

3
2ej



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE ENVIO DE APLICACIONES

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION

P R E S E N T A N

ALEMAN CRESPO JOSE HUGO
RODRIGUEZ GUTIERREZ IGNACIO
VALEZ MAYA ALBERTO

DIRECTOR DE TESIS,
ACTUARIO SERGIO CASTRO RESINES



MEXICO, D. F.

FALLA DE COPIA

1990



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE TEMATICO

	PAG.
I.- INTRODUCCION.	1
II.- ANALISIS.	2
II.1 Definición del problema.	2
II.2 Análisis de requerimientos.	4
II.3 Alternativas de solución.	4
II.4 Solución propuesta.	5
II.4.1 Especificaciones de hardware y software.	5
II.5 Análisis	7
II.5.1 Selección de la aplicación a enviar.	7
II.5.2 Preparación de envío.	7
II.5.3 Activación.	9
II.5.4 Depuración y restauración.	9
II.5.5 Comunicaciones.	10
II.6 Diccionario de datos.	15
III.- DISEÑO.	17
IV.- DESARROLLO.	27
IV.1 Programación.	27
V.- IMPLANTACION Y PRUEBAS.	29
VI.- ENLACE MICRO-MAINFRAME.	31
VI.1 Ambiente Mainframe.	31
VI.1.1 Sistema de comunicación de datos.	31
VI.1.2 Controladores de comunicación.	33
VI.1.3 Manejador de base de datos.	34
VI.1.4 Comunicaciones.	36
VI.2 Ambiente micro.	38
VI.2.1 Norma RS232.	38
VI.2.2 Modems.	53
VI.2.3 Comandos HAYES.	55
VI.2.4 Comunicaciones.	57
VI.3 Protocolo utilizado.	68
VII.- INTEGRACION DEL SISTEMA EN OSI.	78
VIII.- CONCLUSIONES.	81

APENDICES.	82
A.1 Sistema operativo DOS para PC.	83
A.2 Sistema Operativo MVS para Mainframe.	88
A.3 Interrupciones.	92
A.4 Diagramas de flujo de datos.	96
A.5 Manual de usuario.	97
A.6 Bibliografia.	98

I.- INTRODUCCION

El objetivo principal del envío de aplicaciones es la actualización y activación de sistemas de información, desde un nodo remoto. Por sistemas de información nos referimos a programas ejecutables y/o archivos de datos. El motivo de hacer un sistema con estas características está dado por los requerimientos presentados por la empresa automotriz FORD MOTORS COMPANY de México.

Esta empresa automotriz, con su central de operaciones en el Distrito Federal, necesita actualizar constantemente diversos tipos de información (sistemas, bases de datos, archivos conteniendo listas de precios, inventarios, etc.) a sus concesionarios que se encuentran distribuidos en toda la república. Para esto necesita un sistema que le permita con confiabilidad y confidencialidad llevar a cabo esta actualización.

A continuación mostraremos un panorama general del contenido de este trabajo. Las palabras que aparecen en letras itálicas, son aquellas que son parte del ámbito de comunicaciones y que son conocidas de la misma manera en inglés y en español.

Capítulo II - Definiremos el problema presentado por la empresa automotriz, el análisis de los requerimientos necesarios, la solución propuesta y los diagramas de flujo de datos obtenidos del análisis.

Capítulo III - Presentaremos el diseño del sistema.

Capítulo IV - Trataremos sobre el lenguaje de programación y se presentará el Manual de Usuario.

Capítulo V - En este capítulo comentaremos sobre las pruebas llevadas a cabo y los ajustes realizados al protocolo de comunicaciones.

Capítulo VI - Este es uno de los capítulos más importantes ya que se exponen los conceptos necesarios para el establecimiento del enlace Micro-Mainframe.

Capítulo VII - Se mencionará cómo se encuentra integrado nuestro sistema dentro del modelo OSI.

Capítulo VIII - Aquí expondremos nuestras conclusiones.

Finalmente se presentará un apéndice conteniendo conceptos básicos sobre los Sistemas Operativos en PC y en Mainframe, Interrupciones, Normas de comunicación, etc.

II.- ANALISIS.

II.1 DEFINICION DEL PROBLEMA.

En cualquier organización de tipo centralizado, el establecimiento de un canal de comunicación entre los elementos de la red es indispensable. Tradicionalmente, esta comunicación ha sido efectuada a través de servicios de mensajería, teléfono, y más recientemente, líneas conmutadas para el uso de Fax y paquetes de comunicación punto a punto entre computadoras.

Un ejemplo típico de este tipo de organización son los fabricantes de automóviles. En este documento se tomará como ejemplo a FORD.

FORD es una empresa que tiene sus oficinas centrales en el Distrito Federal, su planta de producción en Cuautitlán, Estado de México, y una red de más de 150 concesionarios esparcidos por todo el país. Cada uno de ellos distribuye refacciones y automóviles completos de la línea FORD. Para ello, FORD ha desarrollado un sistema de administración interna de pedidos, y cada concesionario cuenta con una microcomputadora y una copia del mismo.

La comunicación en este tipo de organización puede ser:

- Entre concesionarios, para localización o intercambio de refacciones y/o unidades.
- Del concesionario a la central, para la generación de pedidos.
- De la central a uno o más concesionarios (quizá todos), para envío de mensajes específicos a un concesionario, listas generales de precios y, principalmente, actualización de versiones del sistema de administración de pedidos.

Así mismo, se ha detectado que el mayor porcentaje del flujo de información se genera entre la central y los concesionarios, y viceversa. Con los métodos convencionales, las actividades asociadas al envío de información de la central hacia los concesionarios, pueden ser:

- Generación de una copia para cada uno de los concesionarios, y su envío por servicios de mensajería; o bien

- Enlace por línea conmutada con cada uno de los 150 concesionarios utilizando Fax o paquete de comunicaciones punto a punto entre computadoras.

Como es lógico suponer, los problemas asociados con estos métodos son:

Para el primer método:

- Organización de la información a enviar.
- Tiempo requerido para la elaboración de las copias y armado de juegos.
- Retraso en la entrega de la información.
- Pérdida de la información o maltrato.
- Para varias copias de información, necesidad de un servicio de distribución de mensajería.
- En caso de no contratar un servicio de mensajería profesional se tendrá que derogar en gastos para viáticos, etc.

Para el segundo método:

- Infraestructura para la recepción y envío simultáneo de información desde o hacia varios concesionarios (líneas telefónicas, personal).
- Necesidad de *software* (y quizá *hardware*) especial para el manejo de comunicaciones para cada concesionario.
- *Hardware* dedicado para la espera de llamadas.
- Pérdida de tiempo en el envío.
- Para el envío de archivos con caracteres especiales se requiere de un paquete de comunicación específico.

y para ambos:

- Poca confidencialidad en el manejo de la información.

II.2 ANALISIS DE REQUERIMIENTOS.

Con base en los problemas que se determinaron en el capítulo anterior, se describieron los siguientes requerimientos:

- Envío de cualquier tipo de información.
- Capacidad de envíos múltiples. Esto es, la posibilidad de enviar la misma información a varios destinos.
- Integridad en la información manejada. La información no debe sufrir alteración alguna durante su tránsito.
- Confidencialidad en el manejo de la información. Debe contar con un sistema de seguridad, donde solamente el personal responsable pueda enviar y recibir información, además de un sistema de ligas que controle el enrutamiento de mensajes.
- La información enviada debe considerar una fecha de activación automática.
- Si la información existe previamente en el nodo remoto, deberán obtenerse respaldos para su posible restauración.

II.3 ALTERNATIVAS DE SOLUCION.

Las alternativas de solución se reducen drásticamente debido al tipo de requerimientos. Se presentan dos alternativas de solución que los satisfacen.

La primera alternativa es por medio de un sistema de mensajería, en donde los principales pasos a llevar cabo serían los siguientes:

- Organizar la información que será enviada.
- Programar o calendarizar, en la medida de lo posible, el envío de la información dado que el medio de transporte puede ser diverso.
- Para la actualización de la información en el nodo remoto se deberá especificar perfectamente como debe llevarse a cabo, pues quizá sea necesario obtener respaldos, o generar una estructura (directorios en la microcomputadora) específica.

La segunda alternativa contempla una infraestructura de comunicación, y en base a ella, el desarrollo de un sistema de computación que cumpla con los requerimientos antes presentados. Con esta alternativa se pretende hacer de forma automática la actualización de la información en el nodo remoto. Para la implantación de esta solución se deberán tomar en cuenta los siguientes puntos:

- Desarrollo de un programa de aplicación que manipule y controle la información.
- Equipo de cómputo requerido.
- Equipo de comunicación.
- Desarrollo de un programa de comunicaciones que efectúe la transferencia de la información.
- Disponer del medio de comunicaciones necesario, que en este caso sería la red pública telefónica.

II.4 SOLUCION PROPUESTA.

II.4.1 ESPECIFICACIONES DE HARDWARE Y SOFTWARE.

La solución que se propone corresponde a la segunda alternativa presentada. Se basa en una infraestructura de comunicaciones que permite el intercambio de información entre áreas, realizando transferencias de cualquier tipo de archivos (de datos, ejecutables, etc.) controlando el flujo de los mismos, garantizando su integridad y adecuada distribución.

Para utilizar los recursos técnicos y humanos existentes, se propuso el siguiente esquema:

- Un centro de cómputo con una computadora *Mainframe* IBM 4381, que se encargará de realizar el almacenamiento centralizado de la información, con equipo de comunicaciones para el soporte de los 150 concesionarios.
- Cada concesionario contará con una computadora personal IBM PC o compatible, con 512 KB de memoria (mínimo), disco duro (opcional), y un *modem* (interno o externo) compatible con comandos HAYES.
- Una configuración tipo estrella (centralizada), donde cada uno de los usuarios (concesionarios) puede establecer enlace con el computador central (*Mainframe*), para enviar y/o recibir información.

- Un sistema de software en la microcomputadora que lleve a cabo la administración de la información que requiere ser enviada, o bien que ha sido recibida; un programa de comunicaciones que establezca el enlace con el ambiente Mainframe, y lleve a cabo la transferencia de archivos.
- En Mainframe, un sistema de base de datos para el manejo de la información, y que se encargue del control de ligas de comunicación entre destinatarios.
- La red pública telefónica como canal de comunicación.

Para generalizar, definimos una aplicación como un conjunto de archivos, interrelacionados entre sí, que pueden contener información de tipo texto (datos), ejecutable (programas) u otros (bases de datos, hojas de cálculo, etc.).

II.5 ANALISIS.

El objetivo principal del envío de aplicaciones es que se actualicen y activen sistemas completos de información desde un nodo remoto. Para llevar a cabo este objetivo se deben de tomar en cuenta los siguientes puntos:

- Selección de la aplicación a enviar.
- Preparación de envío.
- Envío y recepción.
- Activación.
- Depuración y restauración.
- Comunicaciones.

Los diagramas de flujo de datos correspondientes a estos módulos se presentan en el apéndice A.4 .

A continuación se muestra un análisis que abarca cada uno de estos puntos.

II.5.1 SELECCION DE LA APLICACION A ENVIAR.

Para seleccionar la aplicación que será enviada se deberán indicar los directorios y archivos que la conforman, así como los directorios destino en donde ésta deberá ser actualizada. Se deberá pedir al usuario que identifique el programa, que en forma opcional deberá ser ejecutado en el momento en que la aplicación sea activada, en la microcomputadora remota.

II.5.2 PREPARACION DE ENVIO.

Para realizar la actualización de una aplicación no es suficiente enviar los archivos que la conforman. También se requiere información asociada a la estructura de directorios en la cual deberá llevarse a cabo esta actualización, fecha de realización, programa inicial a ejecutar, etc. Con base en estos puntos, se determinaron que los archivos complementarios son los siguientes:

- Archivo *.APL : Contendrá una imagen de la aplicación que será enviada. Tomando como ejemplo de aplicación al Sistema DBASE III PLUS, el contenido de este archivo sería el siguiente:

Contenido

Significado

CON001,1	núm aplicación, prioridad
02\$,004	destinatarios
DBASE III PLUS	nombre de la aplicación
D C:\DBASE	directorio
A DBASE.MSG	archivo
A DBASEINL.OVL	archivo
A DBASE.COM	archivo
A DBASE.OVL	archivo
A DBASE.LD1	archivo
A ASSIST.HLP	archivo
D C:\DBASE\PRG	directorio
A INICIA.EXE	archivo
A INICIA.PRG	archivo
A CATALOGO.PRG	archivo
D C:\DBASE\DBF	directorio
A AGENDA.DBF	archivo
C C:\DBASE\PRG\INICIA.EXE PARAMETROS	programa a ser ejecutado
890724	fecha de activación

- Archivos *.NNN : Contendrán a los archivos de la aplicación (*.EXE, *.DAT, etc.), que serán segmentados para su envío. NNN representa un número de tres dígitos. Cada uno de estos archivos cuenta con un encabezado con el siguiente formato:

Nombre real del archivo.
 Archivo siguiente.
 Nombre temporal.
 <separador>
 Destinatarios.
 <separador>
 Número de la aplicación.

- Archivo *.RGR : Contendrá información necesaria para la actualización de la aplicación en la microcomputadora receptora. La estructura del contenido de este archivo es:

Número de la aplicación, Destinatarios
 *.APL
 *.001
 *.002
 .
 .
 .
 .
 *.NNN
 *.RGR

- Archivo OUTLIST.DAT : Contendrá la lista de los archivos que serán enviados. La estructura de su contenido deberá ser:

*.APL
*.NNN
*.RGR
*.APL
*.NNN
*.RGR
.
.
.
*.RGR

II.5.3 ACTIVACION.

La activación de una aplicación puede ser manual o automática.

Forma automática: cuando se recibe en forma completa una aplicación, su fecha de activación deberá grabarse en la base de datos APLICA.*. Existirá un proceso que estará sensando esta base de datos para determinar en que momento deberá ser activada.

En caso de que al momento de la recepción de una aplicación se obtenga que la fecha de activación es anterior o igual a la fecha actual en que se recibe, en ese momento la aplicación será activada.

Forma manual: se lleva a cabo mediante la selección, por parte del usuario, de la aplicación a ser activada.

Cuando se activa una aplicación se genera un respaldo de la misma, si es que ésta existía previamente. También será actualizada la base de datos APLICA.* en los campos correspondientes a la fecha de arranque y el subdirectorío en donde se hizo el respaldo.

II.5.4 DEPURACION Y RESTAURACION.

Por depuración nos referimos al borrado del respaldo generado en el momento en que la aplicación fue actualizada. Se lleva a cabo mediante la selección, por parte del usuario, de la aplicación a ser depurada.

Cuando una aplicación sea depurada, el registro asociado a ella en la base de datos APLICA.* será borrado.

La restauración de una aplicación será exclusivamente manual, y consiste en regenerar el ambiente de la aplicación a partir del respaldo obtenido poco antes de haber efectuado la actualización. El usuario deberá seleccionar aquella aplicación que desee restaurar.

II.5.5 COMUNICACIONES.

La etapa de envío de la información requiere de la existencia de los archivos PARAM.TXT y ENTRA.TXT. El primero contiene los datos necesarios para establecer la comunicación, tales como el puerto, la velocidad, los teléfonos, el identificador y *password* del usuario. El segundo archivo (ENTRA.TXT) es un archivo de referencias necesario para formar el archivo ENTRA.TSK que tendrá los comandos para poder llevar a cabo el firmado (LOGON) en *Mainframe*. Una vez realizado el firmado, se interactuará con el programa de comunicaciones en el *Mainframe* para la transferencia de archivos (COMM1_2).

Para la recepción de la información se actuará en base a la información recibida. Los archivos físicamente se recibirán en un subdirectorío (\TRANSFER por ejem.) y lógicamente en el archivo INLIST.DAT.

Si se recibe un archivo con extensión APL se generará un subdirectorío cuyo nombre será el número de aplicación de que se trate, y el archivo *.APL se moverá a este subdirectorío. Si se recibe un archivo de datos (*.NNN) se extraerá de su encabezado la información correspondiente a la aplicación a la que pertenece para así ubicarlo en el subdirectorío de aplicación que le corresponda. Ahora bien, si se recibe un archivo con extensión RGR se tomará como fin de la aplicación, y se grabarán en la base de datos de aplicaciones APLICA.* los siguientes datos:

Nombre de la aplicación.

Fecha en que la aplicación fue recibida.

Fecha en la que deberá ser activada.

Fecha en la que fue activada.

Número de aplicación que identifica al subdirectorío en donde ésta se encuentra.

Lugar donde se encuentra el respaldo de la aplicación.

El archivo *.RGR será ubicado en el subdirectorío de aplicación que le corresponda y se expandirán, en base a la información que éste nos proporcione, todos los archivos de la aplicación.

Para finalizar la sesión de comunicaciones con el *Mainframe* se requiere la existencia del archivo SALE.TSK, que contiene los comandos para llevar a cabo el proceso de salida (LOGOFF) del *Mainframe*.

Es necesario mencionar que en los procesos de envío y recepción de información se llevará a cabo la actualización de los registros estadísticos (ver análisis de estadísticas).

Especificaciones para el módulo de comunicaciones.

1. Manejo de dos modos para el envío de mensajes :

- a) Modo General : Este modo se caracterizará por un proceso de encriptación de toda la información previo a su envío, y otro de desencriptación, posterior a la etapa de recepción.
- b) Modo optimizado : Este modo encriptará únicamente aquellos caracteres no transmitibles, realizando este proceso en el momento de la transmisión, y los desencriptará durante la recepción.

El uso de estos modos depende del contenido de los archivos que se desean enviar. Si en alguno de ellos la cantidad de caracteres no transmitibles es mayor al 15% del tamaño total del mismo, se recomienda usar el modo general, de lo contrario se deberá usar el modo optimizado.

Para especificar los archivos que serán enviados en modo general, se deberá generar un archivo que contenga los nombres de éstos en una línea, y en la siguiente los destinatarios. Al ser enviados al *Mainframe*, serán catalogados con una marca de tipo de archivo especial ('@') para que puedan ser identificados y desencriptados al momento de ser entregados al usuario. Para el envío optimizado se sigue el mismo procedimiento, pero con un archivo de nombre y un tipo de archivo diferente.

2. Integrar un módulo de estadísticas de comunicación.

El manejo e interpretación de las estadísticas se lleva a cabo únicamente en *Mainframe*, pero los datos se envían desde la micro de manera transparente al usuario, ya que los registros se generan durante la sesión. Los datos que se manejan son:

- Identificación del usuario (UID)
- Fecha y hora de inicio de la sesión.
- Lista de teléfonos marcados y número de intentos.
- Teléfono por el cual se logró el enlace y velocidad de conexión.
- Código de terminación.
- Número de retransmisiones en envío y recepción.

- Número de bytes transmitidos (virtuales y reales).
- Número de bytes recibidos (virtuales y reales).
- Número de archivos enviados por modo.
- Número de archivos recibidos por modo.
- Fecha y hora de terminación.

Por bytes transmitidos y recibidos reales nos referimos al número de bytes que conforman el archivo perteneciente al usuario, es decir, al tamaño de este archivo al momento de residir en disco duro o flexible.

Por bytes transmitidos y recibidos virtuales nos referimos a la cantidad total de bytes desplazados a través de la línea de transmisión, es decir, los bytes reales más los caracteres de control necesarios para llevar a cabo dicha transmisión (pertenecientes al protocolo de comunicación).

Los datos para la elaboración de estas estadísticas son proporcionados por diversos módulos del sistema, y en un formato abierto que permite inclusiones posteriores. Los datos que cada módulo proporciona son:

MODULO

DATOS

Firmado	UID, Teléfonos marcados, teléfono exitoso, velocidad de conexión y hora de inicio.
Entrega de Archivos	Número de archivos enviados por modo, número de retransmisiones en envío, bytes enviados (virtuales y reales).
Recepción de archivos	Número de archivos recibidos por modo, número de retransmisiones en la recepción, número de bytes recibidos. Código de terminación.
Fin de sesión	Fecha y hora de terminación, código de terminación.
Controlador de procesos	Fecha y hora de terminación (en caso de error).

Un código de línea, es decir, un prefijo seguido de un número, fue añadido al inicio de cada línea para facilitar la interpretación de los datos. Los códigos, así como su significado se muestran a continuación :

PREFIJO	FORMATO	DESCRIPCION
I1	UID	Identificación.
I2	Teléfono	Intento en el teléfono.
I3	Vel,teléfono	Velocidad de conexión y teléfono exitoso.
I4	AAMMDDHHMMSS	Fecha y hora de inicio.
I5	Real,Virtual,Retrans.	Tamaño real y virtual del archivo en modo general y número de retransmisiones (Envío).
I6	Real,Virtual,Retrans.	Tamaño real y virtual del archivo en modo optimizado y número de retransmisiones (Envío).
I7	Real,Virtual,Retrans.	Tamaño real y virtual del archivo en modo general y número de retransmisiones (Recepción).
I8	Real,Virtual,Retrans.	Tamaño real y virtual del archivo en modo optimizado y número de retransmisiones (Recepción).
I9	Código,AAMMDDHHMMSS	Código y fecha de terminación.

Los códigos de terminación son :

CODIGO	DESCRIPCION
0	Terminación Normal.
1	Número de retransmisiones posibles excedido.
2	Pérdida de la señal portadora.
3	Transmisión abortada por el usuario.
4	Sin respuesta del Mainframe.

II.6 DICCIONARIO DE DATOS.

*.APL : Son los archivos que indican los directorios donde se encuentra la aplicación y los archivos que ésta contiene. En otras palabras, los archivos *.APL contienen una imagen de la aplicación que se encuentra en la PC. El nombre de cada uno de ellos siempre comienza con una 'A'.

NUM APL : Número de la aplicación.

OUTLIST.DAT : Es un archivo cuyo contenido es una lista de los archivos que se enviarán a la *Mainframe*. Esta lista contendrá archivos *.APL, archivos de la aplicación (*.NNN) y archivos que indican fin de la aplicación (*.RGR).

*.NNN : Archivos segmentados de la aplicación. El nombre de cada uno de ellos siempre comienza con una 'N'.

*.RGR : Este es un archivo que contiene información necesaria para regenerar la aplicación en el lado de recepción. La estructura del contenido de este archivo es:

NUM APL, DESTINOS

*.APL

*.001

*.002

.

.

.

*.NNN

*.RGR

El nombre del archivo comienza siempre con una 'R'.

FAT : Directorio del sistema.

PARAM.TXT : Es un archivo que contiene los parámetros de comunicación, tales como el puerto, la velocidad, teléfonos; además contiene datos del usuario tales como el UID y PWD.

ENTRA.TXT : Es un archivo que contiene las referencias necesarias para formar el archivo de comandos que se utiliza para el firmado (LOGON) en *Mainframe*.

ENTRA.TSK : Es el archivo de comandos para llevar a cabo el firmado en *Mainframe*.

INLIST.DAT : Es un archivo que contiene una lista de los archivos que fueron recibidos. Esta lista contendrá archivos *.APL, *.NNN y *.RGR.

SALE.TSK : Archivo de comandos necesario para llevar a cabo la terminación (LOGOFF) en Mainframe.

OUTOK.DAT : Archivo cuyo contenido es una lista de los archivos que fueron enviados con éxito.

ESTADIST.DAT : Es un archivo con registros de información estadística que será enviada al Mainframe.

APLICA.* (.DAT y .NDX) : Es una base de datos que contiene la información necesaria para identificar una aplicación, y el estado en el que ésta se encuentra.

APLICA.TMP : Es un archivo temporal que se utiliza para generar el archivo *.APL.

RESPALDO.DAT : Cuando se obtienen respaldos de una aplicación, los nombres de los archivos respaldados son grabados en este archivo.

REGISTROS DE ESTADISTICAS : Anexarán al archivo de estadísticas los datos que el módulo maneje. Su orden es el siguiente:

Número de registro (*)	Datos
1	I01,I02,I03,I04.
2	I05,I07,I08,I11,I12.
3	I06,I09,I13,I14.
4	I15,I16.

(*) La relación entre el número de registro y los módulos que los generan se muestra en los diagramas de flujo de datos contenidos en el apéndice A.4 .

III.- DISEÑO.

Partiendo del análisis, condensando la información recabada, y agrupándola en módulos, se decidió dividir al sistema, acorde a sus funciones, en 7 módulos principales.

Finalmente el diseño se determinó como sigue :

Se presentará un menú con las siguientes opciones:

- Parámetros de comunicación.
- Nueva aplicación.
- Activación.
- Restauración.
- Depuración.
- Comunicaciones.

Parámetros de comunicación.

En este módulo se deberán actualizar los siguientes datos:

Puerto
Velocidad
Identificación
Password
Comando para obtener línea
6 teléfonos (máximo).

Grabar estos datos en el archivo PARAM.TXT

Nueva aplicación.

Obtener nombre de la aplicación.

Obtener los destinatarios.

Obtener drive donde residen los archivos de la aplicación a enviar.

Obtener directorios destino (un máximo de 30).

Obtener la prioridad de envío de la aplicación.

Obtener fecha de activación de la aplicación.

Por cada directorio destino:

Pedir máscara de marcado.

Presentar archivos para que sean seleccionados.

Grabar en el archivo APLICA.TMP el directorio destino y los archivos seleccionados.

Visualización de la imagen de la aplicación.

Se presentará una imagen de cómo quedará la aplicación en la PC receptora.

Aceptar información.

Peticion de programa a ejecutarse en el momento de activarse la aplicación (opcional). Si el usuario proporcionó el nombre de un programa que deberá ejecutarse en el momento de activar la aplicación, este nombre de programa, precedido por una letra C, se grabará en el archivo APLICA.TMP. Por ejemplo:

C INICIO.BAT

Preparación de envío.

Generación de un número de aplicación.

El número de aplicación será generado con una clave random. La semilla para este random serán los segundos del tiempo de la PC. Su longitud será de 8 caracteres.

Generación del archivo *.APL.

El * se sustituye por el número de la aplicación, a partir del archivo APLICA.TMP. El archivo *.APL tendrá como estructura la que se muestra en el siguiente ejemplo:

CON001,1	núm aplicación, prioridad
02\$,004	destinatarios
CONTACTO	nombre de la aplicación
D C:\CONTACTO	directorio
A USUARIO.EXE	archivo
A CORREO.LIB	archivo
A HOST.EXE	archivo
A HOST.OVR	archivo
D C:\CONTACTO\COMMS	directorio
D C:\CONTACTO\HOST	directorio
A LLAMA.TXT	archivo
C INICIA.BAT PARAMETROS	programa a ser ejecutado
890724	fecha de activación

Segmentación de archivos.

En base al tamaño de bloque definido en la configuración (o el default) se segmentan los archivos que componen la aplicación, aumentando 200 bytes por los encabezados que contienen la siguiente información :

Nombre real del archivo.
Nombre del archivo siguiente.
Nombre del archivo segmentado.
*
Destinatarios.
*
Número de la aplicación.

Cada archivo es primero codificado a 6 bits ya que el envío es en modo general. Los nombres de los archivos segmentados se generan como sigue:

<PREFIJO> + <UID> + <SECUENCIA>.<EXTENSION>

donde :

<PREFIJO> : Es una letra que identifica el tipo de archivo. Se manejará una 'X'.

<UID> : Consta de tres letras y es el mismo que se utiliza como *password* en el archivo de configuración 'PARAM.TXT'.

<SECUENCIA> : Consta de 4 posiciones, y es un número consecutivo que se conserva en un archivo llamado 'SECUENCIA.DAT'. Cada vez que se genere un nombre de archivo, se incrementará en una unidad este valor y será actualizado en el archivo SECUENCIA.DAT.

<EXTENSION> : Consta de tres posiciones, y es un número secuencial de tres dígitos. El primer archivo tendrá asignado el '000', y el último el '999'. En caso de existir un sólo archivo se le asignará éste último, es decir, '999'.

Generación del archivo OUTLIST.DAT.

Grabar el nombre del archivo segmentado, junto con la prioridad, en el archivo OUTLIST.DAT.

Generación del archivo *.RGR.

En base a los nombres de los archivos generados anteriormente, se forma la lista de aquellos que conforman la aplicación. Para ello, el * se sustituye por el número

de la aplicación. La estructura de su contenido será la siguiente:

```
Número de la aplicación,destinatarios
<NOMBRE>.APL
<NOMBRE>.001
<NOMBRE>.002
.
.
.
<NOMBRE>.999
<NOMBRE>.RGR
```

donde <NOMBRE> indica el nombre del archivo, y es diferente para cada uno de ellos, pues es generado en forma random.

Grabar el nombre del archivo <NOMBRE>.RGR en el archivo OUTLIST.DAT junto con la prioridad.

Comunicaciones.

La sección de comunicaciones estara formada por los siguientes módulos (la relación entre ellos se muestra en el apéndice A.4).

Sub-Módulo 3.1. Verificación y generación de archivo de control.

Verificar por los archivos y programas necesarios para llevar a cabo la sesión. Estos son :

ENTRA.TXT, PARAM.TXT, SALE.TSK.

Leer archivo de configuración (PARAM.TXT).

Generar archivo base (CONNECT.TXT) para conexión con Mainframe, a partir del archivo de referencia (ENTRA.TXT).

Generar archivo de comandos (ENTRA.TSK) a partir de archivo base (CONNECT.TXT).

Inicializar Modem.

Anexar el archivo de estadísticas de las últimas conexiones (ESTADIST.DAT), en la lista de archivos a subir (OUTLIST.DAT).

Lanzar al proceso 3.2, pasándole como parámetros el puerto y la velocidad.

Sub-Módulo 3.2. Mcp.

Lanzar a los procesos 3.3, 3.4, 3.5, 3.6, pasándoles como parámetros el puerto y la velocidad, recibiendo código de terminación y ejecutándolos en ese orden.

Anexar en el archivo de estadísticas el código de terminación. Si no hay error lanzar proceso 4.

Sub-Módulo 3.3. Logon.

Leer y ejecutar el archivo generado por el proceso 3.1 (ENTRA.TSK), anexando al momento los registros I01, I02, I03, e I04 en el archivo de estadísticas (ESTADIST.DAT). Regresar al proceso 3.2 un código de terminación.

Sub-Módulo 3.4. Sube.

Lanzar al proceso 3.5 en Mainframe para inicializar el protocolo.

Enviar los archivos que están en la lista (OUTLIST.DAT), y anexar los registros I05, I07, I08, e I11 al archivo de estadísticas (ESTADIST.DAT).

Registrar en la lista de archivos enviados exitosamente (OUTOK.DAT) los indicados por el proceso 3.5.

Regresar código de terminación al proceso 3.2.

Sub-Módulo 3.5. Com1_2.

Inicializar el protocolo.

Recibir los archivos enviados por el módulo 3.4, y enviar confirmación por cada uno de dichos archivos recibido correctamente, ruteándolos al momento.

Enviar los archivos correspondientes al usuario, al módulo 3.6, y borrarlos cuando se recibe de éste la confirmación de recepción correcta.

Sub-Módulo 3.6. Baja.

Recibir los archivos que envía el módulo 3.5, registrándolos en una lista (INLIST.DAT) para su posterior procesamiento. Físicamente, estos archivos serán ubicados en el subdirectorío "TRANSFER\".

Enviar confirmación de recepción correcta por cada uno de los archivos recibidos correctamente.

Anejar los registros I06, I09, I10, e I13 al archivo de estadísticas (ESTADIST.DAT).

Regresar código de terminación al proceso 3.2.

Sub-Módulo 3.7. Logoff.

Leer y ejecutar el archivo de comandos (SALE.TSK) para terminar la sesión con el Mainframe.

Anejar el registro I15 al archivo de estadísticas (ESTADIST.DAT).

Regresar código de terminación al proceso 3.2.

Activación (Modo automático, vía módulo de comunicaciones).

Sub-Módulo 4.4. Ruteo.

Leer la lista de archivos recibidos (INLIST.DAT), y en base a su extensión, catalogar como sigue:

- *.APL : Aplicación.
- *.NNN : Archivo de datos.
- *.RGR : Fin de aplicación.

donde "*" es un nombre hasta de 8 caracteres, y "NNN" un número de tres dígitos. De acuerdo al tipo de archivo, se le dará un trato especial a cada uno de ellos.

- Aplicación.

A partir de un archivo de aplicación, obtener la siguiente información:

- Nombre de la aplicación.
- Número de aplicación.
- Fecha de activación.
- Comando para instalación.

Crear un subdirectorío para esta aplicación, donde serán almacenados todos los archivos que la componen, incluyendo los *.APL y *.RGR. Este subdirectorío se genera en base al número de aplicación.

Grabar un registro en la base de datos, el cual contendrá la siguiente información:

```
Nom_Apl      : String[30] ; {Nombre de la aplicación}
Num_Apl      : String[8]  ; {Número de aplicación}
Comando      : String[50] ; {Comando para instalación}
F_Activacion : String[6]  ; {Fecha de activación}
F_Recepcion  : String[6]  ; {Fecha en que grabó el RGR}
F_Arrangue   : String[6]  ; {Fecha en la que se activó}
Respaldo     : String[8]  ; {Subdirectorío de respaldo}
```

Los cuatro primeros datos se obtienen del *.APL; los restantes se obtienen de otros módulos recibidos posteriormente, y por ello se graban temporalmente nulos.

Este registro se graba en el archivo de datos (APLICA.DAT), y se indexa por medio de una llave, formada por el nombre concatenado con el número de aplicación. Esta relación de índices se mantiene en el archivo de índices (APLICA.NDX) de la base de datos.

Este archivo siempre deberá ser el primero en bajar de todos los que componen la aplicación.

- Archivos que componen a la aplicación.

Obtener de su encabezado la siguiente información:

```
Nombre real del archivo.
Nombre del archivo siguiente.
Nombre del archivo segmentado.
<Separador>.
Destinatarios.
<Separador>.
Número de aplicación.
Prioridad.
<Fragmento del archivo encriptado>.
```

Desde "Nombre real del archivo" hasta "Prioridad" se utilizan 200 bytes.

En base al "Número de aplicación", copiar el archivo al subdirectorío correspondiente y borrar inmediatamente.

- Fin de aplicación.

Este archivo indica que todos los archivos que componen la aplicación se encuentran residentes en la PC en modo de envío, es decir, encriptados y segmentados. Este archivo contiene:

Número de aplicación, Destinatarios, Prioridad

*.APL

*.NNN

*.RGR

En la recepción de un archivo de este tipo, rutear al subdirectorio correspondiente y comenzar la fase de descripción:

Actualizar en la base de datos (APLICA.*) el registro correspondiente, grabando la fecha de recepción completa (F_Recepcion).

Leer del *.RGR todos los archivos de datos, descriptándose al momento, por lo que los archivos de la aplicación deberán quedar con sus nombres y tamaños reales. El proceso de descripción actúa de la siguiente forma:

1. Leer el encabezado.

2. Si el archivo siguiente no tiene como extensión 'END', entonces:

Insertar al archivo siguiente entre el encabezado y el fragmento de datos, el fragmento de datos del archivo actual.

De otra manera:

Obtener el nombre real del archivo.

Descriptar el fragmento de datos y grabar el resultado en un archivo con el nombre real.

3. Eliminar el archivo actual.

Si la fecha de activación es anterior o igual a la fecha de recepción (F_Activacion <= F_Recepcion), entonces se activa automáticamente llamando al módulo 4.3.

Activación (desde el menú de actividades)

Pedir una selección de la aplicación a activar.

Los items de selección están formados por el nombre de la aplicación, y serán obtenidos de los archivos APLIC.A.*.

Desplegar la imagen de la aplicación (*.APL), para confirmar la aplicación seleccionada.

Obtener de los archivos APLICA.* el número de la aplicación que permite identificar el subdirectorío en el que ésta se encuentra.

Si la aplicación por activar ya existe, se hará un respaldo de directorios y archivos. El nombre del subdirectorío en donde se hará el respaldo será un número *random* (la semilla se obtendrá de la hora de la PC), verificando que este nombre no exista previamente. Los archivos de la aplicación anterior serán reubicados en este directorío.

Grabar en el archivo APLICA.* la fecha en que la aplicación fue activada y el nombre del subdirectorío en donde se llevó a cabo el respaldo (si es que se hizo una actualización).

Mover el archivo *.APL de la aplicación por activar al subdirectorío de respaldo.

En caso de que no se haya hecho respaldo, se borrará el registro correspondiente a la aplicación activada de los archivos APLICA.*. Mover (no copiar) los archivos de la aplicación a activar a los directorios destino que les correspondan. Borrar los archivos y directorios de la aplicación a activar.

Verificar en el archivo *.APL de la aplicación, si ésta tiene un programa que deba ser ejecutado en el momento de la activación.

Restauración.

Seleccionar la aplicación a restaurar.

Las aplicaciones a seleccionar sólo serán las que fueron activadas y además se obtuvieron respaldos de ellas. Los ítems de selección serán los nombres de las aplicaciones que serán obtenidos de los archivos APLICA.*.

Obtener de los archivos APLICA.* el subdirectorío en donde se llevó a cabo el respaldo.

Desplegar la imagen de la aplicación activada (*.APL), para confirmar que la aplicación seleccionada será restaurada. El *.APL se encuentra en el subdirectorío de respaldo.

Borrar los archivos y directorios de la aplicación activada.

Mover los directorios y archivos que se encuentran en el subdirectorio de respaldo hacia el(los) directorio(s) en donde se encontraban originalmente.

Borrar de los archivos APLIC.A.* la aplicación restaurada.

Depuración.

Seleccionar la aplicación a depurar.

Las aplicaciones a seleccionar serán las que fueron activadas y se obtuvo respaldo de ellas, y las que no han sido activadas. Los ítems de selección serán los nombres de las aplicaciones y se obtendrán de los archivos APLIC.A.* .

Para una aplicación que ha sido respaldada:

Obtener de APLIC.A.* el directorio del respaldo. Borrar los directorios y archivos de este respaldo.

Para una aplicación que no ha sido activada:

Obtener de APLIC.A.* el directorio temporal donde reside la aplicación, y borrar tanto el contenido como el directorio.

Borrar el registro correspondiente en los archivos APLIC.A.* .

IV.- DESARROLLO.

IV.1 PROGRAMACION.

A medida que la computadora ha ido evolucionando, se ha observado la aparición de varias generaciones de lenguajes. La primera generación corresponde al lenguaje de máquina, que fue el primero en usarse. La segunda generación se refiere a los lenguajes de bajo nivel, que requieren un proceso de interpretación y que han sido llamados ensambladores. La tercera generación involucra a los lenguajes de alto nivel, independientes de la máquina utilizada, y que requieren un proceso de traslación realizado por los compiladores. La cuarta generación se caracteriza por manejar conjuntos de registros en lugar de registros individuales, uso de potentes macroinstrucciones y de fácil manejo para el usuario. Tienen por objetivo disminuir el tiempo de programación, y optimizar, en la medida de lo posible, los recursos del sistema.

Para la implantación del sistema en el ambiente de la microcomputadora, se pensó en dos lenguajes de alto nivel para la programación: Turbo Pascal y C.

Con estos dos lenguajes se tiene un control sobre interrupciones, puertos del procesador, área de memoria, etc; control que es necesario para poder llevar a cabo la programación del módulo de comunicaciones.

De estos dos lenguajes se eligió Turbo Pascal por las siguientes razones:

- Dado que el sistema fue elaborado por más de una persona, y cada una tiene un estilo distinto de programación, el acoplamiento y revisión de rutinas se torna más complejo en C, pues el código, dado la versatilidad de programación en este lenguaje, es en algunas ocasiones difícil de interpretar por otras personas que no desarrollaron la rutina.
- Existe mayor experiencia de programación en Pascal.
- Para Turbo Pascal se cuenta con una utilería que permite el manejo de archivos como si correspondieran a una base de datos.

Lenguaje utilizado:

Turbo Pascal Versión 5.0

Utilería para manejo de archivos:

Turbo Pascal Database Toolbox Versión 4.0

Para el ambiente en *Mainframe* se utilizó el lenguaje de cuarta generación NATURAL, que es nativo al manejador de base de datos ADABAS. Este lenguaje proporciona un conjunto de macroinstrucciones que permiten desarrollar en forma rápida y eficiente aplicaciones involucradas con el manejo de grandes cantidades de datos. Así mismo, facilita la comunicación directa con el monitor de teleproceso y con rutinas de propósito específico que hayan sido desarrolladas en otros lenguajes de programación.

Esto último se debe a la integración entre el monitor de teleproceso COM-PLETE, el manejador de bases de datos ADABAS y el lenguaje de programación NATURAL, que fueron desarrollados por un mismo fabricante.

V.- IMPLANTACION Y PRUEBAS.

Dadas las características del sistema, las pruebas cubrieron básicamente dos aspectos : comunicaciones y tratamiento de la información.

La parte de comunicaciones consistió a la vez en una serie de pruebas para primero cubrir el objetivo del envío de aplicaciones y posteriormente optimizar recursos. Este módulo cubre tres etapas :

- 1.- Firmado (Logon).
- 2.- Transferencia de información (Sesión).
- 3.- Fin de sesión (Logoff).

La parte del firmado contempla desde la inicialización del *modem*, el marcaje de teléfonos, así como el acceso a COM-LETE y NATURAL. Aquí se presentó la variante de los *modems*, ya que aunque algunos especifican ser compatibles a todos los comandos HAYES, no lo son completamente, por lo que no reconocen algunos comandos básicos para su inicialización. Se realizaron pruebas con distintos modelos de *modems* tales como RACAL VADIC, SISCO, MULTITECH, ECSA, MICROCOMM, etc., y se obtuvo una manera común para inicializarlos. En la parte del marcaje de teléfonos se tuvo que crear una rutina que actuara de manera inteligente, ya que se deben contemplar acciones para los siguientes casos :

- Si se necesita marcar un número en especial para obtener tono para marcar. Este caso se da en los teléfonos que son extensiones de un número de grupo (conmutador).
- Si al descolgar la bocina, no se obtiene tono para marcar.
- Si el teléfono al que se marcó está ocupado.
- Si el teléfono al que se marcó no responde.
- Si el teléfono al que se marcó contesta adecuadamente.

Agrupando los casos anteriores, se creó un algoritmo general con las características necesarias para tomar las acciones correspondientes, ya sea para uso línea directa y no es necesario el marcaje, o bien se intenta el enlace vía TELEPAC.

La parte de transferencia es aquella donde la micro entrega la información correspondiente a la *Mainframe*, y viceversa. Aquí se tuvieron que contemplar cuatro estados del proceso:

- a) Recibiendo. La micro esta entregando información al *Mainframe*.

- b) Recibida. La micro ya entregó información al *Mainframe*.
- c) Entregando. La *Mainframe* esta entregando la información a la micro.
- d) Entregada. La *Mainframe* entregó la información a la micro.

Estos estados del proceso debieron de manejarse con mucho cuidado para evitar que se transfirieran datos incompletos o repetidos, por lo que siempre se pide una confirmación por parte del sistema remoto si ya recibió la información correctamente, y así poder registrarla como enviada o recibida. Esto es : cuando la micro comienza a enviar un archivo, la *Mainframe* genera un registro y le pone estado de recibiendo; cuando se terminó de recibir el archivo, lo cataloga como recibido y le envía un mensaje a la micro, quien a su vez lo registra como enviado exitosamente para su posterior depuración. El proceso de bajada es el mismo, solo que se invierten los papeles entre la micro y la *Mainframe* como receptor y emisor, respectivamente.

La parte de fin de sesión (Logoff) consiste en cancelar el protocolo con *Mainframe* e indicar la terminación de sesión tanto en NATURAL como en COM-LETE. En esta parte el principal problema es el colgar el teléfono, ya que aunque COM-LETE corta la portadora al fin de sesión, y los *modems* son configurados para colgar el teléfono cuando esto ocurra, algunos no lo hacen y en algunas ocasiones se queda descolgado hasta que se apague la micro. Fué indispensable entonces generar una rutina para lograr esto con resultados satisfactorios.

La parte de tratamiento de la información cubre tres etapas: reconocimiento, regeneración e instalación.

El reconocimiento se lleva a cabo cuando una vez recibida la información se rutea al subdirectorío correspondiente, es por esto que se maneja dentro del archivo información concerniente a su destino. Si no cumple con el formato del sistema, se desecha la información. Dentro del formato antes mencionado se tienen comandos para la regeneración e instalación de la aplicación. En este punto se debe tener cuidado en el manejo de los archivos, ya que si no hay espacio en disco para regenerar o instalar la aplicación e inclusive sacar respaldos, se puede perder información. Esto se solucionó con un algoritmo que estima el espacio necesario para proceder, y si no se tiene éste, no se procede.

VI.- ENLACE MICRO-MAINFRAME.

VI.1 AMBIENTE MAINFRAME.

VI.1.1 SISTEMA DE COMUNICACION DE DATOS.

En el ambiente de comunicaciones existen básicamente dos tipos de configuración de una red. La elección de una de ellas depende del tipo de aplicación que se desee implantar.

Como primer tipo tenemos una red de computadoras donde originalmente todas ellas se encuentran en estado de espera de llamada (fig. 1). En el momento en que alguna desee establecer comunicación con otro elemento de la red, deberá cambiar a un estado de originador. Al término del enlace asumirá nuevamente el estado de espera.

El segundo tipo corresponde a una configuración tipo estrella, en donde existe una computadora central que siempre se encuentra en espera de llamadas. Dichas llamadas serán efectuadas por las computadoras periféricas cuyo estado fijo es de originador. Aquí la información se deposita temporalmente en la central para que sea recibida por la computadora destinataria en el momento en que ésta se conecte.

Nuestro sistema se ajusta a las características del segundo tipo de configuración antes mencionado, en donde el papel de computadora central esta desempeñado por un ambiente Mainframe y el de periféricas por computadoras personales (fig. 2).

Las características principales de cualquier sistema de comunicación de datos son:

- Entrada/salida remota.

Las estaciones remotas leen y/o modifican los datos de la computadora central por medio de servicios y facilidades de comunicación.

- Entrada aleatoria.

Cualquier estación remota puede acceder las facilidades de la computadora central en cualquier momento.

- Procesamiento inmediato de las transacciones.

Cada evento que pueda modificar los datos existentes (transacción), deberá procesarse individual e inmediatamente que éste ocurra.

- Corto tiempo de respuesta.

COMPUTADORAS QUE SE ENCUENTRAN EN ESTADO DE ESPERA DE LLAMADA.

31 A

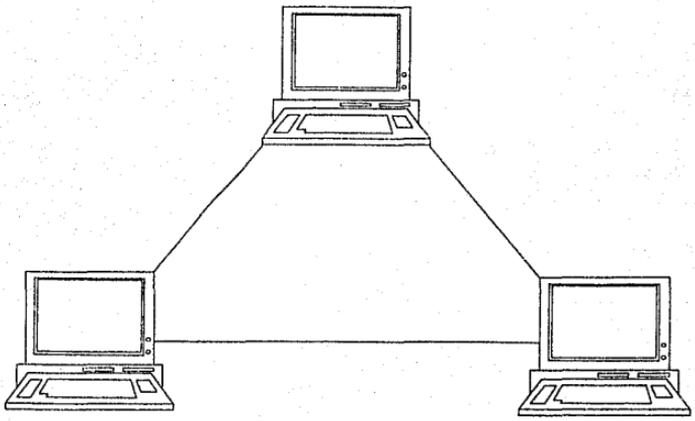
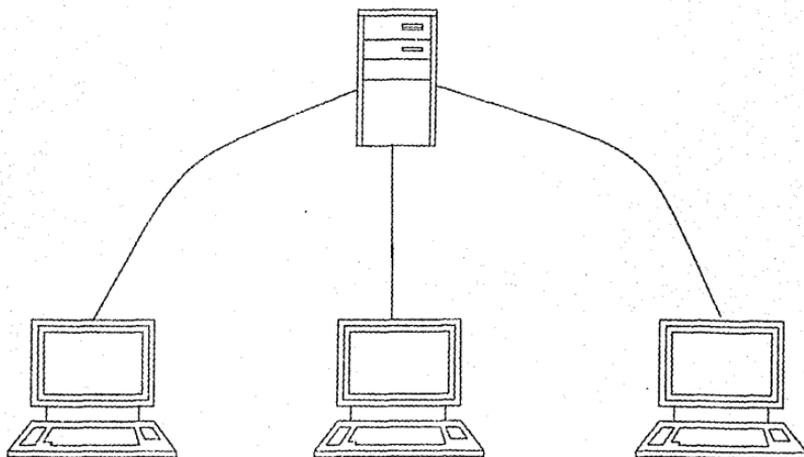


FIGURA 1

CONTESTADOR (AMBIENTE MAINFRAME)



ORIGINADORES DE LLAMADA (PC'S)

FIGURA 2

Responder en el menor tiempo posible a la terminal remota o aplicación.

- Concurrencia.

Atención a varios usuarios en forma simultánea.

- En línea.

Los datos de entrada son recibidos directamente desde el punto del cual se originan, y/o se envían hacia el punto donde serán usados.

- Tiempo real.

La respuesta a los datos de entrada es lo suficientemente rápida para controlar el proceso que los genera y/o influenciar a los datos de entrada subsecuentes. Esta característica, sin embargo, no es indispensable.

Los principales elementos y funciones de un sistema de comunicación de datos son los siguientes:

- Estación terminal remota.

Proporciona los medios para el acceso a las facilidades de la computadora central.

- Facilidades de la computadora central.

Proporciona los recursos de máquina (programas y base de datos) necesarios para llevar a cabo el procesamiento de datos y las funciones propias de la aplicación.

- Servicios y facilidades de comunicación.

Entrega los datos e información de control desde un punto a otro dentro de una red de comunicaciones.

- Unidad de control de comunicaciones.

Permite el enlace lógico y físico entre el computador central y los equipos terminales.

Cada uno de estos elementos está asociado a uno o varios dispositivos físicos, que son los que conforman el sistema de comunicación, y pueden ser identificados con varios nombres.

El primer dispositivo es la terminal, situado en el lugar donde reside el usuario remoto. Los servicios y facilidades de comunicación usados en este proyecto se refieren en parte, a todos aquellos recursos proporcionados por el servicio telefónico público, tales como líneas telefónicas, centrales de conmutación, etc.

Ahora bien, para lograr que las señales generadas por la terminal puedan ser transmitidas a través de las líneas telefónicas, es necesario que algún dispositivo realice un proceso de adecuación a dichas señales, y de la misma forma, otro dispositivo que en el otro extremo realice el proceso inverso. A estos procesos de adecuación para transmisión sobre líneas conmutadas se le denomina modulación y demodulación, respectivamente, y el dispositivo que los realiza recibe el nombre de *MODEM*, como contracción de las dos palabras que describen sus funciones. Así, las facilidades de comunicación constan de un canal de transmisión, y adecuadores de señales en sus extremos.

El siguiente elemento de la red es el sistema central de procesamiento. En este caso, y debido al manejo de varios usuarios y líneas de transmisión, se requiere un dispositivo que asigne adecuadamente las rutas de acceso hacia la computadora central, y los destinos de la información manejada. Este dispositivo recibe el nombre de Unidad de Control de Comunicaciones, y maneja el tráfico de información en la red. Por último, el sistema cuenta con una Unidad Central de Proceso que se encarga de tratar la información recibida, y unidades de entrada/salida, para el manejo de dispositivos auxiliares de almacenamiento e impresión.

A continuación se describen a mayor detalle los elementos pertenecientes al sistema central de procesamiento.

VI.1.2 CONTROLADORES DE COMUNICACION.

Como hemos mencionado los *modems* son dispositivos necesarios para adecuar física y eléctricamente las señales a ser transmitidas por la línea telefónica. Sin embargo, en computadoras grandes, donde se tienen varias líneas y múltiples terminales, se hace necesaria la presencia de un dispositivo capaz de atender estos recursos en forma óptima.

En forma general, una unidad de control es aquella que se encarga de adaptar lógicamente a una computadora con las características particulares de un dispositivo de entrada/salida, por medio de circuitos y programas especiales, estableciendo una ruta para el intercambio de datos y señales de control entre ellos. Manteniendo este enfoque, una unidad de control de comunicaciones sirve para adaptar y unir una red de terminales con la computadora central.

VI.1.3 MANEJADOR DE BASES DE DATOS.

Una Base de Datos es un sistema cuyo propósito general es registrar y mantener información. Incluye cuatro elementos principales: datos, *hardware*, *software* y usuarios.

Los datos son el conjunto de valores con alguna característica común, que deben ser almacenados y manipulados como un todo. La forma en que estos datos se organizan puede seguir cualquiera de los tres enfoques siguientes.

El enfoque relacional considera la organización de los datos en forma de tablas, cumpliendo ciertas restricciones que permiten considerarlas como relaciones matemáticas. Esto ocasiona que la teoría elemental de relaciones pueda ser aplicada en problemas que consideren este tipo de datos.

El enfoque jerárquico organiza los datos en estructuras de árbol, donde el elemento superior pertenece a un tipo de datos (a un archivo), y los que se encuentran por debajo de él a otro tipo (a otro archivo). En este enfoque se requieren procedimientos para la manipulación de datos, que pueden variar considerablemente de un problema a otro.

El enfoque de red utiliza el concepto de liga de registros para organizar los datos. Puede considerarse como una estructura de cadenas interconectadas que, dependiendo del tipo de problema, deberán recorrerse de una forma u otra.

El segundo elemento de un Sistema de Base de Datos es el *hardware*, y esta constituido por los dispositivos físicos de almacenamiento secundario donde residen los datos. Como ejemplos se tienen a los discos y tambores, y los dispositivos asociados a ellos como son los controladores de disco, canales, etc.

El tercer elemento es el *software*, identificado como el Sistema Manejador de la Base de Datos (DBMS), cuya función es aislar al usuario del nivel de manipulación del *hardware* en el uso de los datos. Este sistema proporciona al usuario un Sublenguaje de Datos (DSL), conformado por el lenguaje de definición de datos (DDL) y el lenguaje de manipulación de datos (DML). Este sublenguaje puede ser utilizado fácilmente por el programador para la generación de aplicaciones.

El cuarto elemento es el usuario. Es la persona que se encarga de definir las estructuras para el soporte de los datos, o bien el que los accesa. El usuario cuenta con un lenguaje de programación que le permite generar sus propias aplicaciones, ya sea de procesos en lote (*batch*) o en línea (*on-line*).

En forma general, el manejador de base de datos mantiene una relación dependiente del monitor de teleproceso de la instalación, pues es quien le permite establecer y mantener comunicación con el usuario final. La mayoría de las veces esta comunicación es transparente al usuario final, y el programador tiene una relación superficial en ella.

En el sistema que presentamos, el papel del manejador de base de datos es primordial, pues se encarga de almacenar y manipular toda la información que fluye a través de la red. El manejador de base de datos utilizado considera el enfoque relacional, y el lenguaje de cuarta generación usado es el nativo de dicho manejador.

La lógica empleada para el manejo de la información se asemeja a un sistema de correo: cada usuario cuenta con una clave de identificación que permite que el espacio ocupado por los mensajes almacenados para él sea considerado como un buzón. Asimismo, con ello es posible definir quién o hacia quién se envía cualquiera de los mensajes existentes en la base de datos.

Son dos los archivos importantes dentro de esta estructura. El primero contiene la información de identificación de cada mensaje, tal como las claves de los usuarios fuente y destino, el identificador o nombre, prioridad de entrega, tipo y fechas de entrega y recepción. El segundo almacena el texto de cada mensaje, evitando la duplicidad, donde cada registro contiene una página de información. Contiene el identificador del mensaje, que es la liga entre los dos archivos, y el número de página para asegurar la secuencia correcta al momento de la entrega.

La parte aplicativa en Mainframe se describe como un proceso de dos etapas. Desde el punto de vista Mainframe se tiene una primera etapa de recepción, donde la microcomputadora envía todos los mensajes listos para tal efecto. Al ser recibidos por el manejador de base de datos se actualiza toda la información de identificación del mensaje, excepto la fecha de entrega al usuario destino.

Una vez que se han recibido todos los mensajes, se lleva a cabo la etapa de entrega, donde el manejador de base de datos es requerido para buscar los mensajes pendientes de ser entregados al usuario que ha establecido la conexión. Una vez hecho lo anterior, la conexión termina.

La estructura general de la sesión es la siguiente:

INICIO DE SESION
Establecimiento del protocolo

RECEPCION

Etapas de conmutación

ENTREGA

FIN DE SESION

Debido al uso de un protocolo especial se cuenta con módulos para control de comunicaciones, principalmente rutinas de lectura y escritura directa desde o hacia los *buffers* de datos del monitor de teleproceso. Estas rutinas sustituyen a aquellas normalmente utilizadas para la comunicación entre el monitor y el manejador de base de datos. Esto permite el manejo de caracteres de control propios para la identificación de bloques, y verificación de errores.

VI.1.4 COMUNICACIONES.

Es importante mencionar que el ambiente de *software* utilizado pertenece a un fabricante específico, y que es posible encontrar fuertes diferencias en la implementación de nuestro esquema en un ambiente diferente. Se describirá este manejo de manera conceptual, tratando de evitar la dependencia al producto.

El encargado del manejo de los *modems* para la atención de líneas de transmisión es el monitor de teleproceso, quien recibe la información que llega por la línea, y la almacena en espacios temporales de memoria o *buffers*. Sobre este monitor de teleproceso se encuentra una aplicación que se encarga de llevar a cabo la transmisión y la recepción de la información. Dicha aplicación o programa consta de 8 módulos que a continuación se describen:

#INIT_COMMS Establece el protocolo con la microcomputadora, e inicializa todos los registros de trabajo necesarios.

#RECIBE Realiza el proceso de recepción de archivos, almacenándolos en la base de datos (ADABAS) por medio de instrucciones propias del manejador de base de datos (NATURAL).

#SWITCH_COMMS Realiza el cambio en el protocolo de modo recepción a modo envío.

#ENVIA_1 Realiza el proceso de envío de archivos, buscando en la base de datos, para el usuario que estableció la sesión, los archivos pendientes que éste no haya recibido.

#END_COMMS Finaliza el protocolo y la sesión.

#ENVIA_2 Envía la fecha y la hora registrada en el *Mainframe* hacia la microcomputadora. Esto se utiliza para conocer el tiempo y fecha exactos de conexión del usuario.

#READ Esta es una función que regresa al programa que la utiliza una línea de información enviada por el usuario. Se considera una línea de información a aquella a aquella que se encuentra contenida entre dos caracteres <Carriage Return>. Internamente esta función transfiere la información del *buffer* de recepción del monitor de teleproceso en bloques de 2400 caracteres a su *buffer* propio. Cada llamada a la función revisa este *buffer* propio. Si está vacío o si no encuentra un <Carriage Return> después de la información que éste contiene, lee otro bloque del *buffer* del monitor. En caso contrario, busca información hasta antes del primer <Carriage Return> que encuentre, y la entrega al programa en forma de parámetro de la función.

#WRITE Es una función que permite enviar una línea de información hacia el usuario. Realiza el proceso inverso de la función #READ, es decir, la línea por enviar es proporcionada como parámetro a la función y esta la almacena temporalmente en un *buffer* propio. Cada vez que este *buffer* se llena se hace uso de otro función que permitirá escribir el bloque de información en el *buffer* de envío del monitor de teleproceso.

VI.2 AMBIENTE MICRO.

VI.2.1 NORMA RS232.

Los circuitos de interfaz definidos aquí son los requeridos para la conexión de *modems* para línea privada. Estas definiciones se tomaron del EIA-232-D. Las definiciones del CCITT fueron tomadas de la Recomendación V.24. Para hacer las definiciones más legibles, se hicieron dos cambios:

1. Los circuitos CCITT están identificados por nombre así como por número. Por ejemplo, la referencia al "Circuito 103" se ha cambiado por "Circuito 103 (Datos Transmitidos)".
2. Comentarios adicionales u otras referencias se encuentran entre paréntesis cuadrados [].

EIA-232-D
Definiciones

Circuito AB - Señal de tierra o Retorno común (CCITT 102)
Dirección - No aplicable

Este conductor establece el potencial de referencia de tierra común para para todos los circuitos de intercambio. Dentro del DCE, este circuito debe estar libre en un punto de tal forma que sea posible conectar este punto a Tierra de Protección por medio de una banda de acero dentro del equipc. Esta banda puede estar conectada o no en la instalación, así como ser requerida para conocer las regulaciones aplicables o para minimizar la introducción de ruido al circuito electrónico. La banda puede estar en paralelo con otra impedancia (altas a bajas frecuencias) para proporcionar inmunidad a descargas estáticas o para otros propósitos cuando la banda sea desconectada. El requerimiento de una banda removible no trata de cambiar el estado de los DCE que proveen, de acuerdo a la EIA-422-A, una conexión del circuito AB hacia Tierra de Protección por medio de un resistor de 100 ohms.)

Circuito BA - Datos Transmitidos (CCITT 103)
Dirección - Hacia el DCE

Las señales en este circuito son generadas por el DTE y se transfieren al convertidor local de señales de transmisión para transmisión de datos hacia los DTE remotos, para mantenimiento o control del convertidor local de señales de transmisión.

El DTE retiene el Circuito BA (Datos Transmitidos) en condición de marca siempre que no se esten transmitiendo datos.

En todos los sistemas, el DTE no transmitirá datos a menos que una condición de PRENDIDO se presente simultaneamente en los siguientes cuatro circuitos:

1. Circuito CA (Petición para envío)
2. Circuito CB (Limpio para envío)
3. Circuito CC (DCE listo)
4. Circuito CD (DTE listo)

Círculo BB - Datos Recibidos (CCITT 104)
Dirección - Desde el DCE

Las señales en este círculo son generadas por el convertidor de señales de recepción en respuesta a las señales de datos recibidas desde el DTE remoto por medio del convertidor de señales de transmisión remoto, o bien por el convertidor de señales de recepción en respuesta a la señal de mantenimiento o de control del DTE local. El círculo BB (Datos Recibidos) se mantendrá en condición de UNO binario (estado de marca) mientras el Círculo CF (Detector de Señal de Línea Recibido) se encuentra en condición APAGADO.

En un canal *half-duplex*, el Círculo BB mantendrá la condición binaria UNO (condición de marca) cuando el círculo CA (Petición para envío) esté en condición de PRENDIDO, y durante un breve intervalo a continuación de la transición de PRENDIDO a APAGADO en el Círculo CA para permitir que la transmisión termine completamente (ver el círculo BA - Datos Transmitidos) y la atenuación de reflejos en la línea.

Círculo CA - Petición para envío (CCITT 105)
Dirección - Hacia el DCE

Este círculo se usa para condicionar al DCE local para transmitir datos y, en un canal *half-duplex*, controlar la dirección de la transmisión de datos del DCE local.

En canales de una o dos vías, la condición de PRENDIDO mantiene al equipo de comunicación de datos en modo de transmisión. La condición APAGADO mantiene al DCE en modo de no transmisión.

En un canal *half-duplex*, la condición PRENDIDO mantiene al DCE en modo de transmisión e inhibe el modo de recepción. La condición APAGADO mantiene al DCE en modo de recepción.

Una transición de APAGADO a PRENDIDO ordena que el DCE pase a modo de transmisión. El DCE responde efectuando tal acción e indicando su terminación colocando el Círculo CB (Limpio para Enviar) en PRENDIDO, indicando así al DTE que los datos pueden ser transferidos a través del punto interfase con el círculo de intercambio BA (Datos Transmitidos).

Una transición de PRENDIDO a APAGADO ordena al DCE completar la transmisión de todos los datos que fueron previamente transferidos a través del punto interfase con el Círculo BA y

entonces asume un modo de no transmisión o de recepción, según sea el caso. El DCE responde a esta instrucción colocando en APAGADO al Circuito CB (Limpio para Enviar) cuando esta preparado para responder a una subsecuente condición de PRENDIDO del Circuito CA.

NOTA - Un modo de no transmisión no implica que todas las señales de línea hayan sido removidas del canal de comunicación.

Cuando el Circuito CA este en APAGADO, no deberá ser cambiado a PRENDIDO hasta que el Circuito CB este en APAGADO por el equipo de comunicación de datos.

Esta permitido cambiar el Circuito CA a PRENDIDO en cualquier momento que el Circuito CB esté en APAGADO sin importar la condición de cualquier otro circuito de intercambio.

Circuito CB - Limpio para envío (CCITT 106)
Dirección - Desde el DCE

Las señales de este circuito son generadas por el DCE para indicar si esta listo o no para transmitir datos.

La condición de PRENDIDO junto con la condición de PRENDIDO de los circuitos CA, CC y, cuando exista, CD, es una indicación al DTE de que las señales presentes en el Circuito BA (Datos Transmitidos) serán transmitidos al canal de comunicación.

La condición APAGADO es una indicación al DTE de que no debe transmitir datos a través de la interfaz con el circuito de intercambio BA.

La condición de PRENDIDO del Circuito CB es una respuesta a la ocurrencia simultánea de la condición PRENDIDO en los circuitos CC (DCE Listo) y CA (Petición para envío), retardada lo suficiente para que el DCE establezca un canal de comunicación de datos. Esto incluye quitar la señal de MARCA del circuito de intercambio del DCE remoto.

Cuando no exista el Circuito CA (Petición de Envío) en el DCE con capacidad de transmisión, el Circuito CA deberá asumirse como si estuviera en condición de PRENDIDO todo el tiempo, y el Circuito CB responderá de acuerdo a ello.

[La señal Limpio para Envío tiene un significado ligeramente diferente en distintos modems. En modems full-duplex que no están equipados con un sistema de detección de errores, indica

que la portadora esta siendo recibida desde el modem remoto y es así una buena indicación de que existe un canal de comunicacion. En modems full-duplex con sistema de deteccion de errores, indica que los buffers internos del modem están listos para aceptar datos por transmitir. En algunos modems más antiguos (serie 103 Bell), las señales Limpio para Envio y Detector de Portadora están fusionadas, aunque pueden funcionar de manera independiente si ha sido instalada la opción "Separar CB-CF". En modems half-duplex, aún cuando se usan con facilidades full-duplex, la señal Limpio para Envio es solamente una versión retrasada de la señal Petición de Envio. El control de interfaz del modem emite una Petición de Envio, un reloj en el modem que detecta esta señal espera hasta 200 milisegundos (dependiendo de la opción elegida) y regresa una señal de Limpio para Envio a la interfaz del modem. En este caso, la señal Limpio para Envio realmente significa "Probablemente Limpio para Envio". }

Circuito CC - DCE Listo (CCITT 107)
Dirección - Desde el DCE

Las señales en este circuito son usadas para indicar el estado del DCE local.

La condición PRENDIDO en este circuito se presenta para indicar que:

- a) el DCE local está conectado a un canal de comunicación ("DESCOLGADO" en un servicio conmutado), y
- b) el DCE local no está en habla (alternativa para voz) o, en un servicio conmutado, que no está en modo de marcado (Se considera que el DCE está en modo de marcado cuando los circuitos asociados directamente con la función de generación de la llamada están conectados al canal de comunicación. La función de generación de llamada incluye la señalización hacia la oficina central (el marcado) y el monitoreo del canal de comunicación para continuar la llamada o contestar la señal.), y
- c) en servicio conmutado, el DCE local ha completado, cuando sea aplicable:
 1. cualquier función de reloj requerida por el sistema conmutado para completar el establecimiento de la llamada, y
 2. la transmisión de cualquier tono de contestación discreto, cuya duración es controlada únicamente por el DCE local.

Cuando el equipo local de comunicación de datos no transmite un tono de contestación, o cuando la duración del tono de contestación está controlada por alguna acción del DCE remoto, la condición de PRENDIDO se presenta cuando la condiciones anteriores (a, b y c-1) se satisfagan.

La condición de PRENDIDO de este circuito no deberá usarse para indicar la condición de ningún otro equipo, mas que la del DCE local. No deberá interpretarse ni como indicación de que el canal de comunicación ha sido establecido hacia el DCE remoto, ni de la condición de cualquier otro DCE.

La condición de apagado indica que el DCE local :

1. no está listo para operar,
2. ha detectado una condición de falla (que puede depender de la red o del DCE) que ha durado más que algún periodo de tiempo fijo, que depende de la red, o,
3. en servicio conmutado, ha detectado una indicación de desconexión de la estación remota o de la red.

En el servicio conmutado, cuando ocurre la condición de APAGADO durante la llamada antes de que el Circuito CD esté en condición de APAGADO, el DTE interpretará esto como una pérdida o aborto de la conexión, y toma la acción correspondiente para terminar la llamada. Cualquier condición de PRENDIDO subsecuente en el Circuito CC será considerada una nueva llamada.

En un servicio dedicado, cuando ocurre una condición de APAGADO, el DTE interpretará esto como una falla grave, coloca al Circuito BA en MARCA y terminará cualquier sesión de comunicación pendiente sobre el canal de comunicaciones afectado.

En el servicio conmutado, cuando se utiliza el DCE junto con un Equipo de Llamado Automático (ACE-Automatic Calling Equipment), la transición de APAGADO a PRENDIDO del Circuito CC no será interpretada como indicación de que el ACE ha renunciado al control del canal de comunicación al DCE. Dicha indicación se identifica con un código apropiado en la interfaz del ACE (ver el estándar EIA-366).

NOTA - Se le denomina "atención" al hecho de que, en un servicio dedicado, si la llamada para transmisión de datos se interrumpe por una comunicación alterna de voz, el Circuito CC estará en condición de APAGADO durante el tiempo que dure la comunicación de voz. La transmisión o recepción de las señales requeridas para condicionar el canal de comunicación o al DCE en

respuesta a la condición de PRENDIDO del circuito de intercambio CA (Petición de Envío) del DTE transmisor tendrá lugar después de que el Circuito CC esté en PRENDIDO, pero antes de que la condición de PRENDIDO se presente en los circuitos CB (Limpie para Envío) o CF (Detector de Señal de Línea Recibida).

Circuito CF - Detector de Señal de Línea Recibida (CCITT 109)
Dirección - Desde el DCE

La condición de PRENDIDO en este circuito se presenta cuando el DCE esta recibiendo una señal que cumple un determinado criterio. Este criterio es establecido por el fabricante del DCE.

La condición de APAGADO indica que no se recibe señal o que la señal recibida no cumple con el criterio establecido en el DCE.

La condición de APAGADO del Circuito CF (Detector de Señal de Línea Recibida) ocasionará que el Circuito BB (Datos Recibidos) sea puesto en condición de Uno Binario (MARCA).

Las indicaciones en este circuito seguirán al inicio o pérdida de la señal por retardos de tiempo apropiados.

En los canales *half-duplex*, el Circuito CF se retiene en la condición de APAGADO siempre y cuando el Circuito CA (Petición de Envío) esté en la condición de PRENDIDO y por un breve lapso de tiempo a continuación de la transición de PRENDIDO a APAGADO del circuito CA (ver el Circuito BB).

[El Detector de Señal de Línea Recibida, comúnmente llamado "Detector de Portadora", indica que hay un tono apropiado que está siendo recibido del *modem* remoto. En un arreglo *full-duplex*, existirá siempre y cuando exista el canal de comunicación y el *modem* remoto tenga su Petición de Envío activado, por ejemplo, estando transmitiendo. En aplicaciones *half-duplex*, existirá siempre y cuando la portadora (un tono) esté en la línea y la Petición de Envío del *modem* local esté en APAGADO; si el *modem* local no transmite el tono deberá hacerse desde el otro *modem*. Una excepción a esta regla es el *modem* Sistema Bell 202C. En este *modem* la señal del Detector de Portadora es independiente del estado de la señal de Petición de Envío local y así, existirá siempre y cuando el *modem* local o el remoto estén transmitiendo. La señal del Detector de Portadora usualmente tiene una inmunidad para muy pocas pérdidas de portadora. Pérdidas de menos de 20 +/- 10 milisegundos no se reflejan, por lo general, en la señal de Detector de Portadora.]

Circuito CD - DTE Listo (CCITT 108/2)
[Formalmente llamado Terminal de Datos Lista]
Dirección - Hacia el DCE

Las señales en este circuito son usadas para controlar la conexión del DCE al canal de comunicación. La condición de PRENDIDO prepara al DCE para ser conectado al canal de comunicación y mantener la conexión establecida por medios externos (por ejemplo, establecimiento manual de una llamada, contestación manual de llamadas o establecimiento automático de llamadas). Cuando la estación cuenta con equipo de contestación automática de llamadas y se encuentra en modo de contestación automática, la conexión a la línea ocurre sólo como respuesta a una combinación de la señal de llamada y la condición PRENDIDO en el Circuito CD (DTE Listo); sin embargo, se permite que el DTE presente la condición PRENDIDO en el Circuito CD siempre y cuando esté listo para transmitir o recibir datos, excepto en las condiciones que a continuación se enumeran.

La condición de APAGADO provoca que el DCE sea desconectado del canal de comunicación después de la terminación de cualquier transmisión en proceso. Ver el Circuito BA (Datos Transmitidos). La condición APAGADO no debería deshabilitar la operación del Circuito CE (Indicador de Llamada).

En aplicaciones sobre redes conmutadas, cuando el Circuito CD está en APAGADO, no debe ser puesto en PRENDIDO hasta que el Circuito CC (DCE Listo) sea puesto en APAGADO por el DCE.

Circuito CE - Indicador de Llamada (CCITT 125)
Dirección - Desde el DCE

La condición de PRENDIDO de este circuito indica que una señal de llamada está siendo recibida en el canal de comunicación.

La condición PRENDIDO aparecerá aproximadamente al mismo tiempo que el segmento de PRENDIDO del ciclo de campaneos (durante los campaneos) en el canal de comunicación.

La condición APAGADO debiera mantenerse durante el segmento de APAGADO del ciclo de campaneos (entre los campaneos) y todo el demás tiempo que éstos no, se reciban. La operación de este circuito no debería deshabilitarse por la condición APAGADO en el Circuito CD (DTE Listo).

Definiciones CCITT V.24

Circuito 102 - Señal de Tierra o Retorno Común

Este conductor establece la señal común de retorno para circuitos de intercambio desbalanceados con características eléctricas correspondientes a la recomendación V.28 y al potencial de referencia de c.d. para circuitos de intercambio de acuerdo a las recomendaciones V.10, V.11 y V.35.

En el interior del DCE, este circuito deberá establecerse en un punto de tal forma que sea posible conectarlo a tierra de protección por medio de una banda metálica dentro del equipo.

Esta banda metálica podrá conectarse o desconectarse de la instalación, ya sea para cumplir las regulaciones correspondientes o para minimizar la introducción del ruido a los circuitos electrónicos del equipo. Se debe prevenir el establecimiento de ciclos de tierra que sean portadores de corrientes altas.

Circuito 103 - Datos Transmitidos Dirección : Hacia el DCE

Las señales de datos emitidos por el DTE:

1. para ser transmitidas por medio del canal de datos a una o más estaciones remotas,
2. para ser transferidas hacia el DCE con el propósito de efectuar pruebas de mantenimiento bajo el control del DTE, o
3. para la programación o control del llamado serial automático de los DCE,

se transfieren por este circuito hacia el DCE.

Circuito 104 - Datos Recibidos Dirección : Desde el DCE

Las señales de datos generadas por el DCE:

1. como respuesta a las señales de línea del canal de datos recibidas desde la estación de datos remota,
2. como respuesta a las señales de prueba por mantenimiento, o
3. las señales de datos generadas por un DCE de llamado automático serial, como respuesta a señales de programación o de control desde el DTE,

se transfieren por este circuito hacia el DTE.

NOTA - Las condiciones de recepción para las señales de prueba por mantenimiento se especifican en el circuito 107.

Circuito 105 - Petición para Envío
Dirección : Hacia el DCE

Las señales en este circuito controlan la función de transmisión del DCE a través del canal de datos.

La condición de PRENDIDO ocasiona que el DCE asuma el modo de transmisión a través del canal de datos.

La condición de APAGADO ocasiona que el DCE asuma el modo de no transmisión a través del canal de datos, una vez que todos los datos transferidos al circuito 103 (Datos Transmitidos) hayan sido transmitidos.

Interrelación de los Circuitos 103, 105 y 106

El DTE notifica su intención de transmitir datos poniendo en condición de PRENDIDO el circuito 105 (Petición para Envío). Es entonces responsabilidad del DCE asumir el modo de transmisión, es decir, estar preparado para transmitir datos, y también alertar al DCE remoto y condicionarlo para recibir datos. La forma en que el DCE asume el modo de transmisión y alerta y condiciona al *modem* remoto se describe en la recomendación correspondiente para el DCE.

Cuando el DCE transmisor pone en condición de PRENDIDO al circuito 106 (Listo para Envío) con el circuito 107 (Modem Listo) en la condición de PRENDIDO, se le permite al DTE transferir datos a través de la interfaz del Circuito 103 (Datos Transmitidos). Colocando en la condición de PRENDIDO al circuito 106 con el circuito 107 también en PRENDIDO, implica que todos los datos transferidos a través de la interfaz antes de que cualquiera de los cuatro circuitos (105, 106, 107 y 108/1 o

108/2) se coloque de nuevo en condición de APAGADO, serán transferidos a la línea; sin embargo, la condición de PRENDIDO del circuito 106 no es necesariamente una garantía de que el DCE remoto esté en modo de recepción. (Dependiendo de la complejidad y sofisticación del convertidor de transmisión de señales, puede existir un retraso que va desde menos de un milisegundo hasta varios segundos entre el tiempo en que un bit es transferido a través de la interfaz hasta el tiempo en que un elemento de señalización de este bit es transmitido sobre la línea.)

Cuando el DCE transmisor coloca al circuito 106 (Listo para Envío) en PRENDIDO, con el circuito 107 (Modem Listo) en condición de APAGADO, se le permite al DTE transferir señales de programación o control al DCE de llamado automático serial a través de la interfaz con circuito 103 (Datos Transmitidos). Durante la transferencia de datos, el DTE no debe poner en APAGADO al circuito 105 (Petición para Envío) antes de que el último bit (bit de datos o elemento de paro) haya sido transferido a través de la interfaz del circuito 103. De forma similar, en ciertas aplicaciones en servicio conmutado full-duplex donde el circuito 105 no está implementado (ver las recomendaciones específicas para el DCE), esta condición se aplica igualmente cuando el circuito 108/1-108/2 es puesto en APAGADO para terminar una llamada en el servicio conmutado.

Cuando el circuito 105 está implementado, las condiciones de PRENDIDO o APAGADO en el circuito 106 durante la fase de transferencia de datos (es decir, el circuito 107 en PRENDIDO) deberá existir una respuesta a las condiciones de PRENDIDO o APAGADO en el circuito 105. Para los tiempos de respuesta del circuito 106, y para la operación del circuito 106 cuando el circuito 105 no está implementado, consultar las recomendaciones relevantes para el DCE.

Para los DCE con llamado automático serial, las condiciones de PRENDIDO y APAGADO en el circuito 106 (Listo para Envío) fuera de la fase de transferencia de datos (es decir, el circuito 107 [Modem Listo] en APAGADO) deberán depender del estado de la interfaz durante la preparación de la llamada automática y los procedimientos asociados. Las transiciones en el circuito 106 para esta aplicación estarán detalladas en la recomendación V.25 bis.

Cuando tanto el circuito 105 (Petición de Envío) como el circuito 106 (Listo para Envío) están en APAGADO, el DTE deberá mantener la condición de uno binario en el circuito 103 (Datos Transmitidos). Cuando el circuito 105 es puesto en APAGADO no deberá ser puesto en PRENDIDO otra vez hasta que el circuito 106 sea puesto en APAGADO por el DCE.

Circuito 106 - Listo para Envío
Dirección : Desde el DCE

Las señales en este circuito indican que el DCE está preparado para aceptar señales de datos para transmisión a través del canal de datos o para pruebas debidas a mantenimiento bajo el control del DTE.

La condición de PRENDIDO indica que el DCE está preparado para aceptar señales de datos provenientes del DTE.

La condición de APAGADO indica que el DCE no está preparado para aceptar señales de datos del DTE.

Circuito 107 - Modem Listo
Dirección : Desde el DCE

Las señales en este circuito indican si el DCE está listo para operar.

La condición de PRENDIDO, cuando el circuito 142 (Indicador de Prueba) esté en APAGADO o no esté implementado, indica que el conversor de señales o el equipo equivalente está conectado a la línea y que el DCE está listo para el intercambio de señales de control con el DTE para iniciar el intercambio de datos.

La condición de PRENDIDO, junto con la condición de PRENDIDO en el circuito 142, indica que el DCE está preparado para intercambiar señales de datos con el DTE para propósitos de pruebas por mantenimiento.

La condición de APAGADO, junto con la condición de PRENDIDO en el circuito 106 (Listo para Envío) indica que el DCE está listo para intercambiar señales de datos asociadas con la programación o control de los DCE con llamado automático serial.

La condición de APAGADO, junto con la condición de APAGADO en el circuito 106, indica:

1. que el DCE no está listo para operar en la fase de transmisión de datos,
2. que se ha detectado una condición de falla que puede depender de la red o del DCE, o

3. que se ha detectado una indicación de desconexión desde la estación remota o desde la red.

NOTA - El proveer las capacidades descritas en los puntos 2 y 3 es un asunto de las Organizaciones Internacionales para servicios de comunicación, a menos que estén especificadas en una recomendación para DCE.

Operación de los circuitos 107, 108/1 y 108/2

Las señales en el circuito 107 (*Modem Listo*) serán consideradas como respuesta a las señales que inician la conexión con la línea, por ejemplo, del circuito 108/1 (*Conectar el Modem a Línea*). Sin embargo, las condiciones de un canal de datos, tales como la igualación y la eliminación de *clamps*, no pueden ser esperados a que ocurran antes de poner al circuito 107 en condición de PRENDIDO.

Quando el DCE está condicionado para la contestación automática de llamadas, la conexión a la línea ocurre solamente en respuesta a la combinación de la señal de llamada y la condición de PRENDIDO en el circuito 108/2 (*Terminal de Datos Lista*).

Una opción alambrada deberá darse dentro del DCE para seleccionar la operación ya sea del circuito 108/1 o del 108/2. En ciertas aplicaciones con líneas rentadas, el circuito 108 puede no estar implementado, en cuyo caso la condición en este circuito deberá asumirse como PRENDIDO en forma permanente.

Bajo ciertas condiciones de prueba, tanto el DCE como el DTE pueden ejercer algunas funciones de los circuitos de intercambio. Así, cuando el circuito 107 (*Modem Listo*) esté en APAGADO, el DTE ignora las condiciones de cualquier circuito de intercambio desde el DCE excepto las del circuito 125 (*Indicador de Llamada*) y las de los circuitos de sincronización, y el DCE ignora las condiciones de cualquier circuito de intercambio del DTE.

La condición de APAGADO en el circuito 108/1 o 108/2 no inhabilitará la operación del circuito 125 (*Indicador de Llamada*).

Quando el circuito 108/2 (*Terminal de Datos Lista*) está en condición de APAGADO, el DTE puede comunicarse con los DCE de llamado automático serial a través de los circuitos 103 (*Datos Transmitidos*) y 104 (*Datos Recibidos*). Este estado se reconoce por la condición de PRENDIDO en el circuito 106 (*Listo para*

Envío).

Bajo las condiciones de prueba de lazo definidas en la Recomendación V.54, el circuito 107 (Modem Listo) estará en la condición de APAGADO y no responderá al circuito 108/1 o 108/2 cuando el DTE no esté involucrado con las pruebas por mantenimiento. El circuito 142 (Indicador de Prueba) estará en la condición de PRENDIDO y el circuito 107 responderá al circuito 108/1 o 108/2 cuando el DTE esté involucrado en pruebas para mantenimiento con el DCE local o remoto.

Cuando el circuito 108/1 o 108/2 (Terminal de Datos Lista) es puesta en condición de APAGADO, no deberá ser puesta en PRENDIDO otra vez hasta que el circuito 107 (Modem Listo) sea puesto en APAGADO por el DCE y el estado de APAGADO del circuito 107 haya sido reconocido por el DTE.

En el caso donde el DCE cambie al circuito 107 a APAGADO primero (ver la NOTA), el DTE deberá considerar abortada la comunicación y deberá proceder como se describe a continuación: (NOTA - El proveer este modo de operación es un asunto de las Organizaciones Internacionales de servicios de comunicación a menos que esté especificada en una recomendación para DCE.)

En el caso del circuito 108/1 (Conectar el Modem a Línea), el DTE pondrá en APAGADO a este circuito con un retraso mínimo y deberá retener al circuito en la condición de APAGADO por lo menos 500 ms y hasta que esté listo para iniciar o contestar una nueva llamada.

En el caso del circuito 108/2, la selección de un protocolo especial se deja para estudios posteriores. Las Organizaciones Internacionales de servicios de comunicación, fabricantes y usuarios están prevenidos de que aquellos protocolos desarrollados para este caso se pueden involucrar situaciones de candado o accesos no autorizados a bases de datos.

Circuito 109 - Detector de Señal de Línea de Canal de Datos
Recibidos

Dirección : Desde el DCE

Las señales en este circuito indican que la señal de línea de canal de datos que ha sido recibida se encuentra dentro de los límites apropiados, como se especifica en las Recomendaciones correspondientes para los DCE.

La condición de PRENDIDO indica que la señal recibida está dentro de los límites apropiados.

La condición de APAGADO indica que la señal recibida no se

encuentra dentro de los límites apropiados.

Circuito 108/2 - Terminal de Datos Lista
Dirección : Hacia el DCE

Las señales en este circuito controlan la conmutación de la conversión de señal o equipo equivalente hacia o desde la línea.

La condición PRENDIDO indica que el DTE está listo para operar, prepara al DCE para conectar la conversión de señal o equipo similar hacia la línea y mantener esta conexión después de que ha sido establecida por medios suplementarios.

Se permite que el DTE presente la condición de PRENDIDO en el Circuito 108/2 siempre y cuando esté listo para transmitir o recibir datos.

La condición APAGADO ocasiona que el DCE remueva la conversión de señal o equipo similar de la línea, cuando la transmisión hacia la línea de todos los datos transferidos previamente al Circuito 103 (Datos Transmitidos) y/o al Circuito 118 (Datos de Canal Alterno Transmitidos) se haya completado.

Circuito 125 - Indicador de Llamada
Dirección : Desde el DCE

Las señales en este circuito indican si una señal de llamada está siendo recibida por el DCE.

La condición PRENDIDO indica que una señal de llamada está siendo recibida.

La condición APAGADO indica que no se está recibiendo señal de llamada, y esta condición puede también aparecer durante las interrupciones de una señal de llamada de pulsos modulados.

VI.2.2 MODEMS.

Definición de modulación.

La técnica de modificar la forma de una señal eléctrica es llamada modulación. Esta señal puede tratarse de información, y en este caso, la señal por donde ésta viaja se le conoce con el nombre de portadora. La modulación cambiará la forma de la portadora de tal manera que el receptor pueda removerla y regenerar la información. El dispositivo eléctrico que ejecuta la modulación en la transmisión es llamado modulador, y el dispositivo que remueve la información de la portadora modulada es llamado demodulador.

El nombre de *MODEM* es producto precisamente de los dos términos anteriores (MODulador/DEModulador).

Usualmente la portadora es un simple tono de frecuencia que es continuo, si la modulación no está presente. Las características que pueden ser modificadas para optimizar la modulación son la amplitud, la frecuencia y la fase de una señal eléctrica u onda. Una modulación más compleja involucra dos o más de estas características.

Banda Base.

La información que es modulada sobre la portadora, ya sea en forma digital o analógica, es llamada la banda base de la señal. Muchos de los *modems* analógicos usados hoy en día, son usados para convertir la comunicación digital en una forma de onda a ser transmitida por medio de una línea telefónica. La información digital proveniente de los puertos de las computadoras es binaria por naturaleza, esto es, solo tiene dos posibles estados. Los dos estados de una señal binaria son llamados 1 y 0, marca y espacio, o hacer-deshacer (*make-break*), término usado en información codificada en modo de "retorno a cero" para flujo corriente y flujo no corriente, respectivamente. La interfaz más popular para la transmisión de información digitalizada es una establecida por EIA (*Electronics Industries Association*) y es eléctricamente polar por naturaleza. Los niveles altos y bajos están en el rango de 0 a 5 volts, usados para describir los estados de 1 y 0 de la terminal o la computadora. El nivel mínimo puede ser modulado sin posibilidad de error; a este caso se le llama el nivel liso de modulación (*slicing level*).

Tipos de modulación.

Los tipos de modulación pueden ser ejecutados sobre una señal portadora con un tipo de información analógica. Como ya

se mencionó, la modulación puede afectar sólo tres parámetros, y en base a éstos recibe el nombre de : modulación en frecuencia (FM de Frequency Modulation), modulación en amplitud (AM de Amplitude Modulation) o modulación en fase (PM de Phase Modulation). Existen también combinaciones de éstos, acorde al contenido de la información digitalizada, que hacen que la modulación tome nombres como por ejemplo de "Modulación Multibit".

Capacidad de manejo de Bits.

La capacidad de manejo de bits de un MODEM usando varios tipos de modulación depende del ancho de banda del canal en que viaja la señal modulada. La eficiencia de utilización del ancho de banda del canal, en el esquema de modulación, es llamado índice de modulación, y es calculado matemáticamente. Para la transmisión de la voz, el ancho de banda del canal deberá ser de 2400 Hz.

Acorde a la teoría, la máxima capacidad para el canal deberá ser dos veces la frecuencia de la portadora.

Características de un MODEM.

La primer característica se refiere a determinar si el modem es analógico o digital. Esto es fundamental, ya que puede usar una portadora analógica como generalmente lo hace la compañía de teléfonos para transmitir la voz y aprovechar estas facilidades, o usar algún tipo de portadora en forma de pulsos para una transmisión digital.

La siguiente característica es la velocidad. Se catalogan como modems de baja velocidad los que operan en un rango de 0 a 300 bps, velocidad media de 300 a 9600 bps, y de alta velocidad de 9600 en adelante.

Otra característica es la forma de transmisión de los datos, sincrónica o asincrónica. La diferencia entre estos tipos de transmisión es que la primera maneja dos señales : una de datos y una de reloj. En este caso, cada caracter ocupa solo los siete u ocho bits con los que se inicializó el puerto serial de comunicaciones, por lo que no se tiene sobrecarga de bits. Para el caso de la transmisión asincrónica se tiene que añadir a cada caracter los correspondientes bits de inicio y fin, lo que aumenta la carga de trabajo, sin embargo, por la característica de que las microcomputadoras tienen implementado un puerto de comunicaciones serial asíncrono, controlado por una UART, este último modo de transmisión es el más popular.

La siguiente característica en el análisis de un *modem* es la capacidad con que cuenta para ser originador y/o contestador. Hasta antes de salir al mercado los *modems* inteligentes (*Smartmodem*), los había dedicados a alguna de las funciones antes mencionadas. Su uso es ahora casi exclusivo para ambientes con controladores de comunicaciones que no requieren la opción de originadores.

VI.2.3 COMANDOS HAYES.

Con el surgimiento de los *modems* inteligentes (*Smartmodem*), se revolucionó la industria de telecomunicaciones. La compañía precursora en la comercialización de tales *modems* fue HAYES, quien sacó al mercado un *modem* capaz de manejar una serie de comandos que facilitan en gran parte los enlaces y permiten realizar algoritmos implementados por software, de tal forma que es posible desde "conversar" con el *modem*, hasta lograr ajustes automáticos de velocidad de acuerdo a la calidad de la línea.

En algunos *modems* este conjunto de comandos ha sido ampliado, y otros no tienen implementados en su procesador todo el conjunto de intrucciones estándar.

Cada comando Hayes inicia siempre con el prefijo AT(Attention), seguido de alguna(s) de las instrucciones que se describen a continuación :

- A : El *modem* descolgará el teléfono y generará un tono para que se enlace otro dispositivo.
- A/ : Repite el último comando.
- B : Selecciona el estándar para enlace : B0 CCITT V.22 y B1 BELL 212A.
- C : Habilitar o deshabilitar (0 o 1 respectivamente) el *switch* para transmitir la portadora.
- D : Indica que se marcará un número telefónico. Entonces este comando deberá ser seguido de los dígitos que se desean marcar. Si se ejecuta sin ningún dígito, actuará como *modem* originador.
- E : Comando de eco. Si se encuentra habilitado (AT E1) el *modem* contestará primero con el mismo comando, y después alguna de las respuestas que se explicarán posteriormente; si está deshabilitado (AT E0), sólo contestará con estas últimas.

- F : Habilita o deshabilita el eco en estado en línea (On-line) anexando un 0 o un 1, respectivamente.
- H : Colgar el teléfono (hang-up).
- I : Anexando un 0 contesta con el código de identificación del producto. Con 1 regresa el checksum de la ROM, y con 2 regresa OK o ERROR si este checksum es correcto o no.
- L : Controla el volumen de la bocina del modem : de 0 a 3 lo aumenta gradualmente.
- M : Controla la activación de la bocina del modem : 0 la apaga, 1 la activa hasta que se tiene portadora y 2 la mantiene siempre activa.
- O : Inicia o termina un ciclo de verificación del modem (0 o 1, respectivamente).
- P : Indica que el marcado del teléfono será efectuado por medio de pulsos.
- Q : Habilita o deshabilita los códigos de respuesta del modem, si el comando se acompaña de un 0 o un 1 respectivamente.
- S : Seguido de un número moverá un apuntador al registro que se indica con el número.
- T : Indica que el número que se va a marcar se hará por tonos.
- V : Controla el formato de respuesta en forma numérica o verbal (0 y 1 respectivamente).
- Z : Actúa como reinicialización, regresando al modem a los valores de fábrica.
- +++ : Regresa al modem a modo comando.
- , : Pausa en el marcaje.

Todos los comandos terminan con un <RETURN>, a excepción del comando +++. Las posibles respuestas del modem son :

Numérico	Verbal	Significado
0	OK	El comando fue ejecutado.
1	CONNECT	Se logró el enlace con algún dispositivo remoto.
2	RING	Se está recibiendo una llamada
3	NO CARRIER	No hay portadora.
4	ERROR	El comando no fue ejecutado por ser desconocido.
6	DIALTONE	Al descolgar la bocina se obtuvo tono para marcado.
7	NO DIALTONE	Al descolgar la bocina no se obtuvo tono para marcado.
8	BUSY	El teléfono que se marcó está ocupado.

VI.2.4 COMUNICACIONES.

UART.

La mayoría de las computadoras usan un circuito integrado conocido como UART (*Universal Asynchronous Receiver Transmitter*), el 8250 de Intel, que es el controlador de comunicaciones asíncrono. Este circuito proporciona muchas facilidades para el programador, ya que contiene registros que facilitan el envío y recepción de información, así como verificación del estado del puerto.

Cada 8250 tiene 10 registros de 1 byte cada uno, la mayoría de los cuales son de inicialización, y sólo unos cuantos se usan regularmente. Todos los registros son accedidos directamente en la dirección de algún puerto de I/O. Esta dirección es calculada mediante una base y un desplazamiento; esta base varía de acuerdo al puerto de comunicaciones que es usado, y sus posibles valores son los siguientes :

Puerto de comunicaciones	Dirección base
COM1	03F8h
COM2	02F8h
COM3	03E8h
COM4	02E8h

Los 10 registros de la UART son accedidos mediante 7 puertos del microprocesador. Dependiendo de la operación por realizar en el puerto (lectura o escritura), la interpretación del puerto será distinta. Los registros se muestran a continuación :

Desplazamiento	LSR bit 7	Significado
0	0	<i>Transmitter Holding Register (THR) y Receiver Data Register (RDR)</i>
0	1	<i>Baud Rate Divisor, low byte (BRDL)</i>
1	1	<i>Baud Rate Divisor, high byte (BRDH)</i>
1	?	<i>Interrupt Enable Register (IER)</i>
2	?	<i>Interrupt Identification Register (IIR)</i>
3	?	<i>Line Control Register (LCR)</i>
4	?	<i>Modem Control Register (MCR)</i>
5	?	<i>Line Status Register (LSR)</i>
6	?	<i>Modem Status Register (MSR)</i>

? - No importa su valor

El THR toma el byte de datos a enviar. Cuando se vaya a escribir un dato en el registro es necesario verificar que este registro está vacío, mediante el bit 5 del LSR, que será explicado posteriormente.

El RDR toma el byte del dato más recientemente recibido. Se puede leer este dato cuando el bit 0 del LSR indica que se ha recibido un dato.

El registro completo del BRD (BRDL y BRDH) está formado por los 16 bits que indican la velocidad de transmisión. Para determinarla, se deberá dividir la velocidad del reloj interno de la UART (1,8432 Mhz) entre el BRD de la siguiente manera :

$$\text{BRD} = \frac{\text{Velocidad del Reloj}}{16 \times \text{Bps deseado}}$$

Por ejemplo para calcular el valor del BRD para una velocidad de 1200 bps se tiene :

$$\text{BRD} = \frac{1843200}{16 \times 1200} = \frac{1843200}{19200} = 96 = 0060h$$

Entonces el BRDL debe ser igual a 00 y el BRDH a 60h.

En la siguiente tabla se muestra la relación entre velocidades y valores del BRD.

Bit Rate	BRDH	BRDL
50	09h	00h
110	04h	17h
300	01h	80h
1200	00h	60h
2400	00h	30h
4800	00h	18h
9600	00h	0Ch
19200	00h	06h

Para alterar este registro es necesario primero cambiar el estado del bit 7 del LCR a 1, y una vez establecida la velocidad, poner el valor del bit en 0.

El IER controla el tipo de interrupción generada por la UART, y cada vez que ocurre una de ellas, su bit correspondiente en el registro es fijado en 1. Es posible que varias interrupciones ocurran a un mismo tiempo. A continuación se muestra una tabla con el significado de cada posible interrupción, y las acciones a tomar.

Bit	Causa	Acción
0	Dato recibido	Leer el RDR
1	THR vacío	Escribir dato de salida al THR
2	Dato erróneo o caída de la línea	Leer el LSR
3	Cambio en el MSR	Leer MSR
4-7	Sin uso, siempre son fijados a 0	

El IIR identifica la ocurrencia de una interrupción, y su tipo. La siguiente tabla informa al respecto.

Bit	Significado
0	Ha ocurrido más de una interrupción.
1-2	Identifican la interrupción :
	00 = Cambio en el MSR
	01 = THR vacío
	10 = Dato recibido
	11 = Error en la recepción o línea caída

El LCR es el controlador primario de la línea serial. Sus bits son asignados de la siguiente manera :

Bit	Significado	Valores	Notas
0-1	Longitud del carácter		
	5 bits	00	
	6 bits	01	
	7 bits	10	
	8 bits	11	

2	Bits de alto		
	1 bit	0	
	1.5 bits	1	Si se usa longitud de 5 bits por caracter
	2 bits	1	Si se usa longitud de 6,7,8 bits por caracter
3-5	Paridad		
	IGNORADA	000	
	PAR	100	
	IMPAR	110	
	MARCA	101	
	ESPACIO	111	
6	Condición para cortar		
	Deshabilitada	0	
	Habilitada	1	
7	Port Toggle		
	Normal	0	Usar THR/RDR y IER
	Alternado	1	Usar BRDL y BRDH

El MCR fija los controles de las líneas para el *modem*, y desde éstas indica a la computadora si está listo para el envío y recepción de caracteres. La asignación de *bits* es la siguiente:

Bit	Significado
0	Línea DTR fijada activa
1	Línea RTS fijada activa
2	Salida de usuario #1
3	Salida de usuario #2
4	UART <i>loopback</i>
5-7	Sin uso, fijados a cero

La línea DTR (*Data Terminal Ready*) indica al *modem* que la computadora está encendida y lista para recibir información. La línea RTS (*Request To Send*) indica al *modem* que la computadora está lista para enviar algún dato. Algunos *modems* podrán ser configurados para ignorarlas, y otros permitirán fijarlas como activas. La salida de usuario #1 es usada por la UART para sus servicios de interrupción, y la salida de usuario #2 es usada sólo por algún *hardware* especializado, como es el caso de los *modem* HAYES que la usan para reinicializar el *modem*. El bit 4 permite al programa de comunicaciones realizar una verificación, de tal manera que el dato que se pone como salida (*output*), es regresado como entrada (*input*).

El LSR tiene el estado de la línea de comunicaciones. Desde este registro es posible determinar el estado de la línea de transmisión durante el envío o recepción de un carácter. La asignación de bits es la siguiente :

Bit	Significado
0	Dato recibido, <i>byte</i> en RDR.
1	Error de <i>overrun</i> . Un <i>byte</i> arribó antes de que el anterior pudiera ser leído.
2	Error de paridad.
3	Error de estructura. Pérdida de sincronía en la transmisión (inexistencia del bit de paro después de que el carácter fue leído).
4	Detección de caída de línea.
5	THR vacío, listo para la salida de un carácter.
6	Transmit Shift Register (TSR) vacío. (El TSR toma el carácter del THR y lo coloca en la línea, bit a bit).
7	Tiempo de espera transcurrido (<i>timeout</i>).

El estado del MSR depende del estado de la línea (alto o bajo), y del cambio de estado de alguna de las líneas del *modem*, desde la última lectura de los registros. Los bits son asignados de la siguiente manera :

Bit	Significado
0	Cambio en <i>Clear to Send</i> (CTS)
1	Cambio en <i>Data Set Ready</i> (DSR).
2	Cambio en <i>Ring Indicator</i> (RI).
3	Cambio en <i>Data Carrier Detected</i> (DCD).
4	<i>Clear to Send</i> (CTS) fijado en alto.
5	<i>Data Set Ready</i> (DSR) fijado en alto.
6	<i>Ring Indicator</i> (RI) fijado en alto.
7	<i>Data Carrier Detected</i> (DCD) fijado en alto.

Las señales del *modem* corresponden a los cambios de una señal eléctrica en la conexión entre la computadora y el dispositivo serial.

Dependiendo del dispositivo, estas señales del *hardware* pueden ser o no usadas. A continuación se da una breve explicación de lo que significa cada señal :

Clear to Send (CTS) : El *modem* está listo para recibir caracteres de la computadora.

Data Set Ready (DSR) : El *modem* está encendido y listo para operar.

Ring Indicator (RI) : La línea telefónica recibe una llamada.

Data Carrier Detect (DCD) : El *modem* está conectado a otro *modem*.

Inicialización del puerto de comunicaciones.

El manejo directo de la UART para la inicialización del puerto de comunicaciones es una tarea compleja, y requiere de una secuencia correcta de valores en los registros para producir efectos específicos. El BIOS tiene una interrupción que facilita en gran medida esta labor. Esta interrupción es la número 14h, y dentro de sus servicios están los de inicialización, envío de un caracter, recepción de un caracter y

revisión del estado del puerto. Únicamente nos encargaremos de la parte de inicialización, que es el servicio 0h. Para realizar dicha inicialización se deben de seguir los siguientes pasos :

- Poner valor de AH=0.
- Poner el número de puerto en DX (0=COM1; 1=COM2).
- Poner valor de AL, en base a los siguientes criterios :

Bit	Significado	Valores
0-1	Longitud de palabra	
	No usado	00
	No usado	01
	7 bits	10
	8 bits	11
2	Bits de alto	
	1 bit	0
	2 bits	1
3-4	Paridad	
	Ninguna	00
	Impar	01
	Ninguna	10
	Par	11
5-7	Velocidad de Bits	
	110 bps	000
	150 bps	001
	300 bps	010
	600 bps	011
	1200 bps	100
	2400 bps	101
	4800 bps	110
9600 bps	111	

- Invocar a la interrupción 14h.

Manejador de comunicaciones.

El programa que controla las comunicaciones en la PC fue escrito en PASCAL, con algunas inclusiones de ensamblador 8086.

Este manejador inicializa el puerto de comunicaciones, acorde a lo señalado en la sección anterior, solicitando únicamente los datos de puerto y la velocidad, ya que siempre es usada la paridad impar (ODD), es decir, siete bits de datos y un bit de paro (STOP).

Para la recepción de datos se sustituye el vector de interrupción correspondiente al puerto con el que se vaya a trabajar (OBH y OCH para el COM1 y COM2, respectivamente), por la dirección de una rutina para manejar los caracteres recibidos, guardando en una variable la dirección original, para posteriormente restaurarla. El algoritmo de esta rutina contempla el manejo de los caracteres del protocolo, así como el control de algunas banderas para verificar tiempo de retraso (timeout), sincronía, y verificación de sobrecupo en el *buffer* de recepción (overrun). Se utiliza un *buffer* circular para recibir los caracteres. Estos van siendo agrupados en forma de paquetes de información, es decir, segmento de datos a partir del <SOH> hasta el <CR> siguiente, y que se analizan para tomar alguna acción.

El envío de información debe efectuarse carácter por carácter, y esperar alguna respuesta por parte del *modem* o del *Mainframe*. A continuación se muestra un algoritmo, escrito en pseudocódigo, que describe las rutinas básicas que se usan en el manejador de comunicaciones.

Inicialización :

- Tomar puerto y velocidad.
- De acuerdo al puerto, guardar la dirección del vector de interrupción correspondiente, y cambiarla por la de la rutina de recepción de caracteres del programa.
- Invocar Interrupción 14 con : puerto, velocidad, 7 bits de datos y 1 de alto, paridad impar.
- Si hay algún estado de error, restaurar el vector de interrupción y terminar el programa.
- Detectar la presencia de un *modem* HAYES, enviando algunos comandos estándar y reconociendo las respuestas.
- Si no es un *modem* HAYES, restaurar el vector de interrupción y terminar el programa.

Envío de información :

- Verificar que el registro de salida esté vacío.
Mientras el bit 5 del LSR no esté en alto
Esperar 1 milisegundo.
- Solicitar envío; sensar DSR, activar RTS, y esperar el CTS.
Poner máscara 0Bh en MCR.
Mientras el bit 1 del MSR esté en bajo
Esperar 1 milisegundo.
- Enviar el carácter.
Poner carácter en el THR

Para recibir información.

- Recibir el carácter en el buffer circular.
Leer el THR.
Si llegó un <Carriage Return>, prender bandera de fin.
- Revisar los apuntadores del buffer, para verificar el sobrecupo (overrun).
- Revisar la presencia de portadora; abortar el proceso en caso de ausencia.
Si el bit 7 del MSR está en bajo aborta proceso.

Transferencia en el protocolo.

Envío de información.

1. Mientras haya información por enviar y no haya error hacer lo siguiente:
 - 1.1 Verifica los caracteres de la línea a enviar, y codifica de acuerdo a KERMIT.
 - 1.2 Calcula Checksum y empaqueta la línea.

- 1.3 Envía la línea
- 1.4 Espera respuesta por parte de *Mainframe*
- 1.5 Si no hay solicitud de retransmisión o no se agotó el tiempo máximo de espera , continúa, de otra manera repite pasos 1.3 y 1.4 hasta que no haya solicitud de retransmisión o el número de retransmisiones se agote.
- 1.6 Si no hay error, continúa
De otra manera, aborta el proceso.

Recepción de información.

1. Mientras no haya error y no termina recepción, hacer lo siguiente:
 - 1.1 Mientras bandera de fin esté apagada y no haya error hacer lo siguiente:
 - 1.1.1 Espera y verifica el retraso
 - 1.2 Si no hubo error hacer lo siguiente:
 - 1.2.1 Calcula el *Checksum*.
 - 1.2.2 Si los *Checksum* no coinciden, solicita retransmisión, si se excede el máximo número de retransmisiones, aborta el proceso.

Abortar proceso.

1. Cuelga el teléfono
 - 1.1 Cancela protocolo
 - 1.2 Envía secuencia para regresar a modo comando ('+++')
 - 1.3 Envía comando para colgar teléfono ('ATH')
2. Restaura vector de interrupción
3. Termina el programa.

VI.3 PROTOCOLO UTILIZADO.

Con la creciente popularidad en el uso de las microcomputadoras tipo PC, se tuvo la necesidad de transferir información entre ellas, sin importar la distancia.

Teniendo en cuenta que el manejar un enlace punto a punto involucra un número de sesiones igual al número de destinatarios, se pensó en diseñar un sistema de transferencia de información centralizado, en el cual se tendrá una red en forma de estrella, con un buzón central capaz de almacenar grandes volúmenes, así como de rutear la información a todos los destinatarios correspondientes en un solo enlace, y que el software residente en la PC consuma pocos recursos, tales como espacio en disco y una configuración estándar de memoria RAM (512 K). Surge así el sistema INFOCOMM, en honor al protocolo del mismo nombre.

INFOCOMM es una combinación de dos protocolos ya establecidos y probados en la transferencia de archivos entre una micro y una mainframe : XMODEM y KERMIT.

XMODEM.

Con el surgimiento de las microcomputadoras CP/M durante los 70's, se tuvo la necesidad de crear un protocolo para transmisión de datos por la vía telefónica. Durante 1977, Ward Christensen escribió la primera versión de XMODEM y se produjo una revisión significativa en 1979; se hizo al programa y el protocolo del dominio público, y se produjo un documento definitivo en 1982 llamado XMODEM.DOC. Existen diferentes versiones de XMODEM para sistemas de CP/M y MSDOS, disponibles junto con el documento en muchos de los Bulletin boards de USA. Con la enorme popularidad de las micros en los 80's, XMODEM fue incorporado en algunos paquetes comerciales.

El principal propósito de desarrollar XMODEM fue el proveer un método eficaz y libre de errores para la transferencia de archivos binarios. XMODEM utilizando una verificación estándar de datos (checksum) detectaba un 99.6% de los errores. Usando una adición posterior de una verificación por redundancia cíclica (CRC) de la CCITT mejoró a 99.997% en la detección de errores.

Caracteres de control.

Son cuatro los caracteres que usa XMODEM como control :

<SOH> Start of header (Inicio de encabezado).

<ACK> Acknowledge (Contestación afirmativa).
<NAK> Negative Acknowledge (Contestación negativa).
<EOT> End of Transmission (Fin de transmisión).

Control en la transferencia de datos.

El formato de los bloques de transmisión de XMODEM se muestra a continuación :

```
-----  
|SOH|Núm. bloque|255-Núm. bloque|128 bytes de datos|Chksum|  
-----
```

El protocolo usa los caracteres de control para el *handshaking* que usa el software para enviar y recibir información. El tamaño máximo del bloque es de 132 caracteres, contando los de control, siendo 128 para los datos (como máximo), manteniendo así la consistencia con la longitud de palabra de CP/M. Cuando comienza la transferencia de un archivo, la máquina receptora indica que está lista enviando un <NAK> a la transmisora; es entonces cuando se envía la transmisión de bloques con el formato antes descrito.

Verificación de errores y recuperación.

Acorde con Christensen, el *checksum* es una simple suma de los valores decimales de los caracteres que forman los datos, despreciando el acarreo.

El número de bloque se encuentra en dos campos distintos. El segundo campo es un complemento del primero. Si los datos están dañados (los *checksum's* no coinciden), la estación receptora envía un <NAK> a la estación transmisora, como petición de retransmisión. Si los datos no se alteraron, se incrementa el número de bloque y se envía el siguiente; en caso de una retransmisión, el bloque se envía exactamente igual. En la siguiente figura se simula una transmisión típica:

```

>-----\NAK\----->
<-----/SOH/BLK #1/BLL #1/-DATOS-/CHECKSUM/-----<
>-----\ACK\----->
<-----/SOH/BLK #2/BLL #2/-DATOS-/CHECKSUM/-----<
>-----\NAK\----->
<-----/SOH/BLK #2/BLL #2/-DATOS-/CHECKSUM/-----<
>-----\ACK\----->
<-----/SOH/BLK #3/BLL #3/-DATOS-/CHECKSUM/-----<
>-----\ACK\----->
<-----\EOT\-----<
>-----\ACK\----->

```

Dado que el *checksum* simple utilizado en la versión original de XMODEM puede llegar a generar datos incorrectos, John Byrnes añadió el uso de un CRC.

El CRC usado hace más robusta la detección de errores, y es una forma alternativa en el uso del protocolo. Los cambios para usar el CRC son integrales (*straightforward*). Byrnes añadió un *handshake* para solucionar el problema de incompatibilidad, ya que en este se identifica si el programa con el que se va a comunicar soporta el CRC, y si es así, que se use esta modalidad durante la transmisión, de lo contrario, se usa la versión original, para no perder la compatibilidad. Byrnes sumo una función "C" para la verificación del CRC.

Esta función calcula el CRC usado en el protocolo. El primer argumento es un apuntador al bloque del mensaje. El segundo argumento es el número de *bytes* que contiene al mensaje. El tamaño de bloque usado por el protocolo es de 128 *bytes*. La función regresa un valor entero conteniendo el CRC. Los primeros 16 *bits* de este entero son los coeficientes del CRC. Dado que el CRC requiere 16 *bits* en lugar de 8, el bloque de transferencia es de 133 *bytes* en lugar de los 132 originales. La estructura del bloque es la siguiente :

|SOH|Núm.bloque|255-Núm. bloque|128 bytes de datos|CRC hi|CRC lo|

El *handshaking* inicial usando CRC empieza cuando el receptor envía una <C>. Si el transmisor responde después de 3 segundos con un <SOH>, se trabajará bajo esta modalidad; de otra manera después de que transcurra un tiempo definido, el transmisor enviará un <NAK>, usando la modalidad estándar. Las versiones nuevas de XMODEM soportan ambos modos para tener interacción con versiones pasadas. Un ejemplo de transmisión con CRC se muestra a continuación :

```
>-----\C\----->
--3 segundos espera; si no SOH, correr el Checksum, enviar NAK--
>-----\NAK\----->
<-----/SOH/BLK #1/BLL #1/-DATOS-/CRChi/CRClo/-----<
>-----\ACK\----->
<-----/SOH/BLK #2/BLL #2/-DATOS-/CRChi/CRClo/-----<
>-----\NAK\----->
<-----/SOH/BLK #2/BLL #2/-DATOS-/CRChi/CRClo/-----<
>-----\ACK\----->
<-----/SOH/BLK #3/BLL #3/-DATOS-/CRChi/CRClo/-----<
>-----\ACK\----->
<-----\EOT\-----<
>-----\ACK\----->
```

Codificación de la información.

XMODEM esta diseñado para trabajar con los 256 caracteres ASCII, el llamado extendido, aunque se usó principalmente para transmisiones de caracteres del ASCII estándar (de 7 bits). Dado que el tamaño del bloque es fijo, no es posible confundir los bytes de datos con los de control.

Transparencia de la información.

XMODEM no tiene una transferencia transparente. Todos los datos son enviados sin separadores o alguna codificación especial para los caracteres de control.

Sincronización.

Cuando llega un número de bloque válido, debe de ser mayor al último o repetirse. Cuaiquier otro caso indica la pérdida de sincronía y se aborta la transmisión enviando un <CAN> (cancelación).

Comunicación Transparente.

XMODEM esta orientado para transmisión asíncrona en la red pública de teléfonos. Esto puede no suceder, o puede surgir la necesidad de parar el flujo de transmisiones para un uso posterior en algunos modernos sistemas de comunicación, que requieren una demanda en el flujo de control fuera del ejercido por el software de comunicaciones en cada fin de conexión.

KERMIT.

Así como XMODEM fue diseñado para transmisiones de archivos libres de errores entre microcomputadoras, KERMIT intenta usarse en un medio "donde haya una diversidad de computadoras (micros, minis, redes, Mainframes, etc.) y distintas arquitecturas". El objetivo del diseño original de KERMIT, fue hecho por Bill Catchings bajo la supervisión de Frank Da Cruz de la Universidad de Columbia, en los 80's, y fué el de proveer un protocolo para la transferencia de archivos libres de errores entre una DEC sistema 20, una Mainframe IBM 370 y diversas microcomputadoras.

El resultado fue un diseño general, aunque las Mainframes IBM fueron los grandes retos. La unidad de transferencia en KERMIT es llamada "paquete", pero este es formado de manera similar a como lo hace XMODEM.

Las características de KERMIT se dirigen hacia varios problemas en la transferencia de archivos, emarcados por las transferencias estándar y las facilidades que proporcionan algunas computadoras. Sus principales características son :

- Permite la comunicación entre varios equipos terminales que manejen comunicación asíncrona estándar.
- La comunicación es excelente en modo *half-duplex*.

- La longitud del paquete es variable, aunque corta (máximo de 96 caracteres) es la mejor por que muchos sistemas no tienen grandes capacidades de *buffering*.
- Los paquetes pueden ser enviados a distintas direcciones si se tienen sistemas que trabajen en modo *half-duplex*.
- Cuando esta disponible, una pausa facilita el reinicio de la transmisión despues del último paquete.
- Todas las transmisiones son código ASCII.
- KERMIT no fue diseñado para programarse en un lenguaje determinado: es un protocolo portable.

Caracteres de Control.

KERMIT esta diseñado para transferir los caracteres ASCII despleables, desde el 32 (espacio) hasta el 126 (la tilde) inclusive. Los caracteres usados como control son :

<SOH> *Start of header* (Inicio de encabezado).

<ACK> *Acknowledge* (Contestación afirmativa).

<NAK> *Negative Acknowledge* (Contestación negativa).

Control de Transferencia de Datos.

La estructura general de KERMIT se muestra a continuación :

```
-----
| MARCA | LONGITUD | SECUENCIA | TIPO | DATOS | CHECKSUM |
-----
```

MARCA : Inicio de encabezado <SOH>.

LONGITUD : Longitud del paquete. Abarca desde este mismo, hasta el último contenido en los datos (excluidos la MARCA, y el CHECKSUM). Su valor se determina mediante la siguiente fórmula :

$$LONGITUD = \text{Char}(LONGITUD)$$

donde:

$$\text{Char}(X) = (\text{Valor decimal del ASCII de } X) + 32$$

SECUENCIA : La secuencia es cíclica de 0 a 64. El carácter que indicará la secuencia se determinará como :

SECUENCIA = Char(SECUENCIA)

con la misma fórmula definida anteriormente.

TIPO : Define el contenido del campo de datos. Puede ser :

D Datos

Y Contestación afirmativa <ACK>.

N Contestación negativa <NAK>.

S Inicia un envío.

R Se inicia una recepción.

B Terminar comunicación <EOT>.

F Encabezado de archivo.

Z Fin de archivo.

E Error.

G Comando genérico. El primer carácter del campo de datos, seguido posiblemente de parámetros, es ejecutado por la computadora receptora. Estos son :

L Fin y termina (Logout).

F Fin pero sin terminar.

D Requerimiento del directorio, seguido de una máscara opcional:

U Requerimiento del uso del disco.

E Borrado de un archivo.

T Desplegar un archivo (Type).

Q Requerimiento del estado del servidor.

C Comando para el host.

X Despliegue del campo de datos.

DATOS : Son un conjunto de caracteres ASCII comprendidos entre el 32 al 126.

CHECKSUM : Es un campo para verificar la integridad de los datos. Contempla todos los del paquete, excepto los de MARCA y los de CHECKSUM. Se define como :

CHECKSUM = Char((SUM+(SUM And 192)/64) And 63)
donde SUM es la suma de los valores de todos los caracteres que contempla el checksum y la función Char(X) es la definida anteriormente.

El campo de DATOS contiene el contexto del paquete, aunque algunos TIPOs de paquetes no requieren ningún tipo de DATOS. Los caracteres que no son despleables (caracteres del 128 al 255) son precedidos de un prefijo, y un caracter repetido tiene un prefijo que indica un contador. El *checksum* puede ser simple con una suma de caracteres o usando un CRC, tal como en XMODEM. Para evitar la incompatibilidad con otros sistemas, KERMIT utiliza una combinación de CR-LF (*Carriage Return-Line Feed*) como terminación de registro.

La manera en que KERMIT encripta la información es la siguiente : si un caracter en el campo de datos es de control (0-31) este se cambia por el caracter resultante al aplicar la función Char(X) antes mencionada, y se le antepone un caracter especial (#). Para el caso de caracteres mayores a 127, el procedimiento es como sigue : el octavo bit se "apaga", quedando un caracter menor a 127, el cual, si es desplegable se le antepone un caracter especial (&), y si resulta que es un caracter de control se antepone este caracter especial y se le da el tratamiento descrito antes. Para transmitir el "#" o el "&", se antepone siempre un "#". Al llegar al lado receptor, se efectua la operación inversa y se recupera la información original.

Una transferencia típica de un archivo usando protocolo KERMIT se muestra a continuación :

```
<-----/SOH/LEN/SEQ/TYPE(Send Init)/DATA/CHKSUM/-----<
>-----/SOH/LEN/SEQ/TYPE(Y-ACK)/DATA/CHKSUM/----->
<-----/SOH/LEN/SEQ/TYPE(File Head)/DATA/CHKSUM/-----<
>-----/SOH/LEN/SEQ/TYPE(Y-ACK)/DATA/CHKSUM/----->
<-----/SOH/LEN/SEQ/TYPE(Data)/DATA/CHKSUM/-----<
>-----/SOH/LEN/SEQ/TYPE(Y-ACK)/DATA/CHKSUM/----->
<-----/SOH/LEN/SEQ/TYPE(Data)/DATA/CHKSUM/-----<
>-----/SOH/LEN/SEQ/TYPE(Y-ACK)/DATA/CHKSUM/----->
<-----/SOH/LEN/SEQ/TYPE(Z-EOF)/DATA/CHKSUM/-----<
>-----/SOH/LEN/SEQ/TYPE(Y-ACK)/DATA/CHKSUM/----->
<-----/SOH/LEN/SEQ/TYPE(B-EOT)/DATA/CHKSUM/-----<
>-----/SOH/LEN/SEQ/TYPE(Y-ACK)/DATA/CHKSUM/----->
```

INFOCOMM.

El protocolo INFOCOMM reúne las principales características de XMODEM y KERMIT. De XMODEM toma el *handshaking*, que es un poco más elaborado que KERMIT, y más confiable, y de éste último toma la manera de encriptar la información, así como la manera de manejar la secuenciación de bloques, añadiéndose otras bondades como el poder manejar una longitud de bloque tan grande como lo permita la línea de transmisión.

Caracteres de control.

Son siete los caracteres que usa INFOCOMM como control :

<SOH> *Start of header* (Inicio de encabezado).

<ACK> *Acknowledge* (Contestación afirmativa).

<NAK> *Negative Acknowledge* (Contestación negativa).

<EOT> *End of Transmission* (Fin de transmisión).

<ETX> *End of Text* (Fin de texto).

<CAN> *Cancel* (Cancelación).

<NUL> *Null* (Nulo).

El registro que envía INFOCOMM tiene el siguiente FORMATO :

```
-----  
|SOH | SEQ | DATOS | ETX | CHKSUM |  
-----
```

donde:

SOH : Inicio de bloque (*Start of Header*).

SEQ : Secuencia de bloque.

DATOS : Información a transmitir.

ETX : Fin de texto (terminan datos).

CHKSUM : Ocupa dos *bytes* y contienen un valor hexadecimal el cual representa la suma de los valores de los caracteres de los DATOS, incluyendo el caracter de SEQ.

El protocolo usa el caracter <CAN> para el *handshaking*, iniciando siempre la microcomputadora, que es la que invoca al programa en *mainframe* y es quien contesta en la forma antes descrita. El tamaño del bloque es definido en los programas aplicativos y es tan grande como lo permita el medio de transmisión, generalmente la red comutada de teléfonos.

Verificación de errores y recuperación.

El *checksum* que se maneja ocupa dos *bytes*, y es un número en formato hexadecimal que corresponde a la suma de los valores decimales de los caracteres que forman los datos, aplicada a la siguiente fórmula :

$$CHKSUM = \text{HEX}(\text{SUM}(\text{DATOS}) \text{ Mod } 255).$$

La secuencia de bloque es de 0 a 9, cíclica y es el primer caracter después del <SOH>. La secuencia del bloque sólo se aplica cuando se transmiten datos, no así cuando se trata de caracteres de sincronía. Si los datos son alterados y los valores de *checksum* no coinciden, el software de *mainframe* envía un <NAK> solicitando una retransmisión; si los datos son correctos, se incrementa el número de bloque de manera cíclica y se espera el siguiente.

Para conservar la compatibilidad, cada registro termina con una combinación de CR-LF (*Carriage Return-Line Feed*).

Una transferencia típica de un archivo usando protocolo INFCOMM se muestra a continuación :

<-----/SOH/CAN/ETX/CHKSUM/-----<
>-----/SOH/NUL/ETX/CHKSUM/----->
<-----/SOH/NUL/ETX/CHKSUM/-----<
>-----/SOH/SEQ1/DATA/ETX/CHKSUM/----->
<-----/SOH/ACK/ETX/CHKSUM/-----<
>-----/SOH/SEQ2/DATA/ETX/CHKSUM/----->
>-----/SOH/NAK/ETX/CHKSUM/----->
>-----/SOH/SEQ2/DATA/ETX/CHKSUM/----->
<-----/SOH/ACK/ETX/CHKSUM/-----<
>-----/SOH/EOT/ETX/CHKSUM/----->
<-----/SOH/ACK/ETX/CHKSUM/-----<
>-----/SOH/NUL/ETX/CHKSUM/----->
<-----/SOH/NUL/ETX/CHKSUM/-----<
<-----/SOH/CAN/ETX/CHKSUM/-----<

VI.3.- INTEGRACION DEL SISTEMA EN OSI.

Como consecuencia de la instalación, en prácticamente todo el mundo, de redes de comunicación de computadoras y con la experiencia que ello ha generado, los diseñadores y los organismos internacionales de estándares identificaron siete niveles o estratos que abarcan todos los elementos que componen un sistema de comunicaciones.

Una de las organizaciones que ha definido estos niveles es la *International Standard Organization (ISO)*, la cual ha planteado como modelo estándar el ampliamente conocido Modelo OSI (*Open System Interconnection*) de siete niveles o estratos, bajo el cual los desarrolladores, proveedores y usuarios de redes de comunicación pueden implantar sus sistemas y tender cada vez más a la adopción de estándares aceptados por los organismos internacionales.

El modelo de siete niveles OSI abarca desde el nivel

básico de comunicación hasta el nivel aplicativo, pretendiendo que cada estrato o nivel sea independiente de los demás, de tal manera que cuando haya cambios en alguno de ellos, no se afecte a los otros niveles.

A continuación se describen dichos niveles, y como son identificados en nuestro sistema :

Nivel 1: Nivel Físico.

Este nivel se encarga en general de las características de tipo eléctrico, mecánico y de procedimientos de la conexión física. Brevemente podemos decir que en éste nivel se lleva a cabo la transmisión de los bits de un punto a otro, sin que importe ni el significado ni la integración de la información. Como normas reglamentarias de este nivel podemos mencionar el RS232C, RS449, de la EIA o sus similares de la CCITT (Comité Consultivo Internacional de Telegrafía y Telefonía) como el X.21, etc.

En nuestro sistema este nivel está representado por las normas RS232C para la microcomputadora y el controlador de comunicaciones para la Mainframe, correspondiendo a los niveles eléctricos. El canal de transmisión es un par telefónico. La parte mecánica se refiere a los elementos físicos que se necesitan para conectar al DTE (Data Terminal Equipment) y al DCE (Data Communication Equipment) entre sí.

Nivel 2: Nivel de eslabonamiento de datos.

Este nivel tiene como función el agrupamiento o eslabonamiento de bits en tramas o bloques para ser transmitidos a través del nivel físico en una forma confiable. En esta capa se realiza la detección y corrección de errores. Es importante comentar que en esta capa se manejan los protocolos de comunicación de datos. Brevemente diremos que un protocolo de comunicación de datos es un procedimiento de control de los datos transmitidos.

En nuestro sistema esta parte se encuentra implantada en el software de comunicaciones, por medio de un protocolo especial.

Nivel 3: Nivel de red.

Este nivel se involucra con el enrutamiento de paquetes de información de un punto a otro a través de una subred de comunicación.

En este nivel se definen las capacidades de servicio y las características que la red proporciona al usuario. Por

ejemplo el protocolo X.25 permite el acceso de usuarios sincrónicos a redes públicas de conmutación de paquetes.

Este nivel se encuentra implícito en los servicios que nos proporciona la red pública telefónica.

Nivel 4: Nivel de transportación.

En este nivel se verifica lo que es el transporte de la información de un nodo a otro de una red, considerando transparente lo que sería el equipo de tránsito, de comunicaciones, etc.

Este nivel lo identificamos en dos partes dentro de nuestro sistema. La primera parte considera al ambiente micro en donde a los archivos por transmitir se les asigna un encabezado que involucra a los destinatarios. La segunda parte considera al ambiente Mainframe en donde se dirigen los archivos para su posterior envío.

Nivel 5: Nivel de sesión.

El nivel de sesión está dedicado a hacer posible la comunicación de un proceso a otro proceso. Aún no existe un estándar internacional que sea realmente aceptado. Este nivel se encuentra involucrado en las siguientes fases del sistema:

- Establecimiento de la sesión en Mainframe.
- Invocación al módulo de envío de información.
- Invocación al módulo de recepción de información.
- Terminación de la sesión en Mainframe.

Nivel 6: Nivel de presentación.

La función de este nivel de presentación es realizar ciertas transformaciones útiles en la información antes de que ésta sea entregada al nivel de sesión. Las transformaciones llevadas a cabo en nuestro sistema son la encriptación, segmentación, integración y desencriptación.

Nivel 7: Nivel de aplicación.

Este nivel corresponde directamente a los desarrolladores de software. No existe todavía una estandarización que en este aspecto haya sido aceptada.

Nuestro sistema utiliza dos manejadores de bases de datos: uno residente en micro y otro en Mainframe. El manejo en micro, por parte del usuario se simplificó al uso de menús.

VIII.- CONCLUSIONES.

La proliferación del uso de las computadoras en el mundo ha sido tan explosiva que es imposible compararla con cualquier otro aspecto del desarrollo social.

La gran demanda que esto representa, ha provocado que los fabricantes se preocupen en desarrollar equipo con cada vez mayor capacidad de procesamiento.

Aunado a esta situación existe otro campo que ha requerido una atención creciente: el de comunicaciones. Los esfuerzos realizados para enlazar todo este universo de equipos han demostrado que el establecimiento y la adopción de estándares deberá ser el camino a seguir. El antiguo enfoque de exclusividad del usuario en el uso de una marca específica tiende a desaparecer, y la colaboración entre fabricantes es cada vez mayor.

Para el usuario, el ambiente de comunicaciones representa ahora, y lo hará con mayor fuerza en el futuro, parte fundamental en la toma de decisiones a nivel corporativo. La información necesaria para este proceso podrá ser concentrada y analizada en puntos estratégicos, y distribuidos eficazmente entre los centros operativos, es decir en forma rápida, segura y barata.

Dentro de este contexto, el presente trabajo, si bien atiende a la problemática de un usuario específico, responde en gran medida a una necesidad de tipo común en los ambientes corporativos: la comunicación eficaz.

Cabe decir que la filosofía de servicio presentada en este trabajo no es nueva, pues ha tenido un desarrollo y un uso importante en otros países, tanto de Europa como de América. Si bien es cierto que esta filosofía permitirá optimizar las actividades de comunicación de muchas empresas, también es cierto que se requiere de una sólida infraestructura de comunicaciones para transmisión electrónica de datos, que en nuestro país aun no es sólida. La red pública telefónica utilizada por nosotros como canal de transmisión, constituyó el único medio masivo de distribución de información, y debido a que no fue diseñada para este fin, representa un elemento sin control en nuestro ambiente.

Por último, cabe destacar la necesidad de capacitación en el área de comunicaciones, para aquellos profesionistas interesados en este aspecto del ambiente de cómputo, ya que consideramos que a corto plazo se requerirá de personal capaz de enfrentar una acelerada expansión en la aplicación y uso de este tipo de sistemas.

APENDICES

A.1 SISTEMA OPERATIVO DOS PARA PC.

El sistema operativo DOS para la computadora personal IBM/PC es un sistema operativo uniusuario y, por lo tanto fue diseñado para cumplir esta tarea; sin embargo, por la historia que antecede a él (sobre todo la existencia de UNIX) sus estructuras no son rígidas y podemos pensar en dejar residentes en memoria programas que se activan, suspendiendo a otros, con la ocurrencia de algún evento en particular como es el caso de SideKick y otros paquetes.

La estructura interna de DOS esta organizada en capas: una relacionada directamente con el hardware (BIOS), la siguiente provee los servicios de máquina extendida (KERNEL), un procesador de comandos (SHELL), y los programas de soporte (EN DISCO) que lo acompañan. Las primeras tres capas son inicializadas por programas cargadores (loaders) que se inicializan a si mismos.

El ROM BIOS arranca al encender la máquina y se encarga de leer de disco el primer sector y ejecutarlo (Boot sector). Este, a su vez, se encarga de leer algunos archivos de disco (MSDOS.SYS e IO.SYS o bien IBMDOS.COM e IBMBIO.COM en PC-DOS) con las principales partes de DOS, y se encarga de la instalación de los manejadores de dispositivos especiales encontrados en ROM, así como los declarados en el archivo CONFIG.SYS. Por último, como parte de MSDOS.SYS, el módulo conocido como SYSINIT inicializa las tablas y buffers de DOS y se descarta a él mismo para dar control al programa de interfaz a usuario (SHELL).

En términos del sistema operativo que nos ocupa, se llama residentes a los dispositivos que están contenidos en IO.SYS, e instalables a aquellos dispositivos que se cargan durante la inicialización del sistema (boot-up) como el caso de el RAMDISK, MOUSE, etc. y se cargan por medio de comandos en el archivo CONFIG.SYS.

BIOS (Basic Input Output System)

Esta es la primera capa del sistema, está especificada con cada computadora en particular y es provista por la empresa que la manufactura dado que tiene a su cargo el manejo de los dispositivos físicos de entrada y salida de información, en particular de los programas manejadores de:

- Despliegue de consola y teclado (CON).
- Impresora de línea (PRN).
- Dispositivos auxiliares (AUX).
- Día y tiempo del sistema (CLOCK).
- Dispositivo de bloque encargado de inicializar el sistema desde el disco (boot disk device, block device).

El BIOS esta localizado en ROM y además es utilizado en diagnósticos y la inicialización del sistema.

La siguiente capa del sistema operativo (conocido como el Kernel de MS-DOS) se comunica con este sistema manejador de dispositivos a través de paquetes de requerimientos de entrada/salida, estos manejadores traducen los requerimientos en los comandos propios para los diferentes controladores de hardware.

EL KERNEL.

El llamado kernel del sistema operativo es el sistema operativo en sí, que es visto por los programas de aplicación, y es quien proporciona los servicios de máquina extendida, en forma de servicios independientes del hardware. Estas incluyen:

- Creación y manejo de programas (manejo de procesos).
- Manejo de memoria (*memory admon*).
- Manejo de periféricos (reloj, impresora, etc).
- Manejo de archivos, directorios y registros (*file system*).

El kernel está contenido en el archivo MSDOS.SYS. Los programas accesan las funciones provistas por el kernel usando la instrucción de interrupción de software que generalmente es INTR.

Las interrupciones del sistema operativo que estan reservadas para este propósito son:

Interrupción	Descripción
20H	Para terminar la ejecución de un programa.
21H	Funciones generales del sistema operativo.
22H	Dirección de terminación de rutinas.
23H	Dirección del manejador del Ctrl-C.
24H	Dirección del manejador de errores críticos.
25H	Para leer de la unidad de disco.
26H	Para escribir en la unidad de disco.
27H	Para terminar la ejecución de un programa y que permanezca residente en memoria.
28H-3FH	No documentadas.

La interrupción 21H es la principal fuente de servicios del sistema operativo.

SHELL.

El SHELL o intérprete de comandos es un programa que se carga en memoria justo después del BIOS y el KERNEL. Provee la interfaz entre este último y el usuario. Este SHELL se llama COMMAND.COM y puede ser reemplazado por algún otro comercial o alguno desarrollado por un programador. El COMMAND.COM puede ejecutar un conjunto de comandos internos, cargar y ejecutar un programa e interpretar archivos batch. Los programas que ejecuta el COMMAND.COM son archivos con extensión .COM, .EXE o .BAT. Estos últimos proveen un lenguaje de programación limitado junto con una serie de comandos del sistema.

Cuando se inicializa una microcomputadora el COMMAND.COM es cargado en memoria en dos porciones: una residente y otra no. Cuando la ejecución de un programa ha terminado, la porción residente, antes de regresar el control al sistema, verifica si el área de memoria en donde se encuentra la porción no residente, fué utilizada por el programa, pues este pudo haber requerido de más memoria para su ejecución. Si la

porción residente determina que si ha sido utilizada la porción no residente, una copia de la porción no residente del COMMAND.COM es cargada nuevamente del disco y así el COMMAND.COM continúa con su operación normal.

La razón de que pueda ser utilizada la porción no residente del COMMAND.COM por cualquier programa ejecutable es que ésta contiene las rutinas para el procesamiento de los archivos batch y ciertos comandos que no son importantes en el momento de la ejecución de un programa. Recordemos que la porción residente es la que se encarga de regresar el control al sistema después de la ejecución de un programa.

El COMMAND.COM interpreta los comandos proporcionados por el usuario, primero verificando si el comando especificado concuerda con el nombre de alguno de los comandos internos del SHELL. Si es así ejecuta el comando, si no, busca en el directorio del disco por un archivo .COM, .EXE o .BAT (en ese orden) con el nombre especificado. Si encuentra este archivo, mediante la función EXEC del sistema operativo (interrupción 21 función 4BH) lo carga en memoria y lo ejecuta.

A.2 SISTEMA OPERATIVO MVS PARA MAINFRAME.

Un sistema operativo es una colección de programas que se encargan de manejar el trabajo interno de una computadora. La finalidad del sistema operativo deberá ser mantener los recursos de la máquina tan ocupados como sea posible, asegurando que sea procesada la máxima cantidad de trabajo, y que cada unidad de trabajo se procese eficientemente. MVS trata de hacer esto.

MVS es el resultado del proceso evolutivo de los sistemas operativos diseñados para computadoras grandes IBM. Sus siglas provienen de Memoria Virtual Múltiple (*Multiple Virtual Storage*). Este sistema operativo tiene la capacidad de procesar trabajos de tamaño mayor que la capacidad de almacenamiento interno con que cuenta la computadora, debido a que mantiene un control especial de él, de tal forma que hace que su capacidad de almacenamiento parezca más grande de lo que realmente es. Por ello se denomina virtual, pues es sólo una simulación. El término multiple significa que varios usuarios pueden usar todo el espacio virtual, mencionado anteriormente, a un mismo tiempo. Algunas de sus principales características se mencionan a continuación.

MULTIPROGRAMACION

La multiprogramación es una técnica para mejorar el uso del procesador de la máquina. Cuando un trabajo no puede usarlo, posiblemente por estar esperando datos de un dispositivo de entrada/salida, el sistema tiene la facultad de suspender o interrumpir dicho trabajo; entonces, el procesador puede destinarse a atender otro trabajo, en tanto el anterior tiene disponible los recursos necesarios. Una vez que esto se ha cumplido, el sistema regresa la atención al primer trabajo, que se ejecuta desde el punto en que fué suspendido. Esto se debe a que el sistema, al momento de la suspensión, almacenó toda la información relevante del ambiente en que se ejecutaba el trabajo.

MEMORIA VIRTUAL

Existen dos tipos de almacenamiento físico. El primero se refiere a la memoria situada dentro de la computadora, y que es llamada memoria real. El segundo tipo es externo al procesador, y se ubica en los dispositivos de acceso directo, principalmente manejadores de disco, y en dispositivos de acceso secuencial, como son las unidades de cinta. Son conocidos como memoria auxiliar, y el procesador tiene acceso a ellos por medio de canales de entrada/salida.

Por otro lado, para que una instrucción de un trabajo pueda ser ejecutada por el procesador, dicha instrucción y los datos asociados a ella deben estar situados en memoria real. Sin embargo, no se requiere que la totalidad del programa o trabajo se encuentre localizado en memoria real. Por esta razón se diseñó una técnica en la cual se mueven piezas del programa hacia memoria real cuando el procesador está listo para ejecutarlas, y se mueven hacia memoria secundaria cuando éste no las necesita. De esta forma, un sistema operativo puede ejecutar más programas, y más grandes, en forma concurrente.

Para lograr esto, el sistema operativo crea un espacio de direccionamiento virtual de tamaño constante para el programa o trabajo, donde cada dirección representa un byte de dicho espacio.

Además, el almacenamiento físico se divide en áreas separadas, todas ellas del mismo tamaño y accesibles por medio de una dirección virtual única. En memoria real estas áreas se llaman *frames*, y en memoria auxiliar *slots*. Con esto, cada *frame* o *slot* es mapeado con el espacio de direccionamiento previamente definido, de tal forma que ahora tendrá asociada la dirección virtual que ocupa en dicho espacio.

De igual forma, el sistema operativo divide al programa o trabajo en piezas del mismo tamaño que los *slots* o *frames*, que son llamadas páginas, y asigna a cada una de ellas la dirección única del *slot* o *frame* que la contiene.

Con toda esta información el sistema genera un conjunto de tablas que le permiten saber en qué tipo de memoria y dónde se encuentra cada una de estas páginas, y colocarla o quitarla de memoria real cuando sea necesario. Este movimiento de páginas entre *slots* de memoria auxiliar y *frames* de memoria real es llamado paginación.

Conceptualmente, la paginación unificó a la memoria real y a la auxiliar, haciendo que la memoria disponible no se limite al tamaño de memoria real de la computadora, sino al número de direcciones virtuales con que cuenta el sistema.

MEMORIA VIRTUAL MULTIPLE

Para lograr que cada usuario pudiera tener acceso al rango completo de direcciones virtuales del sistema, cada espacio de direccionamiento cuenta con una tabla de memoria virtual. Esta solución fue llamada Memoria Virtual Multiple (MVS), y bajo este concepto, cada espacio de direccionamiento consta de 16 megabytes de memoria virtual, con 24 bits de direccionamiento.

A medida que el MVS fue utilizado para atender ambientes cada vez más complejos, estos 16 megabytes empezaron a ser insuficientes. La Arquitectura Extendida (MVS/XA), con 31 bits de direccionamiento permitió que cada espacio tuviera 128 veces más direcciones disponibles, es decir, 2 gigabytes de memoria virtual. Esta mejora originó la necesidad de un aumento en la cantidad de memoria real disponible en el equipo. Aún cuando la arquitectura extendida permitió procesar trabajos más grandes, la carga de tiempo ocasionada por las operaciones de entrada/salida para acceso de datos siguió siendo la misma.

La Arquitectura de Sistemas Corporativos (MVS/ESA) permite a un espacio de direccionamiento acceder uno o más espacios de datos. Este nuevo concepto se refiere a un área de memoria virtual adicional, con un tamaño de 2 gigabytes de direccionamiento. En él es posible almacenar programas del sistema, del usuario o datos, teniendo como condición que esta información no puede ser ejecutada directamente. En otras palabras, aunque se almacenen programas, el sistema los considera datos. Esta facilidad permite ahorrar tiempo a aquellos programas que procesan los mismos datos en forma repetitiva, ya que en lugar de acceder un archivo directamente, éstos se obtienen de memoria virtual.

Por último cabe decir que hasta antes de la aparición de la arquitectura de sistemas corporativos, el crecimiento en el tamaño de memoria virtual había sido vertical. Con esta última, el crecimiento también es horizontal, y marca una nueva era en la historia de los sistemas operativos para máquinas grandes IBM.

A.3 INTERRUPCIONES.

Una interrupción es la suspensión momentánea de la secuencia de un procedimiento o un programa que permite que otro proceso se ejecute. Una interrupción puede ser generada por un evento externo al proceso actual, o ésta puede ser generada como resultado de una acción del propio proceso (por proceso actual nos referimos al proceso a ser interrumpido).

Fisicamente la interrupción esta dada por una señal enviada al microprocesador que hace que este suspenda su labor actual y transfiera el control al programa encargado de ejecutar la interrupción, conocido como *Interrupt Handler*.

El *Interrupt Handler* determina la causa de la interrupción, toma la acción apropiada y regresa el control al proceso original que fue suspendido.

TIPOS DE INTERRUPCIONES

Las interrupciones pueden ser agrupadas dentro de dos categorías :

- Interrupciones de *Hardware*
- Interrupciones de *Software*

A su vez las interrupciones de *hardware* se pueden dividir en internas y externas.

Interrupciones de *Software*:

Las interrupciones de *software* son precisamente las generadas por un programa que manda ejecutar la instrucción de interrupción (generalmente INT) para un cierto propósito. Por ejemplo la interrupción 10H nos permite manejar el video de una PC, la 14H el puerto de comunicaciones etc.

Interrupciones de *Hardware*:

Las Interrupciones de *Hardware* son causadas por eventos fisicamente ligados con el diseño de *Hardware*. Por ejemplo el teclado de la computadora está ligado con la interrupción 9. Cuando presionamos alguna tecla el procesador interrumpe al proceso actual y transfiere el control al *Interrupt Handler* (la dirección del *Interrupt Handler* se encuentra almacenada en el vector de interrupción asociado con la interrupción 9).

Interrupciones Internas de *Hardware*:

Estas interrupciones son generadas por ciertos eventos ocurridos durante la ejecución de un programa como por ejemplo

una división entre cero. Estas interrupciones son dadas dentro del mismo procesador de la computadora.

Interrupciones Externas de Hardware:

Estas interrupciones son generadas por dispositivos controladores de periféricos (teclado, video, impresora) o por coprocesadores (8087/80287). Algunas de estas interrupciones externas son canalizadas a través de un controlador de interrupciones programable (Intel 8259A) mejor conocido como PIC.

Vectores de interrupción.

Existen 256 diferentes interrupciones dentro de la arquitectura de una PC, numeradas de la 0 a la ffh. Algunas de estas interrupciones están definidas para uso del procesador (por ejemplo la interrupción 0 es la interrupción dada por la división entre cero). Otras interrupciones están definidas para invocar funciones dentro del ROM-BIOS, y otras más están definidas para uso del Sistema Operativo. Las interrupciones restantes están disponibles para programas aplicativos y dispositivos externos de hardware.

Cada interrupción es representada por un vector de dirección de 4 byte's almacenado en la memoria de la computadora. Estos vectores se encuentran en las localidades 0 a la 0fff y se conocen como Vectores de Interrupción. En otras palabras los Vectores de Interrupción contienen las direcciones de memoria en donde comienzan los diferentes *Interrupt Handler*.

Mapa de la memoria de una PC bajo DOS.

COMMAND PROCESSOR
(COMMAND.COM Transient Portion)

TRANSIENT PROGRAM AREA
(TPA)

COMMAND PROCESSOR
(COMMAND.COM Resident Portion)

DEVICE DRIVERS

DOS (BIOS y Kernel)

INTERRUPT VECTORS

Las primeras 400h localidades de memoria estan reservadas para los vectores de interrupción. Arriba de los vectores se encuentra DOS. A continuación de DOS se encuentran los programas manejadores de dispositivos que son cargados cuando DOS es inicializado. Por ejemplo, si usamos un RAM Disk o el archivo ANSI.SYS (manejador de teclado y monitor) estos manejadores de dispositivos son cargados después de DOS. A continuación de los manejadores de dispositivos se encuentra la porción residente del programa procesador de comandos de DOS (COMMAND.COM).

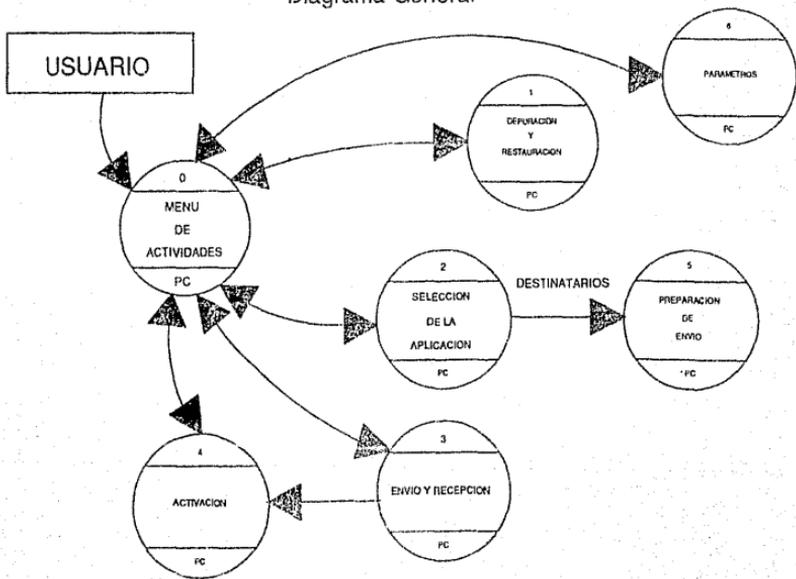
Este programa procesa los comandos en línea de DOS (por ejemplo un dir *.*) y ejecuta programas. El área de memoria reservada para la ejecución de programas de usuario conocida como TPA (Transient Program Area) se encuentra después de la porción residente del programa procesador de comandos de DOS.

Cuando un usuario corre un programa éste es cargado en la TPA. Esta área es transitoria, porque una vez que el programa termina su ejecución, es liberada. Por último se encuentra la porción transitoria del programa procesador de comandos de DOS. El programa de un usuario puede utilizar la parte de memoria que ocupa la parte transitoria del programa procesador de comandos, si esto llega a ocurrir la parte residente del programa procesador de comandos relocaliza a la parte transitoria en donde el programa del usuario termine.

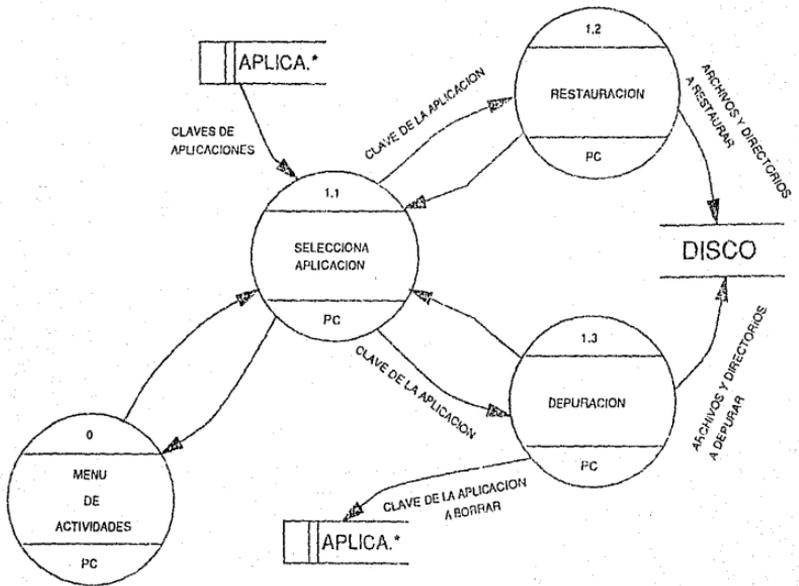
A.4 DIAGRAMAS DE FLUJO DE DATOS

ENVIO DE APLICACIONES

Diagrama General

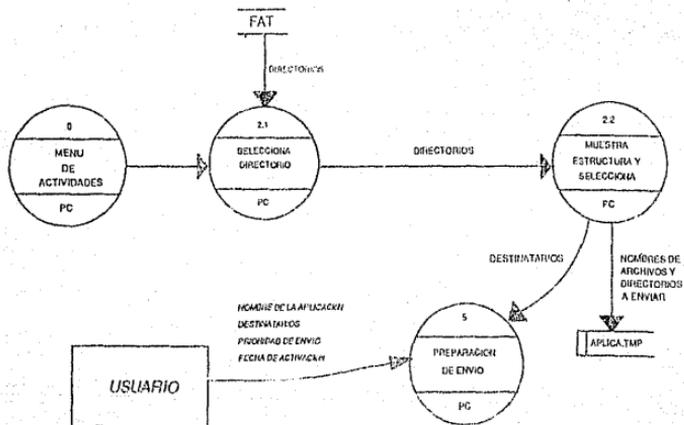


MODULO 1 : DEPURACION Y RESTAURACION

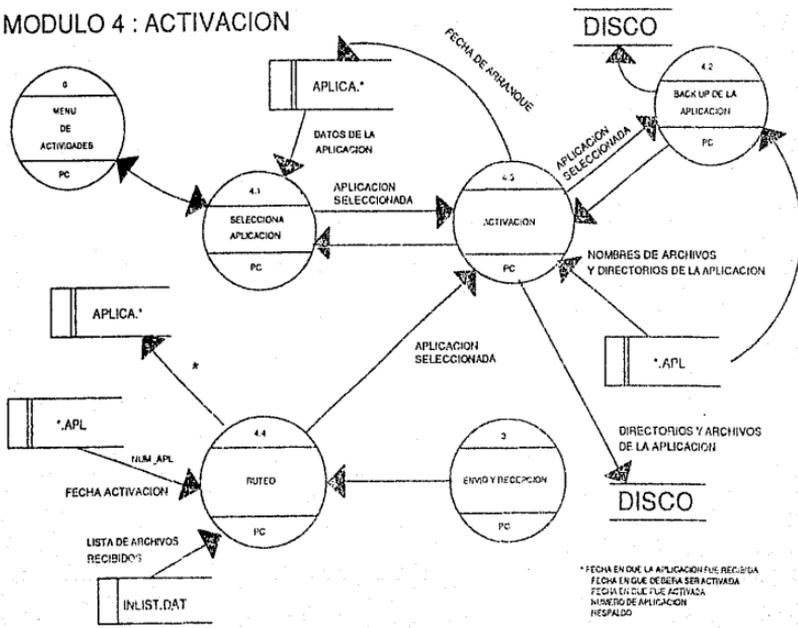


MODULO 2 : SELECCION DE LA APLICACION

96 C

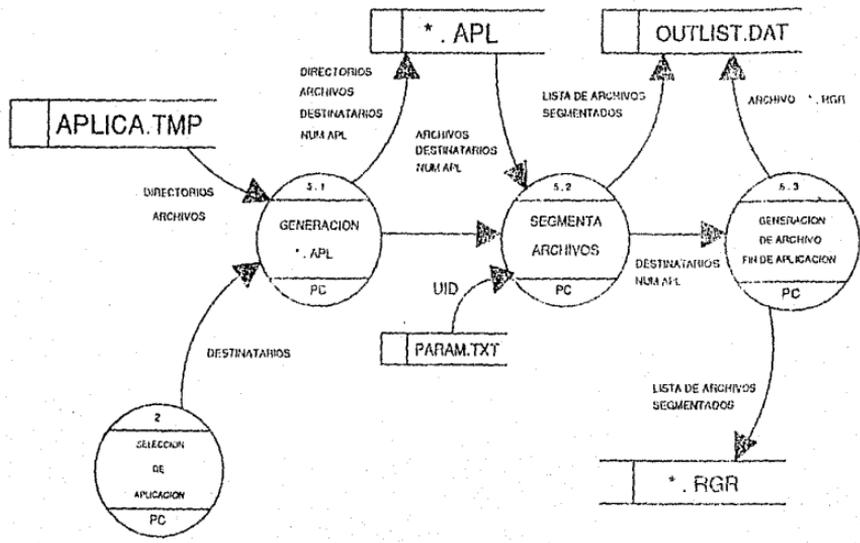


MODULO 4 : ACTIVACION

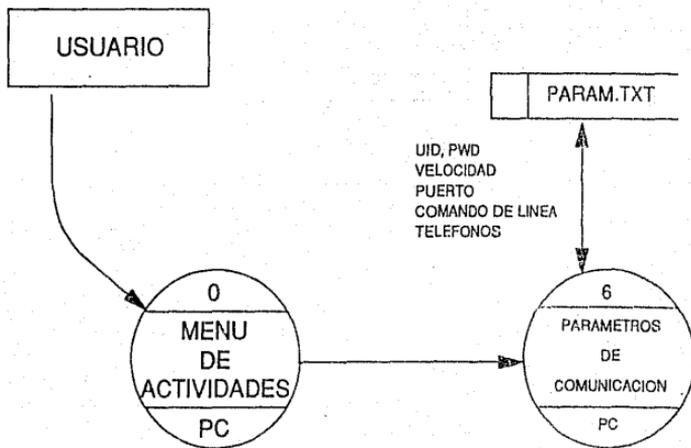


96

* FECHA EN QUE LA APLICACION FUE RECIBIDA
 FECHA EN QUE DEBERIA SER ACTIVADA
 FECHA EN QUE FUE ACTIVADA
 NOMBRE DE APLICACION
 RESPALDO



MODULO 5: PREPARACION DE ENVIO



MODULO 6: PARAMETROS DE COMUNICACION

A.5 MANUAL DE USUARIO

SISTEMA ENVIO DE APLICACIONES

MANUAL DE USUARIO

CONTENIDO

	Pág.
Introducción	2
Instalación	5
Iniciar el sistema	6
Capítulo 1 - Parámetros de comunicación	8
Capítulo 2 - Nueva aplicación	11
Capítulo 3 - Activación	14
Capítulo 4 - Restauración	17
Capítulo 5 - Depuración	20
Capítulo 6 - Comunicaciones	22

Presentación.

El presente manual tiene como objetivo proporcionar al usuario los elementos necesarios para la operación del sistema Envío de Aplicaciones.

Se presenta una introducción que muestra las características del sistema, un texto para instalarlo, uno para iniciar su operación y seis capítulos que muestran la operación de cada uno de los módulos que lo integran.

Para la mejor comprensión de este manual se recomienda que sea lea utilizando conjuntamente el paquete de software.

INTRODUCCION

Como la intención de este manual es facilitarle al usuario la utilización de este sistema, a continuación haremos una breve descripción del funcionamiento del mismo y también una referencia a conceptos que deben conocerse antes de utilizarlo.

El objetivo principal del envío de aplicaciones es contar con un sistema que permita no solo el envío de archivos entre dos nodos remotos, si no el control sobre conjuntos de ellos en cuanto a su actualización, su fecha de inicio de utilización, directorios en donde deben almacenarse, manejo de distintas versiones, confidencialidad, que se obtengan respaldos de la información actualizada y que se pueda depurar cualquier aplicación.

Para actualizar una aplicación en una microcomputadora remota se deben de seleccionar los archivos que la conforman y los directorios destino en donde será actualizada. Se puede de manera opcional dar un comando para que en el momento de la actualización sea ejecutado (por ejemplo un programa, un archivo batch o algún comando del sistema operativo).

Las aplicaciones tendrán una fecha de activación. Por lo tanto se pueden activar de forma manual o automática. Si al momento de recibir la aplicación la fecha de activación es anterior o igual a la actual la aplicación será activada automáticamente.

Cuando se activa una aplicación se verifica que no existan previamente los archivos que la conforman. Si existen se obtiene un respaldo de seguridad de estos archivos.

Si la aplicación actualizada es satisfactoria y ya no se necesitan los respaldos, estos pueden ser depurados; en caso contrario cualquier aplicación puede ser restaurada a su ambiente original.

La transmisión y recepción de aplicaciones se lleva a cabo mediante un enlace Micro-Mainframe. Todas las aplicaciones son enviadas a la Mainframe que se encarga del control de distribución de las mismas.

Definiciones:

Por Aplicación denominamos al conjunto de archivos que forman un sistema completo de información como son paquetes de software, sistemas a la medida, conjuntos de archivos de datos, etc.

Teclado

Para el sistema Envío de Aplicaciones se definió un conjunto compacto de teclas que permiten su operación:

Return o Enter - Tecla de selección.
Notifica al sistema que el elemento sobre el cual la barra o cursor está posicionado ha sido seleccionado.

Flechas - Teclas de posicionamiento.
Permiten elegir un elemento de una lista, posicionando una barra o un cursor sobre él.

Esc - Tecla de cancelación.
Cancela el proceso que se este llevando a cabo. Sobre un menú actual regresa el control al menú anterior.

F1 - Tecla de ayuda.
Dependiendo del módulo en que se encuentre el usuario, muestra una pantalla con información auxiliar.

Cabe mencionar que estas son las teclas básicas de operación del sistema, pero se cuenta con otras teclas que también operan, tales como la barra espaciadora (mismo funcionamiento que la tecla de flecha hacia abajo), las teclas de PgUp y PgDn (en campos de captura permiten ir al primer campo y/o al último respectivamente).

Cursor

Debido a que en el sistema se utilizan menús, se han determinado dos tipos de cursor para la obtención de información. La barra en video inverso indica que debe elegirse una opción o varias, por lo que se tienen que utilizar las teclas de posicionamiento, selección o cancelación.

El guión bajo será utilizado en aquellos lugares en donde sea necesario proporcionar información explícitamente.

Convenciones

A lo largo de este manual se hace referencia a distinta información que se debe introducir al sistema, por lo que se han usado las siguientes convenciones.

Cuando se indica una tecla, el nombre de esta la escribiremos entre paréntesis cuadrados y con letra mayúscula. Por ejemplo, encontraremos el siguiente texto:

Para seleccionar presione la tecla [ENTER], para cancelar presione [ESC].

INSTALACION

1.- Ponga el disco flexible que se le proporcionó en la unidad de disco A: y verifique que este cerrada.

2.- Teclee:

```
A:[ENTER]
INSTALA C:[ENTER]
```

donde C: corresponde al disco en donde quedará instalado el sistema. Si su disco no es 'C' sustituya esta letra por la correspondiente. Asegure que la ruta (path) que tiene en su disco sea el correcto.

INICIAR EL SISTEMA

Para iniciar el sistema Envío de Aplicaciones, el usuario deberá ubicarse en el directorio en donde haya sido instalado, y teclear APL seguido de un [ENTER]. A continuación se presenta en su pantalla un menú general que muestra los 6 módulos principales de operación. Si la fecha de su microcomputadora es 01/Ene/80 deberá actualizarla, para ello se presenta una pantalla con campos de captura para que usted introduzca la fecha, la hora y una opción para que el sistema interprete la información. Si desea usted seleccionar la fecha por medio de un calendario, en el campo de captura destinado para la fecha teclee una 'C' y un [ENTER]; a continuación se visualiza un calendario indicando el mes y día en que se este trabajando, los cuales son seleccionables, y para modificarlos se deberán utilizar las teclas de [+] y [-] o la barra espaciadora. En la parte inferior del calendario se encuentran las palabras Modificar y Aceptar, que se pueden elegir, dependiendo de si se desea modificar la fecha que indica el calendario o si se acepta. Para abandonar el calendario acepte la información o teclee [ESC].

Envío de aplicaciones

Selección una opción :

PARAMETROS DE CONFIGURACION

NUEVA APLICACION
ACTIVACION
RESTAURACION
DEPURACION
COMUNICACIONES
SALIR DEL SISTEMA

Enter F1=Ayuda Esc=Cancelar

PARAMETROS DE COMUNICACION: En esta sección establece usted los datos necesarios para que se pueda realizar la comunicación.

NUEVA APLICACION: En este módulo se especifican al sistema los datos necesarios para el envío y activación en uno o varios nodos remotos.

ACTIVACION: En este módulo se lleva a cabo la activación de aplicaciones recibidas y que están listas para su puesta en marcha.

RESTAURACION: Este módulo permite restaurar aquellas aplicaciones que hayan sido respaldadas.

DEPURACION: Borrar aplicaciones, respaldadas o sin activar.

COMUNICACIONES: En esta sección usted realiza el envío y recepción de aplicaciones y/o información pendiente.

Posicione en la opción que usted desea y tecle [ENTER], o bien [ESC] si desea salir del sistema.

CAPITULO 1 - Parámetros de comunicación.

Parámetros de comunicación

Teclée la nueva información y presione Enter.

PUERTO (1),(2) :
 VELOCIDAD (3)99,(1)200,(2)400,(4)800,(9)600 :
 IDENTIFICACION :
 PASSWORD :
 COMANDO PARA OBTENER LINEA :

TELEFONOS :

ACEPTAR INFORMACION

Enter F1=Ayuda Esc=Cancelar

En esta sección se presenta una pantalla con campos de captura en donde deberá introducirse información necesaria para que se pueda realizar la comunicación:

PUERTO : Para llevar a cabo la comunicación entre el usuario y la Mainframe se debe indicar en que puerto de comunicaciones se encuentra instalado el modem. Como opciones para teclear se tienen los digitos 1 y 2. En general el 1 es para modem externo y el 2 es para modem interno. Cualquier otro dígito no será interpretado por el sistema.

VELOCIDAD : En este campo se puede configurar la velocidad de transmisión de los datos. Para la velocidad de 300 bauds se debe introducir el dígito 3, para 1200 bauds se debe introducir un 1 y así para las velocidades de 2400, 4800 y 9600 se deberá introducir el primer dígito que les corresponda.

IDENTIFICACION y PASSWORD : Para que usted pueda enviar o recibir de la Mainframe una aplicación es necesario que introduzca su identificación y password previamente asignados.

COMANDO PARA OBTENER LINEA : Para que se pueda transmitir la información por medio de un enlace 'DIAL UP' (este enlace se lleva a cabo por medio de una línea telefónica, y dura únicamente el tiempo que el usuario este activo en la sesión de Mainframe) se necesita introducir un comando para que el modem obtenga línea. Revise en el manual de su modem la sintáxis de este comando. En general para los modems Hayes y compatibles se recomiendan los siguientes comandos:

Para un teléfono por pulsos el comando es 'ATDPW' para línea directa y 'ATDPW9' para conmutador.

Para un teléfono por tonos el comando es 'ATDTW' para línea directa y 'ATDTW9' para conmutador.

En los comandos para el conmutador el último dígito varía dependiendo del conmutador precisamente.

Si el marcado lo quiere usted hacer en forma manual el comando es 'ATA'.

TELEFONOS : Teclee los teléfonos a los cuales usted puede comunicarse. Tiene la opción de introducir hasta seis diferentes números telefónicos. En el momento de la comunicación el sistema intentará realizar el enlace por cualquiera de estos teléfonos.

Para que los datos anteriores sean interpretados por el sistema usted deberá seleccionar la opción ACEPTAR INFORMACION y presionar la tecla [ENTER]. Si algún campo no tiene información (excepto los destinados para los números telefónicos, pues para marcar el teléfono en forma manual no se necesitan) el sistema le indicará mediante una pantalla de error que los datos están incompletos.

Parámetros de comunicación

Teclee la nueva información y presione Enter.

PUERTO	(1),(2) :	
VELOCIDAD	(3)00,(1)200,(2)400,(4)800,(3)600 :	
IDENTIFICACION	:	
PASSWORD	:	123456
COMANDO PARA OBTENER LINEA :	ATDPW	

TELEFONOS :

S4ESPEV		
Datos Incompletos		
Enter = Esc=Cancelar		

Enter F1=Ayuda Esc=Cancelar

Para posicionarse en cualquier campo de captura o en la opción ACEPTAR INFORMACION utilice la tecla de [ENTER] o bien las flechas de direccionamiento.

Para abandonar este módulo sin cambios teclee [ESC].

CAPITULO 2 - Nueva aplicación.

En esta sección puede usted seleccionar la aplicación que desea enviar. En primera instancia se presenta una pantalla con campos de captura en donde se debe introducir la siguiente información:

NOMBRE DE LA APLICACION: Corresponde al nombre con el cual usted desea que sea identificada la aplicación.

DESTINATARIOS: Son los usuarios a los que usted puede enviar la aplicación. Pueden ser grupos de usuarios.

RUTA PARA SELECCION DE ARCHIVOS: Indique la ruta o directorio en donde se encuentran los archivos que conforman a la aplicación que desea enviar.

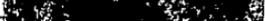
PRIORIDAD DE ENVIO DE APLICACION: Si usted desea enviar una aplicación con carácter de urgente teclee la letra 'U'. En caso contrario teclee la letra 'N' que corresponde a una prioridad de envío normal.

FECHA DE ACTIVACION: Fecha en que será activada la aplicación en el nodo remoto.

PROPORCIONE LOS DIRECTORIOS DESTINO: Introduzca los directorios en los cuales será actualizada la aplicación en la microcomputadora destino.

 Nueva aplicación

Teclee la nueva información y presione Enter.

Nombre de la aplicación	:	
Destinatarios	:	
Ruta para selección de archivos	:	
Prioridad de envío de la aplicación	:	
Fecha de activación	:	
Proporcione los directorios destino :		
1.-		
2.-		
3.-		
4.-		
5.-		
6.-		
7.-		
8.-		
9.-		
10.-		
		

ACEPTAR INFORMACION

Para que los datos anteriores sean interpretados por el sistema debe seleccionar la opción ACEPTAR INFORMACION y presionar la tecla [ENTER]. Si algún campo no tiene información, excepto algunos de los 20 que están asignados para los directorios destino, el sistema le indicará mediante una pantalla de error aquel que se encuentre incompleto. También, si alguno de los directorios destino tiene un formato inválido el sistema lo indicará con una pantalla de error.

Para posicionarse en cualquier campo de captura o en la opción ACEPTAR INFORMACION utilice la tecla de [ENTER] o bien las flechas de direccionamiento.

Para abandonar este módulo teclee [ESC].

Después de que ha sido aceptada la información, se presenta un menú que tiene como opciones de selección los directorios destino que usted proporcionó, y además las opciones VISUALIZAR ESTRUCTURA y ACEPTAR INFORMACION.

Nueva aplicación

Posicione la barra en el directorio y presione Enter.

```

┌───┐
│DBASE│
└───┘

```

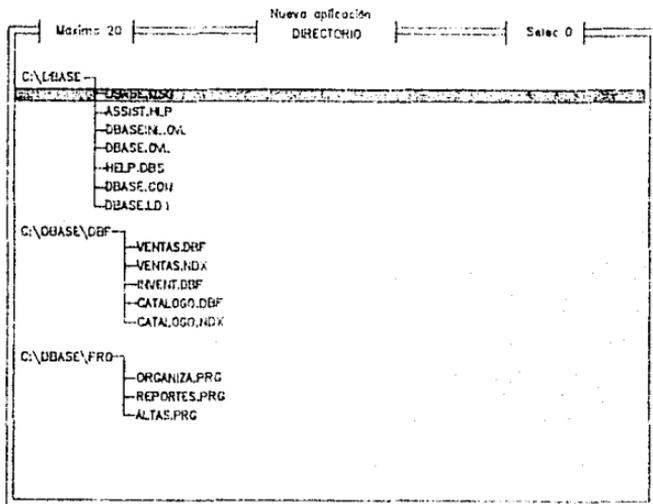
VISUALIZAR ESTRUCTURA

ACEPTAR INFORMACION

Enter F1=Ayuda Esc=Cancelar

Para seleccionar alguna de estas opciones utilice las flechas de direccionamiento, posicione la barra en la opción que desee y presione la tecla [ENTER].

Para cada directorio destino que es seleccionado se presenta una pantalla con una lista de los archivos que se encuentran en la unidad de disco que usted seleccionó anteriormente. Para seleccionar los archivos que serán actualizados en la microcomputadora remota posicione la barra de selección en el archivo que elija y presione [ENTER]. Cada uno de los archivos seleccionados quedarán en color inverso. Una vez que haya completado su selección de archivos presione la tecla [ESC].



En la opción VISUALIZAR ESTRUCTURA puede ver una imagen de como será actualizada la aplicación que envía en el o los nodos remotos.

Es importante que mencionemos que usted puede tener directorios destino sin haber seleccionado archivos para ellos, pues al ser activada la aplicación en la microcomputadora remota estos directorios destino "vacíos" serán simplemente creados.

Una vez que usted este de acuerdo con la estructura de la aplicación seleccione la opción ACEPTAR INFORMACIÓN.

A continuación se presenta una pantalla con el rubro PROPORCIONE EL COMANDO PARA INSTALACION y un campo de captura. Si usted requiere que al activarse la aplicación se ejecute algún archivo (*.BAT, *.EXE, *.COM) o algún comando del sistema operativo, introduzca el nombre del mismo y presione [ENTER]. En caso contrario nada más presione [ENTER]. A continuación verá usted una pantalla indicándole los archivos que son procesados para el envío de la aplicación.

CAPITULO 3 - Activación.

La activación de una aplicación se lleva a cabo de dos formas, manual y automáticamente. El sistema verifica cuando la fecha de activación de una aplicación es anterior o igual a la actual, y automáticamente la activa.

En esta sección se explica como puede usted llevar a cabo la activación de una aplicación en forma manual. Se presenta una pantalla con las opciones SELECCIONAR APLICACION, VISUALIZAR APLICACION, ACTIVAR.

Activación

Seleccione una opción:

SELECCIONAR APLICACION	<INDEFINIDA>	VISUALIZAR APLICACION
ACTIVAR		

Enter F1=Ayuda Esc=Cancelar

Para seleccionar alguna de estas opciones utilice las flechas de direccionamiento, posicione la barra en la opción que desea y presione la tecla [ENTER].

Al escoger la opción SELECCIONAR APLICACION se presenta una lista de las aplicaciones que pueden ser activadas. Utilizando las flechas de dirección y la tecla de [ENTER] elija la aplicación que desea activar. En caso de que no se tengan aplicaciones para ser activadas se presenta una pantalla de aviso.

Activación

Seleccione una opción:

SELECCIONAR APLICACION :	BASES DE DATOS	(00000001)
	DISPLAY WRITE 4	(00000003)
	ENABLE	(00000004)
	LOTUS 1-2-3	(00000002)
	INDICE ANTERIOR	

Enter Esc=Cancelar

La opción VISUALIZAR APLICACION permite ver una imagen de la aplicación que será activada. Si se eligió esta opción y no ha sido seleccionada ninguna aplicación el sistema dará un mensaje de error.

La siguiente figura nos muestra como se vería la imagen de una aplicación que será activada.

Activación

APLICACION SELECCIONADA	
C:\DBASE	<ul style="list-style-type: none"> -DBASE.MSG -ASSIST.RLP -DBASEK.OVL -DBASE.OVL -HELP.DBS -DBASE.COM -DBASE.LST
C:\DBASE\DEF	<ul style="list-style-type: none"> -VENTAS.DBF -VENTAS.INX -\$VENT.DBF -CATALOGO.DBF -CATALOGO.INX
C:\DBASE\PHG	<ul style="list-style-type: none"> -ORGANIZA.PHG -REPORTES.PHG -ALTAS.PHG

Si usted ya seleccionó una aplicación elija la opción ACTIVAR para que pueda utilizar la nueva versión de su aplicación. Si eligió ACTIVAR sin haber seleccionado una aplicación el sistema le indicará mediante una pantalla de error que seleccione primero la aplicación.

Para abandonar este módulo presione la tecla [ESC].

Cuando se activa una aplicación cada uno de los archivos que la conforman son respaldados si existen previamente.

CAPITULO 4 - Restauración.

En esta sección se explica como puede usted restaurar una aplicación en caso de que desee regresar a una versión previa. Sólo los archivos que fueron respaldados son los que se restaurarán.

Se presenta una pantalla con las opciones SELECCIONAR APLICACION, VISUALIZAR APLICACION, RESTAURAR.

Restauración

Seleccione una opción:

SELECCIONAR APLICACION	<INDEFINIDA >	VISUALIZAR APLICACION
RESTAURAR		

Enter F1=Ayuda Esc=Cancelar

Para seleccionar alguna de estas opciones utilice las flechas de direccionamiento, posicione la barra en la opción que desee y presione la tecla [ENTER].

Al escoger la opción SELECCIONAR APLICACION se presenta una lista de las aplicaciones que fueron respaldadas y que pueden restaurarse. Utilizando las flechas de dirección y la tecla de [ENTER] elija la aplicación que desee restaurar. En caso de que no se tengan aplicaciones para restaurar se presenta una pantalla de aviso.

Restauración

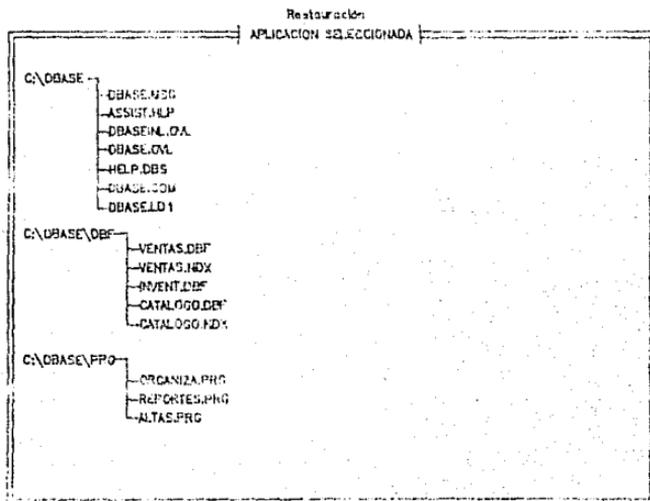
Seleccione una opción:

SELECCIONAR APLICACION	BASE II (25000000)
	INDICE ANTERIOR
	RESTAURAR

Enter Esc=Cancelar

La opción VISUALIZAR APLICACION permite ver una imagen de como se encuentra la aplicación actualmente. Si se eligió esta opción y no ha sido seleccionada ninguna aplicación el sistema dará un mensaje de error.

La siguiente figura nos muestra como se vería la imagen de una aplicación que será restaurada.



Si usted ya seleccionó una aplicación elija la opción RESTAURAR para que los respaldos obtenidos de la aplicación anterior sean nuevamente actualizados. Si eligió RESTAURAR sin haber seleccionado una aplicación el sistema le indicará mediante una pantalla de error que seleccione primero la aplicación.

Para abandonar este módulo presione la tecla [ESC].

CAPITULO 5 - Depuración.

En esta sección se explica como puede usted depurar una aplicación.

Se presenta una pantalla con las opciones: SELECCIONAR APLICACION, VISUALIZAR APLICACION, DEPURAR.

 Depuración

Seleccione una opción:

SELECCIONAR APLICACION: <INDEFINIDA>	VISUALIZAR APLICACION:
DEPURAR	

Enter F1=Ayuda Esc=Cancelar

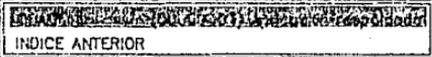
Para seleccionar alguna de estas opciones utilice las flechas de direccionamiento, posicione la barra en la opción que desee y presione la tecla [ENTER].

Al escoger la opción SELECCIONAR APLICACION se presenta una lista de las aplicaciones que no han sido activadas y aquellas que al activarse fueron respaldadas.

Utilizando las flechas de dirección y la tecla de [ENTER] elija la aplicación que desee depurar. En caso de que no se tengan aplicaciones para depurar se presenta una pantalla de aviso.

Depuración

Seleccione una opción:

	
DEPURAR	

Enter Esc=Cancelar

La opción VISUALIZAR APLICACION permite ver una imagen de como se encuentra la aplicación actualmente. Si se eligió esta opción y no ha sido seleccionada ninguna aplicación el sistema dará un mensaje de error.

Si usted ya seleccionó una aplicación elija la opción DEPURAR. Si la aplicación seleccionada es una que fue activada y se obtuvo respaldo, estos serán borrados y si la aplicación no ha sido activada se borrarán los archivos que la conforman.

Si eligió DEPURAR sin haber seleccionado una aplicación el sistema le indicará mediante una pantalla de error que seleccione primero la aplicación.

Para abandonar este módulo presione la tecla [ESC].

CAPITULO 6 - Comunicaciones.

En este módulo usted envía y/o recibe aplicaciones. El proceso de comunicaciones consta de 5 fases que se ejecutan automáticamente, y cada una de ellas emite mensajes al usuario indicándole la etapa que se esta llevando a cabo.

A continuación se hace una descripción de cada una de estas fases.

Fase 1 - Parámetros de comunicación.

En esta fase se configura al equipo con los parámetros de comunicación que el usuario proporcionó al sistema, de tal forma que son utilizados para establecer el enlace con el computador central. Así mismo se realiza la organización y segmentación de los archivos que se desean enviar como parte de una aplicación.

Fase 2 - Conexión con sistema remoto.

Se inicia el proceso de enlace por medio de la línea telefónica hacia el computador central. Se realiza la inicialización del modem, el marcado del teléfono y la entrada al sistema (Complete) en el computador central.

Fase 3 - Envío de archivos.

En caso de que exista una aplicación por enviar, se realiza su lectura y envío hacia el computador central. En caso contrario esta fase cede el control a la siguiente.

Fase 4 - Recepción de archivos.

Se realiza la recepción de los archivos que conforman una aplicación y se prepara una lista con el nombre de cada uno de ellos. Así mismo, se organiza y rutea estos archivos para dejarlos disponibles al sistema.

Fase 5 - Finalizando sesión.

Terminación del enlace con el computador central. Se finaliza la sesión de usuario y se cuelga el teléfono.

En esta parte del sistema el usuario no participa activamente con la microcomputadora. Solo podrá teclear [ESC] en caso de un error durante la comunicación con la Mainframe.

Para abandonar el sistema ubíquese en el menú principal de actividades. Posicione en la opción SALIR DEL SISTEMA y tecla [ENTER] o bien presione [ESC]. En seguida se presentará una pantalla de confirmación con las opciones SI y NO. Para salir posicione en la opción SI y presione [ENTER].

Envío de aplicaciones

Seleccione una opción :

```
PARAMETROS DE COMUNICACION
NUEVA APLICACION
AC
RE  SALIR DEL SISTEMA
DE  SI
CO  NO
SA
```

Enter F1=Ayuda Esc=Cancelar

A.6 BIBLIOGRAFIA

- ADVANCED MSDOS
Ray Duncan
MICROSOFT PRESS
- THE MS-DOS ENCYCLOPEDIA
Ray Duncan
MICROSOFT PRESS
- THE PETER NORTON PROGRAMMER'S GUIDE TO THE IBM PC
Peter Norton
MICROSOFT PRESS
- THE MS-DOS HANDBOOK
Richard Allen King
SYBEX
- MULTI MODEM 224 Y 224E. OWNER'S MANUAL
MULTI-TECH SYSTEMS, INC.
- DATA COMMUNICATIONS
John Macnamara
ADDISON WESLEY
- PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS
William M. Newman, Robert F. Sproull
MCGRAW-HILL
- AN INTRODUCTION TO MVS
IBM, Information Development.
- COMMUNICATIONS SYSTEMS CONCEPTS
Study Guide
IBM, Direct-Education.
- C PROGRAMMER'S GUIDE TO SERIAL COMMUNICATIONS
Joe Campbell
HOWARD W. SAMS & COMPANY
- DOS PROGRAMMER'S REFERENCE
Dettman
QUE
- MICRO-MAINFRAME CONNECTION
Thomas Wm. Madron
HAYDEN BOOKS, 1987
- KERMIT A FILE TRANSFER PROTOCOL
Frank Da Cruz
DIGITAL PRESS

- INTRODUCCION A LOS SISTEMAS DE BASES DE DATOS
Date
ADDISON-WESLEY IBEROAMERICANA

- DISEÑO DE BASES DE DATOS
Widerhold
MCGRAW-HILL