canal4;
     end;
    end;
 end;{End Case }
 TabAnterior:=TabSet1.TabIndex;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 33 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método por el que se solicita el envío de un paquete de reinicio a través del canal selecionado en el editor.

```
procedure TTerminal.ReiniClick(Sender: TObject);
begin
{ Solicitud de reinicio }
If Cursor = crDefault then
 begin
 TxComandos:=DH( Canal )+DH(27)+'0000';
 HayComandosTx:=True;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 34 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método por el que se solicita el envío de un paquete de rearranque a través del canal selecionado en el editor.

```
procedure TTerminal.RearranClick(Sender: TObject);
begin
{ Solicitud de Rearranque }
If Cursor = crDefault then
 begin
 TxComandos:=DH(0)+DH(251)+'0000';
 HayComandosTx:=True;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 35 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Solicitud de envío de un archivo. Se selecciona un archivo de los que hay en el subdirectorio de Envíos, especificado en la configuración de directorios y este método indica al procedimiento Control que cada vez que esté libre de tareas lea 128 caracteres y los envíe.

```
procedure TTerminal.Archivo1Click(Sender: TObject);
Const NombreArchivo : TFileName='*.TXT';
begin
If Cursor= crDefault then
 begin
 With OpenDialog1 do
  begin
  Title:='Enviar archivo';
  InitialDir:=Direct;
  Filter :='Archivos de texto *.TXT|*.TXT|'+
        'Todos          *.* |*.*';
  FileName:=NombreArchivo;
  If Execute then
   begin
   NombreArchivo:=Filename;
   M:='10';
   Permiso:=True;
   Panel8.Caption:=' Enviando Archivo '+NombreArchivo;
   Case TabSet1.TabIndex of
   0:
   begin
   If Canal0<>0 then
    begin
    EnvCanal:=Canal0;
    AssignFile( EnvArch,NombreArchivo );
    Reset( EnvArch );
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  EnvActivo:=True;
 Archivo1.Enabled:=False;
 RedX25.Visual( Nombre+'Comienza la Tx del archivo '+NombreArchivo
                        +' por el canal '+IntToStr(EnvCanal ));
 end;
end;
1:
begin
If Canal1<>0 then
 begin
 EnvCanal:=Canal1;
 AssignFile( EnvArch,NombreArchivo );
 Reset( EnvArch );
 EnvActivo:=True;
 Archivo1.Enabled:=False;
 RedX25.Visual( Nombre+'Comienza la Tx del archivo '+NombreArchivo
                        +' por el canal '+IntToStr(EnvCanal));
 end;
end;
2:
begin
If Canal2<>0 then
 begin
 EnvCanal:=Canal2;
 AssignFile( EnvArch,NombreArchivo );
 Reset( EnvArch );
 EnvActivo:=True;
 Archivo1.Enabled:=False;
 RedX25.Visual( Nombre+'Comienza la Tx del archivo '+NombreArchivo
                        +' por el canal '+IntToStr( EnvCanal));
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 3:
 begin
 If Canal3<>0 then
  begin
  EnvCanal:=Canal3;
  AssignFile( EnvArch,NombreArchivo );
  Reset( EnvArch );
  EnvActivo:=True;
  Archivo1.Enabled:=False;
  RedX25.Visual( Nombre+'Comienza la Tx del archivo '+NombreArchivo
                              +' por el canal '+IntToStr(EnvCanal));
  end;
 end;
 4:
 begin
 If Canal4<>0 then
  begin
  EnvCanal:=Canal4;
  AssignFile( EnvArch,NombreArchivo );
  Reset( EnvArch );
  EnvActivo:=True;
  Archivo1.Enabled:=False;
  RedX25.Visual( Nombre+'Comienza la Tx del archivo '+NombreArchivo
                              +' por el canal '+IntToStr( EnvCanal ));
  end;
 end;
 end{ Case }
 end;
 end;{ End With }
end;
If Cursor =crDrag then RedX25.LlamaAyuda( 36 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de insertar líneas en el editor. Si hay más de 200 líneas en el editor, se sobreescribe la primera, a continuación la segunda y así susesivamente.

```
Procedure TTerminal.InsLinea2( Linea:String );
Var M : Integer;
begin
If Memo2.Lines.Count>=MaxLineas then
 begin
 If Puntero2>MaxLineas-1 then Puntero2:=0;
 Memo2.Lines[ Puntero2 ]:=Linea;
 For M:=1 to length( Linea ) do
  If Linea[ M ]=Char( 13 ) then Inc( Puntero2 );
 If Puntero2=MaxLineas-1 then
  Memo2.Lines[Puntero2+1]:=' '
 Else
  Memo2.Lines[Puntero2+1]:='>>';
  Inc( Puntero2 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo2.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de insertar líneas en el editor. Si hay mas de 200 líneas en el editor, se sobreescribe la primera, a continuación la segunda y así susesivamente.

```
Procedure TTerminal.InsLinea4( Linea:String );
Var M : Integer;
begin
If Memo4.Lines.Count>=MaxLineas then
 begin
 If Puntero4>MaxLineas-1 then Puntero4:=0;
 Memo4.Lines[ Puntero4 ]:=Linea;
 For M:=1 to length( Linea ) do
  If Linea[ M ]=Char( 13 ) then Inc( Puntero4 );
 If Puntero4=MaxLineas-1 then
  Memo4.Lines[Puntero4+1]:=' '
 Else
  Memo4.Lines[Puntero4+1]:='>>';
  Inc( Puntero4 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo4.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de insertar líneas en el editor. Si hay más de 200 líneas en el editor, se sobreescribe la primera, a continuación la segunda y así susesivamente.

```
Procedure TTerminal.InsLinea6( Linea:String );
Var M : Integer;
begin
If Memo6.Lines.Count>=MaxLineas then
 begin
 If Puntero6>MaxLineas-1 then Puntero6:=0;
 Memo6.Lines[ Puntero6 ]:=Linea;
 For M:=1 to length( Linea ) do
  If Linea[ M ]=Char( 13 ) then Inc( Puntero6 );
 If Puntero6=MaxLineas-1 then
  Memo6.Lines[Puntero6+1]:=' '
 Else
  Memo6.Lines[Puntero6+1]:='>>';
  Inc( Puntero6 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo6.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de insertar líneas en el editor. Si hay más de 200, se sobreescribe la primera, a continuación la segunda y así susesivamente.

```
Procedure TTerminal.InsLinea8( Linea:String );
Var M : Integer;
begin
If Memo8.Lines.Count>=MaxLineas then
 begin
 If Puntero8>MaxLineas-1 then Puntero8:=0;
 Memo8.Lines[ Puntero8 ]:=Linea;
 For M:=1 to length( Linea ) do
  If Linea[ M ]=Char( 13 ) then Inc( Puntero8 );
 If Puntero8=MaxLineas-1 then
  Memo8.Lines[Puntero8+1]:=' '
 Else
  Memo8.Lines[Puntero8+1]:='>>';
  Inc( Puntero8 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo8.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de insertar líneas en el editor. Si hay más de 200, se sobreescribe la primera, a continuación la segunda y así susesivamente.

```
Procedure TTerminal.InsLinea10( Linea:String );
Var M : Integer;
begin
If Memo10.Lines.Count>=MaxLineas then
 begin
 If Puntero10>MaxLineas-1 then Puntero10:=0;
 Memo10.Lines[ Puntero10 ]:=Linea;
 For M:=1 to length( Linea ) do
  If Linea[ M ]=Char( 13 ) then Inc( Puntero10 );
 If Puntero10=MaxLineas-1 then
  Memo10.Lines[Puntero10+1]:=' '
 Else
  Memo10.Lines[Puntero10+1]:='>>';
  Inc( Puntero10 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo10.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que junto a los anteriores controla la inserción de líneas en los editores.

```
Procedure TTerminal.Miralinea2( Cadena : String );
Var Linea : String;
      N : Integer;
begin
Linea :='';
For N:= 1 to length( Cadena ) do
 begin
 If Cadena[ N ]=Char( 13 ) then
  begin
  Inslinea2( Linea );
  Linea:='';
  end
 Else
  begin
  If Cadena[ N ]<>char( 10 ) then
    linea:=Linea+Cadena[ N ];
  end;
 end;{ For }
If Linea<>'' then Inslinea2( Linea );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que junto a los anteriores controla la inserción de líneas en los editores.

```
Procedure TTerminal.Miralinea4( Cadena : String );
Var Linea : String;
      N : Integer;
begin
Linea :='';
For N:= 1 to length( Cadena ) do
 begin
 If Cadena[ N ]=Char( 13 ) then
  begin
  Inslinea4( Linea );
  Linea:='';
  end
 Else
  begin
  If Cadena[ N ]<>char( 10 ) then
    linea:=Linea+Cadena[ N ];
  end;
 end;{ For }
If Linea<>'' then Inslinea4( Linea );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que junto a los anteriores controla la inserción de líneas en los editores.

```
Procedure TTerminal.Miralinea6( Cadena : String );
Var Linea : String;
      N : Integer;
begin
Linea :='';
For N:= 1 to length( Cadena ) do
 begin
 If Cadena[ N ]=Char( 13 ) then
  begin
  Inslinea6( Linea );
  Linea:='';
  end
 Else
  begin
  If Cadena[ N ]<>char( 10 ) then
    linea:=Linea+Cadena[ N ];
  end;
 end;{ For }
If Linea<>'' then Inslinea6( Linea );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que junto a los anteriores controla la inserción de líneas en los editores.

```
Procedure TTerminal.Miralinea8( Cadena : String );
Var Linea : String;
      N : Integer;
begin
Linea :='';
For N:= 1 to length( Cadena ) do
 begin
 If Cadena[ N ]=Char( 13 ) then
  begin
  Inslinea8( Linea );
  Linea:='';
  end
 Else
  begin
  If Cadena[ N ]<>char( 10 ) then
    linea:=Linea+Cadena[ N ];
  end;
 end;{ For }
If Linea<>'' then Inslinea8( Linea );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que junto a los anteriores controla la inserción de líneas en los editores.

```
Procedure TTerminal.Miralinea10( Cadena : String );
Var Linea : String;
      N : Integer;
begin
Linea :='';
For N:= 1 to length( Cadena ) do
 begin
 If Cadena[ N ]=Char( 13 ) then
  begin
  Inslinea10( Linea );
  Linea:='';
  end
 Else
  begin
  If Cadena[ N ]<>char( 10 ) then
    linea:=Linea+Cadena[ N ];
  end;
 end;{ For }
If Linea<>'' then Inslinea10( Linea );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite seleccionar el valor del bit D

```
procedure TTerminal.BitD01Click(Sender: TObject);
begin
If Cursor = crDefault then
 begin
 If BitD01.Caption='&Bit D=0'then
  begin
  BitD01.Caption:='&Bit D=1';
  IGF:=PonerBitD( IGF );
  end
 Else
  begin
  BitD01.Caption:='&Bit D=0';
  IGF:=QuitarBitD( IGF );
  end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 38 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Limpia el editor de su contenido actual.

```
procedure TTerminal.Limpiapantalla1Click(Sender: TObject);
begin
If Cursor = crDefault then
 begin
 Case Tabset1.Tabindex of
 0:
  begin
  If canal0<>0 then
   begin
   If Not( HayDatosTx ) then
    begin
    Memo1.Clear;
    TxDatos:=DH(128 or 64 or HD(IGF))+DH( Canal0 )+'00'+'CLEAR';
    HayDatosTx:=True;
    end;
   end;
  end;
 1:
  begin
  If Canal1<>0 then
   begin
   If Not( HayDatosTx ) then
    begin
    Memo3.Clear;
    TxDatos:=DH(128 or 64 or HD(IGF))+DH( Canal1 )+'00'+'CLEAR';
    HayDatosTx:=True;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      end;
    end;
   end;
  2:
   begin
   If Canal2<>0 then
   begin
   If Not( HayDatosTx ) then
    begin
    Memo5.Clear;
    TxDatos:=DH(128 or 64 or HD(IGF))+DH( Canal2 )+'00'+'CLEAR';
    HayDatosTx:=True;
    end;
   end;
   end;
  3:
   begin
   If Canal3<>0 then
   begin
   If Not( HayDatosTx ) then
    begin
    Memo7.Clear;
    TxDatos:=DH(128 or 64 or HD(IGF))+DH( Canal3 )+'00'+'CLEAR';
    HayDatosTx:=True;
    end;
   end;
   end;
  4:
   begin
   If Canal4<>0 then
   begin
   If Not( HayDatosTx ) then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
   begin
   Memo9.Clear;
   TxDatos:=DH(128 or 64 or HD(IGF))+DH( Canal4 )+'00'+'CLEAR';
   HayDatosTx:=True;
   end;
  end;
  end;
 end { Case }
end;
If Cursor =crDrag then RedX25.LlamaAyuda( 37 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Lee la configuración en el Registro a partir del archivo cuyo nombre se le pasa como parámetro.

```
Procedure TTerminal.LeerConfig( Archivo : String );
Const ColorCvp : TColor = ClAqua;

Var Cont : Integer;
begin
AssignFile( ArchCfg, Archivo );
{$I-}
Reset( ArchCfg );
{$I+}
If IOresult<>0 then
 begin
 Showmessage( 'No se ha podido abrir el archivo '+Archivo );
 end
Else
 begin
 Read( Archcfg, Parametros );
 CloseFile( ArchCfg );
 TabSet1.Tabs.Clear;
 For Cont:=1 to CuentaCanales( Parametros ) do
  begin
  TabSet1.Tabs.Add(' ');
  end;
 With Parametros do
  begin
  { Si hay canales permanentes }
  For Cont:=1 to 5 do
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 If ( CVPIRDestino[ Cont ]<>'' )and( CVPOrigen[Cont
]<>0)and(CVPDestino[Cont]<>0)then
  begin
  TabSet1.Tabs[CVPOrigen[ Cont ]-100]:= CVPIRDestino[ Cont ]+':'+DH(
CVPOrigen[Cont]);
  TabSet1.TabIndex:=CVPOrigen[Cont]-100;
  TabAnterior:=Tabset1.TabIndex;
  Case TabSet1.TabIndex of
  0:begin
    Memo1.Visible:=True;
    Memo2.Visible:=True;
    Memo1.Color:=ColorCvp;
    Memo2.Color:=ColorCvp;
    Canal0:=CVPOrigen[ Cont ];
    TipoC0:='CVP';
    end;
  1:begin
    Memo3.Visible:=True;
    Memo4.Visible:=True;
    Memo3.Color:=ColorCvp;
    Memo4.Color:=ColorCvp;
    Canal1:=CVPOrigen[ Cont ];
    TipoC1:='CVP';
    end;
  2:begin
    Memo5.Visible:=True;
    Memo6.Visible:=True;
    Memo5.Color:=ColorCvp;
    Memo6.Color:=ColorCvp;
    Canal2:=CVPOrigen[ Cont ];
    TipoC2:='CVP';
    end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    3:begin
    Memo7.Visible:=True;
    Memo8.Visible:=True;
    Memo7.Color:=ColorCvp;
    Memo8.Color:=ColorCvp;
    Canal3:=CVPOrigen[ Cont ];
    TipoC3:='CVP';
    end;
   4:begin
    Memo9.Visible:=True;
    Memo10.Visible:=True;
    Memo9.Color:=ColorCvp;
    Memo10.Color:=ColorCvp;
    Canal4:=CVPOrigen[ Cont ];
    TipoC4:='CVP';
    end;
  end;{ Case }
  end;{ End For }
 {ShowMessage( IntToStr( NCanales ));}
 end;{ End With }
end;
If Parametros.NGCU<>'' then
 begin
 GCU1.Caption:='GCU-'+Parametros.NGCU;
 GCU1.checked:=true;
 end
Else
 begin
  gcu1.Caption:='No GCU';
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que indica cuantos canales tiene el terminal configurados.

```
Function TTerminal.CuentaCanales( Par : CFG ): Integer;
Var P,E,N,S : Integer;
   M : Integer;
begin
P:=0;E:=0;N:=0;S:=0;
With Par do
 begin
 For M:=1 to 5 do
  If CVPOrigen[ M ]<>0 then Inc( P );
 If ( CVCELI<>0 )and( CVCELS<>0 ) then E:=CVCELS-CVCELI+1;
 If ( CVCESLI<>0 )and( CVCESLS<>0 ) then N:=CVCESLS-CVCESLI+1;
 If ( CVCSLI<>0 )and( CVCSLS<>0 ) then S:=CVCSLS-CVCSLI+1;
 CuentaCanales := P+E+N+S;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que permite seleccionar o no Grupo Cerrado de Usuarios si el terminal se configurara con GCU con acceso salida.

```
procedure TTerminal.GCU1Click(Sender: TObject);
begin
{ Seleccion  de llamada con o sin GCU }
{ si hubiese gcu con acceso salida }
If parametros.ngcu<>'' then
{ Si hay grupo cerrado de usuarios }
 begin
 If Parametros.gcuas then
 { Si hay acceso de salida }
  begin
  if gcu1.checked then
   begin
   { Si esta seleccionado quita la seleccion }
   GCU1.Checked:=False;
   end
  Else
   begin
   { Si no esta seleccionado seleccionalo }
   GCU1.Checked:=True;
   end;
  end;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que extrae el campo de datos de una llamada con selección rápida y la muestra en pantalla.

```
Function TTerminal.RxSeleccionRapida( Comando:String ; Var Hora : String ):Boolean;
Var NFac : Integer;
   CadFac : String;
   N    : Integer;
   Hay  : Boolean;
begin
{ Si en un paquete de llamada o acep.llamada hay la facilidad de seleccion rapida}
{ saca la cadena de la seleccion rapida y ponla como la hora que es }
Hay:=False;
NFac :=HD( Copy( Comando,15,2 ));
CadFac:=Copy( Comando, 15,length( Comando )-14 );
RedX25.Visual( Nombre+' '+Comando+' '+CadFac+' '+IntToStr( NFac ) );
For N:=1 to NFac div 2 do
 begin
 If ( Copy( CadFac,2+N+( N-1 )*3,4 )='0180')or
    (Copy( CadFac,2+N+( N-1 )*3,4 )='0181') then
    begin
    Hay:=True;
    RedX25.Visual( Nombre+' Detecta seleccion rapida '+Comando);
    end;
 end; { For }
 If Hay then
  begin
  Hora :=Copy( Comando,17+NFac*2,length( Comando )-(15+NFac*2) );
  RedX25.Visual( Nombre+' '+Hora );
  For N:=1 to length( Hora ) do
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    if Hora[ N ]='A' then Hora[ N ]:=':';
  end;
RxSeleccionRapida:=Hay;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Resumen de la configuración del terminal para mostrarla en el Panel de Estados.

```
Procedure TTerminal.ResumenConfig;
Var M : Integer;
 begin
 RedX25.Visual( '***** '+Nombre );
  With Parametros do
  begin
  RedX25.Visual('***** -------- Nivel 2 --------');
  RedX25.Visual('***** Ventanas de Nivel 2 Tx/Rx '+IntToStr(VN2Tx)+' '
            +IntToStr(VN2Rx));
  RedX25.Visual('***** Temporizadores y Contadores T1 '+IntToStr( T1 )+' T3 '
            +IntToStr( T3 )+' N2 '+IntToStr( N2 ) );
  RedX25.Visual('***** -------- Canales Virtuales Permanentes --------');
  RedX25.Visual('***** Ventanas de N3 para CVP Tx/Rx '+IntToStr( VCVPTX )+' '
            +IntToStr( VCVPRX ));
  For M:=1 to 5 do
   begin
   RedX25.Visual('***** CVP Origen '+IntToStr(CVPOrigen[ M ])+
            ' IR Destino '+CVPIRDestino[ M ]+
            ' CVP Destino '+IntToStr( CVPDestino[ M ] ));
  end;{ fin CVP ocupados }
  RedX25.Visual('***** -------- Canales Virtuales Conmutados -------- ');
  RedX25.Visual('***** Ventanas de N3 para CVC Tx/Rx '+IntToStr( VCVCTX )+' '
            +IntToStr( VCVCRX ));
  RedX25.Visual('***** Limite Inferior de CVC Salientes '+IntToStr( CVCSLI ));
  RedX25.Visual('***** Limite Superior de CVC Salientes '+IntToStr( CVCSLS ));
  RedX25.Visual('***** Limite Inferior de CVC Entrantes '+IntToStr( CVCELI ));
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  RedX25.Visual('***** Limite Superior de CVC Entrantes '+IntToStr( CVCELS ));
  RedX25.Visual('***** Limite Inferior de CVC Entrantes/Salientes '
          +IntToStr( CVCESLI ));
  RedX25.Visual('***** Limite Superior de CVC Entrantes/Salientes '
          +IntToStr( CVCESLS ));
  RedX25.Visual('***** Facilidades de Red ');
  If PLLS then RedX25.Visual('***** Prohibicion de Llamadas Salientes ');
  If PLLE then RedX25.Visual('***** Prohibicion de Llamadas Entrantes ');
  If GCU then RedX25.Visual('***** Grupo Cerrado de Usuarios '+NGCU );
  If GCUAS then RedX25.Visual('***** Grupo Cerrado de Usuarios con Acceso de
Salida' );
  If GCUAE then RedX25.Visual('***** Grupo Cerrado de Usuarios con Acceso de
Entrada' );
  If GCUPLLS then RedX25.Visual('***** Grupo Cerrado de Usuarios con '+
                  'Prohibicion de LLamadas Salientes' );
  If GCUPLLE then RedX25.Visual('***** Grupo Cerrado de Usuarios con'+
                  'Prohibicion de Llamadas Entrantes');
  If ACR then RedX25.Visual('***** Aceptación de Cobro Revertido ');
  If ASR then RedX25.Visual('***** Aceptación de Selección Rápida ');
  If NPCF then RedX25.Visual('***** Negociación de Parámetros de Conbtrol de
Flujo');
  If Sr then RedX25.Visual('***** Selección Rápida ');
  If SCR then RedX25.Visual('***** Selección de Cobro Revertido ');

  RedX25.Visual('*****');
  end;{ End With }

 end;
end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad CCTerm

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

unit CCTerm;

Unidad en la que se implementan los objetos Tcanal y TCCT. El último de ellos es el objeto que constituye la capa de Nivel 3 del Protocolo X25 en el lado del Terminal de usuario. El objeto Tcanal es utilizado dentro del Objeto TCCT para implementar las variables y procedimientos que constituyen cada uno de los canales.

interface

Uses WinTypes,WinProcs,Classes,Controls,Dialogs,SysUtils,
Calculo, CfgNodo ;

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Definición de la Clase Tcanal. A partir de esta clase se crearan los objetos que formarán los canales. Todos los canales descendientes de esta clase se crearan en el objeto TCCT como una matriz.

```
Type TCanal = Class( TObject )
 IR          : String;
 Numero      : Byte;
 Tipo                                : String;
 PR ,PS ,M   : Byte;
 Canal       : Byte;
 VT,VR       : Byte;
 VentN3Tx    : Byte;
 VentN3Rx    : Byte;
 EstadoN3    : String;
 Constructor Create( TipoC : String; Par : CFG );
end; { TCanal }

Type Datos     = String;
    Paquetes  = String;
    TipoCanal = Array[100..104]of TCanal;
    TipoCanalActivo =Array[100..104]of Boolean;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Objeto que forma el controlador de comunicaciones de Nivel 3 del Terminal de usuario. Se comunicará con la capa superior o Terminal de usuario y con la capa inferior o capa de Nivel 2.

```
Type TCCT = Class( TObject )
 Nombre : String;
 Canal : TipoCanal;
 CanalActivo:TipoCanalActivo;
 DatosRx :Datos;
 DatosTx :Datos;
 ComandosRx :Datos;
 ComandosTx :Datos;
 PaqueteTx:Paquetes;
 PaqueteRx:Paquetes;
 HayDatosRx  : Boolean;
 HayDatostx : Boolean;
 HayComandosRx  : Boolean;
 HayComandostx : Boolean;
 HayPaqueteTx : Boolean;
 HayPaqueteRx : Boolean;
 N2Activo :^Boolean;
 N2ActivoAnterior: Boolean;
 IGF : String;
 Parametros : CFG;
 ArchCfg : File of CFG;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

A continuación se enumeran los métodos que constituyen esta Clase.

```
Constructor Create;
Function TxDatos:String;
Procedure RxDatos( RxDatos:String );
Function TxComandos:String;
Procedure RxComandos( RxComandos:String );
Function TxPaquete:String;
Procedure RxPaquete( RxPaquetes : String );
Procedure Control;
Function RRN3( IGFRR:String;CanalRR,Secuencia : Byte ):String;
Function RNRN3( CanalRNR,Secuencia : Byte ): String;
Procedure LeerConfig( Archivo : String );
Function HayCVCS : Boolean;
Function DecideCVCS( Var CSaliente : Byte ) : String ;
Function AceptaCVCE( CEntrante : Byte ) : String;
Function HayNPCF( Paq : String ;Var Vrx ,Vtx : Integer ): Boolean;
Procedure PonVentanaRx( Vr : Integer;Var Paq : String );
Procedure PonVentanaTx( Vt : Integer;Var Paq : String );
Function AceptaFacilidades( Paq : String ; Var Lib : String ):Boolean;
End;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Comienza la implementación del código de los distintos métodos del objeto TCCT y TCanal.

```
implementation

Uses SimX25;

Var CanalAct : Byte;
   CanalActH : String ;
   VNpcftx ,Vnpcfrx : Integer;{ Variables provisionales para extraer los }
                      {tamaños de ventana del paquete de llamada }
                      { en la negociacion de parametros de control}
                      { de flujo }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Al crearse un objeto Tcanal, el procedimiento Create del mismo, reinicia las variables del canal desde el registro de configuración del terminal.

```
Constructor TCanal.Create( TipoC : String; Par : CFG );
begin
TObject.Create;
PR:=0;
PS:=0;
M:=0;
Canal:=0;
 VT:=0;
 VR:=0;
 EstadoN3:='P1';
 IR :='';
 Tipo:=TipoC;
 If Tipo='CVP' then
  begin
   VentN3Tx:=Par.VCVPTx;
   VentN3Rx:=Par.VCVPRx;
  end
 Else
  begin
   VentN3Tx:=Par.VCVCTx;
   VentN3Rx:=Par.VCVCRx;
  end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Estos dos métodos comunican esta capa con la capa superior a nivel de paquetes de datos de usuario. La capa superior la implementa el Terminal de usuario en la Unidad Term1. Estos métodos se ejecutan desde el procedimiento Idle de la Unidad principal SimX25.

```
Function TCCT.TxDatos:String;
Begin
HayDatosTx:=False;
TxDatos:=DatosTx;
DatosTx:='';
end;
Procedure TCCT.RxDatos( RxDatos :String );
begin
DatosRx:=RxDatos;
HayDatosRx:=True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Métodos con los que esta unidad se comunica con la capa superior a nivel de comandos de Nivel 3. Se entienden por comandos todos aquellos paquetes de nivel tres distintos de paquetes de datos y RR.

```
Function TCCT.TxComandos:String;
Begin
HayComandosTx:=False;
TxComandos:=ComandosTx;
ComandosTx:='';
end;
Procedure TCCT.RxComandos( RxComandos :String );
begin
ComandosRx:=RxComandos;
HayComandosRx:=True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Los métodos siguientes permiten el intercambio de paquetes con la capa de Nivel 2. Esta capa envía hacia la superior el contenido de las tramas INFO de Nivel 2 y añade los encabezados necesarios al Paquete enviado por el objeto TCCT para que tenga entidad como trama de Nivel 2.

```
Function TCCT.TxPaquete:String;
Begin
HayPaqueteTx:=False;
TxPaquete:=PaqueteTx;
PaqueteTx:='';
end;
Procedure TCCT.RxPaquete( RxPaquetes : String);
begin
PaqueteRx:=RxPaquetes;
HayPaqueteRx:=True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que se ejecuta desde el procedimiento Idle si hay algún dato, comando o paquete recibido desde cualquiera de los Niveles con los que se conecta este Objeto. Es el método encargado de decidir las acciones que van a constituir el Protocolo X25 en su Capa de Nivel de Red.

```
Procedure TCCT.Control;
{Var N: Integer;}
Var CausaLib : String;
begin
If N2Activo^ then
begin
```

Comenzamos analizando los Comandos recibidos desde el Terminal de Usuario.

```
If HayComandosRx then
 begin
 { Extrae el canal del paquete }
 CanalAct:=HD( Copy(ComandosRx,1,2) );
 CanalActH:=Copy( ComandosRx,1,2 );
 ComandosRx:=Copy( ComandosRx,3 ,Length( ComandosRx )-2 );
 Case HD( Copy( ComandosRx ,1,2 ) ) of
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Si el octeto de control es 11 en decimal, el paquete recibido es una solicitud de llamada cursada desde el terminal con dirección hacia la Red.

```
11:{ Solicitud de llamada }
  begin
  { Si hay canal disponible }
  If HayCVCS then
   begin
   If Not( HayPaqueteTx ) then
    begin
    CausaLib :=DecideCVCS( CanalAct );
     if CausaLib='00' then
      begin
      Canal[ CanalAct ]:=TCanal.Create('CVC',Parametros);
      CanalActivo[ CanalAct ]:=True;
      Canal[ CanalAct ].IR:=Copy( ComandosRx,9,4 );
      Canal[ CanalAct ].EstadoN3:='P2';
      PaqueteTx:='10'+DH( CanalAct )+ ComandosRx;
      HayPaqueteTx:=True;
      ComandosRx:='';
      HayComandosRx:=False;
      RedX25.Visual( Nombre+' Rx Comando LLamada.Envia llamada por el canal
'+IntToStr(CanalAct));
      RedX25.Visual( Nombre+' '+PaqueteTx );
      end
     Else
      begin
      ComandosRx:='';
      HayComandosRx:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      ShowMessage(' Llamada no permitida .No se encuentran canales  '+CausaLib );
      RedX25.Visual( Nombre+' Rx Comando LLamada.El canal indicado no es
configurable');
       end;
      end;
     end
    Else
     begin
     ComandosRx:='';
     HayComandosRx:=False;
     Showmessage(' No hay canales disponibles ' );
     RedX25.Visual( Nombre+' Rx Comando LLamada.No hay canales disponibles ');
     end;
    end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Se recibe una solicitud de liberación desde el Terminal con dirección hacia la Red.

```
19:{ Solicitud de liberacion }
  begin
  if ( CanalAct>99 )and( CanalAct<105 ) then
  If ( CanalActivo[ CanalAct ] )and(Not( HayPaqueteTx ))
                  and(Canal[CanalAct].Tipo<>'CVP' )then
   begin
   {CanalActivo[ CanalAct ]:=False;}
   {Canal[ CanalAct ].Free;}
   { tiene que saber que canal libera }
   PaqueteTx:='10'+CanalActH+ComandosRx;
   HayPaqueteTx:= True;
   ComandosRx:='';
   HayComandosRx:=False;
   Canal[CanalAct].EstadoN3:='P6';
   RedX25.Visual( Nombre+' Rx Comando liberacion.Envia liberacion
'+IntToStr(CanalAct));
   end;
  end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Se recibe un paquete de Solicitud de Reinicio desde el Terminal de Usuario con sentido hacia la Red.

```
27:{ Solicitud de Reinicio }
  begin
  if ( CanalAct>99 )and( CanalAct<105 ) then
  If ( CanalActivo[ CanalAct ] )and( Not( HayPaqueteTx )) then
   begin
   PaqueteTx:='10'+CanalActH+DH( 27 )+'0000';
   HayPaqueteTx:=True;
   ComandosRx:='';
   HayComandosRx:=False;
   Canal[ CanalAct ].EstadoN3:='D2';
   RedX25.Visual( Nombre+' Rx Comando Reinicio.Envia reinicio por el canal
'+IntToStr(CanalAct));
   Canal[ CanalAct ].Pr:=0;
   Canal[ CanalAct ].Ps:=0;
   Canal[ CanalAct ].Vt:=0;
   Canal[ CanalAct ].Vr:=0;
   end;
  end;
```

Se recibe una Solicitud de Rearranque desde el Terminal de Usuario para ser cursada hacia la Red.

```
251:{ Solicitud de Rearranque }
  begin
  if ( CanalAct=0 ) then
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  { Si hay al menos un canal activo }
  If (( CanalActivo[ 100 ] )or
     ( CanalActivo[ 101 ] )or
     ( CanalActivo[ 102 ] )or
     ( CanalActivo[ 103 ] )or
     ( CanalActivo[ 104 ] ) )And( Not( HayPaqueteTx )) then
   begin
   PaqueteTx:='10'+'00'+DH( 251 )+'0000';
   HayPaqueteTx:=True;
   ComandosRx:='';
   HayComandosRx:=False;
   RedX25.Visual( Nombre+' Rx Comando Rearranque.Envia Rearranque  ');
   end;
  end;
 end;{ End Case }
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

A continuación comienza el estudio de los datos de usuarios recibidos desde el Terminal de usuario para ser enviados hacia la Red.

```
If ( HayDatosRx )and( CanalActivo[HD(Copy( DatosRx,3,2 ))] ) then
 begin
 If Not( HayPaqueteTx )and
( Canal[HD(Copy( DatosRx,3,2 ))].Vt<Canal[HD(Copy( DatosRx,3,2
))].VentN3tx)then
  begin
  IGF:=Copy( DatosRx,1,2);
  CanalAct:=HD(Copy( DatosRx,3,2 ));
  CanalActH:=Copy( DatosRx,3,2 );
  If Copy( DatosRx,5, 2 )='00' then
    Canal[ CanalAct ].M:=0
  Else
    Canal[ CanalAct ].M:=1;
  DatosRx:=Copy( DatosRx,7,length( DatosRx )-6 );
  HayPaquetetx:=True;
  Paquetetx:=IGF+CanalActH+(PrMPsDH( Canal[CanalAct].Pr,
                    Canal[CanalAct].M,
                    Canal[CanalAct].Ps ))+DatosRx ;
  if Canal[CanalAct].Ps < 7
      {Canal[CanalAct].VentN3Tx-1} then
      Canal[CanalAct].Ps:=Canal[CanalAct].Ps+1
  Else Canal[ CanalAct ].Ps:=0;
  If VerBitD( IGF ) then
  { Si el bit D esta a uno espera un RR con bit D=1 }
  Canal[ CanalAct ].Vt:=Canal[ CanalAct ].VentN3Tx
  Else
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 Inc( Canal[ CanalAct ].Vt ); {Espero el siguiente en la secuencia }
 { Si se llega al limite de la ventana espera un RR }
 Canal[ CanalAct ].Vr:=0;{ Estoy en el Humbral de la ventana de Rx }
 HayDatosRx:=False;
 DatosRx:='';
 RedX25.Visual( Nombre+' Rx Datos.Envia Paquete de datos por el canal
'+IntToStr(CanalAct));
 end;
end;
If ( HayDatosRx )and( Not( CanalActivo[HD(Copy( DatosRx,3,2 ))] ) ) then
 begin
 DatosRx:='';
 HayDatosRx:=False;
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Comienza el estudio de los Paquetes recibidos desde la Red con destino el Terminal de Usuario.

```
If HayPaqueteRx then
 begin
 { Extrae el canal del paquete }
 CanalAct:=HD(Copy( PaqueteRx,3,2 ));
 CanalActH:= Copy( PaqueteRx,3,2 );
```

Si el paquete recibido es un Paquete de Datos de Usuario de Nivel 3 comprueba que es correcto, contesta con RR si fuera necesario y se pasa al Nivel superior o Terminal de usuario.

```
 If  HD( Copy( PaqueteRx,5,2)) mod 2=0 then
  begin
  { Es un paquete de datos }
  { Comprobar que es el que se esperaba }
  { Comprobar que es un canal valido }
  If CanalActivo[ CanalAct ] then
   begin
   If PsHD( Copy( PaqueteRx,5,2 ))=Canal[CanalAct ].Pr then { Si es el que se esperaba
}
     begin
     DatosTx:=PaqueteRx;
     HayDatosTx:=True;
     if Canal[ CanalAct ].Pr< 7
          {Canal[ CanalAct ].VentN3Rx-1}
    then Canal[ CanalAct ].Pr:=Canal[ CanalAct ].Pr+1
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  Else Canal[ CanalAct ].Pr:=0;
  Canal[ CanalAct ].Vt:=0;
    { Si el bit D esta a uno }
  If VerBitD(Copy(PaqueteRx,1,2)) then
    begin
     If Not( HayPaqueteTx ) then
      begin
      PaqueteTx:=RRN3('50',CanalAct,Canal[CanalAct].Pr );
      HayPaqueteTx:=True;
      Canal[ CanalAct ].Vr:=0;{ Actualiza la ventana de recepcion }
      RedX25.Visual( Nombre+' Rx paquete de datos con P=1.Tx RR por el canal
'+IntToStr(CanalAct));
      end;
    end
   Else{ Si el bit D no esta a uno }
    If Canal[ CanalAct ].Vr<
          Canal[ CanalAct ].VentN3Rx-1 then { Si no ha llegado al limite de la ventana}
     begin
     Inc( Canal[ CanalAct ].Vr );
     RedX25.Visual( Nombre+' Rx paquete de datos con P=0.Esta dentro de ventana
'+IntToStr(CanalAct));
     end
    Else
      begin { Si ha llegado al limite de la ventana }
      If Not( HayPaqueteTx ) then
        begin
        PaqueteTx:=RRN3('10',CanalAct,Canal[ CanalAct ].Pr);
        HayPaqueteTX:=TRue;
        Canal[ CanalAct ].Vr:=0;{ Actualiza la ventana de recepcion }
        RedX25.Visual( Nombre+' Rx paquete de datos con P=0.'
            +'Se cumple la ventana yTx un RR '+IntToStr(CanalAct));
        end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        end;
      end { fin de 'es el que se esperaba '}
    Else
      begin
      { Si no es el que se espera reinicia el canal }
      RedX25.Visual(Nombre+' ERROR.Rx de Paquete de la Red fuera de secuencia ')
      end;
    end
   Else
    begin
    { LLega un paquete de datos por un canal inexistente }
    { Se ignora }
    HayPaqueteRx:=False;
    PaqueteRx:='';
    RedX25.Visual(Nombre+' ERROR.Se recibe un paquete de Datos por un canal no
establecido ');
    end;
   end
 Else
  begin
  { Es otra cosa }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Aqui finaliza el análisis del paquete de datos de usuario de Nivel 3 y comienza el análisis de RR de Nivel 3.

```
If HD('0'+ Copy( PaqueteRx,6,1))=1 then
 begin
 { Es un RR }
 { Actualizar humbral de ventana de Tx }
 Canal[ CanalAct ].Vt:=0;
 RedX25.Visual( Nombre+' Rx paquete RR .Actualiza la ventana
'+IntToStr(CanalAct));
 end;
If HD('0'+ Copy( PaqueteRx,6,1 ))=5 then
 begin
 { Es un RNR }
 { Parar la tx hasta que se reciba un RR }
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Se recibe una solicitud de llamada por un canal. Se comprueba si el canal es válido y si las facilidades son aptas. Se decide si se acepta la llamada o se libera en cualquiera de los casos, se contesta con el paquete pertinente a la capa inferior y, si procede, se avisa a la superior de la recepción de la solicitud de llamada y por tanto, de su aceptación.

```
If Copy( PaqueteRx,5,2 )='0B' then
 begin
 { Es una solicitud de llamada }
 {Comprueba si el canal es valido y esta libre }
 If ( CanalAct >=100 )and( CanalAct  <=104 ) then
 {Si la solicitud es por un canal no valido no hace nada }
  begin
  CausaLib:=AceptaCVCE( CanalAct );
  if  CausaLib='00' then
  begin
  { Si es asi envia una aceptacion de la llamada }
  If AceptaFacilidades( PaqueteRx ,CausaLib ) then
   begin
   If Not( HayPaqueteTx ) then
    begin
    Canal[ CanalAct ]:=TCanal.Create('CVC',Parametros );
    CanalActivo[ CanalAct ]:=True;
    Canal[ CanalAct ].IR:=Copy( PaqueteRx,9,4);
    If HayNPCF( PaqueteRx , Vnpcfrx, Vnpcftx ) then
     begin
     If Canal[ CanalAct].VentN3tx > Vnpcfrx then
      begin
      { Si la ventana de Rx asignada a el canal es mayor que la que}
      { solicita el remoto utiliza la del remoto }
      Canal[ CanalAct ].VentN3tx := Vnpcfrx;
      RedX25.Visual( Nombre+' Negociacion de ventana de tx,toma la del Etd remoto');
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        end;
       If Canal[ CanalAct].VentN3tx < Vnpcfrx then
        begin
        { Si la ventana de Rx asignada a el canal es menor que la }
        { que solicita el remoto ,utiliza la asignada y pon este }
        { valor en el paquete de aceptacion de llamada }
        PonVentanaRx( Canal[ CanalAct].VentN3tx, PaqueteRx );
        RedX25.Visual( Nombre+' Negociacion de ventana de Tx,toma la del Etd local');
        end;
       If Canal[ CanalAct].VentN3Rx > Vnpcftx then
        begin
        { Si la ventana de Tx asignada a el canal es mayor que la que}
        { solicita el remoto utiliza la del remoto }
        Canal[ CanalAct ].VentN3Rx :=  Vnpcftx;
        RedX25.Visual( Nombre+' Negociacion de ventana de Rx,toma la del Etd
remoto');
        end;
       If Canal[ CanalAct].VentN3Rx < Vnpcftx then
        begin
        { Si la ventana de Rx asignada a el canal es menor que la }
        { que solicita el remoto ,utiliza la asignada y pon este }
        { valor en el paquete de aceptacion de llamada }
        PonVentanaTx( Canal[ CanalAct].VentN3Rx, PaqueteRx );
        RedX25.Visual( Nombre+' Negociacion de ventana de Rx,toma la del Etd local');
        end;
       end;
      ComandosTx:=Copy( PaqueteRx,3,length( PaqueteRx )-2 );
      HayComandosTx:=True;
      Canal[ CanalAct ].EstadoN3:='P4';
      PaqueteTx:='10'+CanalActH+'0F'+Copy( PaqueteRx,7,length( PaqueteRx )-6 );
      HayPaqueteTx:=True;
      PaqueteRx:='';
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
{CanalActivo[ CanalAct ]:=True;}
RedX25.Visual( Nombre+' Rx paquete Llamada.Si el canal esta libre Tx una
aceptacion '+IntToStr(CanalAct));
end;
end
Else
begin
{ Si no se aceptan las facilidades libera }
If Not( HayPaqueteTx ) then
begin
Canal[ CanalAct ].EstadoN3:='P6';
PaqueteTx:='10'+CanalActH+'13'+CausaLib;
HayPaqueteTx:=True;
HayPaqueteRx:=False;
PaqueteRx:='';
RedX25.Visual( Nombre+' Rx paquete Llamada.Si la facilidad no es valida'
                    +' Tx una liberacion ');
end
end;
end
Else
Begin
{ Si no es asi envia una liberacion }
If ( CausaLib<>'') then
begin
If Not( HayPaqueteTx ) then
begin
if causalib<>'1322' then
begin
Canal[ CanalAct ].EstadoN3:='P6';
end
Else
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
        begin
        RedX25.Visual(Nombre+' ERROR en la configuración de canales salientes');
        end;

      PaqueteTx:='10'+CanalActH+'13'+CausaLib;
      HayPaqueteTx:=True;
      HayPaqueteRx:=False;
      PaqueteRx:='';
      RedX25.Visual( Nombre+' Rx paquete Llamada.Si el canal no esta libre Tx  una
liberacion '+IntToStr(CanalAct));
       end;
      end
     Else
      begin
      { Ignora el paquete de llamada }
      HayPaqueteRx:=False;
      PaqueteRx:='';
      end;
     end;
    end
   Else
    begin
    { Si no es por un canal entre el 100 y el 104 ignoralo}
    HayPaqueteRx:=False;
    PaqueteRx:='';
    end;
   end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Se recibe un Paquete de Aceptación de Llamada como respuesta a un paquete de Solicitud de llamada. Se comprueba que es válido y se pasa el canal al estado de transferencia de información informando a la capa superior.

```
If ( Copy( PaqueteRx,5,2 )='0F' )
    and( Canal[HD( Copy( PaqueteRX,3,2))].EstadoN3='P2' )then
begin
{ Es una  llamada aceptada}
CanalActivo[ CanalAct ]:=True;
Canal[ CanalAct ].EstadoN3:='P4';
If HayNPCF( PaqueteRx , Vnpcfrx, Vnpcftx ) then
  begin
  Canal[ CanalAct ].VentN3Rx:=Vnpcfrx;
  Canal[ CanalAct ].VentN3Tx:=Vnpcftx;
  RedX25.Visual( Nombre+' Modificada ventana por NPCF '+IntToStr(Vnpcfrx)
  +' '+IntTostr( VnpcfTx ));
  end;
ComandosTx:=Copy( PaqueteRx,3,length( PaqueteRx )-2 );
HayComandosTx:=True;
RedX25.Visual( Nombre+' Rx paquete Aceptacion.Pasalo al Terminal
'+IntToStr(CanalAct));
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Se recibe un paquete de Solicitud de liberación. Comprueba que es válido, procede a liberar el objeto Tcanal pertinente e informa a la unidad superior a la vez que envía hacia la Red una aceptación de liberación.

```
If Copy( PaqueteRx,5,2 )='13' then
 begin
 { Es una solicitud de liberacion }
 { Comprobar si el canal esta activo y desactivarlo }
 If ( CanalActivo[ CanalAct ] )and( Canal[ CanalAct ].Tipo<>'CVP') then
  begin
  if not( HayPaqueteTx ) then
   begin
    CanalActivo[ CanalAct ]:=False;
    Canal[ CanalAct ].Free;
    ComandosTx:=Copy(PaqueteRx,3,10);
    HayComandosTx:=True;
   { Enviar una confirmacion de liberacion }
    { Si no ha habido confirmacion de llamada toma la }
    {solicitud de liberacion como respuesta a la de llamada }
    { y no responde con una aceptacion de liberacion }
   PaqueteTx:='10'+CanalActH+'17';
   HayPaqueteTx:=True;
   Canal[ CanalAct ].EstadoN3:='P1';
   RedX25.Visual( Nombre+' Rx paquete Liberacion.Pasalo al terminal '
   +',Tx una aceptacion y libera el canal'+IntToStr(CanalAct));
  end;
 end;
end;
If ( Copy( PaqueteRx,5,2 )='17')and( Canal[ CanalAct ].EstadoN3='P6') then
 begin
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
If ( Not( HayPaqueteTx ) )and(Canal[ CanalAct ].Tipo<>'CVP') then
 begin
  ComandosTx:=Copy( PaqueteRx,3,4 );
  HayComandosTx:=True;
  Canal[ CanalAct ].Free;
  CanalActivo[ CanalAct ]:=False;
 { Es una confirmacion de liberacion }
 RedX25.Visual( Nombre+' Rx confirmacion de liberacion.Pasalo al Terminal y libera
el Canal '+IntToStr(CanalAct));
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Si se recibe un paquete de Solicitud de Reinicio por un canal válido, responde con una aceptación, informa al Nivel superior y pone todas las variables de ventana del canal correspondiente a cero.

```
If Copy( PaqueteRx,5,2 )='1B' then
   begin
   { Es un Reinicio }
   { Pon las variables del canal a cero }
   { Envia el comando al Terminal }
   {y envia una confirmacion de reinicio }
   If CanalActivo[ CanalAct ] then
    begin
    With Canal[ CanalAct ] do
     begin
     Pr:=0;
     Ps:=0;
     Vt:=0;
     Vr:=0;
     EstadoN3:='P4';
     end;{ End with }
    ComandosTx:=copy( PaqueteRx ,3,8 );
    HayComandosTx:=True;
    PaqueteTx:='10'+CanalActH+'1F';
    HayPaqueteTx:=True;
    RedX25.Visual( Nombre+' Rx Sol.Reinicio.Lo pasa al terminal.'
    +'Pone las  ventanas a cero y responde con una aceptacion'+IntToStr(CanalAct));
    end;
   end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Si se recibe una confirmación de Reinicio como respuesta a una Solicitud, se ponen las variables de ventana del Canal correspondiente a cero.

```
If Copy( PaqueteRx,5,2 )='1F' then
 begin
 { Es una Confirmacion de reinicio  }
  With Canal[ CanalAct ] do
   begin
   Pr:=0;
   Ps:=0;
   Vt:=0;
   Vr:=0;
   EstadoN3:='P4';
   end;{ End with }
  ComandosTx:=copy( PaqueteRx, 3,4 );
  HayComandosTx:=True;
  RedX25.Visual( Nombre+' Rx Conf de reinicio.Lo pasa al terminal.Pone las ventanas
a cero '+IntToStr(CanalAct));
  end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Se recibe una solicitud de Rearranque. Si hay algún canal establecido se libera o reinicia dependiendo si es conmutado o permanente, se contesta al terminal remoto con una Aceptación de Rearranque. Informa al Nivel superior. Esta situación sería anómala ya que la Red no progresaría un Rearranque del Terminal Remoto sino una Liberación por cada uno de sus canales Conmutados y un Reinicio por sus Permanentes. Sólo se recibe un Rearranque por inicio de Nivel 3.

```
If Copy( PaqueteRx,5,2 )='FB' then
 begin
 { Es un Rearranque }
 if ( CanalAct=0 ) then
  begin
 { Si no hay canal activo es un rearranque por inicio de nivel tres }
  If not( HayPaqueteTx )then
   begin
   PaqueteTx:='10'+'00'+DH( 251 )+'0000';
   HayPaqueteTx:=True;
   HayPaqueteRx:=False;
   RedX25.Visual( Nombre+' Rx Sol.Rearranque.Rearranque de sesion.'
   +'Responde con Sol de Rearranque ');
   end;
 end;
{ Si hay al menos un canal activo }
If (( CanalActivo[ 100 ] )or
    ( CanalActivo[ 101 ] )or
    ( CanalActivo[ 102 ] )or
    ( CanalActivo[ 103 ] )or
    ( CanalActivo[ 104 ] ) )And( Not( HayPaqueteTx )) then
 begin
  PaqueteTx:='10'+'00'+DH( 255 );
  HayPaqueteTx:=True;
  ComandosRx:='';
  HayComandosRx:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 RedX25.Visual( Nombre+' Rx Sol.Rearranque.La pasa al terminal.'
+'Libera todos los canales y responde con una Aceptacion ');
 { Libera todos los canales }
 For CanalAct:=100 to 104 do
  begin
  If ( CanalActivo[ CanalAct ])and( Canal[ CanalAct ].Tipo<>'CVP' ) then
   begin
   Canal[ CanalAct ].Free;
   CanalActivo[ CanalAct ]:=False;
   RedX25.Visual( Nombre+' Libera canal '+IntToStr(CanalAct));
   end;
  end;
 ComandosTx:=copy( PaqueteRx, 3,8 );
 HayComandosTx:=True;
end;
 ComandosTx:=copy( PaqueteRx, 3,8 );
 HayComandosTx:=True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Contestación de Aceptación de Rearranque de la Red a una Solicitud de Rearranque hecha por el Terminal de Usuario y progresada por esta capa de Nivel 3. Si hay algún canal conmutado se libera y si hay permanentes se reinician, también se informa al Terminal de usuario.

```
If Copy( PaqueteRx,5,2 )='FF' then
 begin
 { Es una Confirmacion de Rearranque }
 if ( CanalAct=0 ) then
 { Si hay al menos un canal activo }
 If (( CanalActivo[ 100 ] )or( CanalActivo[ 101 ] )or
    ( CanalActivo[ 102 ] )or( CanalActivo[ 103 ] )or
    ( CanalActivo[ 104 ] ) )And( Not( HayPaqueteTx )) then
  begin
  { Libera todos los canales }
  RedX25.Visual( Nombre+' Rx Conf.Rearranque.La pasa al terminal.'
  +'Libera todos los canales y responde con una Aceptacion ');
  For CanalAct:=100 to 104 do
   begin
   If ( CanalActivo[ CanalAct ] )and( Canal[ CanalAct ].Tipo<>'CVP' ) then
    begin
    Canal[ CanalAct ].Free;
    CanalActivo[ CanalAct ]:=False;
    RedX25.Visual( Nombre+' Libera el canal '+IntToStr( CanalAct ));
   end;
   end;
  end;
  ComandosTx:=copy( PaqueteRx, 3,4 );
  HayComandosTx:=True;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
{ Si el paquete es un RR o un Data actua }
{Si el paquete es otra trama pasala al terminal}
HayPaqueteRx:=False;
PaqueteRx:='';
end;
end;{fin de If N2Activo then }
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que inicia las variables del objeto TCCT a la hora de crearse éste.

```
Constructor TCCT.Create;
Var N: Integer;
begin
TObject.Create;
DatosRx :='';
DatosTx :='';
ComandosRx :='';
ComandosTx :='';
PaqueteTx:='';
PaqueteRx:='';
HayDatosRx:=False;
HayDatostx :=False;
HayComandosRx:=False;
HayComandostx :=False;
HayPaqueteTx :=False;
HayPaqueteRx :=False;
IGF :='10';
For n:=100 to 104 do
 begin
 CanalActivo[ n ]:=False;
end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Métodos que crean las cadenas de texto que forman paquetes RR y RNR de Nivel 3.

```
Function TCCT.RRN3( IGFRR:String;CanalRR,Secuencia : Byte ):String;
begin
 RRN3:=IGFRR+DH(CanalRR)+PrRRDH( Secuencia,0,1 );
end;
Function TCCT.RNRN3( CanalRNR,Secuencia : Byte ): String;
begin
 RNRN3:=IGF+DH(CanalRNR)+PrRRDH( Secuencia,0,1 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que lee la configuración del Terminal del Archivo de Configuración y actualiza las variables de configuración del Nivel 3.

```
Procedure TCCT.LeerConfig( Archivo : String );
Var Cont : Integer;
begin
AssignFile( ArchCfg, Archivo );
{$I-}
Reset( ArchCfg );
{$I+}
If IoResult<>0 then{ Si hay error en la apertura del archivo }
 ShowMessage( ' No se ha podido abrir el Archivo '+Archivo )
ELse
 begin
 Read( ArchCfg,Parametros );
 CloseFile( Archcfg );
  With Parametros do
  begin
  { Si hay canales permanentes }
  For Cont:=1 to 5 do
  If ( CVPIRDestino[ Cont ]<>" )and( CVPOrigen[Cont
]<>0)and(CVPDestino[Cont]<>0)then
   begin
   Canal[ CVPOrigen[ Cont ] ]:=TCanal.Create('CVP',Parametros );
   CanalActivo[ CVPOrigen[ Cont ] ]:= TRue;
   Canal[ CVPOrigen[ Cont ] ].Ir:=CVPIRDestino[ Cont ];
   Canal[ CVPOrigen[ Cont ] ].EstadoN3:='P4';
   end;{ Fin de Si hay canales permanentes}
  end;{ With }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que interroga si hay Canales Virtuales Conmutados disponibles para una llamada saliente.

```
Function TCCT.HayCVCS : Boolean;
Var N : Integer;
   Hay : Boolean;
begin
Hay :=False;
{ Mira primero los canales virtuales salientes disponibles }
If ( Parametros.CVCSLI<>0 )and( Parametros.CVCSLS<>0 )then
For N:=Parametros.CVCSLI to Parametros.CVCSLS do
 begin
 If Not( CanalActivo[ N ] ) then Hay:=true;
 end;
{ Despues mira los canales normales disponibles }
If ( Parametros.CVCESLI<>0 )and( Parametros.CVCESLS<>0 )then
For N:=Parametros.CVCESLI to Parametros.CVCESLS do
 begin
 If Not( CanalActivo[ N ] ) then Hay:=True;
 end;
HayCVCS:=Hay;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que decide cual de los Canales Virtuales Conmutados Salientes o Entrantes/Salientes se toma para cursar la llamada.

```
Function TCCT.DecideCVCS( Var CSaliente : Byte ) : String ;
Var Hay : Boolean;
   N : Integer;
begin
Hay :=False;
If Not( Parametros.PllS ) then
 begin
 { Mira primero los canales virtuales salientes disponibles }
 If ( Parametros.CVCSLI<>0 )and( Parametros.CVCSLS<>0 )then
 For N:=Parametros.CVCSLI to Parametros.CVCSLS do
  begin
  If Not( CanalActivo[ N ] ) then
   begin
   CSaliente:=N;
   Hay:=true;
   DecideCVCS:='00';
   end;
  end;
 If Not( Hay ) then
  begin
  { Despues mira los canales normales disponibles }
  If ( Parametros.CVCESLI<>0 )and( Parametros.CVCESLS<>0 )then
   For N:=Parametros.CvcESLI to Parametros.CVCESLS do
    begin
    If Not( CanalActivo[ N ] ) then
     begin
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    CSaliente:=N;
    Hay:=True;
    DecideCVCS:='00';
    end;
   end;
 end;{ If No( Hay )}
If NOt( Hay ) then
 begin
 If (( Parametros.CVCESLI<>0 )and( Parametros.CVCESLS<>0 ))or
    (( Parametros.CVCSLI<>0 )and( Parametros.CVCSLS<>0 )) then
     begin
     CSaliente:=0;
     DecideCVCS:='01'
     end
    Else
     begin
     CSaliente:=0;
     DecideCVCS:='21';
     end;
  end;
 end
Else
 begin
 CSaliente:=0;
 DecideCVCS:='0B';
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que estudia si es posible aceptar un canal indicado en laSolicitud de Llamada Entrante.

```
Function TCCT.AceptaCVCE( CEntrante : Byte ) : String;
Var N: Integer;
   Resultado : String;
begin
{ Mira si no se hace la llamada por un CVP}
Resultado:='';
For N:=1 to 5 do
 begin
 { Libera por error de procedimiento local hace llamadas por un CVP }
 If Parametros.CVPOrigen[ N ]=CEntrante then ACeptaCVCE :='1323';
 end;
If Not( Parametros.PLLE ) then{ Si no hay prohibicion de llamadas entrantes }
 begin
 If ( Parametros.CVCESLI<>0 )and( Parametros.CVCESLS<>0 ) then
  begin
  If ( CEntrante >= Parametros.CVCESLI )and
    ( CEntrante <= Parametros.CVCESLS )then
   begin
   { El canal por el que se hace la llamada es salientey esta libre}
   If ( Not( CanalActivo[ CEntrante ] )) then
      Resultado:='00'
   Else
      Resultado:='0147';
   end;
  end;
 If ( Parametros.CVCELI<>0 )and( Parametros.CVCELS<>0 ) then
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
begin
If ( CEntrante >= Parametros.CVCELI )and
  ( CEntrante <= Parametros.CVCELS )then
  begin
  If ( Not( CanalActivo[ CEntrante ] )) then
    Resultado:='00'
  Else
    Resultado:='0147';
  end;
end;
If Resultado<>'00' then
 begin
 If ( Parametros.CVCSLI<>0 )and( Parametros.CVCSLS<>0 ) then
  begin
  If ( CEntrante >= Parametros.CVCSLI )and
    ( CEntrante <= Parametros.CVCSLS )then
   begin
   { Se pretende hacer una llamada por un canal Saliente }
   Resultado:='1322';
   end;
  end;
 end;
end
Else
 begin
 { Prohibicion de llamadas salientes .se libera causa 0B Acceso prohibido }
 Resultado:='0B47';
 end;
 AceptaCVCE:=Resultado;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que estudia si hay Negociación de los Parametros de Control de Flujo en la Llamada Entrante.

```
Function TCCT.HayNPCF( Paq : String ;Var Vrx ,Vtx : Integer ): Boolean;
Var N :Integer;
   NFac : Integer;
   adFac : Integer;
   CadFac : String;
begin
NFac:=HD( Copy( Paq,17,2 ) );
CadFac:=Copy( Paq, 17,length( Paq )-16 );
HayNpcf:=False;
For N:=1 to NFac div 2 do
 begin
 If Copy( CadFac,2+N+( N-1 )*3,2 )='43' then
  begin
   Vrx:=HD( Copy( CadFac, 4+N+( N-1)*3,2 ));
   Vtx:=HD( Copy( CadFac, 6+N+( N-1)*3,2 ));
   HayNPCF:=true;
  end;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que modifica el valor de la ventana de recepción en el paquete de Aceptación de llamada tras una Negociación de los Parametros de Control de Flujo.

```
Procedure TCCT.PonVentanaRx( Vr : Integer ;Var Paq : String );

Var Numero  : String;
    NFac    : Integer;
    CadFac  : String;
    PaqLlam : String;
    N       : Integer;
begin
PaqLlam :=Copy( Paq, 1,16 );
NFac:=HD( Copy( Paq,17,2 ) );
CadFac:=Copy( Paq, 17,length( Paq )-16 );
For N:=1 to NFac div 2 do
 begin
 If Copy( CadFac,2+N+( N-1 )*3,2 )='43' then
  begin
   Numero := IntToStr( Vr );
   CadFac[5+N+( N-1 )*3 ]:=Numero[ 1 ];
  end;
 end;
 Paq:=PaqLlam+CadFac;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que modifica el valor de la ventana de transmisión en el paquete de Aceptación llamada tras una Negociación de los Parametros de Control de Flujo.

```
Procedure TCCT.PonVentanaTx( Vt : Integer ;Var Paq : String );

Var Numero  : String;
   NFac    : Integer;
   CadFac  : String;
   PaqLlam : String;
   N       : Integer;
begin
PaqLlam :=Copy( Paq, 1,16 );
NFac:=HD( Copy( Paq,17,2 ) );
CadFac:=Copy( Paq, 17,length( Paq )-16 );
For N:=1 to NFac div 2 do
 begin
 If Copy( CadFac,2+N+( N-1 )*3,2 )='43' then
  begin
   Numero := IntToStr( Vt );
   CadFac[7+N+( N-1 )*3 ]:=Numero[ 1 ];
  end;
 end;
 Paq:=PaqLlam+CadFac;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que comprueba si las facilidades que se solicitan en el paquete de Solicitud llamada Entrante son válidos. En caso de que no sean válidos, modifica el valor de la variable Lib para indicar en ella la causa de la Liberación.

```
Function TCCT.AceptaFacilidades( Paq : String ; Var Lib : String ):Boolean ;
Var Numero  : String;
    NFac    : Integer;
    CadFac  : String;
    PaqLlam : String;
    N       : Integer;
begin
AceptaFacilidades:=True;
PaqLlam :=Copy( Paq, 1,16 );
NFac:=HD( Copy( Paq,17,2 ) );
CadFac:=Copy( Paq, 17,length( Paq )-16 );
For N:=1 to NFac div 2 do
 begin
 If Copy( CadFac,2+N+( N-1 )*3,2 )='43' then
  begin
  If Parametros.NPCF then
   begin
   AceptaFacilidades:=True;
   end
  Else
   begin
   AceptaFacilidades:=False;
   Lib:='0300';
   end;
  end;
If Copy( CadFac,2+N+( N-1 )*3,4 )='0180' then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 begin
 If Parametros.ASR then
  begin
  AceptaFacilidades:=True;
  end
 Else
  begin
  AceptaFacilidades:=False;
  Lib:='0000';
  end;
 end;
If Copy( CadFac,2+N+( N-1 )*3,4 )='0181' then
 begin
 If ( Parametros.ASR )and( Parametros.ACR ) then
  begin
  AceptaFacilidades:=True;
  end
 Else
  begin
  AceptaFacilidades:=False;
  Lib:='0000';
  end;
 end;
If ( Copy( CadFac,2+N+( N-1 )*3,4 )='0101')and
  ( Copy( CadFac,N+( N-1 )*3,2 )<>'43' ) then
 begin
 If Parametros.ACR then
  begin
  AceptaFacilidades:=True;
  end
 Else
  begin
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  AceptaFacilidades:=False;
  Lib:='0000';
  end;
 end;
end;
end;
end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad CCTerm2

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

unit CCTerm2;

Unidad CCTerm2. Implementa la capa de Nivel 2 del protocolo X25. Crea el objeto TCCN2.

interface

```
Uses  WinTypes,WinProcs,Classes,Controls,Dialogs,SysUtils,
    Calculo,
    Cfgnodo ;

Type TCCN2 = Class( TObject )
 Nombre : String;
 Estado : String;
 DirLocal : String;
 DirRemota : String;
 DirActual : String;
 T1Activo  : Boolean;
 T1Inicial : LongInt;
 T1        : Integer;
 T3Activo  : Boolean;
 T3Inicial : Real;
 T3        : Integer;
 N2        : Integer;
 ContN2    : Integer;

 Nr,P,Ns   : Byte;
 Vt,Vr     : Byte;
 VentN2Tx  : Byte;
 VentN2Rx  : Byte;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
TramasRx  :String;
TramasTx :String;
PaqueteTx:String;
PaqueteRx:String;

HayTramasRx  : Boolean;
HayTramastx : Boolean;
HayPaqueteTx : Boolean;
HayPaqueteRx : Boolean;
N2Activo : Boolean;

Parametros : CFG;
ArchCfg    : File of CFG;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

A continuación se enumeran los métodos pertenecientes al objeto TCCN2.

```
Constructor Create;
Function TxTramas:String;
Procedure RxTramas( RxTramas:String );
Function TxPaquete:String;
Procedure RxPaquete( RxPaquetes : String );
Procedure Control;
Procedure DisparaT1;
Function FinalT1 : Boolean;
Procedure DisparaT3;
Function FinalT3 : Boolean;
Function SABM( Direc : String; P:Boolean ): String;
Function UA( Direc : String; P:Boolean ): String;
Function Disc( Direc : String; P:Boolean ) : String;
Function DM( Direc : String; P:Boolean ):String;
Function REJ( Direc : String; P:Boolean; NRX : Integer ):String;
Function FRMR( Direc : String; P:Boolean ):String;
Function RR( Direc : String; P:Boolean;NRX : Integer ):String;
Function Info( Direc: String;NRx,Px,NSx:Byte;Paq:String ):String;
Function TipoTrama( Trama : String ): String;
Function BitP( Trama : String ):Boolean;
Function ExtraePaq( Trama : String ): String;
Function Rearranque : String;
Procedure LeerConfig( Archivo : String );
End;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Comienza la Implementación de los métodos del Objeto TCCN2.

```
implementation

Uses SimX25;

Constructor TCCN2.Create;
 begin
 TObject.Create;
 Estado:='E1';
 T1Activo:=False;
 T3Activo:=False;
 T1:=3;
 T3:=10;
 ContN2:=0;
 N2:=5;
 Nr:=0;
 P:=0;
 Ns:=0;
 Vt:=0;
 Vr:=0;
 VentN2Tx:=7;
 VentN2Rx:=7;
 HayTramasTx:=False;
 HayTramasRx:=False;
 HayPaqueteTx:=False;
 HayPaqueteRx:=False;
 T1Inicial:=0;
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
T3Inicial:=0;
N2Activo:=False;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que contesta afirmativamente si ha concluido la temporización T1.

```
Function TCCN2.FinalT1:Boolean;
Var H,M,S,Ms : Word;
   Tiempo : LongInt;
begin
If T1Activo then
 begin
 DecodeTime( Time ,H,M,S,Ms );
 Tiempo:=H*60;
 Tiempo:=Tiempo*60;
 Tiempo:=Tiempo+M*60;
 Tiempo:=Tiempo+S;
 If ( Tiempo )-T1Inicial>=T1 then
  begin
  T1Activo:=False;
  FinalT1:=True;
  {T1Inicial:=0;}
  end
  Else FinalT1:=False;
 end
Else FinalT1:=True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que inicia el temporizador T1.

```
Procedure TCCN2.DisparaT1;
Var H,M,S,Ms : Word;
 begin
 If Not( T1Activo )then
  begin
  T1Activo:=True;
  DecodeTime(Time,H,M,S,Ms );
  T1Inicial:=H*60;
  T1Inicial:=T1Inicial*60;
  T1Inicial:=T1Inicial+M*60;
  T1Inicial:=T1Inicial+S;
  end;
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que contesta afirmativamente si ha concluido el temporizador T3.

```
Function TCCN2.FinalT3:Boolean;
Var H,M,S,Ms : Word;
   Tiempo:LongInt;
begin
If T3Activo then
 begin
 DecodeTime( Time ,H,M,S,Ms );
 Tiempo:=H*60;
 Tiempo:=Tiempo*60;
 Tiempo:=Tiempo+M*60;
 Tiempo:=Tiempo+S;
 If ( Tiempo )-T3Inicial>=T3 then
  begin
  T3Activo:=False;
  FinalT3:=True;
  {T3Inicial:=0;}
  end
 Else FinalT3:=False;
 end
Else FinalT3:=True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento que inicia el temporizador T3.

```
Procedure TCCN2.DisparaT3;
Var H,M,S,Ms : Word;
 begin
 If Not( T3Activo ) then
  begin
  T3Activo:=True;
  DecodeTime(Time,H,M,S,Ms );
  T3Inicial:=H*60;
  T3Inicial:=T3Inicial*60;
  T3Inicial:=T3Inicial+M*60;
  T3Inicial:=T3Inicial+S;
  end;
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función y procedimiento encargados de intercambiar tramas con la capa física implementada en el Objeto EtFi.

```
Function TCCN2.TxTramas:String;
 Begin
 HayTramasTx:=False;
 TxTramas:=TramasTx;
 TramasTx:='';
 end;
Procedure TCCN2.RxTramas( RxTramas :String );
 begin
 TramasRx:=RxTramas;
 HayTramasRx:=True;
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función y procedimiento encargados de intercambiar paquetes con la capa superior del protocolo implementada sobre el objeto CCTerm.

```
Function TCCN2.TxPaquete:String;
 Begin
 HayPaqueteTx:=False;
 TxPaquete:=PaqueteTx;
 PaqueteTx:='';
 end;
Procedure TCCN2.RxPaquete( RxPaquetes : String);
 begin
 PaqueteRx:=RxPaquetes;
 HayPaqueteRx:=True;
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento encargado de implementar el Nivel 2 del protocolo. Se le llama desde el procedimiento Idle de la Unidad SimX25.

```
Procedure TCCN2.Control;
 begin
 { Si concluye T1 sin respuesta }
 {If ( Not( HayTramasRx ))and( T1Activo )and
   (Estado<>'E1')and( FinalT1 )then
 begin
  Estado:='E1';
 end;}
```

Desconexión.Estado E1

```
 If Estado = 'E1' then
  begin
  If ( HayTramasRx ){and( FinalT1 )and( FinalT3 )}then
   begin
   { Si es un DISC contesta DM Sigue en estado E1}
   If TipoTrama( TramasRx )='DISC' then
    begin
    If not( HayTramasTx ) then
     begin
     Estado:='E1';
     N2Activo:=False;
     TramasTx:=DM( DirRemota,BitP( TramasRx ));
     HayTramasTX:=True;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 HayTramasRx:=False;
 TramasRx:='';
 {DisparaT1;}
 RedX25.Visual( Nombre+' Recepcion de DISC en estado E1');
 end;
end;
{ Si es un SABM contesta UA Pasa a estado E4 }
If TipoTrama( TramasRx )='SABM' then
 begin
 If not( HayTramasTx ) then
  begin
  Estado:='E4';
  N2Activo:=True;
  TramasTx:=UA( DirRemota,BitP( TramasRx ));
  HayTramasTx:=True;
  HayTramasRx:=False;
  TramasRx:='';
  {DisparaT1;}
  RedX25.Visual( Nombre+' Recepcion de SABM en estado E1.Pasa a Estado E4');
  end;
 end;
If (TipoTrama( TramasRx )<>'DISC')and( TipoTrama( TramasRx )<>'SABM')then
 begin
{ Si es una trama con P=1 contesta DM con F=1 y sigue en E1 }
 If BitP( TramasRx ) then
  begin
  If not( HayTramasTx ) then
   begin
   Estado:='E1';
   N2Activo:=False;
   TramasTx:=DM( DirRemota,BitP( TramasRx ));
   HayTramasTX:=True;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    HayTramasRx:=False;
    TramasRx:='';
    {DisparaT1;}
    RedX25.Visual( Nombre+'Recibe Trama con P=1 contesta DM y sigue en E1 ');
    end;
   end;
  { Si es una trama con P=0 se descarta y sigue en E1 }
  If Not( BitP( TramasRx )) then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E1';
    N2Activo:=False;
    HayTramasRx:=False;
    TramasRx:='';
    {DisparaT1;}
    RedX25.Visual( Nombre+'Recibe trama con P=0, no contesta y sigue en E1');
   end;
   end;
  end;
  end
 Else
  begin
  { Envia SABM dispara T1 y pasa a E2 }
  If Not( T1Activo ) then
   begin
   TramasTx:=SABM( DirLocal,False );
   HayTramasTX:=True;
   DisparaT1;
   Estado:='E2';
   N2Activo:=False;
   RedX25.Visual( Nombre+'En estado E1 envia un SABM dispara T1 y pasa a E2');
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    end;
  end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Indicación de Iniciación. Estado E2.

```
If Estado = 'E2' then
 begin
  If ( HayTramasRx ){and( FinalT1 )and( FinalT3 )} then
  begin
  { Si es un DISC contesta DM y pasa a estado E1 }
  If TipoTrama( TramasRx )='DISC' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E1';
    N2Activo:=False;
    TramasTx:=DM( DirRemota,BitP( TramasRx ));
    HayTramasTX:=True;
    HayTramasRx:=False;
    TramasRx:='';
    ContN2:=0;
    RedX25.Visual( Nombre+' Recepcion de DISC en E2.Contesta DM y pasa a E1');
    end;
   end;
  { Si es un DM pasa a estado E1 }
  If TipoTrama( TramasRx )='DM' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E1';
    N2Activo:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 HayTramasRx:=False;
 TramasRx:='';
 ContN2:=0;
 RedX25.Visual( Nombre+' Rx de DM en E1 .Pasa a E1');
 end;
end;
{ Si es un SABM contesta UA y sigue en estado E2 }
If TipoTrama( TramasRx )='SABM' then
 begin
 If not( HayTramasTx ) then
  begin
  Estado:='E2';
  N2Activo:=False;
  TramasTx:=UA( DirRemota,BitP( TramasRx ));
  HayTramasTx:=True;
  HayTramasRx:=False;
  TramasRx:='';
  ContN2:=0;
  RedX25.Visual( Nombre+' Rx de SABM en  E2.Contesta UA y sigue en E2');
  end;
 end;
{ Si es un UA pasa a estado E4 }
If TipoTrama( TramasRx )='UA' then
 begin
 If not( HayTramasTx ) then
  begin
  Estado:='E4';
  N2Activo:=True;
  HayTramasRx:=False;
  TramasRx:='';
  ContN2:=0;
  If Not( HayTramasTx ) then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    begin
    If DirLocal='03' then
     begin
     TramasTx:=Rearranque;
     HayTramasTx:=True;
     If Ns=7 then Ns:=0 Else Inc( Ns );
     Inc( Vt );
     Vr:=0;
     end;
    RedX25.Visual( Nombre+' Rx de UA en E2.Pasa a E4 y envia un
INFO=Rearranque');
    end;
    end;
   end;
  end
 Else
  begin
  { Si  T3 no activo }
  If Not( T3Activo )then
   begin
   { Si T1 activo pregunta si ha finalizado }
   { Si ha finalizado incrementa ContN2 y envia SABM }
   If FinalT1 then
    begin
    if Not(HayTramasTx)then
     begin
     TramasTx:=SABM( DirLocal,False );
     HayTramasTX:=TRue;
     Inc( ContN2 );
     RedX25.Visual( Nombre+' Envio de SABM en E2 por vencimiento de T1 ');
     { Si ContN2=N2 dispara T3 si no dispara T1 }
     If ContN2>=N2 then
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      begin
      DisparaT3;
      ContN2:=0;
      end
     Else
      begin
      DisparaT1;
      end;
     end;
    end;{FinalT1}
  end{ T3NoActivo }
 Else
  begin
  { Si T3 activo pregunta si ha finalizado }
  If FinalT3 then
   begin
  { Si Ha finalizado }
  { Pon ContN2 a cero }
  If Not( HayTramasTx ) then     { Envia SABM }
   begin
   TramasTx:=SABM( DirLocal ,True );
   HayTramasTx:=TRue;
   RedX25.Visual( Nombre+' Envio de SABM por vencimiento de T3 ');
    DisparaT1;
   Inc( ContN2 );
   end;
   end;
  end;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Indicación de desconexión. Estado E3.

```
If Estado = 'E3' then
 begin
  If HayTramasRx then
  begin
  { Si es un SABM contesta con DM y sigue en E3 }
  If TipoTrama( TramasRx )='SABM' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E3';
    TramasTx:=DM(DirRemota,BitP( TramasRx ));
    HayTramasTx:=True;
    HayTramasRx:=False;
    TramasRx:='';
    RedX25.Visual( Nombre+' Rx de SABM en E3 .Envia DM y sigue en E3 ');
    end;
   end;
  { Si es un DISC contesta con DM y pasa a E1 }
  If TipoTrama( TramasRx )='DISC' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E1';
    TramasTx:=DM( DirRemota,BitP( TramasRx ));
    HayTramasTx:=True;
    HayTramasRx:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 TramasRx:='';
 RedX25.Visual( Nombre+' Rx de DISC en E3.Envia DM y pasa a E1 ');
 end;
end;
{ Si es un UA pasa a E1 }
If TipoTrama( TramasRx )='UA' then
 begin
 If not( HayTramasTx ) then
  begin
  Estado:='E1';
  HayTramasRx:=False;
  TramasRx:='';
  RedX25.Visual( Nombre+' Rx de UA en E3.Pasa a E1 ');
  end;
 end;
{ Si es un DM pasa a E1 }
If TipoTrama( TramasRx )='DM' then
 begin
 If not( HayTramasTx ) then
  begin
  Estado:='E1';
  HayTramasRx:=False;
  TramasRx:='';
  RedX25.Visual( Nombre+' Rx de DM en E3.Pasa a E1 ');
  end;
 end;
end
Else
 begin
 { Si no hay nada en recepcion }
 { Si T1 activo }
 If T1Activo then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
If FinalT1 then
 begin
{ Mira si ha concluido }
 { Si ContN2=N2 pasa a estado E1 }
 if ContN2>=N2 then Estado:='E1'
 Else
  begin
 { Si ContN2<N2 }
  { Si ha concluido envia DISC }
  TramasTx:=DISC( DirLocal,False );
  HayTramasTx:=True;
  { Dispara T1}
  DisparaT1;
  { Incrementa ContN2}
  Inc( ContN2 );
  end;
 end;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Transferencia de Información. Estado E4.

```
If Estado = 'E4' then
 begin
  If HayTramasRx then
  begin
  { Si es un SAB contesta UA y sigue en E4 }
  If TipoTrama( TramasRx )='SABM' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E4';
    TramasTx:=UA(DirRemota,BitP( TramasRx ));
    HayTramasTx:=True;
    HayTramasRx:=False;
    TramasRx:='';
    RedX25.Visual( Nombre+' Rx de SABM en E4.Contesta UA y sigue en E4 ');
    end;
   end;
  { Si se recibe un DM pasa a estado E1 }
  If TipoTrama( TramasRx )='DM' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E1';
    HayTramasRx:=False;
    TramasRx:='';
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 RedX25.Visual( Nombre+' Rx de DM en E4.Pasa a E1 ');
 end;
end;
{ Si se recibe FRMR contesta SABM y pasa a E2 }
If TipoTrama( TramasRx )='FRMR' then
begin
If not( HayTramasTx ) then
 begin
 Estado:='E2';
 TramasTx:=SABM( DirRemota,BitP( TramasRx ));
 HayTramasTx:=True;
 HayTramasRx:=False;
 TramasRx:='';
 RedX25.Visual( Nombre+' Rx de FRMR en E4,Envia SABM y pasa a E2 ');
 end;
end;
{ Tratamiento de infos y RR }
If TipoTrama( TramasRx )='INFO' then
begin
{Pasa el info recibido al nivel 3 }
If Not( HayPaqueteTx ) then
 begin
 PaqueteTx:=ExtraePaq( TramasRx );
 DirActual:=Copy( TramasRx,3,2 );
 HayPaqueteTx:=True;
 RedX25.Visual( Nombre+' Rx un INFO en E4 y lo pasa a N3 ');
 { Si T1 estaba activo desactivalo }
If T1Activo then T1Activo:=False;
{ Si hay paquetes que enviar envia un Info y dispara T1}
If Nr = 7 then Nr:=0 Else Inc( Nr );
Inc( Vr );
Vt:=0;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
If HayPaqueteRx then
 begin
 { Si hay algo que enviar manda un info con direccion remota }
 { Si el bit P esta activo manda un Info con F activo }
 If BitP(TramasRx)then
  begin
  If ( Not(HayTramasTx ))and( Vt<VentN2Tx ) then
   begin
   If DirActual= DirRemota then
   TramasTx:=INFO( DirRemota,Nr,1,Ns,PaqueteRx )
   Else
   TramasTx:=INFO( DirLocal,Nr,1,Ns,PaqueteRx );
   DirActual:=DirLocal;
   HayTramasTx:=True;
   HayTramasRx:=False;
   TramasRx:='';
   HayPaqueteRx:=False;
   PaqueteRx:='';
   RedX25.Visual( Nombre+' Si Rx INFO con P=1 y Hay un INFO que mandar
Envialo con F=1 ' );
   If Ns=7 then Ns:=0 Else Inc( Ns );
   Vr:=0;
   Inc( Vt );
   end;
  end
 Else{ Si el bit P no esta activo manda un Info con F no activo }
  begin
  If ( Not(HayTramasTx ))and( Vt<VentN2Tx ) then
   begin
   If DirActual= DirRemota then
   TramasTx:=INFO( DirRemota,Nr,0,Ns,PaqueteRx )
   Else
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      TramasTx:=INFO( DirLocal,Nr,0,Ns,PaqueteRx );
      DirActual:=DirLocal;
      HayTramasTx:=True;
      HayTramasRx:=False;
      TramasRx:='';
      HayPaqueteRx:=False;
      PaqueteRx:='';
      RedX25.Visual( Nombre+' Si Rx INFO con P=0 y Hay un INFO que mandar
Envialo con F=0 '+tramastx);
      If Ns=7 then Ns:=0 Else Inc( Ns );
      Vr:=0;
      Inc( Vt );
      end;
     end;
     {DisparaT1;}
    { Incrementa la ventana de transmision }
    end
   Else
    begin
   { Si no hay algo que enviar y se cumple la ventanamanda o el bit P esta}
   { activo envia un RR ( Con el bit F activo si P esta activo )y dispara T1}
    If Not(HayTramasTx) then
     begin
     If BitP( TramasRx ) then
      begin
      { Si el bit P esta activo responde con un RR inmediatamente }
      If DirActual=DirRemota then
      TramasTx:=RR(DirRemota,BitP(TramasRx),Nr)
      Else
      TramasTx:=RR(DirLocal,BitP(TramasRx),Nr);
      DirActual:=DirLocal;
      HayTramasTx:=True;
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      HayTramasRx:=False;
      TramasRx:='';
      Vr:=0;
      RedX25.Visual( Nombre+' Si Rx INFO con P=1 y no hay un INFO que mandar
Envia RR con F=1 ');
      end
     Else
      begin
      If Vr=VentN2Rx then
       begin
       { Si el bit P no esta activo espera a que se cumpla la ventana }
       TramasTx:=RR(DirRemota,BitP(TramasRx),Nr);
       HayTramasTx:=True;
       Vr:=0;
       RedX25.Visual( Nombre+' Rx INFO con P=0 y no hay un INFO que mandar Envia
'
                              +'RR con F=0 cuando se cumpla la ventana');
      end;
      HayTramasRx:=False;
      TramasRx:='';
      end;
     end;
    end;
    end;
   end;
   If TipoTrama( TramasRx )='RR' then
   begin
   { Si T1 estaba activo desctivalo }
   If T1Activo then T1Activo:=False;
   { Pon a cero el contador de la ventana de Tx }
   Vt:=0;
   { Si hay algo que enviar envia un Info y dispara T1}
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
If HayPaqueteRx then
 begin
 If ( not( HayTramasTx ))and( Vt<VentN2Tx ) then
  begin
   If DirActual= DirRemota then
   TramasTx:=INFO( DirRemota,Nr,0,Ns,PaqueteRx )
   Else
   TramasTx:=INFO( DirLocal,Nr,0,Ns,PaqueteRx );
   DirActual:=DirLocal;
   HayTramasTx:=True;
   HayTramasRx:=False;
   TramasRx:='';
   HayPaqueteRx:=False;
   PaqueteRx:='';
   If Ns=7 then Ns:=0 Else Inc( Ns );
   Vr:=0;
   Inc( Vt );
   RedX25.Visual( Nombre+' Si Rx RR y hay un INFO que mandar Envia un INFO ');
   end;
 end
Else{ Si no hay algo que enviar manda un RR y dispara T1}
 begin
 { Si es un comando }
 if Copy(TramasRx,3,2 )=DirRemota then
  begin
  if {(FinalT1)and}(Not( HayTramasTx ))then
   begin
   TramasTx:=RR(DirRemota,BitP(TramasRx),Nr);
   HayTramasTx:=True;
   HayTramasRx:=False;
   TramasRx:='';
   {DisparaT1;}
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
      Vr:=0;
      RedX25.Visual( Nombre+' Si Rx RR( Comando) y no hay un INFO que mandar
Envia un RR ');
      end;
     end;
    {Si es una  respuesta }
    if Copy(TramasRx,3,2 )=DirLocal then
     begin
      TramasRx:='';
      HayTramasRx:=False;
      {Vr:=0;}
      RedX25.Visual( Nombre+' Si Rx RR( Respuesta) no responde nada ');
      {DisparaT1;}
     end;
    end;
   end;
  If TipoTrama( TramasRx )='REJ' then
   begin
   end;
  end
 Else
  begin
  { Si no hay nada en recepcion }
  { Si hay paquetes que enviar y T1 esta cumplido }
  { y esta dentro de la ventana de Tx envia un Info }
  If HayPaqueteRx then
   begin
   If {( FinalT1 )and}(Not(HayTramasTx))and( Vt<VentN2Tx ) then
    begin
    HayPaqueteRx:=False;
    TramasTx:=INFO( DirActual,Nr,P,Ns,PaqueteRx );
    DirActual:=DirLocal;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    HayTramasTx:=True;
    PaqueteRx:='';
    If Ns=7 then Ns:=0 Else Inc( Ns );
    { Incrementa la ventana de Tx }
    Inc(Vt);
    Vr:=0;
    RedX25.Visual( Nombre+' Si no hay nada en Rx y hay  INFO que enviar envialo ' );
    { y Dispara T1 }
    {DisparaT1;}
    end;
   end
  Else{ Si no hay nada que enviar y T1 esta cumplido envia un RR }
   begin
   If ( FinalT3 )and(Not(HayTramasTx))and( DirLocal='03')then
    begin
    { Polling del nodo cada t3 segundos }
    { y Dispara T3}
    If DirActual = DirRemota then
    TramasTx:=RR(DirRemota,False,Nr)
    Else
    TramasTx:=RR(DirLocal,False,Nr);
    DirActual:=DirLocal;
    HayTramasTx:=True;
    TramasRx:='';
    HayTramasRx:=False;
    DisparaT3;
    Vr:=0;
    RedX25.Visual( Nombre+' Si no hay nada que enviar y nada en recepcion manda un
RR y dispara T3 ');
    end;
   end;
  end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Indicación de rechazo. Estado E5.

```
If Estado = 'E5' then
 begin
  If HayTramasRx then
  begin
  { Si es un SABM contesta UA y pasa a E4}
  If TipoTrama( TramasRx )='SABM' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E4';
    TramasTx:=UA(DirRemota,BitP( TramasRx ));
    HayTramasTx:=True;
    HayTramasRx:=False;
    TramasRx:='';
    RedX25.Visual( Nombre+' Si Rx SABM en E5 .Envia UA y pasa a E4 ');
    end;
   end;
  { Si es un DISC contesta DM y pasa a E1 }
  If TipoTrama( TramasRx )='DISC' then
   begin
   If not( HayTramasTx ) then
    begin
    Estado:='E1';
    TramasTx:=DM( DirRemota,BitP( TramasRx ));
    HayTramasTx:=True;
    HayTramasRx:=False;
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 TramasRx:='';
 RedX25.Visual( Nombre+' Si Rx DISC en E5 .Envia DM y pasa a E1 ');
 end;
end;
{ Si es un FRMR contesta SABM y pasa a E2}
If TipoTrama( TramasRx )='FRMR' then
 begin
 If not( HayTramasTx ) then
  begin
  Estado:='E2';
  TramasTx:=SABM( DirRemota,BitP( TramasRx ));
  HayTramasTx:=True;
  HayTramasRx:=False;
  TramasRx:='';
  RedX25.Visual( Nombre+' Si Rx FRMR en E5 .Envia DM y pasa a E2 ');
  end;
 end;
{ Si es un DM pasa a estado E1 }
If TipoTrama( TramasRx )='DM' then
 begin
 If not( HayTramasTx ) then
  begin
  Estado:='E1';
  HayTramasRx:=False;
  TramasRx:='';
  RedX25.Visual( Nombre+' Si Rx DM en E5 .Pasa a E4 ');
  end;
 end;
end
Else
 begin
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama SABM a partir de las variables que se le entregan como parámetros.

```
Function TCCN2.SABM( Direc : String ; P:Boolean): String;
Var Trama : String;
begin
Trama:=Direc;
if P then Trama:=Trama+'3F';
If Not( P ) then Trama:=Trama+'2F';
SABM:= '7E'+Trama+CRC( Trama )+'7E';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama UA a partir de las variables que se le entregan como parámetros

```
Function TCCN2.UA( Direc : String ; P:Boolean): String;
Var Trama : String;
begin
Trama:=Direc;
If P then Trama:=Trama+'73';
If Not( P ) then Trama:=Trama+'63';
UA:='7E'+Trama+CRC( Trama )+'7E';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama DISC a partir de las variables que se le entregan como parámetros.

```
Function TCCN2.Disc( Direc : String; P:Boolean ) : String;
Var Trama : String;
begin
Trama:=Direc;
If P then Trama:=Trama+'53';
If Not( P ) then Trama:=Trama+'43';
DISC:='7E'+Trama+CRC( Trama )+'7E';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama DM a partir de las variables que se le entregan como parámetros.

```
Function TCCN2.DM( Direc : String; P:Boolean ):String;
Var Trama : String;
begin
Trama:=Direc;
If P then Trama:=Trama+'1F';
If Not( P ) then Trama :=Trama+'0F';
DM:='7E'+Trama+CRC( Trama )+'7E';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama REJ a partir de las variables que se le entregan como parámetros.

```
Function TCCN2.REJ( Direc : String; P:Boolean; NRX : Integer ):String;
Var Trama : String;
begin
Trama:=Direc;
If P then Trama:=Trama+NrPRR(NRX,1,9);
If Not( P ) then Trama:=Trama+NrPRR(NRX,0,9 );
REJ:='7E'+Trama+CRC( Trama )+'7E';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama FRMR a partir de las variables que se le entregan como parámetros.

```
Function TCCN2.FRMR( Direc : String; P:Boolean ):String;
Var Trama : String;
begin
Trama:=Direc;
If P then Trama:=Trama+'97';
If not( P ) then Trama:=Trama+'87';
FRMR:='7E'+Trama+CRC( Trama )+'7E';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama RR a partir de las variables que se le entregan como parámetros.

```
Function TCCN2.RR( Direc : String; P:Boolean; NRX : Integer ):String;
Var Trama : String;
begin
Trama:=Direc;
If P then Trama:=Trama+NrPRR(NRX,1,1);
If Not( P ) then Trama:=Trama+NrPRR(NRX,0,1 );
RR:='7E'+Trama+CRC( Trama )+'7E';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que construye una trama de Información a partir de las variables que se le entregan como parámetros.

```
Function TCCN2.INFO( Direc: String;NRx,Px,NSx:Byte;Paq:String ):String;
Var Trama : String;
begin
{RedX25.Visual( Nombre+'ERROR Paquete 1 '+Paq );}
If ( HD(Copy(paq,5,2))mod 2<>0 )and(Copy(Paq,6,1)<>'1')then Px:=1;
Trama:=Direc+NrPNsDH( Nrx,Px,Nsx)+Paq;
Trama:='7E'+Trama+INFOCRC( Trama )+'7E';
{RedX25.Visual( Nombre+'ERROR Trama 1 '+Trama );}
Info:=Trama;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que entrega una cadena que identifica el tipo de trama entregada como parámetro.

```
Function TCCN2.TipoTrama( Trama : String ): String;
Var Control: String;
   Tipo   : String;
begin
If Trama<>''then
begin
Control:=Copy( TRama,5,2 );
If ( HD( Control ) mod 2 <>0 )then
 begin
 If ( Control='3F' )or( Control='2F' )then Tipo:='SABM';
 If ( Control='73' )or( Control='63' )then Tipo:='UA';
 If ( Control='53' )or( Control='43' )then Tipo:='DISC';
 If ( Control='1F' )or( Control='0F' )then Tipo:='DM';
 If ( Control='97' )or( Control='87' )then Tipo:='FRMR';
 If ( copy(Control,2,1)='9' )then Tipo:='REJ';
 If ( copy(Control,2,1)='1' )then Tipo:='RR';
 end
Else Tipo:='INFO';
TipoTrama:=Tipo;
end
Else TipoTrama:='';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que interroga si la trama recibida tiene el bit P activo o no.

```
Function TCCN2.BitP( Trama : String ):Boolean;
Var Control: String;
   Cntl  : Byte;
begin
Control:=Copy( TRama,5,2 );
Cntl:=HD( Control );
If ( Cntl and 16 )=16 then BitP:=True Else BitP:=False;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que entrega la subcadena de la cadena que representa la trama que es a su vez la parte correspondiente al paquete de nivel de Red.

```
Function TCCN2.ExtraePaq( Trama : String ):String;
Var Cad : String;
begin
{RedX25.Visual( Nombre+'ERROR Trama 2 '+Trama );}
Cad:=Copy( Trama,7,length( Trama )-6);
{RedX25.Visual( Nombre+'ERROR Trama 3 '+Cad );}
Cad:=Copy( Cad,1,Length( Cad )-6 );
ExtraePaq:=Cad;
{RedX25.Visual( Nombre+'ERROR Paquete 4 '+Cad );}
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función encargada de generar una cadena que representa un Rearranque de nivel tres para el inicio del nivel de Red una vez que se establece el nivel de enlace.

```
Function TCCN2.Rearranque: String;
Var cad : String;
begin
Rearranque :=INFO( DirLocal,Nr,1,Ns,'1000'+DH( 251 )+'0700');
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Procedimiento encargado de leer la configuración de nivel 2 del archivo de configuración.

```
Procedure TCCN2.LeerConfig( Archivo : String );
begin
AssignFile( ArchCfg, Archivo );
{$I-}
Reset( ArchCfg );
{$I+}
If IOResult<>0 then
 ShowMessage(' No se ha podido abrir el archivo '+Archivo )
Else
 begin
 Read( ArchCfg, Parametros );
 CloseFile( ArchCfg );
 VentN2Tx:=Parametros.VN2Tx;
 VentN2Rx:=Parametros.VN2Rx;
 T1:=Parametros.T1;
 T3:=Parametros.T3;
 N2:=Parametros.N2;
 RedX25.Visual(Nombre+'VN2Tx '+IntToStr( VentN2Tx )+' VN"Rx '+IntToStr(
VentN2Rx )
                +' T1 '+IntToStr(T1)+' T3 '+IntToStr(T3)+' N2 '+IntToStr(N2));
 end;{ Se a leido correctamente el archivo }
end;
end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad EtFi

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

unit EtFi;

Unidad que implementa el medio de transmisión. Define la interconexión entre los dos interfaces DTE/DCE. En un circuito constituido por pares metálicos, esta unidad implementaría el Modem o DCE de un extremo, el par físico que lo uniría con el Modem remoto y el Modem remoto hasta el Interfaz con el terminal. Es por esto que se le ha dado el nombre de Etapa Física. No se implementan Flags (7E).

```
interface

Uses WinTypes,WinProcs,Classes,Controls;

Type
  Frame = String;

TEF = Class( TObject )
 TramasNT : Frame;
 TramasTN : Frame;
 IntTN : Boolean;{ Si hay tramas con sentido Terminal Nodo }
 IntNT : Boolean;{ Si hay Tramas con sentido Nodo Terminal }
 Constructor Create;
 Function TxEFT:String;
 Procedure RxTEF( Trama:String ) ;
 Function TxEFN:String;
 Procedure RxNEF( Trama: String );
end;

implementation
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función Transmisión desde la Etapa Física hacia el Terminal y Procedimiento Recepción desde el Terminal hacia la Etapa Física. Estos son los métodos que se encargan de intercambiar tramas entre el Terminal o DTE y el Modem o DCE.

```
Function TEF.TxEFT: String;
begin
 IntNT :=False;
 TxEfT:=TramasNT;
 TramasNt:="
end;
Procedure TEF.RxTEF( Trama:String ) ;
begin
 TramasTN:=Trama;
 IntTN := True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función Transmisión desde la Etapa Física hacia el Nodo y Procedimiento Recepción desde el Nodo hacia la Etapa Física. Procedimientos que permiten al Nivel 2 del Nodo intercambiar información con la capa Física. El Nodo sería el DTE y el DCE la Etapa Física.

```
Function TEF.TxEFN: String;
begin
 IntTN:=False;
 TxEfN:=TramasTN;
 TramasTN:='';
end;
Procedure TEF.RxNEF( Trama : String);
begin
 TramasNT:=Trama;
 IntNT := True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método de iniciación de las variables del objeto EtFi. Se invoca cuando se crea este objeto al iniciarse el programa. Concretamente se crea el objeto EtFi cuando se crea el objeto RedX25 de la unidad SimX25.

```
Constructor TEF.Create;
begin
TObject.Create;
TramasNT :='';
TramasTN :='';
IntTN :=False;
IntNT :=False;

end;


end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad Analizad

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

unit Analizad;

Unidad soporte del objeto visual TAnalizador.Este objeto simula un Analizador de Protocolo conectado a la línea del terminal , entre el ETD y el ETCD.En el se pueden visualizar los distintos niveles ISO del protocolo X25.Se implementan tambien otras funciones como la de Filtro,Captura en Archivos,Trampas, etc..

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,

Forms, Dialogs,Buttons, Menus, StdCtrls, ExtCtrls, Tabs,

Calculo,AnFil,tram;

A continuación se define el Tipo Trm. Este registro definirá las condiciones y acciones de la opción "Trampa" del Analizador.

type

Trm = Record

Dir : String;

Cont2 : String;

Cont2MB : String;

IGF : String;

IGFMB : String;

Can : String;

Cont3 :String;

Cont3MB : String;

ParарAnal : Boolean;

ComenzarAnal : Boolean;

ComenzarCaptura : Boolean;

ParарCaptura : Boolean;

VentanaAviso : Boolean ;

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
ArchTrampas : String;
end;{ Record }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

A continuación se define la clase Tanalizador que será la que cree los analizadores de cada uno de los Terminales de Usuario así como los Analizadores de Ruta.

```
type
  TAnalizador = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;

    MainMenu1: TMainMenu;
    Nivel1: TMenuItem;
    Filtro1: TMenuItem;
    Fisico1: TMenuItem;
    Enlace1: TMenuItem;
    Red1: TMenuItem;
    TabSet1: TTabSet;
    Memo1: TMemo;
    Memo2: TMemo;
    Memo3: TMemo;
    Memo4: TMemo;
    Panel3: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Parar1: TMenuItem;

N1: TMenuItem;

N21: TMenuItem;

Todos1: TMenuItem;

Archivos1: TMenuItem;

Guardar1: TMenuItem;

Capturar1: TMenuItem;

NivelFisico1: TMenuItem;

NiveldeEnlace1: TMenuItem;

NiveldeRed1: TMenuItem;

TodoslosNiveles1: TMenuItem;

SaveDialog1: TSaveDialog;

N2: TMenuItem;

Trampas1: TMenuItem;

N3: TMenuItem;

LimpiarAnalizador1: TMenuItem;

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Se enumeran a continuación los métodos que definen las opciones realizables por el Analizador de Protocolo X25

```
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
Procedure Captura( Trama : String );
procedure TabSet1Click(Sender: TObject);
FuncTion PresentaN3( H:String ):String;
Function PresentaN2( H:String ):String;
Function EventoTrampa( Frame : String ): Boolean;
Function ComparaMascara( Mascara,Octeto :String ):Boolean;
Procedure InsLinea1( Linea:String );
Procedure InsLinea2( Linea:String );
Procedure InsLinea3( Linea:String );
Procedure InsLinea4( Linea:String );
procedure Parar1Click(Sender: TObject);
procedure Cambio( TabNuevo: Integer );
procedure Fisico1Click(Sender: TObject);
procedure Enlace1Click(Sender: TObject);
procedure Red1Click(Sender: TObject);
procedure Todos1Click(Sender: TObject);
procedure N21Click(Sender: TObject);
procedure NivelFisico1Click(Sender: TObject);
procedure NiveldeEnlace1Click(Sender: TObject);
procedure NiveldeRed1Click(Sender: TObject);
procedure TodoslosNiveles1Click(Sender: TObject);
procedure Capturar1Click(Sender: TObject);
procedure LimpiarAnalizador1Click(Sender: TObject);
procedure Trampas1Click(Sender: TObject);
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
{ procedure Memo3MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);}
{ procedure Memo2MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);}
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Siguen a partir de aqui las variables definidas dentro del objeto.Destacan las variables encargadas de dar soporte a la opción de Filtro

```
private
  { Private declarations }
public
{ Public declarations }
 Boton    : TSpeedButton;
 Activado : ^Boolean;
 Puntero1 : Integer;
 Puntero2 : Integer;
 Puntero3 : Integer;
 Puntero4 : Integer;
 Maxlineas : Integer;
{ Comienzan las variables que definen el filtro }
 FDTE     : Boolean;{ Filra DTE}
 FDCE     : Boolean;{ Filtra DCE}
 FSABM    : Boolean;{ Filtra SABM}
 FUA      : Boolean;{Filtra UA}
 FDISC    : Boolean;{ Filtra DISC}
 FDM      : Boolean;{Filtra DM}
 FRR      : Boolean;{ Filtra RR}
 FRNR     : Boolean;{ Filtra RNR}
 FINFO    : Boolean;{ Filtra INFO}
 FFRMR    : Boolean;{Filtra FRMR}
 FSLLAM   : Boolean;{Filtra solicitud de llamada }
 FALLAM   : Boolean;{ Filtra aceptación de llamada}
 FSLIBE   : Boolean;{Filtra solicitud de liberación}
 FALIBE   : Boolean;{Filtra aceptación de liberación }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  FSREIN    : Boolean;{ Filtra solicitud de Reinicio}
  FAREIN    : Boolean;{ Filtra aceptación de Reinicio }
  FSINTE    : Boolean;{ Filtra solicitud de interrupción}
  FAINTE    : Boolean;{Filtra aceptación de interrupción }
  FSREAR    : Boolean;{ Filtra solicitud de rearranque }
  FAREAR    : Boolean;{ Filtra aceptación de rearranque}
  FRRN3     : Boolean;{ Filtra RR de nivel 3}
  FRNRN3    : Boolean;{ Filtra RNR de nivel 3}
  FDATA     : Boolean;{ Filtra paquetes de Datos }
  ACaptura  : Text;
  Direct    : String;
  VTrampas  : Trm;
 end;

var
  Analizador: TAnalizador;
  TabAnterior : Integer ;

Const
  Tab : Char=Char(9);

implementation

{$R *.DFM}

Uses SimX25;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método invocado en la creación del objeto. Se encarga de iniciar todas las variables del objeto, así como de iniciar los objetos que contiene el Analizador.

```
procedure TAnalizador.FormCreate(Sender: TObject);
begin
Memo1.Clear;
Memo2.Clear;
Memo3.Clear;
Memo4.Clear;
Memo1.ReadOnly := True;
Memo2.ReadOnly := True;
Memo3.ReadOnly := True;
Memo4.ReadOnly := True;
TabAnterior := 0;
Label1.Caption:='';
Label2.Caption:='';
Label3.Caption:='';
Label4.Caption:='';
Label5.Caption:='';
Label6.Caption:='';
Label7.Caption:='';
Puntero1:=0;
Puntero2:=0;
Puntero3:=0;
Puntero4:=0;
MaxLineas:=200;
Panel3.Visible:=False;
FSABM  :=True;
FUA    :=True;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
FDISC :=True;
FDM   :=True;
FRR   :=True;
FRNR  :=True;
FINFO :=True;
FFRMR :=True;
FSLLAM :=True;
FALLAM :=True;
FSLIBE :=True;
FALIBE :=True;
FSREIN :=True;
FAREIN :=True;
FSINTE :=True;
FAINTE :=True;
FSREAR :=True;
FAREAR :=True;
FRRN3 :=True;
FRNRN3 :=True;
FDATA :=True;
FDTE  :=True;
FDCE  :=True;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método invocado cuando se cierra el Analizador.Prepara las opciones del menú de la pantalla principal para que se pueda llamar de nuevo al Analizador.Cuando se llamó al analizador estas opciones fueron desavilitadas para impedir que se abrieran varios analizadores del mismo Terminal de Usuario.

```
procedure TAnalizador.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Boton.Enabled := True;
Boton.Visible := True;
Activado^:=False;
RedX25.FormPaint( Sender );
RedX25.CierraAnaliz( Caption );
{ Si la captura continua activa cierra el fichero }
If Capturar1.Checked= True then CloseFile( ACaptura );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método encargado de Importar los datos desde el exterior a el objeto Analizador. Este procedimiento actua dentro del método Idle de la unidad principal "SimX25".Cada vez que hay una trama que intercambiar entre la Etapa Física y el controlador de Nivel 2 del Terminal de Usuarios, el analizador captura dicha trama mediante este metodo.A continuación llama a los métodos que interpretan cada uno de los Niveles del Protocolo. También se miran si se dan las condiciones de disparo de la trampa implementada si la hubiere y las condiciones de filtrado.

```
procedure TAnalizador.Captura( Trama : String );
begin



{ Filtro DTE DCE }
If(( FDTE )and(Pos('DTE',Trama)<>0))Or(( FDCE )and(Pos('DCE',Trama)<>0))then
begin
If Parar1.Checked= False then
 begin
{ Nivel 1 ,Nivel Fisico }
InsLinea1( Trama );
{ Nivel 2 ,Nivel de Enlace }
{ Dir ,tipo, Nr, P/F, Ns, CRC }
If PresentaN2( Copy(Trama,6,length( Trama )-5)) <>'' then
InsLinea2(Copy( Trama,1,5 )+PresentaN2( Copy(Trama,6,length( Trama )-5)) );
{ Nivel 3 ,Nivel de Red }
{ D ,Q ,Canal ,Tipo,Pr ,M ,Ps}
If PresentaN3( Copy(Trama,6,length( Trama )-5))<>'' then
InsLinea3(Copy( Trama,1,5 )+PresentaN3( Copy(Trama,6,length( Trama )-5)) );
{ Todos los niveles }
If ( PresentaN2( Copy(Trama,6,length( Trama )-5)) <>'' )then
 begin
 If ( HD(Copy( Trama,10,2))Mod 2=0 ) then
  begin
  If (PresentaN3( Copy(Trama,6,length( Trama )-5))<>'')then
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
   begin
   InsLinea4(Trama );
   InsLinea4('   N2'+PresentaN2( Copy(Trama,6,length(Trama)-5))) ;
   end;
  end
Else
  begin
  InsLinea4(Trama );
  InsLinea4('   N2'+PresentaN2( Copy(Trama,6,length(Trama)-5))) ;
  end;
 end;

If PresentaN3( Copy(Trama,6,length( Trama )-5))<>'' then
InsLinea4('   N3'+PresentaN3(Copy(Trama,6,length( Trama )-5)));
 end;{ If Parar1.Checked }
 end; { Filtro DTE,DCE }

 { Aqui se mira si se cumple la trampa }
{ Mirar si se produce la condicion de la trampa }
If EventoTrampa( Trama ) then
 begin
 With( VTrampas ) do
 begin
 If ( PararAnal )and( not( Parar1.Checked )) then
  begin
  Parar1.Checked:=True;
  end;
 If ( ComenzarAnal )and( Parar1.Checked ) then
  begin
  Parar1.Checked:=False;
  end;
 If ( ComenzarCaptura )and(not( Capturar1.Checked ))then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 begin
   {NombreArchivo:=ArchTrampas;}
   AssignFile( ACaptura,Direct+ArchTrampas );
   ReWrite( ACaptura );
 end;
If ( PararCaptura )and( Capturar1.Checked ) then
 begin
  Capturar1.Checked:=False;
  CloseFile( ACaptura );
  SHowMessage( 'Parada la captura en Archivo '+ArchTrampas );
 end;
If VentanaAviso  then
 begin
 {VentanaAviso:=False;}
 Showmessage( ' Trampa disparada ( '+trama+')' );
 end;
end;{ With }
end;{ If }
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que presenta el Nivel 2 del Protocolo X25.Analiza la cadena entregada por el Método "Captura" y entrega una cadena en la que se representa el origen de la trama, el tipo de trama, y las variables asociadas a esta .Comprueba que las variables de filtro asociadas a cada tipo de trama de Nivel 2 ,están activas para poder mostrar ese tipo de trama.

```
Function TAnalizador.PresentaN2(H: String ):String;
Var Cad : String;
   P   : String;
   Dir : String;
   Tipo : String;
   Control : String;
   NRX : String;
   NSX : String;
   Tabuladores : String;
begin
Cad:='';
Tabuladores :=Tab+Tab+tab+tab+tab;
Dir:=Copy(H,3,2);
Control :=Copy( H,5,2 );
If ( Copy( H,5,2 )='3F' )and( FSABM ) then
 begin
 Tipo:='SABM';
 P:='1';
 Cad:=Tab+Dir+Tab+Tipo+Tab+'    '+P+tabuladores;
 end;
If ( Copy( H,5,2 )='2F' )and( FSABM ) then
 begin
 Tipo:='SABM';
 P:='0';
 Cad:=Tab+Dir+Tab+Tipo+Tab+'    '+P+tabuladores;
 end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
If ( Copy( H,5,2 )='73' )and(FUA) then
 begin
 Tipo:='UA';
 P:='1';
 Cad:=Tab+Dir+Tab+Tipo+Tab+'   '+P+tabuladores;
 end;
If ( Copy( H,5,2 )='63' )and(FUA) then
 begin
 Tipo:='UA';
 P:='0';
 Cad:=Tab+Dir+Tab+Tipo+Tab+'   '+P+tabuladores;
 end;
If ( Copy( H,5,2 )='53' )and(FDISC) then
 begin
 Tipo:='DISC';
 P:='1';
 Cad:=Tab+Dir+Tab+Tipo+Tab+'   '+P+tabuladores;
 end;
If ( Copy( H,5,2 )='43' )and(FDISC) then
 begin
 Tipo:='DISC';
 P:='0';
 Cad:=Tab+Dir+Tab+Tipo+Tab+'   '+P+tabuladores;
 end;
If ( Copy( H,5,2 )='1F' )and(FDM) then
 begin
 Tipo:='DM';
 P:='1';
 Cad:=Tab+Dir+Tab+Tipo+Tab+'   '+P+tabuladores;
 end;
If ( Copy( H,5,2 )='0F' )and(FDM) then
 begin
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
Tipo:='DM';
P:='0';
Cad:=Tab+Dir+Tab+Tipo+Tab+'    '+P+tabuladores;
end;
If ( Copy( H,5,2 )='97' )and(FFRMR) then
begin
Tipo:='FRMR';
P:='1';
Cad:=Tab+Dir+Tab+Tipo+Tab+'    '+P+tabuladores;
end;
If ( Copy( H,5,2 )='87' )and(FFRMR) then
begin
Tipo:='FRMR';
P:='0';
Cad:=Tab+Dir+Tab+Tipo+Tab+'    '+P+tabuladores;
end;
If ( Copy( H,6,1 )='1')and(FRR) then
begin
Tipo:='RR';
P:=IntToStr(PHD( Control ));
NRX:=IntToStr(NrHD( Control ));
Cad:=Tab+Dir+Tab+Tipo+Tab+NRX+'  '+P+Tabuladores;
end;
If ( HD( Copy( H, 5,2 ))Mod 2 =0 )and(FINFO)then
begin
Tipo:='INFO';
P:=IntToStr(PHD( Control ));
NRX:=IntToStr(NrHD( Control ));
NSX:=IntToStr(NsHD( Control ));
Cad:=Tab+Dir+Tab+Tipo+Tab+NRX+'  '+P+'  '+NSX;
{Memo3.Lines.Add(Cab+PresentaN3( Copy(H,7,length( H )-6)));}
{InsLinea3(Cab+PresentaN3( Copy(H,7,length( H )-6)));}
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 end;
PresentaN2:=Cad;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

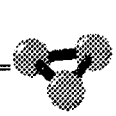
Método invocado desde el método "Captura", encargado de entregar la cadena de texto en la que se representa la información que entrega el Analizador respecto al Nivel 3 de la X25.Se estudia también el origen del paquete ,el tipo de paquete que es y las variables de Nivel 3 asociadas al paquete.Una vez determinado el tipo de paquete que es se estudia la variable de Filtro en la que se indica si dicho paquete pasa o no el Filtro.

```
Function TAnalizador.PresentaN3( H: String ): String;
Var Cad : String;
   Clase : String;
   HH : String;
   Prin: String;
begin
If ( HD( Copy( H, 5,2 ))Mod 2 =0 )then
begin
Cad:='';
HH:=Copy( H, 7,length( H )-6 );
If VerBitQ( Copy( HH,1,2 ) ) then Prin:=Tab+'1' Else Prin:=Tab+'0';
If VerBitD( Copy( HH,1,2 ) ) then Prin:=Prin+'1' Else Prin:=Prin+'0';
Prin:=Prin+VerMod( Copy( HH,1,2 ) );
Prin:=Prin+' '+IntToStr( HD( Copy( H,9,2)));
Clase:=Copy( HH,5,2 );
If ( Clase='0B' )and( FSLlam ) then
 begin
 { Solicitud de llamada }
 Cad:=tab+ 'SOL.LLAM';
 end;
if ( Clase='0F' )and( FALLam ) then
 begin
 { Llamada aceptada }
 Cad:=tab+'ACP.LLAM';
 end;
if ( Clase='13' )and( FSLibe ) then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 begin
 { Solicitud de liberacion }
 Cad:=tab+'SOL.LIBE';
 end;
If ( Clase='17' )and( FALibe ) then
 begin
 { Confirmacion de liberacion }
 Cad:=tab+'ACP.LIBE';
 end;
If ( Clase='23' )and( FSInte ) then
 begin
 { Solicitud de interrupcion }
 Cad:=tab+'SOL.INTE';
 end;
If ( Clase='27' )and( FAInte )then
 begin
 { Confirmacion de interrupcion }
 Cad:=tab+'ACP.INTE';
 end;
If ( Clase='1B' )and( FSRein ) then
 begin
 { Reinicio }
 Cad:=tab+'SOL.REIN';
 end;
If ( Clase='1F' )and( FARein ) then
 begin
 { Confirmacion de reinicio }
 Cad:=tab+'ACP.REIN';
 end;
If ( Clase='FB' )and( FSRear ) then
 begin
 { Solicitud de rearranque }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006



```
Cad:=tab+'SOL.REAR';
end;
If ( Clase='FF' )and( FARear ) then
begin
{ Confirmacion de rearranque }
Cad:=tab+'ACP.REAR';
end;
If ( HD( Clase )Mod 2 = 0 )and( FData ) then
begin
{ Paquete de datos }
Cad:=tab+'DATOS'+tab;
Cad:=Cad+IntToStr(PrHD(Copy( HH,5,2 )))+' ';
Cad:=Cad+IntToStr(MHD(Copy( HH,5,2 )))+' ';
Cad:=Cad+IntToStr(PSHD(Copy( HH,5,2 )));
end;
If ( Copy( HH,6,1 )='1' )and( FRRN3 ) then
begin
{ Es un RR }
Cad:=tab+'RR'+Tab;
Cad:=Cad+IntToStr(PrHD(Copy( HH,5,2 )));
end;
If ( Copy( HH,6,1 )='5' )and( FRNRN3 ) then
begin
{ Es un RNR }
Cad:=tab+'RNR'+Tab;
Cad:=Cad+IntToStr(PrHD(Copy( H,5,2 )));
end;
If Cad<>'' then PresentaN3:=Prin+Cad+tab Else PresentaN3:='';
end
Else
PresentaN3:='';
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método de la librería de Delphi que es llamado cuando se selecciona una de las lenüetas del Analizador.En las lengüetas del analizador se pueden seleccionar los distintos niveles de la Norma X25 o todas ellas a la vez.Se reserva la opción de que sea llamada también la ayuda del programa.

```
procedure TAnalizador.TabSet1Click(Sender: TObject);
begin
If CUrsor=crDefault then Cambio( TabSet1.TabIndex );
If Cursor =crDrag then RedX25.LlamaAyuda( 39 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Este Método se invoca desde el anterior y es el que provoca el cambio de Nivel a mostrar.Para ello cambia el editor que se muestra a la vez que cambia el encabezado de la ventana posterriormente a haber limpiado de la pantalla el editor anterior y el panel de encabezado.

```
Procedure TAnalizador.Cambio( TabNuevo : Integer );
Const Alt : Integer =1;
    Alt2  : Integer =15;
begin
Case TabAnterior of
0:
 begin
 Memo1.Visible:=False;
 Label1.Caption:='';
 Label2.Caption:='';
 Label3.Caption:='';
 Label4.Caption:='';
 Label5.Caption:='';
 Label6.Caption:='';
 Label7.Caption:='';
 Label8.Caption:='';
 Label9.Caption:='';
 Label10.Caption:='';
 Label11.Caption:='';
 Label12.Caption:='';
 Label13.Caption:='';
 Label14.Caption:='';
 end;
1:
 begin
 Memo2.Visible:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  Label1.Caption:='';
  Label2.Caption:='';
  Label3.Caption:='';
  Label4.Caption:='';
  Label5.Caption:='';
  Label6.Caption:='';
  Label7.Caption:='';
  Label8.Caption:='';
  Label9.Caption:='';
  Label10.Caption:='';
  Label11.Caption:='';
  Label12.Caption:='';
  Label13.Caption:='';
  Label14.Caption:='';
  end;
2:
  begin
  Memo3.Visible:=False;
  Label1.Caption:='';
  Label2.Caption:='';
  Label3.Caption:='';
  Label4.Caption:='';
  Label5.Caption:='';
  Label6.Caption:='';
  Label7.Caption:='';
  Label8.Caption:='';
  Label9.Caption:='';
  Label10.Caption:='';
  Label11.Caption:='';
  Label12.Caption:='';
  Label13.Caption:='';
  Label14.Caption:='';
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  end;
3:
  begin
  Memo4.Visible:=False;
  Label1.Caption:='';
  Label2.Caption:='';
  Label3.Caption:='';
  Label4.Caption:='';
  Label5.Caption:='';
  Label6.Caption:='';
  Label7.Caption:='';
  Label8.Caption:='';
  Label9.Caption:='';
  Label10.Caption:='';
  Label11.Caption:='';
  Label12.Caption:='';
  Label13.Caption:='';
  Label14.Caption:='';
  end;
end; { End Case }
Case TabNuevo of
0:
  begin
  Memo1.Visible:=True;
  Panel3.Visible:=False;
  end;
1:
  begin
  Label8.Visible:=False;
  Label9.Visible:=False;
  Label10.Visible:=False;
  Label11.Visible:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  Label12.Visible:=False;
  Label13.Visible:=False;
  Label14.Visible:=False;
  Memo2.Visible:=True;
  Panel3.Visible:=True;
  Panel3.Height:=26;
  Label1.Left:=5;
  Label1.Top:=3;
  Label1.Caption:='DTE/DCE';
  Label2.Left:=65;
  Label2.Top:=3;
  Label2.Caption:='Dir';
  Label3.Left:=137;
  Label3.Top:=3;
  Label3.Caption:='Tipo';
  Label4.Left:=190;
  Label4.Top:=3;
  Label4.Caption:='Nr';
  Label5.Left:=210;
  Label5.Top:=3;
  Label5.Caption:='P ';
  Label6.Left:=225;
  Label6.Top:=3;
  Label6.Caption:='Ns';
  Label7.Left:=500;
  Label7.Top:=3;
  Label7.Caption:='';
end;
2:
  begin
  Label8.Visible:=False;
  Label9.Visible:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
Label10.Visible:=False;
Label11.Visible:=False;
Label12.Visible:=False;
Label13.Visible:=False;
Label14.Visible:=False;
Memo3.Visible:=True;
Panel3.Visible:=True;
Panel3.Height:=26;
Label1.Left:=5;
Label1.Top:=3;
Label1.Caption:='DTE/DCE';
Label2.Left:=65;
Label2.Top:=3;
Label2.Caption:='QDMo';
Label3.Left:=107;
Label3.Top:=3;
Label3.Caption:='Can';
Label4.Left:=147;
Label4.Top:=3;
Label4.Caption:='Tipo';
Label5.Left:=190;
Label5.Top:=3;
Label5.Caption:='Pr';
Label6.Left:=210;
Label6.Top:=3;
Label6.Caption:='M';
Label7.Left:=225;
Label7.Top:=3;
Label7.Caption:='Ps';
end;
3:
begin
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
Memo4.Visible:=True;
Panel3.Visible:=True;
Panel3.Height:=32;
Label7.Visible:=True;
Label8.Visible:=True;
Label9.Visible:=True;
Label10.Visible:=True;
Label11.Visible:=True;
Label12.Visible:=True;
Label13.Visible:=True;
Label14.Visible:=True;
Label1.Left:=5;
Label1.Top:=Alt;
Label1.Caption:='N2';
Label2.Left:=65;
Label2.Top:=Alt;
Label2.Caption:='Dir';
Label3.Left:=137;
Label3.Top:=Alt;
Label3.Caption:='Tipo';
Label4.Left:=190;
Label4.Top:=Alt;
Label4.Caption:='Nr';
Label5.Left:=210;
Label5.Top:=Alt;
Label5.Caption:='P';
Label6.Left:=225;
Label6.Top:=Alt;
Label6.Caption:='Ns';
Label7.Left:=500;
Label7.Top:=Alt2;
Label7.Caption:='';
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 Label8.Left:=5;
 Label8.Top:=Alt2;
 Label8.Caption:='N3';
 Label9.Left:=65;
 Label9.Top:=Alt2;
 Label9.Caption:='QDMo';
 Label10.Left:=107;
 Label10.Top:=Alt2;
 Label10.Caption:='Can';
 Label11.Left:=147;
 Label11.Top:=Alt2;
 Label11.Caption:='Tipo';
 Label12.Left:=190;
 Label12.Top:=Alt2;
 Label12.Caption:='Pr';
 Label13.Left:=210;
 Label13.Top:=Alt2;
 Label13.Caption:='M';
 Label14.Left:=225;
 Label14.Top:=Alt2;
 Label14.Caption:='Ps';
 end;
end;{ End Case }
TabAnterior :=TabSet1.TabIndex;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite insertar líneas de texto en el editor 1 o de Nivel físico.Este método permite insertar las líneas de texto de manera que no se satura la memoria.El máximo de líneas admitidos viene determinado por la variable Maxlineas .Una vez que se llega a este número de líneas este procedimiento sobreescribe la línea primera y a continuación la siguiente.El valor de MaxLineas se inicia como 200.

```
Procedure TAnalizador.InsLinea1( Linea:String );
Var N : Integer;
begin
For N:=1 to length( linea ) do
 begin
 If Linea[ N ]=Char( 10 ) then Linea[ N ]:='@';
 If Linea[ N ]=Char( 13 ) then Linea[ N ]:='$';
 end;
If Memo1.Lines.Count>=MaxLineas then
 begin
 If Puntero1>MaxLineas-1 then Puntero1:=0;
 Memo1.Lines[ Puntero1 ]:=Linea;
 If Puntero1=MaxLineas-1 then
  Memo1.Lines[Puntero1+1]:=' '
 Else
  Memo1.Lines[Puntero1+1]:='>>';
  Inc( Puntero1 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo1.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite insertar líneas de texto en el editor 2 o de Nivel 2.Este método permite insertar las líneas de texto de manera que no se sature la memoria.El máximo de líneas admitidos viene determinado por la variable Maxlineas .Una vez que se llega a este número de líneas este procedimiento sobreescribe la línea primera y a continuación la siguiente.El valor de MaxLineas se inicia como 200.

```
Procedure TAnalizador.InsLinea2( Linea:String );
begin
If Memo2.Lines.Count>=MaxLineas then
 begin
 If Puntero2>MaxLineas-1 then Puntero2:=0;
 Memo2.Lines[ Puntero2 ]:=Linea;
 If Puntero2=MaxLineas-1 then
  Memo2.Lines[Puntero2+1]:=' '
 Else
  Memo2.Lines[Puntero2+1]:='>>';
  Inc( Puntero2 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo2.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite insertar líneas de texto en el editor 3 o de Nivel 3.Este método permite insertar las líneas de texto de manera que no se sature la memoria.El máximo de líneas admitidos viene determinado por la variable Maxlineas .Una vez que se llega a este número de líneas este procedimiento sobreescribe la línea primera y a continuación la siguiente.El valor de MaxLineas se inicia como 200.

```
Procedure TAnalizador.InsLinea3( Linea:String );
begin
If Memo3.Lines.Count>=MaxLineas then
 begin
 If Puntero3>MaxLineas-1 then Puntero3:=0;
 Memo3.Lines[ Puntero3 ]:=Linea;
 If Puntero3=MaxLineas-1 then
  Memo3.Lines[Puntero3+1]:=' '
 Else
  Memo3.Lines[Puntero3+1]:='>>';
 Inc( Puntero3 );
 end
Else { If Memo1.Lines.Count<MaxLineas }
 begin
 Memo3.Lines.Add( Linea );
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite insertar líneas de texto en el editor 4 o de Todos los Niveles . Este método permite insertar las líneas de texto de manera que no se sature la memoria.El máximo de líneas admitidos viene determinado por la variable Maxlineas .Una vez que se llega a este número de líneas este procedimiento sobreescribe la línea primera y a continuación la siguiente.El valor de MaxLineas se inicia como 200.

```
Procedure TAnalizador.InsLinea4( Linea:String );
Var N : Integer;
begin
For N:=1 to length( linea ) do
 begin
 If Linea[ N ]=Char( 10 ) then Linea[ N ]:='@';
 If Linea[ N ]=Char( 13 ) then Linea[ N ]:='$';
 end;
If Memo4.Lines.Count>=MaxLineas then
 begin
 If Puntero4>MaxLineas-1 then Puntero4:=0;
 Memo4.Lines[ Puntero4 ]:=Linea;
 If Puntero4=MaxLineas-1 then
  Memo4.Lines[Puntero4+1]:=' '
 Else
  Memo4.Lines[Puntero4+1]:='>>';
 Inc( Puntero4 );
 end
Else { If Memo4.Lines.Count<MaxLineas }
 begin
 Memo4.Lines.Add( Linea );
 end;
{ Si la captura esta activada se guarda en el archivo }
If Capturar1.Checked=True then
 Writeln( ACaptura,Linea );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Este Método aparece reflejado en la opción "Parar" del menú del Analizador. Detiene la captura de tramas en el Analizador, sin que estas sean ni presentadas en el Analizador ni guardadas en memoria.

```
procedure TAnalizador.Parar1Click(Sender: TObject);
begin
If Cursor=crDefault then
 begin
 If Parar1.Checked=False then
  begin
  Parar1.Checked:=True;
  end
 Else
  begin
  Parar1.Checked:=False;
  end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 40 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Llamada a la presentación del Nivel Físico desde el Menú del Analizador.Produce el mismo evento que si se seleccionara la lengüeta correspondiente a este Nivel con el Ratón. Se tiene en cuenta tambien la opción de ayuda referente a esta selección del menú

```
procedure TAnalizador.Fisico1Click(Sender: TObject);
begin
If cursor=crDefault then
 begin
 Cambio( 0 );
 TabSet1.TabIndex:=0;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 41 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Llamada a la presentación del Nivel de Enlace desde el Menú del Analizador.Produce el mismo evento que si se seleccionara la lengüeta correspondiente a este Nivel con el Ratón. Se tiene en cuenta tambien la opción de ayuda referente a esta selección del menú

```
procedure TAnalizador.Enlace1Click(Sender: TObject);
begin
If Cursor=crDefault then
 begin
 Cambio( 1 );
 TabSet1.TabIndex:=1;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 42 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Llamada a la presentación del Nivel de Red desde el Menú del Analizador.Produce el mismo evento que si se seleccionara la lengüeta correspondiente a este Nivel con el Ratón. Se tiene en cuenta tambien la opción de ayuda referente a esta selección del menú

```
procedure TAnalizador.Red1Click(Sender: TObject);
begin
If Cursor=crDefault then
 begin
 Cambio( 2 );
 TabSet1.TabIndex:=2;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 43 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Llamada a la presentación del Todos los Niveles desde el Menú del Analizador.Produce el mismo evento que si se seleccionara la lengüeta correspondiente a este Nivel con el Ratón. Se tiene en cuenta tambien la opción de ayuda referente a esta selección del menú

```
procedure TAnalizador.Todos1Click(Sender: TObject);
begin
If Cursor=crDefault then
 begin
 Cambio( 3 );
 TabSet1.TabIndex:=3;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 44 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que invoca la ventana de selección de condiciones de Filtro de la Unidad "ANFIL".Desde esta ventana se pueden modificar las variables que contienen las condiciones de filtrado de los recibido por el Analizador.Inicialmente todos los filtros estan activos de manera que se permite que se muestren todos los sentidos de la transmisión ,todos los tipos de tramas y todos los tipos de paquetes.

```
procedure TAnalizador.N21Click(Sender: TObject);
begin
If Cursor=crDefault then
 begin
 With Filtros do
  begin
  ChDTE.Checked :=FDTE;
  ChDCE.Checked :=FDCE;
  ChSabm.Checked:=FSABM;
  ChUa.Checked:=FUA;
  ChDisc.Checked:=FDISC;
  ChDM.Checked:=FDM;
  ChRR.Checked:=FRR;
  ChRNR.Checked:=FRNR;
  ChINFO.Checked:=FINFO;
  ChFRMR.Checked:=FFRMR;
  CHSLLam.Checked:=FSLLAM;
  ChALLam.Checked:=FALLAM;
  ChSlibe.Checked:=FSLIBE;
  ChALibe.Checked:=FALIBE;
  ChSRein.Checked:=FSREIN;
  ChARein.Checked:=FAREIN;
  ChSInte.Checked:=FSINTE;
  ChAInte.Checked:=FAINTE;
  ChSRear.Checked:=FSREAR;
  ChARear.Checked:=FAREAR;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 ChRRN3.Checked:=FRRN3;
 ChRNRN3.Checked:=FRNRN3;
 ChDatos.Checked:=FDATA;
 end;{ End With}
Filtros.ShowModal;
If Filtros.ModalResult=mrOk then
 begin
 With Filtros do
  begin
   FDTE  :=ChDTE.Checked;
   FDCE  :=ChDCE.Checked;
   FSABM :=ChSabm.Checked;
   FUA   :=ChUa.Checked;
   FDISC :=ChDisc.Checked;
   FDM   :=ChDM.Checked;
   FRR   :=ChRR.Checked;
   FRNR  :=ChRNR.Checked;
   FINFO :=ChINFO.Checked;
   FFRMR :=ChFRMR.Checked;
   FSLLAM:= CHSLLam.Checked;
   FALLAM:= ChALLam.Checked;
   FSLIBE:= ChSlibe.Checked;
   FALIBE:= ChALibe.Checked;
   FSREIN:= ChSRein.Checked;
   FAREIN:= ChARein.Checked;
   FSINTE:= ChSInte.Checked;
   FAINTE:= ChAInte.Checked;
   FSREAR:= ChSRear.Checked;
   FAREAR:= ChARear.Checked;
   FRRN3 := ChRRN3.Checked;
   FRNRN3:= ChRNRN3.Checked;
   FDATA := ChDatos.Checked;
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
   end;{ End With }
  end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 45 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite guardar el contenido del Nivel Físico en un Archivo de texto.Se presenta una ventana de diálogo que permite seleccionar el nombre del Archivo y el subdirectorio.Por defecto pone la extensión "FIS" al archivo y lo guarda en el subdirectorio configurado en el programa principal como "Analizadores"

```
procedure TAnalizador.NivelFisico1Click(Sender: TObject);
Const NombreArchivo : TFileName='*.FIS';
begin
If Cursor=crDefault then
 begin
  with SaveDialog1 do
   begin
   InitialDir:=Direct;
   Filename:=NombreArchivo;
   Filter:='Nivel Fisico *.FIS|*.FIS|'+
         'Nivel de Enlace *.ENL|*.ENL|'+
         'Nivel de RED *.RED|*.RED|'+
         'Todos los Niveles *.TOD|*.TOD|'+
         'Todos *.*|*.*';
    if Execute then
    begin
      Memo1.Lines.SaveToFile( Filename);
      NombreArchivo:=FileName;
    end;
   end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 46 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite guardar el contenido del Nivel 2 en un Archivo de texto.Se presenta una ventana de diálogo que permite seleccionar el nombre del Archivo y el subdirectorio.Por defecto pone la extensión "ENL" al archivo y lo guarda en el subdirectorio configurado en el programa principal como "Analizadores"

```
procedure TAnalizador.NiveldeEnlace1Click(Sender: TObject);
Const NombreArchivo : TFileName ='*.ENL';
begin
If Cursor=crDefault then
 begin
  with SaveDialog1 do
   begin
   InitialDir:=Direct;
   FileName:=NombreArchivo;
   Filter:='Nivel de Enlace *.ENL|*.ENL|'+
         'Nivel de RED *.RED|*.RED|'+
         'Todos los Niveles *.TOD|*.TOD|'+
         'Todos *.*|*.*|'+
         'Nivel Fisico *.FIS|*.FIS';
    if Execute then
    begin
      Memo2.Lines.SaveToFile( Filename);
      NombreArchivo:=FileName;
    end;
   end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 47 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite guardar el contenido del Nivel 3 en un Archivo de texto.Se presenta una ventana de diálogo que permite seleccionar el nombre del Archivo y el subdirectorio.Por defecto pone la extensión "RED" al archivo y lo guarda en el subdirectorio configurado en el programa principal como "Analizadores"

```
procedure TAnalizador.NiveldeRed1Click(Sender: TObject);
Const NombreArchivo : TFileName='*.RED';
begin
If Cursor=crDefault then
 begin
  with SaveDialog1 do
   begin
   InitialDir:=Direct;
   FileName:=NombreArchivo;
   Filter:='Nivel de RED *.RED|*.RED|'+
         'Todos los Niveles *.TOD|*.TOD|'+
         'Todos *.*|*.*|'+
         'Nivel Fisico *.FIS|*.FIS|'+
         'Nivel de Enlace *.ENL|*.ENL';
    if Execute then
    begin
      Memo3.Lines.SaveToFile( Filename);
      NombreArchivo:=FileName;
    end;
   end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 48 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que permite guardar el contenido de "Todos los Niveles" en un Archivo de texto.Se presenta una ventana de diálogo que permite seleccionar el nombre del Archivo y el subdirectorio.Por defecto pone la extensión "TOD" al archivo y lo guarda en el subdirectorio configurado en el programa principal como "Analizadores"

```
procedure TAnalizador.TodoslosNiveles1Click(Sender: TObject);
Const NombreArchivo : TFileName = '*.TOD';
begin
If Cursor=crDefault then
 begin
  with SaveDialog1 do
   begin
   InitialDir:=Direct;
   FileName:=NombreArchivo;
   Filter:='Todos los Niveles *.TOD|*.TOD|'+
         'Todos *.*|*.*|'+
         'Nivel Fisico *.FIS|*.FIS|'+
         'Nivel de Enlace *.ENL|*.ENL|'+
         'Nivel de RED *.RED|*.RED';
    if Execute then
    begin
      Memo4.Lines.SaveToFile( Filename);
      NombreArchivo:=FileName;
    end;
   end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 49 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Inicia la Captura en un Archivo de texto.A partir de su invocación se selecciona un nombre de archivo y un subdirectorio, y a partir de aqui todo lo capturado en el editor que representa "Todos los Niveles" se guardará , a la vez en dicho archivo. Si se llama de nuevo a la misma opción se termina la captura a dicho archivo y se cierra este.

```
procedure TAnalizador.Capturar1Click(Sender: TObject);
Const NombreArchivo : TFileName = '*.TOD';
begin
If Cursor=crDefault then
 begin
 If Capturar1.Checked=False then
  begin
  Capturar1.Checked:=True;
   with SaveDialog1 do
   begin
   Title:='Capturar Archivos';
   InitialDir:=Direct;
   FileName:=NombreArchivo;
   Filter:='Todos los Niveles *.TOD|*.TOD|'+
         'Todos *.*|*.*';
  if Execute then
    begin
     Memo4.Lines.SaveToFile( Filename);
     NombreArchivo:=FileName;
     AssignFile( ACaptura,Filename );
     ReWrite( ACaptura );
    end
   Else
    begin
    Capturar1.Checked:=False;
    end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
   end;
 end
 Else
  begin
  Capturar1.Checked:=False;
  CloseFile( ACaptura );
  SHowMessage( 'Parada la captura en Archivo '+NombreArchivo );
  end;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 50 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Elimina todo el contenido de los cuatro editores de texto que constituyen el analizador.

```
procedure TAnalizador.LimpiarAnalizador1Click(Sender: TObject);
begin
If Cursor=crDefault then
 begin
 Memo1.Clear;
 Puntero1:=0;
 Memo2.Clear;
 Puntero2:=0;
 Memo3.Clear;
 Puntero3:=0;
 Memo4.Clear;
 Puntero4:=0;
 end;
If Cursor =crDrag then RedX25.LlamaAyuda( 51 );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que invoca la ventana de diálogo de la unidad "Tram". Esta unidad crea una ventana de diálogo para la edición de condiciones de disparo de la Trampa y eventos que esta debe producir.

```
procedure TAnalizador.Trampas1Click(Sender: TObject);
begin
Trampas.Showmodal;
If Trampas.ModalResult=mrOk then
 begin
 VTrampas.Dir:=Trampas.Edit2.Text;;
 Vtrampas.Cont2:=Trampas.Edit3.Text;
 VTrampas.Cont2MB:=Trampas.Edit4.Text;
 VTrampas.IGF:=Trampas.Edit5.Text;
 VTrampas.IGFMB:=Trampas.Edit6.Text;
 VTrampas.Can:=Trampas.Edit7.Text;
 VTrampas.Cont3:=Trampas.Edit8.Text;
 VTrampas.Cont3MB:=Trampas.Edit9.Text;
 VTrampas.PararAnal := Trampas.CheckBox4.Checked;
 VTrampas.ComenzarAnal :=Trampas.CheckBox3.Checked;
 VTrampas.ComenzarCaptura :=Trampas.CheckBox1.Checked;
 VTrampas.PararCaptura :=Trampas.CheckBox5.Checked;
 VTrampas.VentanaAviso :=Trampas.CheckBox2.Checked;
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que se torna cierto si la Trama que se le entrega como parámetro ,cumple las condiciones , de la trampa que se solicitan en la ventana de selección de condiciones de disparto de la Trampa.

```
Function TAnalizador.EventoTrampa( Frame:String ):Boolean;
Var Act1,act2,act3,act4,act5 : Boolean;

begin
EventoTrampa:=False;
Act1:=False;
Act2:=False;
Act3:=False;
Act4:=False;
Act5:=False;

if ( VTrampas.Dir<>'XX' )or(VTrampas.Cont2<>'XX')or
  (VTrampas.IGF<>'XX')or(VTrampas.Can<>'XX')or
  (VTrampas.Cont3<>'XX')or(VTrampas.Cont2MB<>'XXXXXXXX')
  or(VTrampas.IGFMB<>'XXXXXXXX')or(VTrampas.Cont3MB<>'XXXXXXXX')
then
 begin
 With Vtrampas do
 begin

 { Direccion Dir }
 If Dir ='XX' then Act1:=True
 Else If Dir=Copy( Frame,8,2) then Act1:=true
    Else Act1:=False;

 { Control de nivel 2 }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
if Cont2='XX' then
 begin
 If Cont2MB='XXXXXXXX' then Act2:=true
 Else If ComparaMascara( Cont2MB, Copy( Frame,10,2) ) then Act2:=true
     Else Act2:=False;
 end
Else If Cont2=Copy( Frame,10,2) then Act2:=true
    Else Act2:=False;

If HD( Copy( Frame, 10, 2 ) )mod 2 = 0 then
 begin
 { Identificativo general de formato }
 if IGF='XX' then
  begin
  If IGFMB='XXXXXXXX' then Act3:=true
  Else If ComparaMascara( IGFMB, Copy( Frame,12,2) ) then Act3:=true
      Else Act3:=False;
  end
 Else If IGF=Copy( Frame,12,2) then Act3:=true
     Else Act3:=False;

 { Canal}
 If Can ='XX' then Act4:=True
 Else If Can=Copy( Frame,14,2) then Act4:=true
     Else Act4:=False;

 { Control de nivel 3}
 if Cont3='XX' then
  begin
  If Cont3MB='XXXXXXXX' then Act5:=true
  Else If ComparaMascara( Cont3MB, Copy( Frame,16,2) ) then Act5:=true
      Else Act5:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 end
 Else If Cont3=Copy( Frame,16,2) then Act5:=true
     Else Act5:=False;

 end{ si es un info }
Else
 begin
 If ( IGF='XX' )and( IGFMB='XXXXXXXX' ) then Act3:=True;
 If Can='XX' then  Act4:=True;
 If ( Cont3='XX' )and( Cont3MB='XXXXXXXX')then Act5:=True;
 end;
end;{ With }

If Act1 and act2 and act3 and act4 and act5 then EventoTrampa :=True;

end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Método que compara la mascara de bit , con el octeto de la trama recibida. Si la condición de la mascara de bit ser cumple la función se torna cierta, en caso contrario es falsa.

```
Function TAnalizador.ComparaMascara( Mascara,Octeto :String ):Boolean;
Var T : String;
   N : Integer;
   Resultado : Boolean;
begin
Resultado:=true;
{ pasa a binario la expresion en hexadecimal}
T:=HexBin( Octeto );
For N:=1 to 8 do
 begin
 { si cualquiera de los caracteres de la cadena de ocho bit}
 { no coincide y el caracter de la mascara no es X da el resultado}
 { de la comparacion como falso }
 If ( T[ N ]<> Mascara[ N ] )and( Mascara[ N ]<>'X' ) then
    begin
    Resultado:=False;
    end;
 end;
ComparaMascara :=Resultado;
end;
end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad AnFil

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

unit Anfil;

Unidad en la que se define el objeto que construye una ventana de diálogo en la que se definen las condiciones de Filtro del Analizador de Protocolo.

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,

Forms, Dialogs, StdCtrls, ExtCtrls;

type

TFiltros = class(TForm)

Panel1: TPanel;

Button1: TButton;

ChSabm: TCheckBox;

ChUa: TCheckBox;

ChDisc: TCheckBox;

ChDM: TCheckBox;

ChRR: TCheckBox;

ChRNR: TCheckBox;

ChINFO: TCheckBox;

ChFRMR: TCheckBox;

Label1: TLabel;

CHSLLam: TCheckBox;

ChALLam: TCheckBox;

ChSlibe: TCheckBox;

ChALibe: TCheckBox;

ChSRein: TCheckBox;

ChARein: TCheckBox;

ChSInte: TCheckBox;

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    ChAInte: TCheckBox;
    ChSRear: TCheckBox;
    ChARear: TCheckBox;
    ChRRN3: TCheckBox;
    ChRNRN3: TCheckBox;
    ChDatos: TCheckBox;
    Label2: TLabel;
    Panel2: TPanel;
    Label3: TLabel;
    ChDTE: TCheckBox;
    ChDCE: TCheckBox;
    procedure ChINFOClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Filtros: TFiltros;

implementation

{$R *.DFM}
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Si se elimina el permiso para que las tramas "Info" sean presentadas, debe eliminarse automáticamente la opción de presentarse todos los Paquetes de Nivel 3. De igual manera, si se activa la opción de filtrado de las Tramas "Info" deben activarse también todas las opciones de los Paquetes de Nivel 3. Esto es lo que realiza el siguiente método.

```
procedure TFiltros.ChINFOClick(Sender: TObject);
begin
{ Quita la seleccion de nivel 3 si se elimina el INFO del filtro }
If ChInfo.Checked=true then
 begin
  CHSLLam.Checked:=True;
  ChALLam.Checked:=True;
  ChSlibe.Checked:=True;
  ChALibe.Checked:=True;
  ChSRein.Checked:=True;
  ChARein.Checked:=True;
  ChSInte.Checked:=True;
  ChAInte.Checked:=True;
  ChSRear.Checked:=True;
  ChARear.Checked:=True;
  ChRRN3.Checked:=True;
  ChRNRN3.Checked:=True;
  ChDatos.Checked:=True;
 end
Else
 begin
 CHSLLam.Checked:=False;
 ChALLam.Checked:=False;
 ChSlibe.Checked:=False;
 ChALibe.Checked:=False;
 ChSRein.Checked:=False;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
ChARein.Checked:=False;
ChSInte.Checked:=False;
ChAInte.Checked:=False;
ChSRear.Checked:=False;
ChARear.Checked:=False;
ChRRN3.Checked:=False;
ChRNRN3.Checked:=False;
ChDatos.Checked:=False;
end;
end;

end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad Tram

© Del documento, los autores. Digitalización realizada por ULPGC. Bibliotecas Universitaria, 2006

unit Tram;

Unidad en la que se constituye la ventana de diálogo que permite seleccionar y activar una trampa en el Analizador de protocolo. En este objeto se generan una serie de objetos visuales para la edición de los parámetros que activarán la trampa y unos métodos que permitirán seleccionar, de manera adecuada, las opciones a tomar una vez que se produce el evento esperado.

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,

Forms, Dialogs, StdCtrls, ExtCtrls;

type

TTrampas = class(TForm)

GroupBox1: TGroupBox;

GroupBox2: TGroupBox;

Panel1: TPanel;

Button1: TButton;

Panel2: TPanel;

Panel3: TPanel;

Panel4: TPanel;

Label3: TLabel;

Edit3: TEdit;

Label4: TLabel;

Edit4: TEdit;

Panel5: TPanel;

Label2: TLabel;

Edit2: TEdit;

Panel6: TPanel;

Label5: TLabel;

Label6: TLabel;

Panel7: TPanel;

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    Label7: TLabel;
    Panel8: TPanel;
    Label9: TLabel;
    Label8: TLabel;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Edit9: TEdit;
    Panel9: TPanel;
    CheckBox4: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox1: TCheckBox;
    CheckBox5: TCheckBox;
    CheckBox2: TCheckBox;
    Label1: TLabel;
    Edit1: TEdit;
    procedure CheckBox4Click(Sender: TObject);
    procedure CheckBox3Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Trampas: TTrampas;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Los siguientes métodos impiden que se seleccionen a la vez las opciones "Parar Analizador" y "Comenzar Analizador", o "Comenzar Captura" y "Parar Captura".

```
implementation

{$R *.DFM}

procedure TTrampas.CheckBox4Click(Sender: TObject);
begin
If ( ChecKbox3.Checked=True )and( ChecKbox4.Checked=True )then
  ChecKbox3.Checked:=False;
end;

procedure TTrampas.CheckBox3Click(Sender: TObject);
begin
If ( ChecKbox3.Checked=True )and( ChecKbox4.Checked=True )then
  ChecKbox4.Checked:=False;
end;

procedure TTrampas.CheckBox1Click(Sender: TObject);
begin
If ( ChecKbox1.Checked=True )and( ChecKbox5.Checked=True )then
  ChecKbox5.Checked:=False;

end;

procedure TTrampas.CheckBox5Click(Sender: TObject);
begin
If ( ChecKbox1.Checked=True )and( ChecKbox5.Checked=True )then
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  ChecKbox1.Checked:=False;

end;

end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Unidad Cálculo

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

unit Calculo;

Unidad encargada de crear la librería de procedimientos y funciones implicadas en el cálculo numérico. Estos procedimientos serán invocados por los distintos métodos de las unidades del programa.

```
interface
Uses SysUtils,Dialogs;

Function DH(D:Byte):String;
Function HD(H:String):Byte;
Function PrMPsDH( A,B,C : Byte ):String;
Function PrHD( H: String):Byte;
Function PrRRDH( A,B,C : Byte ):String;
Function MHD( H:String ):Byte;
Function PsHD( H:String ):Byte;
Function VerBitD( H:String ):Boolean;
Function PonerBitD( H:String ):String;
Function QuitarBitD( H:String ):String;
Function VerBitQ( H:String ):Boolean;
Function NrPRR( D,E,F :Byte) : String;
Function NrHD( K:String ):Byte;
Function NsHD( K:String ):Byte;
Function PHD( K:String ):Byte;
Function NrPNsDH( D,E,F : Byte ):String;
Function CRC( Trama : String ):String;
Function ConvCadAH( Cad : String ):String;
Function ConvCadHA( Cad : String ):String;
Function VerMod( Cad : String ):String;
Function INFOCRC( Trama : String ): String;
Function HexBin( H : String ): String;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

implementation

Función que recibe un octeto como parámetro y entrega como respuesta una cadena de dos caracteres como representación Hexadecimal del octeto entregado.

```
Function DH(D:Byte):String;
Var Alto,Bajo :Byte;
   AltoH,BajoH :Char;
begin
 Alto :=( D and 240 )div 16;
 Bajo :=( D and 15 );
 If Alto > 9 then  AltoH:=Char( Alto+55 )
 Else AltoH:=Char( Alto+48 );
 If Bajo > 9 then BajoH:=Char( Bajo+55 )
 Else BajoH:=Char( Bajo+48 );
 DH:=AltoH+BajoH;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que convierte en un valor decimal la cadena de texto de dos caracteres que representa el valor de un octeto en Hexadecimal.

```
Function HD(H:String):Byte;
Var AH,BH : Char;
   A,B : Byte;
begin
If H<>'' then
 begin
 AH:=H[1];
 BH:=H[2];
 If Ord( AH )>64 then A:=Ord( AH )-55
 Else A:=Ord( AH )-48;
 If Ord( BH )>64 then B:=Ord( BH )-55
 Else B:=Ord( BH )-48;
 HD:= 16*A+B;
end
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que crea el octeto de Control de un Paquete de Nivel 3 entregándole como parámetro .los valores decimales de las variables de la secuencia de transmisión, la secuencia de recepción y el bit M.

```
Function PrMPsDH( A,B,C : Byte ):String;
Var D : Byte;
begin
D:=( A*32 )or( B*16 )or( C*2 );
PrMPsDH:=DH( D );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que crea el octeto de control de un Paquete RR de Nivel 3 pasándole como parámetro la secuencia de recepción Pr .

```
Function PrRRDH( A,B,C : Byte ):String;
Var D : Byte;
begin
D:=( A*32 )or( B*16 )or( C );
PrRRDH:=DH( D );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que extrae el octeto Pr del byte de control de un Paquete de Nivel 3.

```
Function PrHD( H: String):Byte;
begin
PrHD:= ( HD( '0'+H[1] ) ) div 2;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que extrae el bit M del octeto de control de un paquete de Nivel 3.

```
{ Extraer M del Byte de Tipo }
Function MHD( H:String ):Byte;
begin
MHD:=HD( '0'+H[1] )and 1;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que extrae el valor de Ps del octeto de Control de un Paquete de Nivel 3.

```
Function PsHD( H:String ):Byte;
begin
PsHD:=HD( '0'+H[2] ) div 2
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que se activa si el bit D del octeto deL IGF de paquete de Nivel 3 está a uno. Si el bit D es cero, la Función no se activa.

```
Function VerBitD( H:String ):Boolean;
Var N:Byte;
   Si:Boolean;
begin
N:= HD( H ) and 64;
If N=0 then Si:=False;
If N=64 then Si:=True;
VerBitD:=Si;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que pone el bit D a uno dentro del IGF de un Paquete de Nivel 3.

```
Function PonerBitD( H:String ):String;
Var N : Byte;
begin
N:=HD( H );
N:=N or 64;
PonerBitD:=DH( N );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que pone el bit D a cero dentro del Identificativo General de Formato de un Paquete de Nivel 3.

```
Function QuitarBitD( H:String ):String;
Var N : Byte;
begin
N:=HD( H );
N:=N and 191;
QuitarBitD:=DH( N );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que muestra el valor del bit Q entregándoselo como parámetro a una cadena de texto que representa el valor en Hexadecimal del Identificativo General de Formato de un Paquete de Nivel 3.

```
Function VerBitQ( H:String ):Boolean;
Var N:Byte;
   Si:Boolean;
begin
N:= HD( H ) and 128;
If N=0 then Si:=False;
If N=128 then Si:=True;
VerBitQ:=Si;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Funciones que realizan las funciones de análisis del byte de control de las tramas de Nivel 2. Utilizan las mismas funciones que para el análisis de Nivel 3.

```
Function NrPRR( D,E,F :Byte) : String;
begin
NrPRR:=PrRRDH( D,E,F );
end;
Function NrHD( K:String ):Byte;
begin
NrHd:=PrHd( K );
end;
Function NsHD( K:String ):Byte;
begin
NsHD:=PsHD( K );
end;
Function PHD( K:String ):Byte;
begin
PHD:=MHD( K );
end;
Function NrPNsDH( D,E,F : Byte ):String;
begin
NrPNsDH:= PrMPsDH( D,E,F );
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función encargada de calcular los dos octetos que constituyen la secuencia de verificación de trama o CRC. La trama para la que se quiere realizar el cálculo se entrega como parámetro y la función entrega los dos octetos en Hexadecimal representados por una cadena de texto de cuatro caracteres. El algoritmo por el cual se calcula el resto de la división se ha sacado del libro "Comunicaciones Serie para PC" de la editorial Anaya Multimedia.

```
Function CRC( Trama : String ):String;

Const PGen : Word =$1021;

Var  Datos,Accum : Word;
    N,M       : Integer;
    Cad       : String;
begin
Accum:=0;
If ( Trama<>'' )and(Length( Trama ) Mod 2 =0 )then
 begin
 Trama:=Trama+'0000';
 Cad:=Trama;
 For N:=1 to Length( Trama )div 2 do
  begin
  Datos:=HD( Copy( Cad,1,2 ));
  Cad:=Copy( Cad,3,length( Cad )-2 );
  Datos:=Datos SHL 8;
  For M:=8 DownTo 1 do
   begin
   If (( Datos XOR Accum )AND $8000 )=$8000 then
     Accum:=( Accum SHL 1 )XOR PGen
   Else
     Accum:=Accum SHL 1;
   Datos:=Datos SHL 1;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
  end;{ End For M}
 end;{ End For N}
 CRC:=DH( Accum Div 256 )+DH( Accum Mod 256 );
end;{ End If }
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Convierte una cadena de texto ASCII en una representación de sus valores Hexadecimales.

```
Function ConvCadAH( Cad : String ):String;
Var HCad : String;
   N   : Integer;
   K   : Integer;
begin
HCad:='';
For N:=1 to length( Cad ) do
 begin
 K:=Ord( Cad[N] );
 HCad:=Hcad+DH( K );
 end;
ConvCadAH:=HCad;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Convierte unacadena de representación de valores Hexadecimales en su correspondiente representaci.n en código ASCII

```
Function ConvCadHA( Cad : String ):String;
Var CadA : String;
   CadH : String;
   N    : Integer;
   Hex  : String;
begin
CadA:='';
CadH:=Cad;
For N:=1 to Length( Cad )div 2 do
 begin
 Hex:=Copy( CadH,1,2);
 CadH:=Copy( CadH,3,Length( CadH )-2);
 CadA:=CadA+Char( HD( Hex ));
 end;
ConvCadHA:=CadA;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Permite ver el Módulo del Identificativo General de Formato de un Paquete de Nivel 3.

```
Function VerMod( Cad : String ):String;
begin
If length( Cad )=2 then
 begin
 If HD( '0'+copy(Cad,1,1 )) mod 2 =0 then
        VerMod:='10'
 Else
        VerMod:='01';
 end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Calcula el CRC de una Trama INFO en la que su contenido de Datos está representado, no en hexadecimal, sino en la representación ASCII de los caracteres de texto. Esta limitación en la representación viene dada por la limitación de la variable "STRING" del lenguaje Pascal de Delphi a 256 caracteres.

```
Function INFOCRC( Trama : String ): String;
Const PGen : Word =$1021;
Var  Datos,Accum : Word;
    Tram      : String;
    N,M       : Integer;
    Cad       : String;
begin
InFoCRC:='';
If ( HD(Copy( Trama,9,2 )) Mod 2 =0 )and(Copy( Trama,7,2 )<>'FF')then
 begin
 { Es un Data de nivel 3 no datagrama }
 Tram:=Copy( Trama,1,10 );
 Accum:=0;
If ( Tram<>'' )and(Length( Tram ) Mod 2 =0 )then
 begin
 Cad:=Tram;
 For N:=1 to Length( Tram )div 2 do
  begin
  Datos:=HD( Copy( Cad,1,2 ));
  Cad:=Copy( Cad,3,length( Cad )-2 );
  Datos:=Datos SHL 8;
  For M:=8 DownTo 1 do
   begin
   If (( Datos XOR Accum )AND $8000 )=$8000 then
     Accum:=( Accum SHL 1 )XOR PGen
   Else
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
    Accum:=Accum SHL 1;
   Datos:=Datos SHL 1;
   end;{ End For M}
  end;{ End For N}
 end;{ End If }

Tram:=Copy( Trama,11,length( Trama )-10 );
If ( Tram<>'' )then
 begin
 Tram:=Tram+Char(0)+Char(0);
 Cad:=Tram;
 For N:=1 to Length( Tram ) do
  begin
  Datos:=Ord( Cad[1]);
  Cad:=Copy( Cad,2,length( Cad )-1 );
  Datos:=Datos SHL 8;
  For M:=8 DownTo 1 do
   begin
   If (( Datos XOR Accum )AND $8000 )=$8000 then
     Accum:=( Accum SHL 1 )XOR PGen
   Else
     Accum:=Accum SHL 1;
   Datos:=Datos SHL 1;
   end;{ End For M}
  end;{ End For N}
  INFOCRC:=DH( Accum Div 256 )+DH( Accum Mod 256 );
 end;{ End If }
 end
Else { Si no es un data de nivel 3 y es un data de Datagrama }
 begin
 If ( Copy( Trama,7,2 ) ='FF' )and( HD(Copy( Trama,29,2 ))MOd 2 =0 )
    and( HD(Copy( Trama,9,2 ))MOD 2 = 0 )
```


© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 and(Copy( Trama,27,2 )<>'0B')
 and(Copy( Trama,27,2 )<>'0F')
 and(Copy( Trama,27,2 )<>'13')
 and(Copy( Trama,27,2 )<>'17')
 and(Copy( Trama,27,2 )<>'23')
 and(Copy( Trama,27,2 )<>'27')
 and(Copy( Trama,27,2 )<>'1B')
 and(Copy( Trama,27,2 )<>'1F')
 and(Copy( Trama,27,2 )<>'FB')
 and(Copy( Trama,27,2 )<>'FF')then
begin
 Tram:=Copy( Trama,1,30 );
 Accum:=0;
If ( Tram<>'' )and(Length( Tram ) Mod 2 =0 )then
 begin
 Cad:=Tram;
 {showmessage( Tram );}
 For N:=1 to Length( Tram )div 2 do
  begin
  Datos:=HD( Copy( Cad,1,2 ));
  Cad:=Copy( Cad,3,length( Cad )-2 );
  Datos:=Datos SHL 8;
  For M:=8 DownTo 1 do
   begin
   If (( Datos XOR Accum )AND $8000 )=$8000 then
    Accum:=( Accum SHL 1 )XOR PGen
   Else
    Accum:=Accum SHL 1;
   Datos:=Datos SHL 1;
   end;{ End For M}
  end;{ End For N}
 end;{ End If Trama<>'' }
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

```
 Tram:=Copy( Trama,31,length( Trama )-30 );
 {showmessage( IntToStr(Length( Tram ))+' '+tram );}
If ( Tram<>'' )then
 begin
 Tram:=Tram+Char(0)+Char(0);
 Cad:=Tram;
 For N:=1 to Length( Tram ) do
  begin
   Datos:= Ord( Cad[ 1 ] );
   Cad:=Copy( Cad,2,length( Cad )-1 );
   Datos:=Datos SHL 8;
   For M:=8 DownTo 1 do
    begin
    If (( Datos XOR Accum )AND $8000 )=$8000 then
      Accum:=( Accum SHL 1 )XOR PGen
    Else
      Accum:=Accum SHL 1;
    Datos:=Datos SHL 1;
    end;{ End For M}
  end;{ End For N}
INFOCRC:=DH( Accum Div 256 )+DH( Accum Mod 256 );
end;{ End IF }
end { End if canal =FF }
Else { Si no es un Data de nivel 3 ni un data del datagrama }
begin
Tram:=Trama+'0000';
INFOCRC:=CRC( Tram );
end;
end;
end;
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Función que convierte un valor hexadecimal, representado por dos caracteres de texto, en su representación binaria, como una cadena de ocho bits.

```
Function HexBin( H : String ): String;
Var Final : String;
   N : Integer;
begin
Final:='';
For N:=1 to 2 do
 begin
 If H[ N ]='0' then Final:=Final+'0000';
 If H[ N ]='1' then Final:=Final+'0001';
 If H[ N ]='2' then Final:=Final+'0010';
 If H[ N ]='3' then Final:=Final+'0011';
 If H[ N ]='4' then Final:=Final+'0100';
 If H[ N ]='5' then Final:=Final+'0101';
 If H[ N ]='6' then Final:=Final+'0110';
 If H[ N ]='7' then Final:=Final+'0111';
 If H[ N ]='8' then Final:=Final+'1000';
 If H[ N ]='9' then Final:=Final+'1001';
 If H[ N ]='A' then Final:=Final+'1010';
 If H[ N ]='B' then Final:=Final+'1011';
 If H[ N ]='C' then Final:=Final+'1100';
 If H[ N ]='D' then Final:=Final+'1101';
 If H[ N ]='E' then Final:=Final+'1110';
 If H[ N ]='F' then Final:=Final+'1111';
 end;
 HexBin:=Final;
end;
end.
```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Anexo 2

Bibliografía

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Bibliografía.

1)

TITULO:TELEINFORMÁTICA Y REDES DE COMPUTADORES

- **AUTOR**: A.ALABAU Y J.RIERA
- **EDITORIAL**: MARCOMBO

2)

TITULO:RECOMENDACIÓN X.25 MIVEL 3 .PROCEDIMIENTO DE CONTROL DE PAQUETES (CCITT84)

- **AUTOR**: SUBDIRECCIÓN GENERAL DE TECNOLOGÍA
- **EDITORIAL**:DOCUMENTACIÓN INTERNA DE TELEFÓNICA

3)

TITULO:RECOMENDACIÓN X.25 MIVEL 2 .PROCEDIMIENTO DE CONTROL DEL ENLACE

- **AUTOR**: SUBDIRECCIÓN GENERAL DE TECNOLOGÍA
- **EDITORIAL**:DOCUMENTACIÓN INTERNA DE TELEFÓNICA

4)

TITULO:REDES DE COMUNICACIONES

- **AUTOR**:JOSÉ MANUEL HUIDOBRO
- **EDITORIAL**:PARANINFO

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

5)

TITULO:REDES DE ORDENADORES,PROTOCOLOS, NORMAS E INTERFACES

- **AUTOR**: UYLESS BLACK
- **EDITORIAL** : RA-MA

6)

TITULO: INTRODUCCIÓN A LA TECNOLOGÍA Y DISEÑO DE SISTEMAS DE COMUNICACIONES Y REDES DE ORDENADORES

- **AUTOR**: JOHN FREER
- **EDITORIAL**: ANAYA MULTIMEDIA

7)

TITULO:LAN TIMES.ENCICLOPEDIA DE REDES.NETWORKING

- **AUTOR**: TOM SHELDON
- **EDITORIAL**: MCGRAW-HILL

8)

TITULO: PROGRAMACIÓN ORIENTADA A OBJETOS

- **AUTOR**: MANUEL ALFONSECA Y ALFONSO ALCALÁ
- **EDITORIAL** : ANAYA MULTIMEDIA

9)

TITULO: DELPHI 1.0. GUIA DE USUARIO(Manual del Programa)

- **AUTOR**:BORLAND
- **EDITORIAL**:BORLAND

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

10)

TITULO: COMUNICACIONES SERIE.GUIA DE REFERENCIAS DEL PROGRAMADOR EN C

AUTOR: JOE CAMPBELL

EDITORIAL: ANAYA MULTIMEDIA

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Anexo 3

Glosario

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Arrastrar. Señalar un elemento de la pantalla y mantener pulsado el botón (normalmente el izquierdo) del ratón mientras se desplaza el ismo.

Bit. (Dígito Binario). La unidad más pequeña de información reconocida por el ordenador. Posee únicamente dos valores en un sistea de mumeración binario: 1 ó 0. Un archivo almacenado en disco no es más que una serie de dígitos binarios.

Hacer clic. Señalar a un elementos de la pantalla, pulsar y soltar rápidamente el botón del ratón o dispositivo de señalización. Al pulsar el botón se percibe un clic.

Hacer doble clic. Señalar a un elementos de la pantalla, pulsar y soltar rápidamente dos veces seguidas el botón del ratón o dispositivo de señalización (normalmente el izquierdo).

Icono. Gráfico o símbolo empleado para representar un archivo, unidad de disco, herramienta o comando.

Imagen digital. Imagen que ha adoptado un formato numérico binario para que el ordenador la pueda leer y manipular.

Megabyte. (Mb. También llamado megaocteto). En términos de memoia de ordenador, aproximadamente un millón de caracteres o mil Kb.

BMP. Formato de archivo gráfico utilizado en los archivos auxiliares de consulta del **Simulador X25**. Es un formato de archivo estándar en Windows y OS/2.

Carácter. Letra, número, signo de puntuación o símbolo.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Tipo de letra. Todos los caracteres de una familia de tipos de letra, dentro de la cual puede haber distintas variedades, como normal, negrita, cursiva, etc.

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006