



105
Zejeu
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO E IMPLANTACIÓN DEL
SISTEMA DE MONITOREO DE
LA RED DE TELEFONÍA DIGITAL DE LA UNAM

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO-ELECTRICISTA

PRESENTAN:

CARLOS JIMÉNEZ CASTILLO
MIGUEL LÓPEZ GUERRERO
HÉCTOR ROSAS FLORES
JESÚS MARIO VALVERDE ROSAS



DIRECTOR DE TESIS:
ING. RICARDO MARTINEZGARZA FERNÁNDEZ

MÉXICO, D.F.

ENERO, 1995

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



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 esta 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.

Dedicatoria

*A mi Mamá, la Sra. Josefina H. Castillo Viosca;
a mi hermano, Gerardo Jiménez Castillo;
a la memoria de mi abuela, la Sra. Ana María de Jesús Viosca Juárez;
a mis tíos Roberto, Carlota, Miguel, Javier, Irene y a toda mi familia;
a Magdalena;
a mis amigos: Francisco Javier, Javier, Gabriel, José, Rafael y Joaquín.*

Carlos Jiménez Castillo

*A mis padres:
Víctor López Zúñiga y Teresa Guerrero Basurto*

Miguel López G.

*A mis padres: Demetrio Rosas Flores y Ma. Martha Flores Vázquez,
a mis hermanos: Martín, Carlos, Juan, Gloria, Lucrecia, Xochitl y Daniel,
a mis abuelos maternos y paternos, así como a cada uno de mis tíos y primos,
a mis amigos Rafael, Javier, José y Joaquín.*

Héctor Rosas Flores.

*A mis padres: Mario Valverde Salvador y Maricela Rosas Ponce,
a mis hermanos: Salvador y Juan Carlos,
a mis familiares y amigos.*

J. Mario Valverde R.

Agradecimientos

A mi Mamá, la Sra. Josefina H. Castillo Viosca y a mi hermano, Gerardo Jiménez Castillo; por todo el amor, apoyo y comprensión que me han brindado en cualquiera de mis empresas. Que este trabajo sea constancia de la profunda gratitud y gran amor que les profeso.

A toda mi familia, porque nunca han perdido la fe en mí y porque sé que siempre podré contar con su apoyo. En especial, a la Sra. Ana María de Jesús Viosca, que con su fuerza y decisión fue y sigue siendo el pedestal de nuestra familia y que nos enseñó a mantenernos unidos bajo cualquier circunstancia; enseñanza que quedará en todos nosotros por el resto de nuestras vidas.

Al Ingeniero Ricardo Martínezgarza Fernández su apoyo, guía y enseñanzas para la culminación de este proyecto.

A la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, porque gracias a sus instalaciones y sobre todo, a sus profesores, pude formar parte de este proyecto que, espero pueda retribuirle de alguna manera los recursos dedicados a mi formación profesional. Este trabajo surgió en la UNAM, por la UNAM y para la UNAM.

Carlos Jiménez Castillo

Agradecimientos

Profunda y sinceramente quiero agradecer a Alicia Mireya Prieto Carreras, Sergio Méndez Rodríguez, Daniel Torres Pelayo, Alberto García Ordoñez, Abraham Pardo Pimentel, Jesús Eloy Nolasco Martínez, Yukihiro Minami Koyama, Félix Núñez Orozco, Craige Mcomb Snaider González, Rolando Peralta Pérez, Juan Carlos Roa Beiza, Víctor Pérez Amador Barrón, Mario Ibarra Pereyra y Rodolfo Lorenzo Bautista, de quienes recibí una formación insuperable.

También a la Dra. Elena López Ruiz que me mostró la importancia de saber valorar la formación recibida en mi *Alma Mater* y de quien he tratado de imitar la actitud de dedicación, esmero y entusiasmo en el trabajo.

Al Ing. Ricardo Martínezgarza Fernández por haber aceptado dirigir esta tesis y el enorme apoyo brindado.

Por último pero no por ello ser menos importante, a Carlos Jiménez Castillo que "me soportó" por más de dos años durante la realización de este trabajo.

A todos ellos expreso mi gratitud y mi reconocimiento.

Miguel López Guerrero, enero 1995

Agradecimientos

A mis padres:

Por su gran apoyo y ayuda para realizar mis estudios, sin ellos no habría sido posible.

A los profesores de la Facultad de Ingeniería:

Porque son la guía fundamental en la preparación, y porque son quienes proporcionan las bases para seguir esta carrera.

A Dios.

En especial al Ingeniero Ricardo Martinezgarza:

Por toda la ayuda brindada y su interés en este trabajo para que saliera adelante.

Mario Valverde

Agradecimientos:

A mis padres:

Doy gracias por sus consejos, apoyo y confianza en el transcurso de mis estudios.

A la Facultad de Ingeniería:

Por su excelente preparación que me dio y ha permitido representarla orgullosamente en el campo de trabajo.

Al Ing. Ricardo Martinezgarza F:

Por darme la oportunidad de colaborar en su equipo de trabajo y por su apoyo en este trabajo de tesis.

A Dios:

Por darme la fortaleza necesaria en todos mis estudios.

Héctor Rosas Flores

Diseño e Implantación del Sistema de Monitoreo de la Red de Telefonía Digital de la UNAM

Introducción	i
Capítulo I, Análisis del problema	
1.1 Descripción de la Red de Telefonía Digital de la UNAM	2
1.2 Planteamiento del problema	7
Capítulo II, Búsqueda de soluciones	
II.1 Requerimientos de la solución	10
II.2 Soluciones propuestas	12
II.2.1 Sistema de alarmas luminosas	12
II.2.2 Sistema de monitoreo con enlaces de radiofrecuencia	14
II.2.3 Sistema de monitoreo con enlaces a través de modem	15
Capítulo III, Diseño e implantación de la solución seleccionada	
III.1 Consideraciones generales de diseño	18
III.2 La elección del lenguaje de desarrollo	21
III.3 Diseño de los sensores del Sistema	24
III.3.1 Sensor de temperatura	24
III.3.2 Sensor de voltaje en las baterías de respaldo	25
III.3.3 Sensores de alarmas mayor y menor	29
III.3.4 Sensor de falla en el suministro de energía eléctrica de AC	32
III.4 Adquisición de los datos	36
III.5 Comunicaciones	44
III.5.1 Arquitectura interna del controlador serial	45
III.5.2 Interfaz con el controlador de comunicaciones seriales	49
III.5.3 Rutinas básicas del comunicación serial	55
III.5.4 Diseño del protocolo de comunicaciones	61
III.5.4.1 Estructura del paquete de información	61
III.5.4.2 Secuencia de transmisión-recepción	63
III.6 Visualización de la información	65
III.6.1 Software para la visualización del estado de la Red	65
III.7 Almacenamiento de la información en la computadora central	72
III.7.1 El contenedor de datos y el archivo de su tamaño	73
III.7.2 El directorio	75
III.7.3 Los archivos de lecturas	77
III.7.4 Archivo de número de registros	79
III.7.5 Archivo de nodos	80
III.7.6 El proceso de almacenamiento de la información	81
III.8 Consultas a la información almacenada	88

III.9	Exportación de la información	92
III.9.1	La estructura de un archivo dBASE	93
III.9.2	El proceso de exportación de la información	94
III.10	Inicialización	102
III.10.1	Verificación del hardware de la PC	102
III.10.2	Creación de las estructuras de administración de los datos	103
III.10.3	Configuración de los elementos de comunicaciones	103
Capítulo IV,	Manual del usuario	
IV.1	Requerimientos del sistema	106
IV.1.1	Características del modem utilizado por el sistema	107
IV.2	Instalación del Sistema de Monitoreo	109
IV.2.1	Instalación del nodo central	109
IV.2.2	Instalación de un nodo remoto	110
IV.3	Ejecución del sistema desde la línea de comandos del MS-DOS	112
IV.4	Visualización del estado de la red	115
IV.4.1	Despliegue en pantalla	116
IV.4.2	Registro y liberación de alarmas	119
IV.5	Interfaz con el usuario en los módulos del sistema	121
IV.6	Módulo de mantenimiento	123
IV.6.1	Altas	125
IV.6.1.1	Altas de nodos ya existentes	125
IV.6.1.2	Creación de un nuevo nodo	127
IV.6.2	Bajas	128
IV.6.3	Cambios	129
IV.6.4	Respaldos	132
IV.6.5	Prueba de los puertos seriales	137
IV.6.6	Cambio del tamaño del área de datos	140
IV.7	Módulo de consultas	142
IV.8	Otras opciones	152
Conclusiones y comentarios		153
Apéndices		
A	Archivos de configuración de modem	157
B	Compilando el proyecto	164
Bibliografía		165

Introducción

Como una respuesta a la apremiante necesidad de la Universidad Nacional Autónoma de México (UNAM) de mejorar la calidad de las comunicaciones que se realizan en ella, se decidió llevar a cabo un proyecto para renovar y expandir el sistema telefónico, empleando los avances que han tenido lugar en materia de telefonía digital. Como resultado se creó un sistema que proporciona el servicio a todas las dependencias de Ciudad Universitaria, las ENEP, FES y casi todos los centros en el interior de la República.

La Dirección General de Servicios de Cómputo Académico (DGSCA) hizo un estudio detallado de las necesidades de la Institución y de las tecnologías existentes en el mercado de las comunicaciones para sustituir el sistema telefónico con equipos de tecnología moderna, que cubrieran la demanda actual y se ajustaran al crecimiento de los próximos años. Como consecuencia nace el Sistema de Telefonía Digital de la UNAM. Algunas de sus características son: marcación abreviada entre los principales *campus* de la Universidad, entrada directa en todos los servicios (del exterior sin pasar por operadora), enlaces redundantes y amplios dispositivos de protección.

Esas características no indican que el equipo quede exento de fallas, las cuales se deben detectar y resolver en el menor tiempo posible, ya que el tipo de servicio que se proporciona así lo requiere. Es por estas razones que surge el presente trabajo de tesis, el cual describe el diseño e implantación de un sistema que permite el monitoreo y análisis del estado de la Red de Telefonía Digital de la UNAM.

Capítulo I

Análisis del problema

I.1 Descripción de la Red de Telefonía Digital de la UNAM

La malla telefónica está conformada por varios equipos de conmutación principales que permiten la intercomunicación entre todas las dependencias; éstos son los que llevan la mayor carga de tráfico y que se identifican como nodos principales (NP), los equipos de conmutación secundarios, llamados subnodos (S), forman una estrella hacia el núcleo de equipos principales y manejan solamente el tráfico de las dependencias que alimentan. Esta configuración permite tener un mejor control y mantenimiento de los equipos de conmutación, asimismo ofrece vías alternas de comunicación entre las extensiones telefónicas.

Para determinar los nodos que formarían la red digital en la Ciudad Universitaria, se consideró básicamente la densidad de población existente y las distancias entre las dependencias para evitar en lo posible cableados de cobre considerablemente largos.

Tomando en consideración la configuración así definida, se interconectaron cinco nodos principales, por los cuales se plantearon diferentes rutas de acceso a las dependencias; los nodos restantes o subnodos, se enlazaron a alguno de los cinco nodos principales. La distribución actual de los conmutadores se muestra en la figura I.1.

Para algunas dependencias en la zona metropolitana y fuera de Ciudad Universitaria, fue necesario instalar conmutadores en ellas. Estas fueron: las tres ENEP: Acatlán, Aragón e Iztacala; las dos FES: Cuautitlán y Zaragoza y por último, la Coordinación de Normatividad y Sistemas Administrativos (Pitágoras). Para la FES Cuautitlán se instalaron conmutadores en dos de sus *campus*. Con estas dependencias, se instalaron un total de 27 equipos de conmutación que actualmente dan servicio a la Universidad.

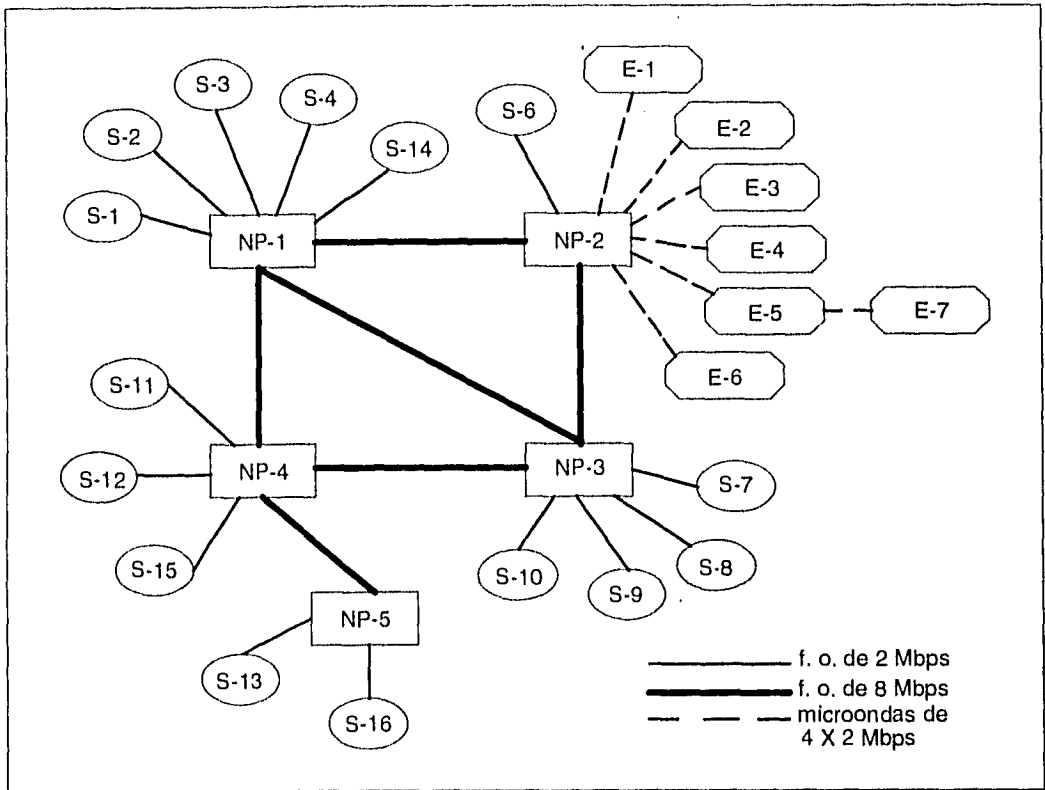


Figura I.1.1 Nodos de la Red de Telefonía Digital de la UNAM

Dentro de la red existen diferentes tipos de enlaces para comunicar sus nodos. En Ciudad Universitaria todos los conmutadores se interconectaron con fibras ópticas ya que ofrecen un medio de comunicación confiable y permiten utilizar un ancho de banda muy amplio. Entre los nodos principales se tiene una velocidad de transmisión de 8.448 Mbps, mientras que para los nodos satélite se cuenta con una de 2.048 Mbps.

Cada una de las extensiones telefónicas, tanto en Ciudad Universitaria como en las unidades multidisciplinarias en el área metropolitana, se conectaron a través de cable de cobre a sus respectivos conmutadores.

Los enlaces con algunas dependencias en el área metropolitana se realizaron a través de microondas, ya que la utilización de la red digital pública implicaba un alto costo de inversión y era limitante debido a que la Universidad tenía que crecer al ritmo de ella. Estos enlaces se realizaron en lugares donde la distancia no fue mayor de 25 km y en donde las características de línea de vista y radio de Fresnel no fueron afectadas. Tomando en cuenta estas condiciones, se unieron a través de microondas la FES Zaragoza, la ENEP Aragón y Pitágoras. También se instaló un enlace de microondas entre los *campus* 1 y 4 de la FES Cuautitlán. Cada uno de los enlaces tiene cuatro canales de 2.048 Mbps y soportan los canales de voz requeridos para enlazar a las dependencias.

Otras dependencias del área metropolitana por no tener línea de vista, no fue posible unirlos por medio de microondas, conectándose originalmente a través de enlaces privados con características PCM de 2.048 Mbps; que posteriormente se cambiaron por enlaces de RDI. Estas dependencias fueron las ENEP Acatlán e Iztacala y la FES Cuautitlán. Los enlaces descritos anteriormente se muestran en la figura I.1.2..

Fue así como quedó constituida la Red de Telefonía Digital de la UNAM cuyas principales características son:

- Transmite voz y datos indistinta y simultáneamente, mediante codificación PCM.
- Las principales instalaciones de la Universidad están integradas a ella. Esto significa que a nivel licenciatura, posgrado e investigación, alrededor del 90% de sus miembros se encuentran en instalaciones cubiertas por la red, independientemente de su ubicación geográfica, incluyendo los planteles en el D.F. y área metropolitana, así como los centros que se encuentran en algunos estados de la República.
- Debido a las experiencias que se tuvieron con la red analógica instalada anteriormente, se decidió emplear un sistema altamente distribuido lo cual determinó la instalación de varios equipos de conmutación ubicados estratégicamente en los diferentes *campus* de la Universidad, formando una topología malla-estrella.

- Es redundante y está integrada por 27 nodos de telecomunicaciones enlazados entre sí, a través de fibra óptica y microondas y la malla de conmutación que conforman es no bloqueable y con enlaces virtuales.
- Tiene una capacidad instalada para 13,000 servicios telefónicos, y cuenta con 2,400 troncales digitales en total (entrada y salida) que se conectaron en los cinco nodos principales.

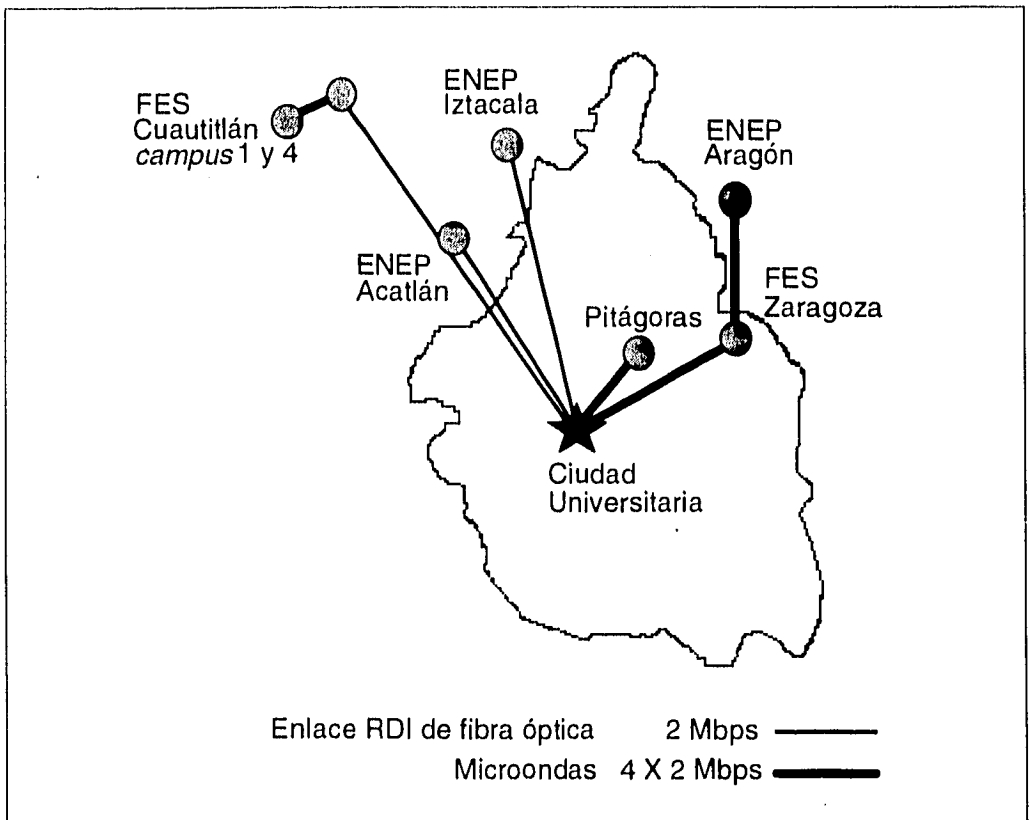


Figura I.1.2 Enlaces de microondas y vía TELMEX

A su vez, las principales características técnicas de los conmutadores son:

- Totalmente programables con diagnóstico, monitoreo y evaluación remota.
- Disponibilidad de enlaces digitales que permiten utilizar la línea telefónica para transmitir datos a velocidades hasta de 64,000 bps.
- Permiten marcación directa (sin pasar por operadora) al llamar del exterior.
- Alimentación de dos hilos a los aparatos telefónicos.

I.2 Planteamiento del problema

Cuando se detectan anomalías en algún equipo de conmutación, generalmente es necesario que el personal se traslade al lugar en el que éstas se presentaron, sin embargo, en algunos casos la falla es de programación, pudiéndose corregir vía modem ya que los conmutadores de la red permiten su operación local o remota por medio del programa propietario de estos equipos. Este programa de monitoreo y configuración consta de varios módulos y algunos parámetros que permite observar son los siguientes: estado de los enlaces y de las líneas, tráfico y alarmas. Asimismo permite dar de alta nuevas extensiones y programar servicios especiales. Esto permite que la red pueda ser monitoreada desde cualquier punto de la Universidad o externo a ella, y en caso de presentarse alguna falla, se puede detectar inmediatamente.

También se pueden generar reportes impresos, en los cuales el formato de salida no es fácil de entender, ya que están codificados y sólo el personal con capacitación especial los puede interpretar. Para personas sin este tipo de entrenamiento, no tienen significado alguno, lo cual es un problema ya que si ocurre alguna falla y en ese momento no se encuentra el personal especializado, no será posible darse cuenta de lo que ocurre y menos corregirla.

Este programa se encuentra instalado en una PC. Por medio de su puerto serial se comunica con el conmutador, obteniendo así los datos provenientes de él, los decodifica y permite visualizarlos en la pantalla. Cada conmutador cuenta con dos puertos seriales, uno que se conecta a la PC y el otro a un modem a través del cual se realiza la operación remota; si cualquiera de ellos se dañara, no se podrían acceder los datos generados por los equipos y no sería posible conocer las condiciones de los mismos.

Una limitante que presenta el programa, es que no es capaz de monitorear condiciones externas al equipo que también afectan su correcto funcionamiento.

La interrupción del suministro de energía eléctrica es un ejemplo de esto. En condiciones normales de operación el sistema se alimenta con la energía que proporciona la Compañía de Luz. Después de una interrupción en el servicio, los equipos continuarán funcionando con baterías de respaldo; sin embargo, si después de ocho horas no se restablece el suministro de energía, el voltaje de las baterías será inferior al requerido, dejando al conmutador fuera de funcionamiento. Bajo estas condiciones, cuando se presenta una falla de este tipo, es importante conocer el momento exacto en que ocurrió y el nivel de voltaje de las baterías en cada instante. Esta información no puede ser proporcionada por el programa propietario de monitoreo.

La temperatura también es un factor muy importante para la operación del equipo, la cual debe estar en un rango aproximado de 20 a 24 °C, de no ser así, los equipos sufren alteraciones en su funcionamiento. Este factor tampoco es monitoreado por ese programa.

Para subsanar estas carencias detectadas en el programa de monitoreo, era necesario un sistema dedicado exclusivamente a monitorear las condiciones externas a los equipos y que además reflejara las posibles causas de las alarmas generadas por los conmutadores.

Capítulo II

Búsqueda de soluciones

II.1 Requerimientos de la solución

El servicio telefónico debe ser proporcionado en forma ininterrumpida, dentro de lo posible. Por ello, entre otras cosas, se necesitaba un sistema que pudiera detectar las fallas y condiciones en cada local de los conmutadores antes de que ocurriera un problema significativo que dejara sin servicio a los usuarios. Dado lo anterior:

- Los datos debían obtenerse oportunamente y mostrar las condiciones bajo las cuales se encontraba operando el equipo.

Era necesario que el sistema pudiera proporcionar información de las condiciones de cada conmutador, de manera que un dispositivo ubicado en el lugar adecuado, desplegara la misma sin tener que desplazarse a cada local, además de que no debía requerir de operadores de los equipos instalados en los lugares remotos. Por ésto, otra de las características que debía tener es:

- Que fuera un sistema de monitoreo centralizado.

Una función importante era mantener un registro de los eventos que hubieran tenido lugar antes de cualquier falla para determinar qué ocurrió. De acuerdo a esto:

- Era necesario registrar los datos obtenidos, almacenándolos para su posterior consulta.

Se consideró que el sistema no debería requerir de una extensa capacitación:

- Su utilización e instalación debían ser sencillas, con una curva de aprendizaje pequeña.

Se pensó en un sistema que no necesitara de grandes recursos económicos para su instalación, que aprovechara la infraestructura ya existente y que funcionara sin problemas las veinticuatro horas del día:

- Debía ser económico y confiable.

Si en algún momento ocurría una falla en los sensores que formarían parte del sistema, era deseable que pudieran repararse o reemplazarse fácilmente:

- El hardware debía ser de mantenimiento sencillo.

Se contempló que pudieran existir cambios futuros, tanto en el plan de numeración como en el crecimiento de la red:

- Necesitaba ser flexible, los parámetros más importantes del sistema deberían poderse modificar y adaptarse a los cambios.

Además se previó que cualquiera de los equipos requeridos por el sistema podría funcionar de manera errónea debido a algún desperfecto, por lo que:

- Debía ser tolerante a fallas.

Tomando en cuenta las características anteriores, se plantearon algunas soluciones que se mencionan en el siguiente tema, para finalmente considerar la más adecuada, y que satisficiera los requerimientos aquí expuestos.

II.2 Soluciones propuestas

Para poder proporcionar un soporte adecuado al programa propietario de monitoreo, se requería de un sistema que detectara el estado de las condiciones generales de los locales de conmutación, algunas características de los equipos y que además cumpliera con los requerimientos descritos en el punto anterior. Para resolver el problema de cómo sería obtenida, almacenada, transmitida e interpretada la información de interés, se plantearon las siguientes tres posibles soluciones.

II.2.1 Sistema de alarmas luminosas

Cada equipo de conmutación cuenta con señales que indican condiciones anormales de su propia operación y entorno, las cuales se concentran en el conmutador de DGSCA y son accesibles a través de uno de sus puertos serie. Esta primera opción de solución planteó la decodificación de dos señales conocidas como alarmas mayor y menor, con un programa desarrollado por nosotros y que estaría instalado en una PC conectada al conmutador.

El funcionamiento de este sistema se basaría en detectar el estado de dichas señales, se mostraría en la pantalla de la PC y finalmente, por medio de un sistema electrónico de potencia conectado al puerto paralelo, se encenderían en un panel de focos correspondientes a cada alarma.

El panel se encontraría en un lugar visible de la Dirección de Telecomunicaciones Digitales (DTD) y así, cualquier persona, aunque no conociera detalladamente el sistema telefónico, podría percatarse de la situación en caso de presentarse el encendido de algún foco.

Esta propuesta se desechó, debido a que una alarma mayor o menor no permite conocer el estado de otros parámetros externos al conmutador, como ya se han citado: temperatura, alimentación de energía de AC y nivel de voltaje en baterías de respaldo, de los cuales se ha dicho, es primordial que se mantengan en determinados niveles para el correcto funcionamiento de los equipos .

Además los conmutadores, para almacenar la información de sus señales, tienen un búfer de capacidad limitada. Una vez lleno, no permite el registro de nuevas alarmas que se generen y hasta que no se vacíe la información del mismo, los datos más recientes no se registrarán. Esto representaba un grave inconveniente a esta propuesta.

En la siguiente figura se muestra esta solución de manera esquemática.

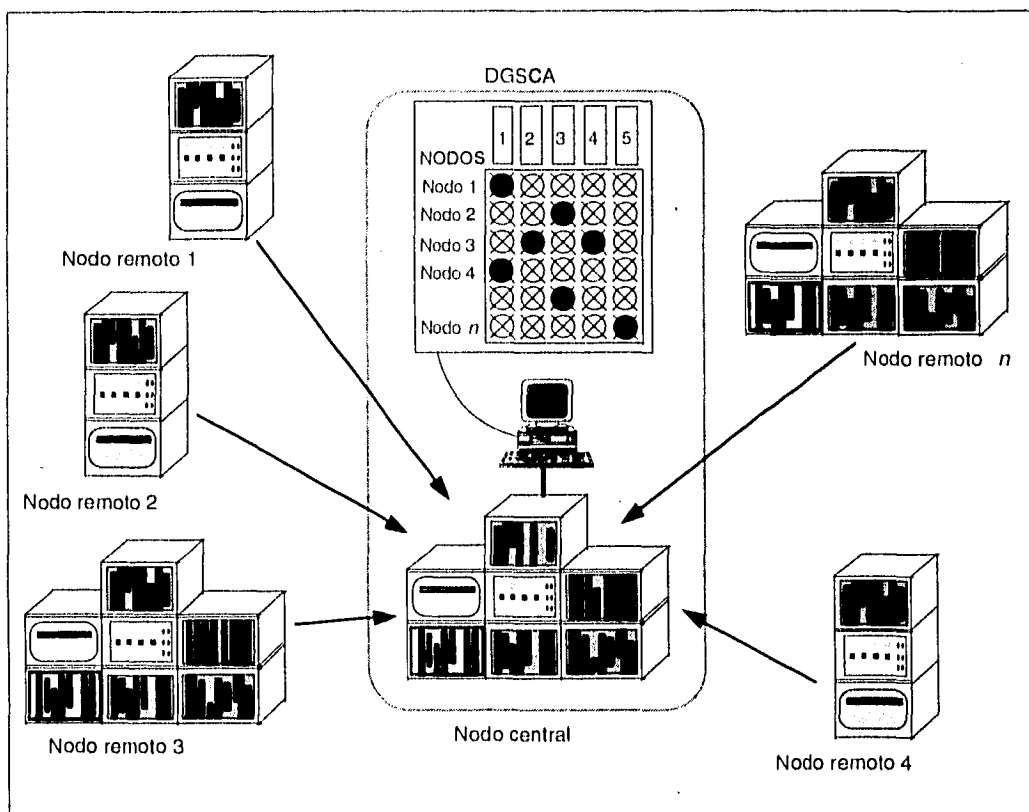


Figura II.2.1 Primera propuesta de solución.- Sistema de alarmas luminosas

II.2.2 Sistema de monitoreo con enlaces de radiofrecuencia

Esta solución consistía en diseñar sensores que detectaran el nivel de voltaje de las baterías de respaldo, suministro de energía eléctrica de AC, temperatura del local y las alarmas mayor y menor generadas por cada conmutador; los datos obtenidos se enviarían a un equipo de radiofrecuencia, y éste a su vez, los transmitiría hacia una antena localizada en DGSCA. El equipo de recepción localizado en esta dependencia, junto con una PC se considerarían como un nodo central, ya que hacia este punto, se enviaría la información de los lugares remotos. Finalmente, los datos recibidos se desplegarían en el monitor de la PC.

Esta alternativa no fue posible realizarla por las siguientes razones:

- El costo del proyecto sería muy elevado por la necesidad de contar en todos los nodos con una antena de radiofrecuencia y su respectivo equipo de comunicaciones.
- Las probabilidades de error en el sistema se acrecentarían, debido a que al utilizar equipo adicional se aumenta el número de puntos de falla.
- El mantenimiento preventivo y correctivo de los equipos de radiofrecuencia es elevado.

Como se observa, los principales problemas de esta propuesta eran el costo de implantación y de mantenimiento; lo cual se oponía al requerimiento de que fuera económico; además, debido a la naturaleza del equipo de comunicaciones, dejaba de ser sencillo. En la siguiente figura, se presenta de manera esquemática el sistema que se pretendía realizar.

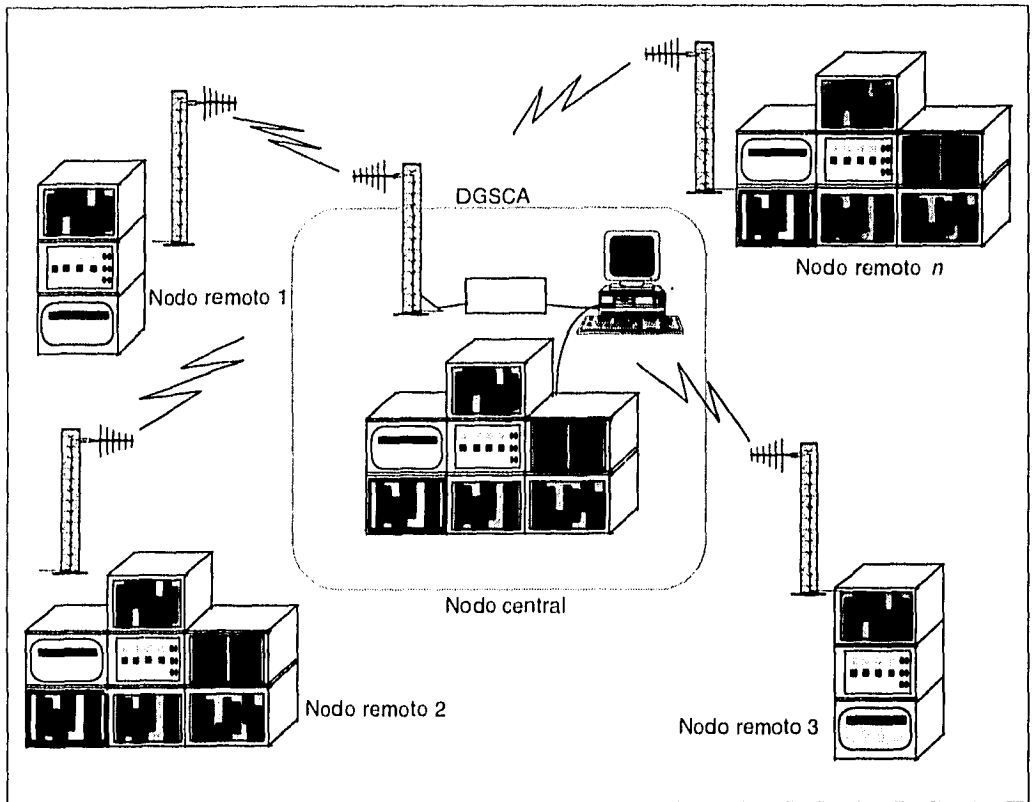


Figura II.2.2 Segunda propuesta de solución.- Sistema con enlaces de radiofrecuencia

II.2.3 Sistema de monitoreo con enlaces a través de modem

Con esta alternativa se propuso monitorear las condiciones de los locales de los conmutadores desde una computadora central localizada en DGSCA. En cada local se encontraría instalado un dispositivo de adquisición de datos, formado por una computadora remota que los leería por su puerto paralelo, obtenidos de un sistema de sensores. La computadora central, se encargaría de realizar una serie de llamadas a través de un modem a cada una de las computadoras remotas. Una vez establecido el enlace, la computadora remota enviaría su información, desplegándose inmediatamente en el nodo central y almacenándose para su posterior consulta.

El uso de una PC en el nodo que concentrara toda la información obtenida por el sistema de monitoreo, ofrece una gran versatilidad para el manejo e interpretación de estos datos, pudiendo su funcionamiento ser tan amigable y práctico como se programara.

El sistema propuesto resultaba además económico: podría utilizar en cada nodo remoto una computadora PC-XT de reuso, incluso sin monitor o disco duro; los elementos que lo componen podrían repararse o sustituirse fácilmente, ya que son equipos y dispositivos comerciales; y además, sería flexible y se podría diseñar con pocas dificultades para su aprendizaje y operación.

En la figura se muestra de manera esquemática esta solución:

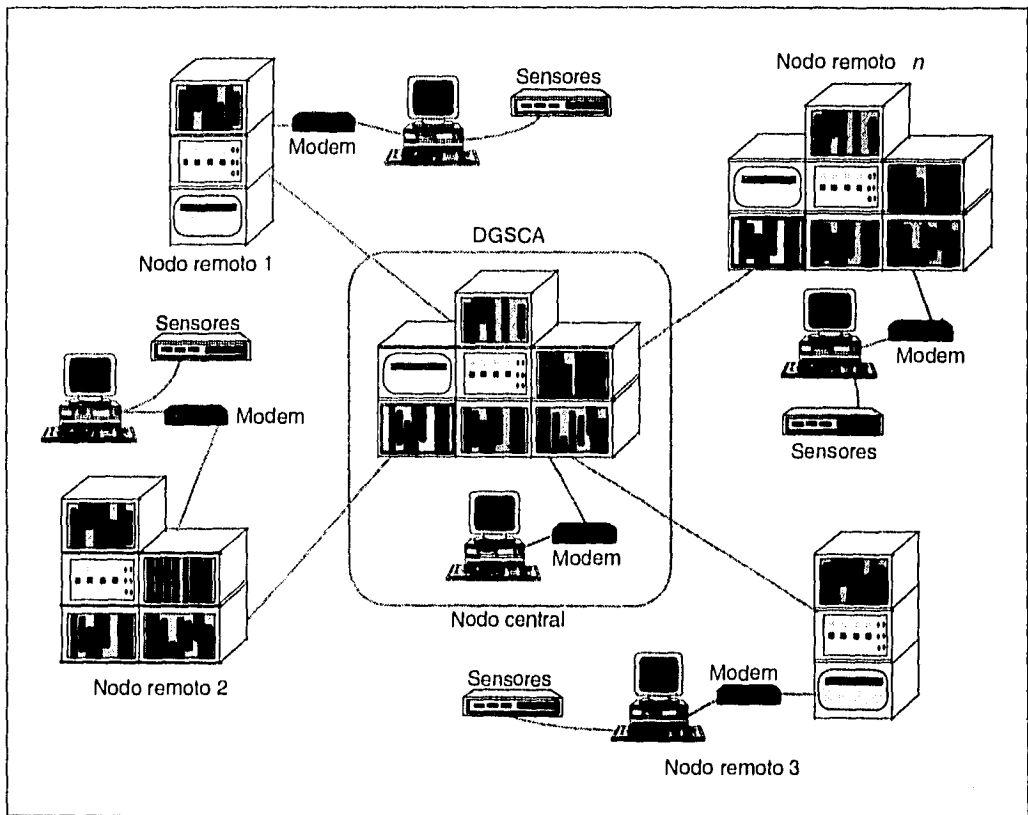


Figura II.2.3 Tercera propuesta de solución.- Sistema con enlaces a través de modem

Capítulo III

Diseño e implantación de la solución seleccionada

III.1 Consideraciones generales de diseño

El sistema que monitoree la Red de Telefonía Digital de la UNAM debe ser una herramienta que coadyuve en el propósito de proporcionar un servicio telefónico ininterrumpido en la medida de las posibilidades.

Qué debe hacer

Debe monitorear el estado del enlace entre los nodos de la Red y reportar las condiciones en las cuales se encuentran funcionando los equipos de conmutación. Para ello, en cada local deberá instalarse un sistema de adquisición de datos formado por una microcomputadora y un conjunto de sensores. La computadora deberá retener en memoria la cantidad de información que permita razonablemente reconstruir la historia de los sucesos que han tenido lugar en el local del conmutador.

La información resultante deberá concentrarse en un nodo central para facilitar su control. Un puerto serial de la computadora instalada en el local del conmutador se conectará a un modem que permita la transferencia de los datos. Existirá un nodo central que continuamente establecerá comunicación con todos los nodos que formen la red de monitoreo. Una vez creado el enlace, el sistema remoto enviará su información desplegándose inmediatamente en la pantalla de la computadora central y almacenándose para su consulta posterior.

El sistema así planteado debe obedecer a las siguientes líneas directrices.

Facilidad de operación

El despliegue de información en pantalla y la presentación de las diversas opciones deberán ser consistentes y claras. El estado de cada proceso debe ser reportado junto con los convenientes mensajes de ayuda, error o advertencia.

Presentación de la información de forma oportuna

Conforme se conozca el estado actual de los nodos remotos se deberá mostrar en pantalla, indicando la prioridad de atención .

Confiabilidad

Se deberá desarrollar un mecanismo de autodiagnóstico que determine que el sistema cuenta con:

- a) los recursos de software y hardware requeridos para su operación.
- b) el canal de comunicaciones preparado para la transmisión de información.

El esquema de transmisión de información que utilice el sistema deberá contemplar límites en los tiempos de espera a la ocurrencia de algún evento y monitorear los posibles errores que se generen durante el enlace a fin de reducir la probabilidad de que el canal de comunicaciones se quede bloqueado.

Será capaz de mantener en sincronía todos los nodos que formen la red de monitoreo.

En caso de falla del nodo central existirá un segundo nodo que tome la iniciativa del monitoreo de la red.

Se le dotará al sistema con la capacidad de autodepuración lo que permitirá incrementar su autonomía.

La asignación de niveles lógicos a las diversas situaciones detectadas por los sensores debe ser tal que, en caso de falla de algún sensor, se deberá obtener un nivel lógico asociado a una situación anormal.

Por otro lado, el Sistema indicará especialmente en la pantalla los nodos en los que se ha presentado una condición extraordinaria. Esta señalización no será eliminada sino hasta que el usuario lo haga personalmente, con lo que se asegurará su reconocimiento.

Mantenimiento sencillo

Debe ser posible modificar fácilmente los parámetros más importantes y cambiantes en la operación del sistema tales como la velocidad de transmisión, la utilización de otro puerto serial, conexión a una línea telefónica diferente, etcétera.

Se evitará el uso de tarjetas internas instaladas en las ranuras de expansión de las computadoras a utilizar, para que el tiempo de reemplazo e instalación se vean reducidos notablemente.

Se preferirá el uso de dispositivos periféricos y componentes electrónicos ampliamente disponibles en el mercado.

Bajo costo

El costo de los elementos externos a la infraestructura de comunicaciones de la Red y de los componentes utilizados para el dispositivo sensor deben ser una preocupación fundamental en la etapa de diseño. Asimismo, se buscará desarrollar una aplicación que no requiera grandes recursos de cómputo para poder funcionar y que de hecho, pueda reutilizar las computadoras que la Universidad está desechando.

III.2 La elección del lenguaje de desarrollo

La elección del lenguaje de programación es una decisión importante, ya que de ella dependerán los alcances y limitaciones del sistema y al mismo tiempo define la facilidad de la implantación y el tiempo de desarrollo. Considerando las opciones más comunes para el desarrollo de sistemas, la primera distinción que debe hacerse entre los lenguajes de programación es la que existe entre el ensamblador y los lenguajes de alto nivel.

Programar en ensamblador tiene ventajas: los programas son pequeños y se ejecutan rápidamente ya que utilizan métodos óptimos para realizar cada paso, también se puede indicar a la computadora que ejecute cualquier función que sea capaz de realizar, mientras que en los lenguajes de alto nivel normalmente no es posible.

Sin embargo, programar al nivel del lenguaje ensamblador es un proceso lento, si cualquier programa se escribiera en ensamblador, constaría de demasiadas rutinas elementales para la computadora. Los lenguajes de alto nivel se diseñan para disminuir el tiempo de desarrollo y la tendencia al error que son naturales en el ensamblador, obligando a la máquina a realizar la mayor parte del trabajo como sea posible, generando una gran cantidad de código en lenguaje de máquina para cada comando y evitando el uso de detalles relacionados con la forma de cómo opera internamente la máquina, pero que no tienen que ver con el trabajo que queremos realizar.

Se ha hablado de los lenguajes de alto nivel como una categoría colectiva, como si todos fueran iguales, tienen mucho en común, pero hay diferencias importantes entre ellos. El próximo paso consiste en analizar de entre la gran diversidad de lenguajes de alto nivel, los más comúnmente utilizados en la computadora personal: BASIC, PASCAL y C.

BASIC es lo más cercano a un lenguaje universal para computadoras personales; la fuerza del BASIC reside en su facilidad para desarrollar programas, y además incluye instrucciones que dan un fácil acceso a la mayoría de las características especiales de la familia PC.

Una enorme desventaja del BASIC es la gran variedad de versiones existentes que utilizan instrucciones completamente incompatibles entre sí. La más popular de ellas es la que viene con el sistema operativo MS-DOS. Esta tiene ciertas limitaciones que se deben conocer: es un intérprete que se ejecuta lentamente y tiene severas limitaciones en relación con el tamaño del programa que se puede escribir, junto con la cantidad de información que puede manejar. Para resolver este problema se han desarrollado compiladores de BASIC, y aún así, algunas versiones tienen una serie de características que van en contra de los esfuerzos para escribir programas estructurados.

Además del BASIC, dos de los lenguajes más conocidos y muy apropiados para la programación del sistema son Pascal y C. Ambos lenguajes tienen características útiles que permiten escribir programas bien estructurados, confiables y fáciles de actualizar o mantener. El Pascal y el C tienen muchas similitudes, incluyendo el mismo tipo de características estructurales que ayudan a la práctica de una buena programación. Ambos son adecuados para la construcción de programas extensos.

El C es favorecido por los programadores que buscan eficiencia en un lenguaje de alto nivel, y por aquellos que quieren que sus programas tengan alguna orientación especial, como hacia el sistema operativo UNIX, que está altamente influenciado por las convenciones del lenguaje C.

Generalmente se considera que el Pascal es un mejor lenguaje, limpio y con mucha menor tendencia al error; por otro lado, C es particularmente bueno para escribir programas que necesiten mayor eficiencia y que trabajan muy relacionados con el BIOS y con el sistema operativo de la computadora. Una característica importante de ambos, es que se pueden usar instrucciones o rutinas en lenguaje ensamblador para realizar funciones que normalmente no podrían realizarse en esos lenguajes.

Debido a que para nuestro proyecto la velocidad de ejecución es un factor crítico, la programación debiera ser realizada en el lenguaje ensamblador de la máquina. Lo anterior no es usual dado que en la mayoría de los programas de comunicación serial las funciones de entrada o salida sólo son una pequeña porción del programa, mientras que la mayor parte del mismo se centra en desarrollo de la interfaz con el usuario y de otras características que soportan, pero que no participan directamente en el proceso de transmisión.

Los pocos libros existentes de comunicaciones seriales dan ejemplos en BASIC o Pascal. Invariablemente los programas así generados son torpes y lentos ya que ambos son fundamentalmente inadecuados para tiempos críticos de ejecución.

El lenguaje de programación C es ideal para nuestras necesidades, es *compacto, conciso y rápido*, sin embargo lo más importante en la lista de características del lenguaje C es el aspecto de la *portabilidad*. Simplemente expuesto, la *portabilidad* significa que, los sistemas y el hardware no se oponen a un código escrito en C, es decir que si un programa se compila y se ejecuta en un determinado ambiente, también lo hará en cualquier otro con pocas modificaciones. Aunando a esto sus tipos de datos y características estructurales, se obtiene un lenguaje de alto nivel, muy eficiente y que permite la construcción de programas extensos, que requieran una relación muy estrecha con el BIOS de la computadora y que sean de fácil mantenimiento. Se dice que el lenguaje C está "más cerca" de la computadora en la cual reside; esto es, sus instrucciones básicas son aquellos operadores típicos del conjunto de instrucciones de un microprocesador. Un programa en C bien escrito y en un compilador avanzado puede rivalizar con uno realizado en ensamblador.

La decisión de optar por el lenguaje de programación C entre los que se podían elegir, es a nuestro punto de vista, la más acertada y, como se verá en las secciones correspondientes a la implantación, realmente se adaptó a nuestras expectativas y necesidades.

El convertidor analógico-digital de ocho bits ADC0804 basa su funcionamiento en el principio de aproximaciones sucesivas R-2R, cuya escala máxima (Full Scale) se alcanza cuando el voltaje de entrada V_{in} llega al valor del voltaje V_{ref} . Este voltaje se le indica al convertidor a través de su terminal $V_{ref}/2$.

Como se ha visto en la descripción del proyecto, una temperatura superior a los 24 °C dentro del local del conmutador es crítica, por lo que se decidió que la escala máxima del convertidor AD se podía alcanzar a 65 °C, la mitad del valor de la temperatura máxima que el transductor puede soportar. Lo anterior quiere decir que cuando el LM35D proporcione 0.65 V, el convertidor debe alcanzar su escala máxima, para ajustar el convertidor a este voltaje se empleó un divisor de tensión, formado por R_1 , R_3 y el potenciómetro R_2 , acoplándose al convertidor AD por medio de un amplificador operacional en configuración seguidor. El valor del voltaje de ajuste ($V_{ref}/2$) que se obtuvo fue entonces:

$$\frac{V_{ref}}{2} = \frac{0.65V}{2} = 0.325 V$$

Para obtener la sensibilidad de este sensor se consideran las características del convertidor AD y la temperatura que fijamos para alcanzar la escala máxima:

$$\frac{\text{Temperatura máxima}}{\text{No. de intervalos de conversión}} = \frac{65^{\circ}C}{255 \text{ intervalos}} = 0.25 \frac{^{\circ}C}{\text{intervalo}}$$

Lo anterior quiere decir que por cada 0.25 °C de variación en la temperatura el convertidor generará un patrón de bits diferente.

III.3.2 Sensor del voltaje en las baterías de respaldo

Para el diseño de este sensor consideramos que cuando las baterías se encuentran en recarga, el voltaje de igualación que se les aplica es de aproximadamente 55 V y cuando ocurre una falla de energía eléctrica de AC éstas operan a 48 V; estos niveles de voltaje no resultan manejables para los circuitos de conversión AD necesarios para que la lectura pueda enviarse a la computadora, por lo que resulta necesaria una etapa previa de adecuación.

Se empleó un amplificador operacional en configuración diferencial, con objeto de adecuar los niveles de voltaje medidos en las baterías, a los que maneja el convertidor AD. El circuito utilizado se muestra en la figura III.3.2.

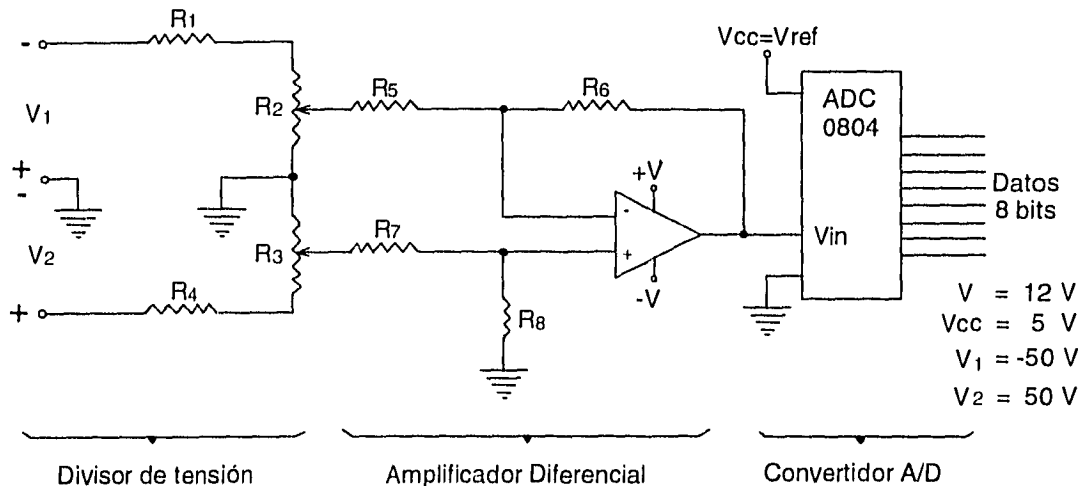


Figura III.3.2 Sensor del voltaje de baterías de respaldo

Cuando no se aplica algún voltaje externo a la terminal $V_{ref}/2$ del convertidor AD, su circuitería interna fija el voltaje de referencia al voltaje de alimentación (5 V), y es al que se debe alcanzar la escala máxima. Para reducir el voltaje en las baterías a este orden de magnitud, se utiliza el siguiente divisor de tensión:

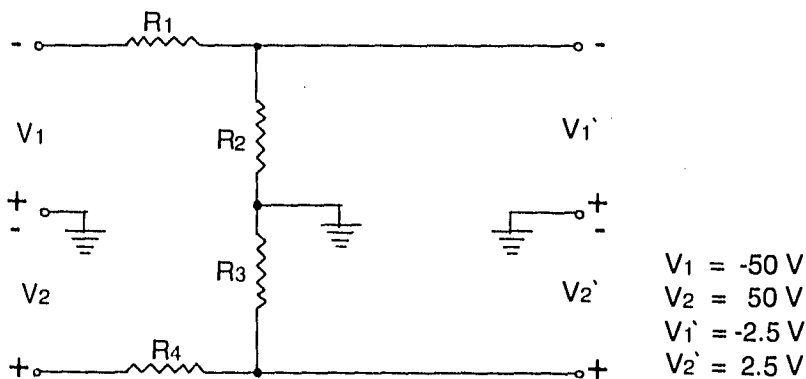


Figura III.3.3 Divisor de tensión para adecuación de niveles

El voltaje total de las baterías se puede dividir en dos voltajes con un punto de referencia. Considerando un valor límite de 100 V en la entrada del divisor (esto es el doble de los valores típicos: -50 V de V_1 a tierra y 50 V de V_2 a tierra), se tiene que el valor del voltaje en la salida debe ser de 5 V (-2.5 V de V_1' a tierra y 2.5 V de V_2' a tierra), para así alcanzar el voltaje de referencia mencionado anteriormente, y por lo tanto la escala máxima.

Para obtener los valores adecuados de las resistencias se considera que en el límite máximo:

$$V_2 = 50 \text{ V}$$

$$V_2' = 2.5 \text{ V}$$

de la expresión del divisor de tensión,

$$V_2' = V_2 \frac{R_3}{R_3 + R_4}$$

sustituyendo los valores del voltaje,

$$2.5 \text{ V} = 50 \text{ V} \frac{R_3}{R_3 + R_4}$$

se obtiene finalmente:

$$R_4 = 19 R_3$$

por la simetría del circuito:

$$R_1 = 19 R_2$$

considerando valores de $R_2 = R_3 = 1 \text{ k}\Omega$ se tiene que $R_1 = R_4 = 19 \text{ k}\Omega$.

Para fines de calibración del sensor, R_2 y R_3 se seleccionaron resistencias variables.

Ahora analicemos la etapa diferencial:

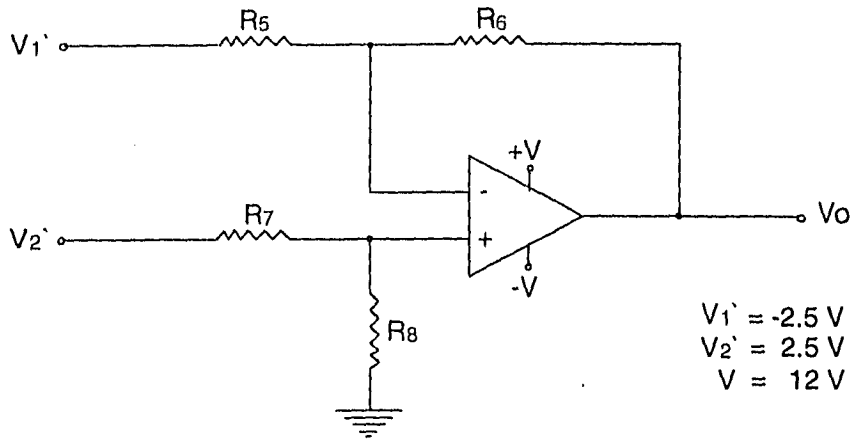


Figura III.3.4 Amplificador diferencial

Aplicando el principio de superposición y considerando que en la configuración del amplificador diferencial básico $R_6 = mR_5$, $R_5 = R_7$ y $R_6 = R_8$:

si la terminal de V_2' está a tierra:

$$V_0 = -\frac{R_6}{R_5} V_1' = -\frac{mR_5}{R_5} V_1'$$

$$V_0 = -mV_1'$$

si la terminal de V_1' está a tierra:

$$V_0 = V_2' \frac{R_8}{R_7 + R_8} (1+m) = V_2' \frac{mR_5}{R_5 + mR_5} (1+m)$$

$$V_0 = mV_2'$$

por lo que al sumar ambas salidas nos da como resultado

$$V_0 = mV_2' - mV_1' = m(V_2' - V_1')$$

si lo que se requiere es que el voltaje a la salida del amplificador diferencial sea directamente la diferencia de V_2' y V_1' , m debe tomar el valor de uno, y por lo tanto:

$$R_5 = R_6 = R_7 = R_8$$

para estas resistencias se seleccionó el valor comercial disponible de 1 k Ω .

La expresión del voltaje de salida finalmente es:

$$V_0 = V_2' - V_1'$$

con respecto a tierra: $V_2' = 2.5$ V y $V_1' = -2.5$ V, así: $V_0 = 2.5V - (-2.5V) = 5$ V y éste es un voltaje que se puede aplicar al convertidor AD, el cual generará una lectura proporcional al voltaje que tengan las baterías en un determinado instante.

Considerando las mismas características para el convertidor AD que en el caso del sensor de temperatura, determinemos la sensibilidad de circuito:

$$\frac{\text{Voltaje máximo}}{\text{No. de intervalos de conversión}} = \frac{100 \text{ V}}{255 \text{ intervalos}} = 0.392 \frac{\text{V}}{\text{intervalo}}$$

Este valor indica que para un incremento de 0.392 V se obtiene una lectura diferente en el convertidor AD.

III.3.3 Sensores de alarmas mayor y menor

Este circuito debe disminuir el voltaje en las alarmas de los conmutadores a niveles compatibles con los utilizados por la computadora. Al mismo tiempo debe aislar el equipo de conmutación con respecto a los circuitos sensores, para ello se utilizó acoplamiento óptico. Se emplearon optoacopladores en dos circuitos iguales, uno para la alarma mayor y otro para la menor.

El circuito utilizado se muestra en la siguiente figura: .

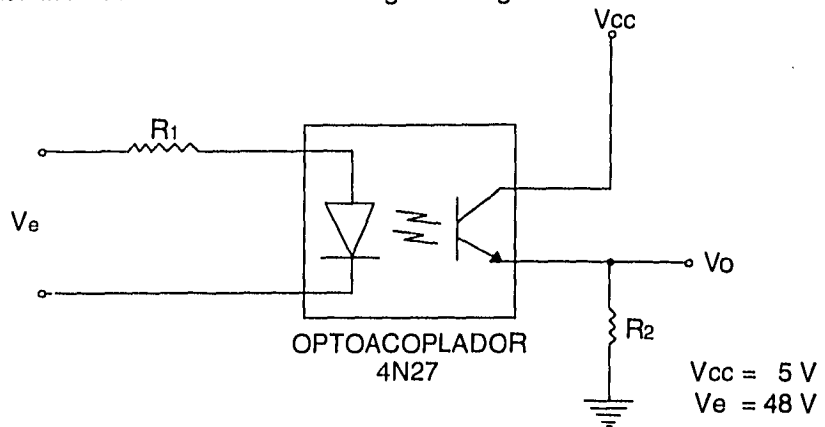


Figura III.3.5 Sensor de alarma mayor o menor

Para el diseño del circuito consideremos las siguientes características del optoacoplador 4N27:

Para el diodo: Voltaje de encendido (V_d) = 1.5 V
Corriente máxima (I_d) = 3 mA

Para el fototransistor en la etapa de salida: $V_{CE\text{ sat}} = 0.2\text{ V}$
 $I_{c\text{ máx}} = 3\text{ mA}$

En el diodo del optoacoplador, el voltaje V_d es constante, por lo que se coloca una resistencia en serie que limite la corriente y reduzca el voltaje:

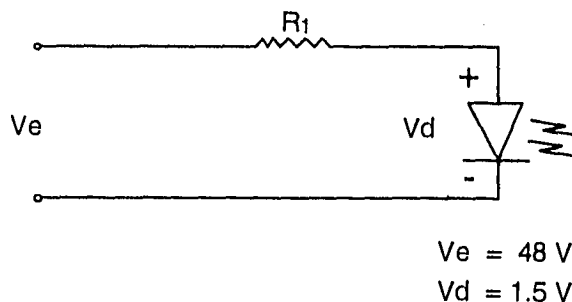


Figura III.3.6 Circuito de entrada del optoacoplador

así la caída de voltaje en la resistencia R_1 es de:

$$V_{R_1} = V_e - V_d$$

$$V_{R_1} = 48V - 1.5V = 46.5 V$$

dado que la corriente máxima que soporta el diodo es de 3 mA, el valor de R_1 es:

$$R_1 = \frac{V_{R_1}}{I_d} = \frac{46.5V}{3mA} = 15.5 k\Omega$$

Dado que sólo nos interesa utilizar el fototransistor del optoacoplador como un interruptor, los cálculos se realizan con su voltaje de saturación y corriente de colector máxima:

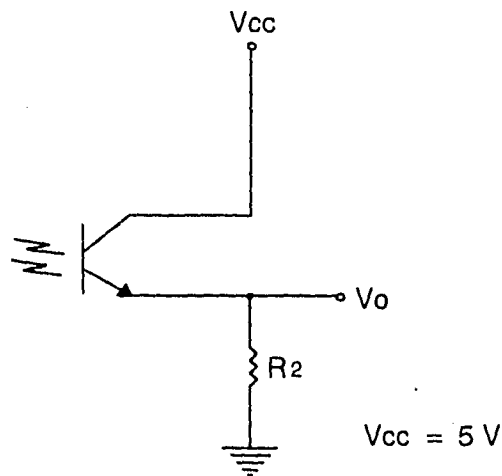


Figura III.3.7 Etapa de salida del optoacoplador

el valor de R_2 será entonces

$$R_2 = \frac{V_{cc} - V_{CE sat}}{I_C máx} = \frac{5V - 0.2V}{3mA} = 1.6 k\Omega$$

Cuando no existe alarma, el diodo emisor de luz del optoacoplador se encuentra apagado, en consecuencia el fototransistor se encuentra en la región de corte; y no hay caída de voltaje en R₂; se producen 0 V a la salida ocasionando un "0" lógico. En el caso de que exista alarma el diodo se enciende y el fototransistor se satura produciendo una caída de 4.8V en R₂, lo cual representa un "1" lógico.

Hay que considerar que al conectar un dispositivo TTL a la etapa de salida del optoacoplador, los voltajes anteriormente calculados se ven afectados. El caso crítico sucede durante la ocurrencia de un nivel lógico de cero a la salida del fototransistor. En este caso existe un valor máximo de la corriente que se puede drenar (-I_{IL}) a través de R₂ y que permite conservar un nivel lógico de "0" (V_{IL}, generalmente menor o igual a 0.8V):

$$-I_{IL} = \frac{V_{IL}}{R_2} = \frac{0.8V}{1.6K\Omega} = 600\mu A$$

En conclusión, el circuito TTL que se conecte a la salida del optoacoplador no debe drenar más de 600μA cuando se le alimenta con un '0' lógico.

El estado de la alarma se codifica de acuerdo a la siguiente tabla:

Valor lógico del optoacoplador	Estado de la alarma*
0	Apagada
1	Encendida

*Alarmas mayor y menor

III.3.4 Sensor de falla en el suministro de energía eléctrica de AC

Al tratar de sensar esta falla, se debía convertir el voltaje de dos fases de la señal de alterna de 220 V_{RMS} a un nivel de DC que fuera compatible con niveles TTL para así poder ser enviado a la computadora. Considerando que uno de los objetivos del circuito era no emplear elementos voluminosos tales como un transformador, se diseñó el siguiente circuito:

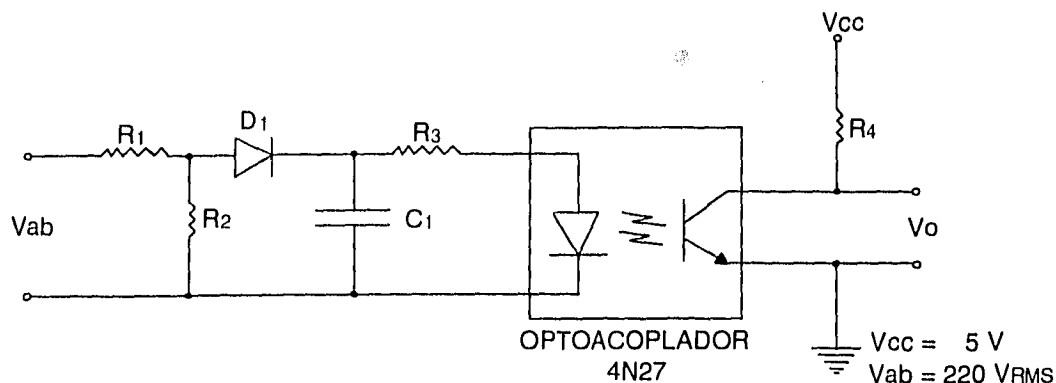


Figura III.3.8 Sensor de energía eléctrica de AC

Se empleó un optoacoplador para aislar la señal de alterna con respecto a los sensores. Dado que el diodo emisor de luz del optoacoplador tiene un voltaje de encendido de 1.5 V, se redujo el voltaje de la línea de 220 VRMS a un valor más manejable, seleccionamos 10 VRMS, de la expresión del divisor de tensión y sustituyendo estos valores:

$$V_{R2} = V_{ab} \frac{R_2}{R_1 + R_2}$$

$$10V_{RMS} = 220V_{RMS} \frac{R_2}{R_1 + R_2}$$

de donde R_1 y R_2 se relacionan por:

$$R_1 = 21 R_2$$

se escogió 1 k Ω para R_1 y por tanto R_2 toma el valor de 21 k Ω .

La señal obtenida se alimentó a un rectificador de media onda, formado por el diodo D_1 y para tener un nivel constante de DC se introdujo el capacitor C_1 .

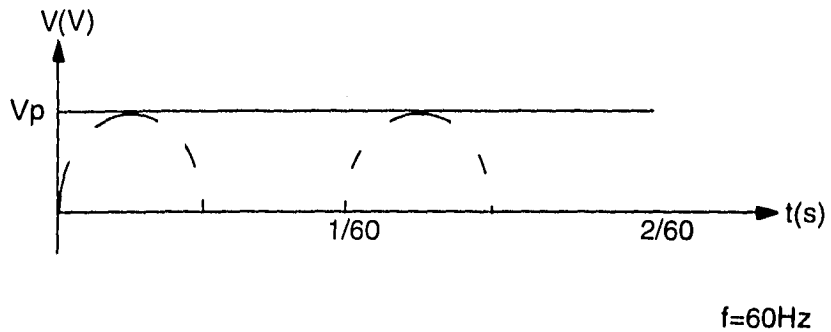


Figura III.3.9 Forma de onda en el capacitor C1

Para calcular el valor de R_3 se debe considerar que el voltaje de DC es aproximadamente el valor pico de la señal de alterna (V_p) en los extremos del capacitor como se muestra en la figura III.3.9, éste vale 14.1V por lo que:

$$R_3 = \frac{V_p - V_d}{I_d} = \frac{14.1\text{V} - 1.5\text{V}}{3\text{ mA}} = 4.2\text{ k}\Omega$$

En el fototransistor se debe tener un voltaje adecuado para ser reconocido como un nivel TTL, con esta observación consideramos en el análisis al transistor como un interruptor.

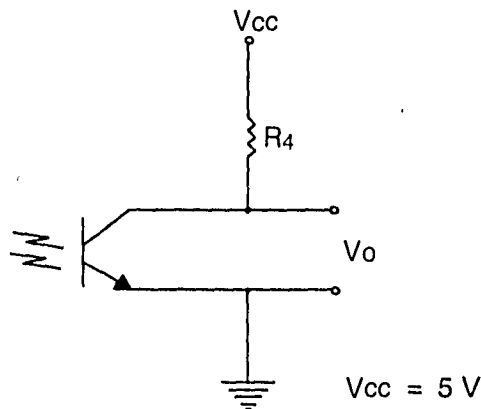


Figura III.3.10 Etapa de salida del optoacoplador

Con referencia en la figura III.3.10, dado que $I_{c \text{ máx}} = 3 \text{ mA}$ y $V_{CE \text{ sat}} = 0.2 \text{ V}$, el valor de R_4 es:

$$R_4 = \frac{V_{CC} - V_{CE \text{ sat}}}{I_{C \text{ máx}}} = \frac{5 \text{ V} - 0.2 \text{ V}}{3 \text{ mA}} = 1.6 \text{ k}\Omega$$

Cuando existe suministro de energía eléctrica de AC el diodo del optoacoplador se enciende, por lo tanto el fototransistor se satura, su voltaje de colector a emisor es de 0.2 V y esto equivale a tener un "0" lógico. Al ocurrir una falla en el suministro de energía eléctrica de AC el diodo del optoacoplador se apaga y el fototransistor cambia a la región de corte, por lo que el voltaje de salida es aproximadamente V_{CC} (5 V) lo que nos lleva a tener un "1" lógico. La siguiente tabla muestra la relación entre el estado de la línea y el valor lógico generado por el optoacoplador:

Valor lógico a la salida del optoacoplador	Estado del suministro de AC
0	No existe falla
1	Existe falla

Dependiendo del tamaño del equipo de conmutación, su alimentación se realiza con dos o tres fases, así que para poder tener la posibilidad de monitorear cualquiera de ambos casos con cambios sencillos y mínimos, se consideró utilizar el siguiente circuito electrónico. En él se puede seleccionar el monitoreo de dos o tres fases a través de un selector. Cuando alguna de las fases falla, se obtiene un "1" lógico a la salida de la compuerta OR.

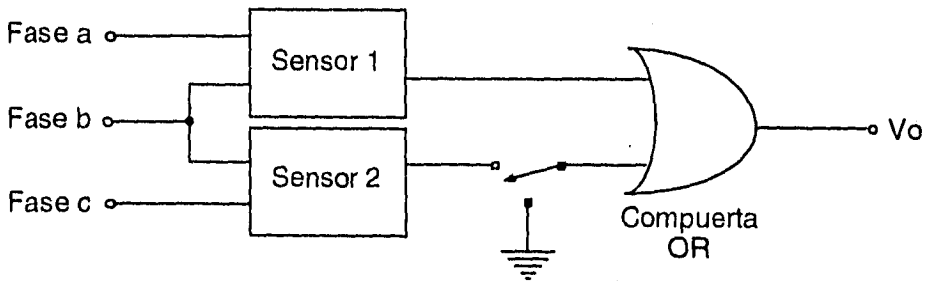


Figura III.3.11 Diagrama esquemático para sensor tres fases de alimentación

III.4 Adquisición de los datos

Con el propósito de obtener la información de los sensores, se pensó en utilizar una opción que aprovechara la circuitería estándar de la PC en lugar de dispositivos adicionales tales como tarjetas de interfaz que se insertarían en una ranura de expansión; esta consideración por sí misma reduce costos, puntos de falla y al mismo tiempo incrementa la facilidad de instalación y mantenimiento.

A pesar de que con una conexión serial se permite transmitir datos bajo ciertas condiciones hasta una distancia de 15 metros, esta opción involucra añadir un circuito controlador de comunicaciones en el dispositivo externo y tener disponibles al menos dos puertos seriales en cada computadora remota; uno para transmisión de la información a la computadora central y otro para la adquisición de los datos. Esta desventaja originó que se decidiera optar por aprovechar el puerto paralelo para leer los datos obtenidos por los circuitos electrónicos externos; sin embargo hay que tomar en cuenta algunas limitaciones que presenta el puerto cuando se le utiliza en una forma no convencional.

Una computadora típica hace 15 años sólo enviaba caracteres a la impresora y el problema más común de ésta en esa época era quedarse sin papel. Por esta razón originalmente se consideró que en una computadora personal, el flujo de datos debería tener una sola dirección, de la PC a la impresora y solamente unas pocas señales de control se dirigirían en sentido contrario. En aquellos días el puerto de impresora estilo *Centronics* ganó popularidad y todas las computadoras personales fueron equipadas con puertos paralelos unidireccionales, esto es, a través de él la PC podía enviar, pero no recibir datos. Después de que se produjeron las primeras PC, IBM replanteó el diseño y empezó a vender puertos paralelos que eran capaces de una operación bidireccional. Sin embargo IBM no soportó oficialmente la operación bidireccional hasta que introdujo la línea PS/2 en 1987.

Sin embargo y a pesar de que los fabricantes prestigiosos de computadoras actualmente instalan puertos bidireccionales en sus productos, los adaptadores de bajo costo y algunas tarjetas multifunción se producen construyendo sus puertos paralelos con búfers unidireccionales. Por esta razón no se puede asegurar que el puerto paralelo sea bidireccional en el 100% de las PC que se encuentren funcionando.

Teniendo presente la anterior limitación, desarrollamos un método confiable para leer datos a través de este puerto. Para ello consideramos que cada puerto paralelo tiene asociadas un conjunto de cuatro direcciones consecutivas de memoria denominadas canales A, B, C y D¹. En el estándar *Centronics* las señales del canal A se consideran datos que viajan de la PC hacia la impresora y desde el punto de vista de la computadora son únicamente de salida; normalmente este canal se utiliza para enviar caracteres a la impresora. A su vez la impresora dispone de cinco señales en el canal B para indicarle su estado de operación a la computadora. Por estas razones, no es posible utilizar más de cinco líneas de entrada hacia la computadora para leer datos en forma completamente general e independiente del diseño particular del puerto paralelo con que se cuente.

Considerando estas condiciones, se decidió multiplexar todas las líneas externas de datos -19 en total- para compartir un canal con un ancho de cuatro bits. Este *bus* se formó con las líneas correspondientes a las siguientes señales de la impresora: ACKNOWLEDGE, BUSY, ERROR y PAPER OUT; en consecuencia se dividieron todas las señales por leer, en grupos de cuatro bits (*nibble*).

Los multiplexores se implantaron a través de circuitos 74LS244, los cuales son búfers TTL de tres estados. Cada uno de estos circuitos está formado por ocho búfers que se activan con lógica negada **en grupos de cuatro**, así que al aplicar una señal de estado bajo en alguna de sus terminalés de habilitación permite la lectura de un *nibble*. Esta característica origina una forma natural de realizar el multiplexaje y regenerar la señal. Las salidas de cada grupo de búfers se conectaron en paralelo para compartir un bus común. Con este esquema, enviando la palabra de control adecuada por el canal A se elige el sensor que se quiere leer y los datos se reciben por el canal B. Esta interfaz se muestra en la figura III.4.1.

Una limitación que hay que tener presente es el tiempo que requieren los búfers para proporcionar datos estables después de haber habilitado sus salidas. Valores típicos de este parámetro se encuentran alrededor de los 10ns.

¹ En teoría el canal D debiera ser utilizado para para la programación de los otros tres si los puertos se fabricaran utilizando un circuito integrado 8250 PPI. Actualmente el canal C no se utiliza.

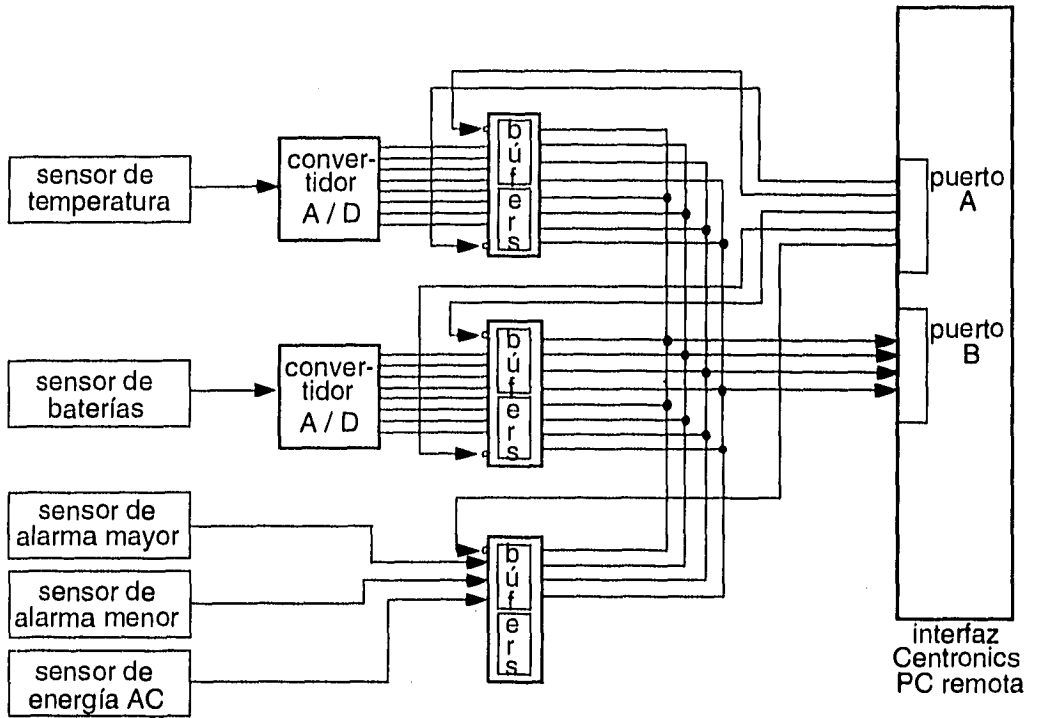


Figura III.4.1 Interfaz para la adquisición de los datos .

La estructura del programa que realiza las lecturas a través del puerto paralelo está fuertemente influenciada por dos aspectos, uno de ellos es la necesidad de multiplexar los sensores y el otro es la de colocar los datos en un formato adecuado que permita una transmisión eficiente.

Durante el proceso de adquisición de datos, a cada lectura que se obtuvo de los sensores, se le dio el formato adecuado para su transmisión posterior a través de modem, utilizando los caracteres de *control lógico* comúnmente empleados en comunicaciones, los cuales marcan la frontera entre los datos y el identificador de la lectura. Estos caracteres de control los definimos en el *header* `asciiesp.h` y son los siguientes:

SOH: *Start of Header*. Este carácter indica el inicio del identificador de la lectura, el cual se encuentra formado por la fecha y hora en que se efectuó.

STX: *Start of Text*. Por medio de este carácter se marca el inicio del bloque de datos y en consecuencia el fin del identificador.

ETX: *End of Text*. Delimita el fin del bloque de datos.

El formato utilizado para cada lectura se puede apreciar en la figura:

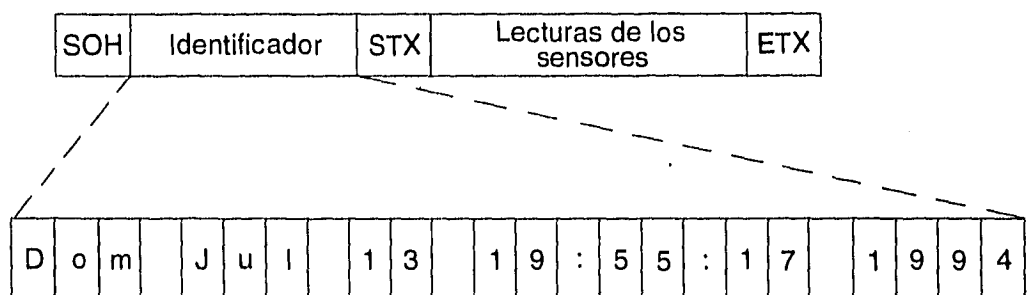


Figura III.4.2 Estructura de una lectura a los sensores

La sección del programa que realiza la función de leer los datos a través del puerto paralelo se muestra a continuación:

```

#include <time.h>
#include <dos.h>
#include <stdio.h>

#include "asciiesp.h"
#include "funcion.h"

#define SEL_TEMP_LO 0xf7
#define SEL_TEMP_HI 0xfb
#define SEL_BATE_LO 0xfd
#define SEL_BATE_HI 0xfe
#define SEL_REST 0xef

#define LPT1_PTO_A 0x0378
#define LPT1_PTO_B 0x0379

/*
Esta función obtiene una lectura de datos de los sensores
*/

void vector( unsigned char register *p_vec_auxiliar )
{
    unsigned char temp_lo, temp_hi, temp;
    time_t fecha;
    char *header;

    delay( 0 );

    *p_vec_auxiliar++ = SOH;

    time( &fecha );
    header = ctime( &fecha );
    traduce_fecha( header );
    while( *header != '\n' )
    {
        *p_vec_auxiliar = *header;
        header++; p_vec_auxiliar++;
    }

    *p_vec_auxiliar++ = STX;
}

```

```

/*
Elige y lee la parte baja del sensor de temperatura
*/
outportb( LPT1_PTO_A, SEL_TEMP_LO );
delay( 100 );
temp_lo = get_nibble( inportb( LPT1_PTO_B ) );

/*
Elige y lee la parte alta del sensor de temperatura
*/
outportb( LPT1_PTO_A, SEL_TEMP_HI );
delay( 100 );
temp_hi = get_nibble( inportb( LPT1_PTO_B ) );

temp_hi <<= 4;
temp = temp_hi | temp_lo;
*p_vec_auxiliar++ = temp;

/*
Elige y lee la parte baja del sensor de baterías
*/
outportb( LPT1_PTO_A, SEL_BATE_LO );
delay( 100 );
temp_lo = get_nibble( inportb( LPT1_PTO_B ) );

/*
Elige y lee la parte alta del sensor de baterías
*/
outportb( LPT1_PTO_A, SEL_BATE_HI );
delay( 100 );
temp_hi = get_nibble( inportb( LPT1_PTO_B ) );

temp_hi <<= 4;
temp = temp_hi | temp_lo;

*p_vec_auxiliar++ = temp;

/*
Elige y lee los sensores de alarmas y potencia
*/
outportb( LPT1_PTO_A, SEL_REST );
delay( 100 );
temp_lo = get_nibble( inportb( LPT1_PTO_B ) ) & 0x7;

```

```

/*
La siguiente línea es necesaria para evitar la
posibilidad de que se transmita un ASCII nulo
*/
*p_vec_auxiliar++ = temp_lo | 0xf0;

*p_vec_auxiliar++ = ETX;
}

```

Figura III.4.3 Listado de la función que lee los datos de los sensores

En la función, el primer conjunto de instrucciones `#define` asignan un nombre simbólico a cada palabra de control utilizada para el multiplexaje de los datos. En estas palabras de ocho bits, a uno de ellos se le asigna un nivel lógico "0", de esa forma, cuando se les utiliza para habilitar los circuitos 74LS244, sólo un grupo de cuatro búfers está seleccionado a la vez. Así por ejemplo, la palabra 0xf7 representada por el nombre `SEL_TEMP_LO` permite la lectura de la parte baja del sensor de temperatura colocando un estado bajo en su terminal de habilitación y al mismo tiempo inhibiendo los demás búfers.

El segundo conjunto de instrucciones `#define` asigna un nombre simbólico a la dirección asociada a dos canales del primer puerto paralelo (dirección base 0x378). Estos nombres se emplean para direccionar la entrada y salida de datos.

A la función `vector` se le envía como parámetro la dirección de un arreglo que se utilizará para almacenar la lectura, este valor se recibe en el apuntador `p_vec_auxiliar`.

La rutina inicia asignando al primer elemento del arreglo el carácter de control SOH. Las instrucciones

```

time( &fecha );
header = ctime( &fecha );

```

generan una cadena de caracteres que contiene la fecha y hora del sistema.

Después de haber copiado esta información al arreglo y generado así el identificador, se coloca un carácter separador STX indicando el inicio de las lecturas de los sensores.

Los datos de los sensores se empiezan a leer iniciando con el de temperatura. Con las instrucciones

```
outportb( LPT1_PTO_A, SEL_TEMP_LO );  
delay( 100 );  
temp_lo = get_nibble( inportb( LPT1_PTO_B ) );
```

por el canal A del puerto paralelo representado con LPT1_PTO_A, se envía la palabra de control SEL_TEMP_LO que permite direccionar los primeros cuatro bits del sensor de temperatura (parte baja). El retardo mostrado tiene por objetivo asegurar que los búfers coloquen datos estables en sus salidas antes de intentar leerlos. La llamada a la función inportb lee los datos presentes en el canal B del puerto direccionado por la constante simbólica LPT1_PTO_B. El byte obtenido contiene datos válidos en cuatro de sus bits; para ordenarlos y agruparlos en un nibble se utiliza la función get_nibble.

Posteriormente y de forma similar, se leen los restantes bits de ese sensor (parte alta) direccionados a través de la máscara SEL_TEMP_HI. Con esta información, se ensamblan los ocho bits que constituyen la lectura del primer sensor.

Con el mismo método y direccionando los búfers adecuados se realiza la lectura del sensor de baterías.

Posteriormente para leer todos los datos dicotómicos se utiliza la máscara de SEL_REST. Dado que en este caso es probable que se produzca una combinación 0x00 (carácter NUL), se hace una OR del valor leído con 0xf0, a fin de evitar la posibilidad de transmitir inadvertidamente ese valor².

Finalmente se añade un carácter ETX que indica el fin de un ciclo de lecturas.

² Esto es debido a que generalmente el carácter NUL se utiliza para el *control físico* de las comunicaciones y su inserción o eliminación puede afectar el comportamiento mecánico de dispositivos tales como una terminal de video o una impresora remota.

III.5 Comunicaciones

Para poder registrar los eventos que se presenten en los locales de los conmutadores, las computadoras remotas ahí instaladas estructuran paquetes de información conforme se detalló previamente en el punto III.4. (*Adquisición de los datos*) y a los cuales denominamos *lecturas*.

La computadora central ubicada en la DGSCA establece enlaces por medio de modem con cada una de las computadoras remotas, crea el canal de comunicaciones mostrado en la figura III.5.1 y recopila las lecturas generadas.

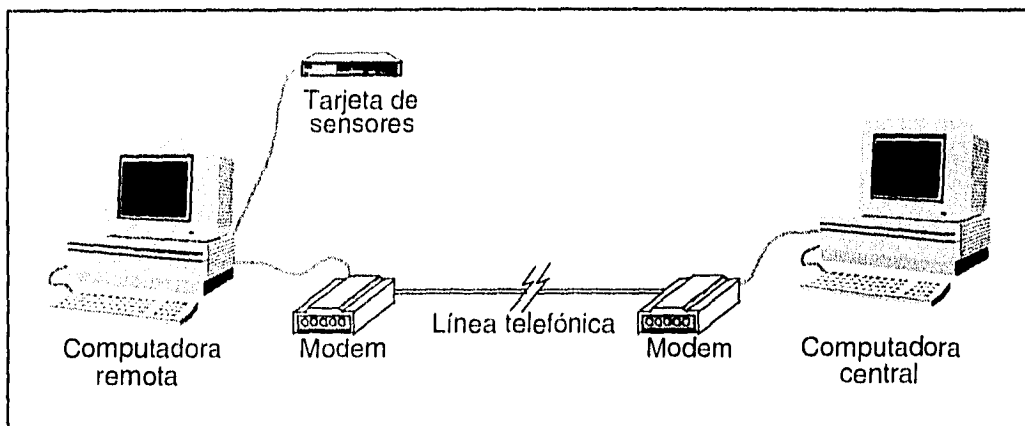


Figura III.5.1 Canal de comunicaciones utilizado para el monitoreo de cada nodo remoto

Una vez que ha recibido toda la información de la computadora remota en turno, la PC central *cierra* el canal de comunicaciones, interpreta la información, la despliega y origina otro enlace. Una vez terminado un ciclo de llamadas a los nodos remotos, guarda la información en disco duro para su posterior consulta.

El aprovechamiento óptimo de esta infraestructura de comunicaciones requiere el control directo del hardware de entrada y salida de las computadoras en las que se ejecuta el Sistema de Monitoreo; por otro lado, como consecuencia de las características operativas de la Red de Telefonía Digital de la UNAM, se deben hacer algunas consideraciones para el uso del modem en el Sistema.

III.5.1 Arquitectura interna del controlador serial

El primer paso para la elaboración de una interfaz entre el Sistema de Monitoreo y el hardware de comunicación serial en las computadoras a utilizar, es el de comprender su constitución y operación.

El circuito integrado que controla la entrada y salida de datos seriales en una PC es el *8250 UART*³. El control del este dispositivo se realiza leyendo y escribiendo en sus registros, que pueden pertenecer a uno de los tres grupos siguientes:

- Entrada y salida de datos
- Programación de los parámetros del controlador
- Verificación del estado de las comunicaciones

El 8250 tiene una operación peculiar. Consta de diez registros; ocho que se accesan directamente mediante las tres líneas de selección (A0, A1 y A2) con que cuenta el UART y otros dos que se pueden accesar únicamente cuando cambia el valor en el bit 7 del registro de formato de datos (o registro 3), llamado *bit de acceso al divisor de frecuencia* o *DLAB*⁴

Cuando el DLAB toma un valor lógico de uno, se crean dos registros *virtuales* adicionales, cambiando las funciones del *registro de transmisión y recepción de datos* (registro 0) y del *habilitador de interrupciones* (registro 1). El registro 0 se convierte en la *parte baja*, y el registro 1 en la *parte alta* de la palabra de 16 bits que utiliza el divisor de frecuencia del 8250 para generar la velocidad de transmisión. La relación entre el divisor así obtenido y esta velocidad está dada por :

$$\text{divisor} = \frac{\text{frecuencia de referencia del reloj}}{16 * \text{la velocidad de transmisión deseada}}$$

En donde, la frecuencia de referencia puede ser de 1.8432 o 3.072 MHz

³ UART es el acrónimo de *Universal Asynchronous Receiver Transmitter*. Aunque algunas computadoras utilizan controladores de mejor rendimiento tales como el 16450 o el 16550, por compatibilidad, emulan un 8250 en su operación; por esta razón, cualquier controlador de comunicaciones seriales en una PC se considerará como un 8250 UART.

⁴ DLAB es acrónimo de *Divisor Latch Access Bit*

La tabla III.5.1 resume la forma en que son direccionados los diez registros del 8250 UART.

Tabla III.5.1 Direccionamiento de los registros del 8250 UART

DLAB	A0	A1	A2	Nombre del registro
0	0	0	0	De transmisión y recepción de datos
0	0	0	1	Habilitador de interrupciones
X	0	1	0	De identificación de interrupciones
X	0	1	1	De formato de los datos
X	1	0	0	De control de salidas RS-232
X	1	0	1	De verificación de estado de la comunicación serial
X	1	1	0	De estado de la entrada RS-232
X	1	1	1	De reserva
1	0	0	0	Parte baja del divisor de frecuencias
1	0	0	1	Parte alta del divisor de frecuencias

X - condición "no importa", generalmente el DLAB tiene el valor de '0' lógico

A continuación se enumeran y describen los registros *reales* del UART (cuando el DLAB tiene el valor de '0'):

0: *Registro de transmisión y recepción de datos.* Este registro permite la entrada y salida de datos seriales, al leer y escribir en él respectivamente.

1: *Registro habilitador de interrupciones.* Este registro puede habilitar los cuatro tipos de interrupciones que existen en el 8250 con un nivel lógico de uno en el bit correspondiente:

Bit 0 (RxRDY). Produce una interrupción cuando en el registro de recepción de datos se encuentra un byte listo para su lectura .

Bit 1 (TBE). El UART genera una interrupción cuando puede aceptar otro byte para su transmisión.

Bit 2 (BREAK). Se genera una interrupción cuando se ha detectado un error en el formato del último dato recibido. Este puede ser un error de paridad, trama, desbordamiento o cuando se presenta una condición de ruptura en el enlace.

Bit 3 (Entradas RS-232). Se genera una interrupción si se produce un cambio en el estado de cualquiera de señales de entrada a la computadora definidas en el estándar RS-232.

Bits del 4 al 7. Siempre tienen un nivel lógico de cero.

2: *Registro de identificación de interrupciones.* Identifica la causas que originaron las interrupciones descritas anteriormente y les asigna una prioridad que no es programable por el usuario. En la tabla III.5.2 se muestra la forma de identificar las interrupciones; la prioridad 0 es la más alta.

Tabla III.5.2 Identificación y prioridad de las interrupciones en el 8250 UART

Bit 2	Bit 1	Bit 0	Prioridad	Causa
0	0	1	-	No hay interrupción
1	1	0	0	BREAK
1	0	0	1	RxRDY
0	1	0	2	TBE
0	0	0	3	Entradas RS-232

3: *Registro de formato de los datos.* Algunos bits que conforman este registro tienen la función de determinar las características del dato a transmitir: su *longitud*, el *número de bits de paro* y la *paridad*. También habilita el control de la condición de ruptura del enlace y adicionalmente, el último bit o DLAB, se utiliza para hacer una extensión del número de registros que pueden ser direccionados con las tres líneas de selección con que cuenta el UART.

4: *Registro de control de las salidas RS-232.* Fundamentalmente permite el acceso a las líneas de control del modem definidas en el estándar RS-232: *DTR* (de *Data Terminal Ready*) señal que indica que la terminal de datos está lista y *RTS* (de *Request To Send*), señal que hace la petición para transmitir. Los bits 2 y 3 son salidas de propósito general (GPO-1 y GPO-2), y el bit 4 habilita la interconexión temporal de entradas y salidas del UART con propósito de pruebas internas. Cuando este bit toma el valor de 1, las cuatro entradas de control RS-232 se conectan directamente a las cuatro salidas: RTS con CTS (de *Clear To Send* -señal que indica que está listo para transmitir-), DTR con DSR (de *Data Set Ready* -que indica que el modem está preparado-), GPO-2 con DCD (de *Data Carrier Detect* -detección de portadora de datos-) y GPO-1 con RI (de *Ring Indicator* -detección de llamada-). Los bits del cinco al siete se mantienen permanentemente en un nivel lógico de cero.

5: *Registro de verificación de estado de la comunicación serial.* Consta de los siguientes bits:

Bit 0 (RxRDY). Este bit toma un valor lógico de uno cuando un byte está disponible para su lectura en el búfer de recepción. Al ser leída la palabra recibida, retorna al valor lógico de cero.

Bit 1 (*Desbordamiento* en la recepción). Un uno lógico en este bit indica que un byte que se encontraba en el búfer de recepción fue sobrescrito por uno nuevo, por lo que el primero se pierde. A este tipo de error se le llama *error por desbordamiento*.

Bit 2 (Error de paridad). Este bit toma el valor de uno lógico cuando el bit de paridad del dato recibido no corresponde con el que se determinó en el registro de formato de los datos. Este bit regresa al valor de cero lógico cuando es leído.

Bit 3 (Error de trama). Si el bit de paro es incorrecto cuando se recibe un byte, este bit toma el valor de un uno lógico y cuando es leído regresa al valor lógico de cero.

Bit 4 (Detección de ruptura en el enlace o BREAK). Un uno lógico aparece en este bit cuando el receptor detecta una condición de espaciamiento más larga de lo normal. Este bit regresa al valor lógico de cero cuando es leído.

Bit 5 (Búfer de transmisión vacío o TBE). Este bit toma un valor de uno lógico cuando el UART está listo para recibir otro byte para su transmisión. Sin embargo, esto no significa que la transmisión del byte previo haya sido totalmente completada.

Bit 6 (Transmisor vacío o TxE). El bit de TxE es un uno lógico cuando se ha completado el envío del último byte a transmitir.

Bit 7 Siempre es un cero lógico.

6: *Registro del estado de las entradas RS-232*. Este registro indica el estado actual de las señales RS-232 y la diferencia existente con su condición previa.

Bits 0 al 3. Un uno lógico en estos bits significá que una señal RS-232 de entrada a la computadora ha cambiado desde la última vez que se leyó.

Bits 4 al 7. Muestran el valor actual de las señales de entrada RS-232:

Bit 4. Señal de listo para transmitir. (CTS).

Bit 5. Modem preparado (DSR).

Bit 6. Detección de llamada (RI).

Bit 7. Detección de señal portadora de datos (DCD).

7: *Registro de reserva*. Este registro no cumple ninguna función. Puede ser utilizado como un byte de memoria. Sin embargo, no está disponible en algunas versiones del 8250.

III.5.2 Interfaz con el controlador de comunicaciones seriales

Como IBM lo proyectó originalmente, los programas no deberían tener el control directo del hardware, en vez de ello, llamarían una rutina en el BIOS. Usando este método, la aplicación primero debe cargar en registros del procesador algunos valores y posteriormente debe enviar la interrupción de software adecuada que le da al BIOS el control para ejecutar las instrucciones requeridas.

Este procedimiento tiene ventajas algunas veces, pero tiende a realizar lentamente muchas funciones de la computadora.

En el panorama actual, no se puede pasar por alto la popularidad de Microsoft Windows, el cual en este momento provee una interfaz de programación, en la que el desarrollador no interactúa con el hardware directamente, sino con el manejador de dispositivos. Sin embargo este método también es lento.

La comunicación de datos a altas velocidades a través del puerto serial, requiere de rutinas que no provee el BIOS. Teniendo en cuenta que la confiabilidad del enlace es inversamente proporcional a la velocidad de transmisión, a fin de lograr el mejor rendimiento posible, resulta necesario desarrollar una interfaz con los puertos seriales, en tanto que, los programas que envían datos más allá de 9,600 bps (o 19,200 en algunas máquinas) definitivamente deben utilizar rutinas de comunicación serial realizadas por el programador de sistemas.

Bajo MS-DOS, se especifican cuatro puertos seriales en la PC, cada uno constituido por ocho direcciones consecutivas y sólo se les asignan dos interrupciones de hardware. Las combinaciones válidas de dirección inicial e interrupción se muestran en la tabla III.5.3:

Tabla III.5.3 Dirección inicial e interrupción para puertos seriales en una PC

Puerto	Dirección e interrupción
COM1	0x3f8, IRQ 4
COM2	0x2f8, IRQ 3
COM3	0x3e8, IRQ 4
COM4	0x2e8, IRQ 3

COM3 y COM4 comparten sus interrupciones con los primeros dos puertos seriales.

El monitoreo y control del UART se realiza accediendo sus registros. En el direccionamiento de puertos de la PC y debido a la construcción interna del controlador, estos registros -cada uno de ocho bits-, ocupan ocho direcciones consecutivas, a partir de la asignada al *de transmisión y recepción de datos* (cf. sección III.5.1).

Para representar estos registros del controlador de comunicaciones, en el programa definimos la estructura `serial` que está constituida por los siguientes campos de tipo `char` (ocho bits de longitud):

```
struct serial {
    char datos;           /*transmisión y recepción de datos*/
    char int_enable;     /*habilitador de interrupciones*/
    char int_id;         /*identificador de interrupciones*/
    char formato;        /*formato de comunicaciones */
    char out_control;    /*líneas de control del modem */
    char status;         /*byte de estado*/
    char i_status;       /*estado de entrada*/
    char scratch;        /*reserva*/
};
```

El objetivo consiste en *apuntar* esta estructura a la dirección inicial ocupada por un puerto serial. Considerando que en la PC se pueden tener hasta 65,536 posibles puertos diferentes, podemos generar un apuntador que sea capaz de direccionarlos aplicándole el modificador `near` (16 bits).

Consideramos suficiente que el Sistema de Monitoreo soporte únicamente los puertos correspondientes a dos interrupciones diferentes. Se seleccionaron los puertos seriales COM1 en 0x3f8 y COM2 en 0x2f8, así que con las siguientes líneas:

```
#define COM1 ((struct serial near *)0x3f8) y
#define COM2 ((struct serial near *)0x2f8)
```

se generan los apuntadores de tipo `near` COM1 y COM2, a estructuras de tipo `serial` en las direcciones 0x3f8 y 0x2f8.

Con estos apuntadores podemos direccionar los registros del UART leyendo y escribiendo bytes sobre las direcciones a las que los campos de la estructura hacen referencia, pero para determinar valores a nivel de bits, es útil crear otros elementos. Cada uno de los bits de los registros del UART está referenciado por una constante simbólica definida en el archivo de cabecera `uart.h`. Aquí se presenta el código correspondiente:

```

/*
Este archivo contiene definiciones y declaraciones que se
utilizan para el control del puerto serial
*/

/* parte baja del divisor de */
#define baud_l datos /* frecuencias */
/* parte alta del divisor de */
#define baud_h int_enable /* frecuencias */

/* Bits del registro habilitador de interrupciones */
#define I_STATUS (1 << 3) /*estado del modem */
#define I_REC_STATUS (1 << 2) /*estado de recepción */
#define I_TRANS_VACIO (1 << 1) /* no hay transmisión */
#define I_CHAR_IN (1 << 0) /* entrada de un caracter */

/* Registro de formato de datos */
/* registros del divisor */
#define F_BAUD_LATCH (1 << 7) /* de frecuencias */
#define F_NORMAL (0 << 7) /* registros normales */
#define F_BREAK (1 << 6) /* break */
#define F_NO_BREAK (0 << 6) /* no break */
#define F_SIN_PARIDAD (0 << 3) /* sin paridad */
#define F_PARIDAD_NON (1 << 3) /* paridad non */
#define F_PARIDAD_PAR (3 << 3) /* paridad par */
#define F_PARIDAD_MARCA(5 << 3) /* bit de paridad 1 */
#define F_PARIDAD_SPACE(7 << 3) /* bit de paridad 0 */
#define F_STOP1 (0 << 2) /* un bit de stop */
#define F_STOP2 (1 << 2) /* dos bits de stop */
#define F_DATA5 0 /* 5 bits de datos */
#define F_DATA6 1 /* 6 bits de datos */
#define F_DATA7 2 /* 7 bits de datos */
#define F_DATA8 3 /* 8 bits de datos */

```

```

/* Registro de control de salidas RS-232 */
#define O_LOOP      (1<<4)   /* habilita prueba local */
#define O_OUT1     (1<<3)   /* propósito general 1 */
#define O_OUT2     (1<<2)   /* propósito general 2 */
#define O_RTS      (1<<1)   /* petición de transmisión */
#define O_DTR      (1<<0)   /* terminal de datos lista */

/* Registro de verificación de la comunicación serial */
#define S_TXE      (1 << 6) /* transmisor vacío */
#define S_TBE      (1 << 5) /* búfer de trans. vacío */
#define S_BREAK    (1 << 4) /* detección de break */
#define S_FR_ERROR (1 << 3) /* error en tramado */
#define S_PARITY_ERROR (1 << 2) /* error de paridad */
#define S_OVERRUN  (1 << 1) /* desbordamiento */
#define S_RXRDY    (1 << 0) /* carácter listo */

/* Registro de verificación de las entradas RS-232 */
#define I_DCD      (1 << 7) /* detección de portadora */
#define I_RI       (1 << 6) /* indicador de llamada */
#define I_DSR      (1 << 5) /* modem listo */
#define I_CTS      (1 << 4) /* limpio para envío */
#define I_DEL_DCD  (1 << 3) /* delta DCD */
#define I_DEL_RI   (1 << 2) /* delta RI */
#define I_DEL_DSR  (1 << 1) /* delta DSR */
#define I_DEL_CTS  (1 << 0) /* delta CTS */

```

Figura III.5.2 Definiciones de bits de los registros del 8250 incluidos en el archivo `uart.h`

El siguiente paso para implantar la interfaz de comunicaciones es el conocer los elementos con que cuenta el lenguaje de desarrollo seleccionado.

Debido a que el procesador de la PC no trata los puertos como una extensión del espacio de memoria, la transferencia de información serial se realiza por medio de *direcciones asignadas a los puertos de hardware*.

La familia Intel 80X86 tiene un conjunto especial de instrucciones en ensamblador que se utilizan para comunicarse con los puertos. Dos de ellas son `OUT` para enviar una palabra e `IN` que la recibe. Turbo C proporciona las instrucciones que hacen exactamente lo que sus contrapartes en lenguaje ensamblador⁵. Estas son:

```
int inport(int puerto);
void outport(int puerto, int dato);

unsigned char inportb(int puerto);
void outportb(int puerto, int dato);
```

La función `inport` lee el `puerto` seleccionado, mientras que `outport` manda un `dato` por el mismo. Las funciones `inportb` y `outportb` son macros que realizan la transferencia de un byte, mientras que `inport` y `outport` operan con una palabra.

La lectura de un byte a los puertos de comunicaciones seriales se realiza sustituyendo el parámetro `puerto` de la sintaxis, por el apuntador de tipo `near` direccionado al campo de la estructura que representa el registro de transmisión y recepción de datos del UART. La línea de código

```
variable = inportb((int)&COMM->datos);
```

asigna a `variable` el valor leído del puerto serial referenciado por la variable `COMM`, que puede ser `COM1` o `COM2`.

El envío de información se hace de manera similar con la función de salida `outportb`. La instrucción

```
outportb((int)&COMM->datos, caracter);
```

envía al registro de entrada y salida serial de datos del puerto direccionado por `COMM` el byte `caracter`.

⁵En otros compiladores del lenguaje de programación C, se tienen funciones diferentes que cumplen con el mismo propósito.

III.5.3 Rutinas básicas de comunicación serial

En la sección precedente se describió la forma en que los bytes se transfieren por el puerto de comunicaciones seleccionado. Sin embargo para que los caracteres tengan un significado, es necesario agruparlos en unidades más grandes. En la implantación de un protocolo de comunicaciones, el siguiente paso es desarrollar las funciones necesarias para el envío y la recepción de datos agrupados como cadenas de caracteres.

A fin de desarrollar una función para el envío consecutivo de bytes, es necesario conocer los elementos relacionados a la etapa de transmisión del UART. Esta sección consiste principalmente de un registro de corrimiento (de entrada en paralelo y salida serie) llamado registro de transmisión, la lógica de control necesaria para cargar un byte en ese registro y un búfer de transmisión en el cual se forma una pequeña cola que permite que los datos puedan ser almacenados temporalmente antes de su serialización.

Cuando un byte del búfer se mueve al registro de transmisión se prende la bandera de *Transmitter Buffer Empty* (TBE), lo cual señala que el UART puede aceptar otro byte. Cuando el registro de transmisión ha finalizado el envío del último bit pendiente, se levanta la bandera de *Transmitter Empty* (TxE). Estos elementos son suficientes para escribir la rutina de transmisión de datos.

La función que realiza la transmisión serial de grupos de caracteres en el Sistema de Monitoreo es:

```
void saca(unsigned char arreglo [ ]);
```

Esta función recibe como parámetro la cadena de caracteres a transmitir, delimitada por un terminador nulo. Antes enviar cada dato, la función espera a que el búfer del transmisor se vacíe revisando el valor lógico del bit `S_TBE` del campo `status`, que corresponde a la bandera TBE del *registro de verificación del estado de la comunicación serial*:

```
do{}while(!inportb((int)&COMM->status) & S_TBE);
```

Después del envío de un byte, se deja un retardo intercarácter y continúa con el siguiente hasta que transmite toda la cadena recibida como parámetro. Finalmente, aguarda a que el registro de transmisión quede vacío por medio de la instrucción:

```
do{}while(!(inportb((int)&COMM->status) & S_TXE));
```

El código completo de la función `saca()` se presenta en la figura III.5.3.

```
#include <dos.h>
#include "uart.h"
#include "funcion.h"

extern struct serial near *COMM;

/*
Esta función envía una cadena de caracteres por el
puerto de comunicaciones
*/

void saca( unsigned char arreglo [] )
{
    unsigned short int i;

    delay( 0 );

    for( i=0; arreglo[i] != '\0'; i++ )
    {
        do{}
        while(!(inportb((int)&COMM->status) & S_TBE));

        /* Retardo intercarácter */
        delay( 70 );
        outportb( (int)&COMM->datos, arreglo [i] );
    }
    do{}
    while(!(inportb((int)&COMM->status) & S_TXE));
}
```

Figura III.5.3 Listado de la función que transmite información a través del puerto serial

En contraparte, para la recepción de información, otra sección del UART constantemente monitorea la línea de comunicación serial esperando por un bit de inicio. Una vez que éste se detecta, los siguientes bits entran a un registro de corrimiento hasta conformar un byte. Este pasa a un búfer y se levanta la bandera de *Data Ready* (RxRDY) indicando con ello que existe un dato listo para su lectura.

En la computadora central, la llegada de datos se detecta realizando ciclos de poleo a la bandera RxRDY y leyendo los datos del puerto hasta completar una cadena de caracteres. Este proceso de recepción se diseñó de tal manera que fuera prácticamente imposible que el programa se quedara bloqueado durante la espera a la ocurrencia de algún evento.

En el Sistema de Monitoreo, la función

```
int toma_lect(unsigned char lectura [ ]);
```

recibe una cadena de caracteres transmitida por las computadoras en los nodos remotos y regresa su estado final al programa que la invocó. La figura III.5.4 muestra el código completo de la función `toma_lect()`:

```
#include <dos.h>
#include <string.h>
#include "uart.h"
#include "bande.h"
#include "funcion.h"

extern struct serial near *COMM;

/*
Esta función obtiene una lectura de la estación remota.
*/

int toma_lect( unsigned char lectura [ ] )
{
    clock_t start_l, start_c, end;
    unsigned char modem_estado, *p_lectura;
    unsigned short int    c_recibidos = 0, error = FALSE,
                        sale = FALSE, sale_l = FALSE;
    int lect_error = FALSE;
```

```

p_lectura = &lectura [0];

start_l = clock( );

do(
  modem_estado = inportb((int)&COMM->status );
  if( modem_estado & S_RXRDY )
  {
    do(
      if( modem_estado & S_RXRDY )
      {
        *(p_lectura++) = inportb((int)&COMM->datos );
        c_recibidos++;

        if( modem_estado &
            (S_BREAK|S_FR_ERROR|S_PARITY_ERROR|S_OVERRUN) )
          error = TRUE;

        start_c = clock( );
      }
      else
      {
        end = clock( );
        if( ((float)(end-start_c)/CLK_TCK)>0.5 )
          sale = TRUE;
      }
      modem_estado = inportb((int)&COMM->status );
    }while( !error && c_recibidos<59 && !sale );
  }
  else
  {
    end = clock( );
    if( ((end-start_l)/CLK_TCK)>1 )
      sale_l = TRUE;
  }
}while( c_recibidos<59 && !error && !sale && !sale_l );

*p_lectura = '\0';

```

```

if( error || sale_1 )
    return( 2 );
else if( strstr( lectura, "\r\n\x4\r\n" ) )
    return( 3 );
else
{
    lect_error = ver_sum( lectura );
    if( !lect_error )
        return( 0 );
    else
        return( 1 );
}
}

```

Figura III.5.4 Código de programa de la función toma_lect

La función inicia fijando un apuntador al primer elemento del vector que se utilizará para almacenar la cadena de caracteres.

A continuación y con las instrucciones:

```

start_1 = clock( );

do{
    modem_estado = inportb((int)&COMM->status );
    if( modem_estado & S_RXRDY )
    {
        /* Recibe la cadena de caracteres */
        .
        .
        .
    }
    else
    {
        end = clock( );
        if( ((end-start_1)/CLK_TCK)>1 )
            sale_1 = TRUE;
    }
}while( c_recibidos<59 && !error && !sale && !sale_1 );

```

se explica el algoritmo utilizado por el Sistema de Monitoreo para tomar en cuenta tiempos de expiración en el protocolo de comunicaciones.

A través de la función de biblioteca `clock()`, en la variable `start_1` se registra la hora en que la rutina comenzó a ejecutarse y se establece un límite en la espera de la llegada del primer byte. El tiempo transcurrido se obtiene comparando `start_1` contra los valores que va tomando la variable `end` al invocar sucesivamente a la función `clock()`. Si después de un segundo no se ha recibido el primer carácter, la función finaliza. La llegada de un byte se detecta guardando en la variable `modem_estado` el contenido del registro de verificación de la comunicación (campo `status`) y determinando el valor de su bit cero (`S_RXRDY`).

Una vez leído el primer byte, el programa espera la llegada de los siguientes hasta completar una cadena que se almacenará en el vector `lectura`. Para cada uno de ellos y de manera similar a como se hizo con el primer byte, se establece un límite de 0.5 segundos en su espera.

Después de la lectura de cada carácter, se verifica que no se hayan detectado errores, utilizando las máscaras definidas en `uart.h`: `S_BREAK`, `S_FR_ERROR`, `S_PARITY_ERROR` y `S_OVERRUN`.

Una vez recibidos todos los bytes de la cadena, se verifica que sea válida por medio de la función

```
int ver_sum(unsigned char [ ]);
```

Los posibles valores que `toma_lect()` puede regresar son:

- 0 La cadena de caracteres fue recibida correctamente y aprobada por el proceso de validación.
- 1 Durante la recepción se perdieron algunos caracteres, quedando una cadena incompleta.

- 2 Al ser monitoreado el *registro de verificación de estado de la comunicación serial* del UART, se detectó un error que pudo ser de paridad, trama, desbordamiento o una condición de ruptura del enlace. Este valor también se puede originar por una interrupción en el proceso de transmisión, siendo detectado por un tiempo anormal de espaciado intercarácter o entre una lectura y la siguiente.
- 3 La computadora remota ha finalizado el envío de información.

III.5.4 Diseño del protocolo de comunicaciones

El protocolo de comunicaciones que utilice el Sistema de Monitoreo debe contemplar las siguientes características:

- Ser lo más simple posible, para minimizar el tamaño en el código del programa y el tiempo de transmisión.
- El canal de comunicaciones no es 100% confiable, por lo que debe realizar algún proceso para garantizar la integridad de los datos recibidos.
- Además debe ser eficiente para facilitar la transmisión de la mayor cantidad información, con el menor número de caracteres de control.

Es imposible tener un protocolo que cumpla plenamente todas estas características, por lo que habrá que implantar uno que satisfaga las más importantes. Por ejemplo, se puede hacer eficiente comprimiendo la información, sin embargo, esto lo hace más complejo. Es aquí donde hay que decidir las cualidades que satisfacen mejor nuestras necesidades.

III.5.4.1 Estructura del paquete de información

En el protocolo del Sistema de Monitoreo la unidad básica de transferencia es el *paquete de información*, una agrupación de varios *campos*. Uno de ellos contiene los datos y los demás, la información requerida por el receptor para identificar el paquete y verificar que esté libre de error.

Decidimos formar un paquete de información por los siguientes elementos:

- El carácter SOH como inicializador de paquete.
- Un identificador de paquete formado por la fecha y hora de cuándo se registró la información por los sensores.

- Tres bytes que contienen las lecturas a los sensores y dos caracteres de control delimitando su principio y fin (STX y ETX, respectivamente) .
- Una suma de verificación o *checksum* que permita saber si la lectura recibida es válida.

Con estos campos, se obtienen paquetes como el que se muestra en la siguiente figura.

SOH	Identificador de lectura (24)	STX	Información obtenida por los sensores (3)	ETX	checksum
-----	----------------------------------	-----	---	-----	----------

Figura III.5.5 Estructura de información transmitida en el Sistema de Monitoreo

El procedimiento para obtener la suma de verificación del paquete de información se muestra en la figura III.5.6:

```
#include "funcion.h"

/*
   Esta función obtiene el byte de checksum de cada
   lectura desde la computadora transmisora
*/

void ver_sum_remota( unsigned char arre [ ],
                    unsigned char *suma )
{
    unsigned short int i;
    *suma = 0;

    for( i = 0; i < 30; i++ )
        *suma = *suma + arre [ i ];
    *suma = ~( *suma );
    *suma += 0x1;
}
```

Figura III.5.6 Programa que genera la suma de verificación en la computadora remota

Las instrucciones

```
for( i = 0; i < 30; i++ )
    *suma = *suma + arre [ i ];
```

suman todos los caracteres del paquete y el resultado se va acumulando en la variable direccionada por el apuntador *suma*. El bit de *carry* se desprecia en cada operación intermedia de manera que, en el resultado se tengan nuevamente ocho bits.

Al finalizar la adición de todos los bytes del paquete, con la instrucción

```
*suma = ~( *suma );    se obtiene el complemento a uno y
*suma += 0x1;          obtiene el complemento a dos.
```

El valor así obtenido en **suma* constituye el byte de la suma de verificación.

III.5.4.2 Secuencia de transmisión-recepción

Después de que el modem ha *creado* el canal de comunicaciones, el nodo central indica que está listo para la recepción de información mandando un carácter ENQ (ASCII 5) al sistema remoto. Durante la transmisión, los datos son "empaquetados", rodeándolos con los caracteres de control necesarios. Después del envío de un paquete, el transmisor espera que el otro extremo le indique su recepción. El protocolo empleado es del tipo de *requerimiento de repetición automática* en la cual, cuando el receptor detecta un error en la información recibida, solicita su retransmisión.

En el nodo central se calcula la suma de verificación sobre los datos recibidos y se compara contra el byte de *checksum* que se encuentra en el paquete. Si el valor local concuerda con el contenido en el paquete recibido, el receptor lo indica enviando un ACK (ASCII 6), en otro caso, transmite un NAK (ASCII 21)⁶.

⁶ ACK, ENQ y NAK son mnemónicos de las palabras inglesas **acknowledge**, **enquire** y **no acknowledge** respectivamente. Sus funciones en el Sistema de Monitoreo siguen la definición oficial del estándar ANSI X3.4.

Después de la recepción de un carácter ACK, el nodo remoto envía el siguiente paquete de información, o bien, si recibe un NAK vuelve a transmitir el último. El número máximo de intentos está limitado por la constante simbólica `MAX_NO_INTENTOS` definida en el archivo cabecera `sist.h`. Cuando se alcanza este valor y no se ha logrado recibir correctamente un paquete, se transmite un carácter ACK para que la computadora en el nodo remoto ignore el error y envíe el siguiente. El proceso de transmisión continúa en esta forma hasta que finaliza el envío de todos los paquetes por el nodo remoto. El diagrama de la figura III.5.7 muestra este proceso simplificado.

Después de que el modem *cierra* el canal de comunicaciones, el total de intentos realizados para recibir el conjunto de paquetes y la cantidad recibida correctamente, se toman en cuenta para determinar el estado del enlace.

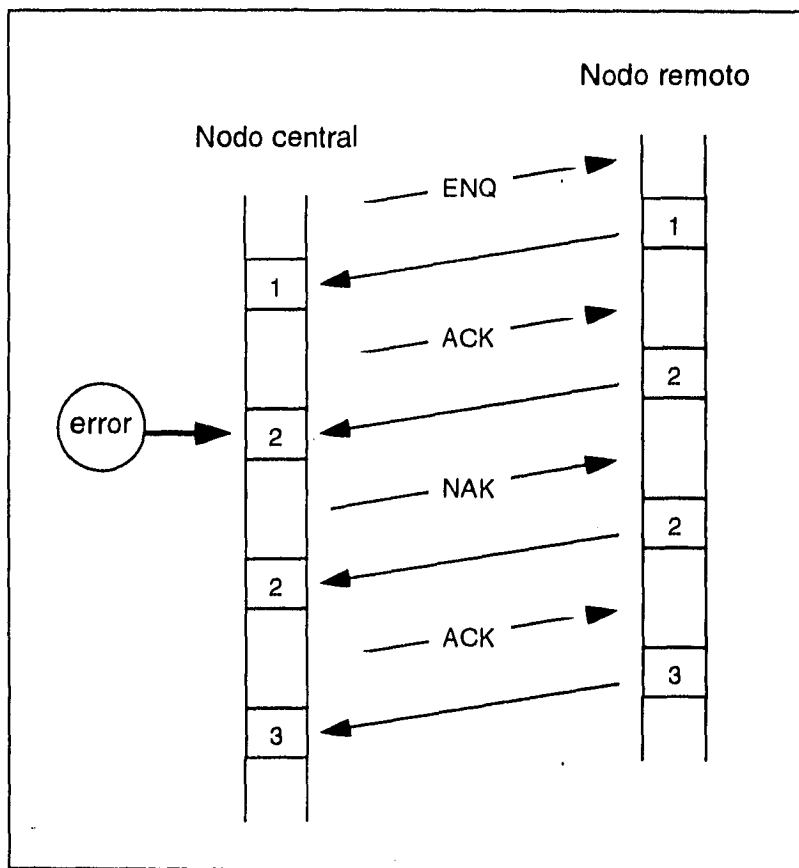


Figura III.5.7 Protocolo de comunicaciones

III.6 Visualización de la información

El Sistema de Monitoreo centraliza en una PC ubicada en la DGSCA la información obtenida de todos los nodos de conmutación, haciendo que sean posibles:

- *Visualización del estado de la Red de Telefonía Digital.*
Es posible conocer en forma *instantánea* el estado en que se encuentran los locales de los equipos de conmutación remotos, desplegando en pantalla las lecturas obtenidas por los sensores instalados en ellos. Además, se registra *especialmente* la última condición anormal que se haya detectado en los locales señalizándola en pantalla y almacenando en memoria la lectura que la represente. La indicación de una condición anormal permanece hasta que se reestablece por el usuario, con lo cual el Sistema de Monitoreo asegura su reconocimiento.
- *Generación de archivos históricos.*
Una vez que se ha completado un ciclo de llamadas a los locales de conmutación registrados en el Sistema, la computadora central transfiere la información alojada en su memoria hacia el disco duro; creando de esta forma archivos de lecturas ordenadas cronológicamente, con lo cual se tiene la posibilidad de consultar y procesar esta información posteriormente.

Los detalles para lograr desplegar en pantalla el estado instantáneo de la Red de Telefonía serán descritos en el siguiente punto, mientras que el proceso de la generación de archivos históricos se explica en la sección III.7.

III.6.1 Software para la visualización del estado de la Red

El Sistema de Monitoreo utiliza estructuras y matrices en memoria para contener temporalmente las lecturas de todos los nodos de conmutación y prepararlas para su interpretación, despliegue en pantalla y almacenamiento en disco duro.

La visualización del estado de la Red se efectúa utilizando el arreglo bidimensional `matriz_panta` y las estructuras `despliegue` y `libera`. Todas ellas definidas en el archivo `matrices.h`.

La matriz_panta, se utiliza para almacenar la información requerida para el despliegue de cada nodo en pantalla. Consta de MAX_NUM_NODOS renglones por COLUMNAS columnas, ambas constantes simbólicas están definidas en el archivo cabecera matrices.h. El parámetro MAX_NUM_NODOS tiene el valor de 39 y corresponde al número máximo de nodos que pueden existir dentro del Sistema.

Cada renglón de esta matriz está formado como se muestra en la figura III.6.1:

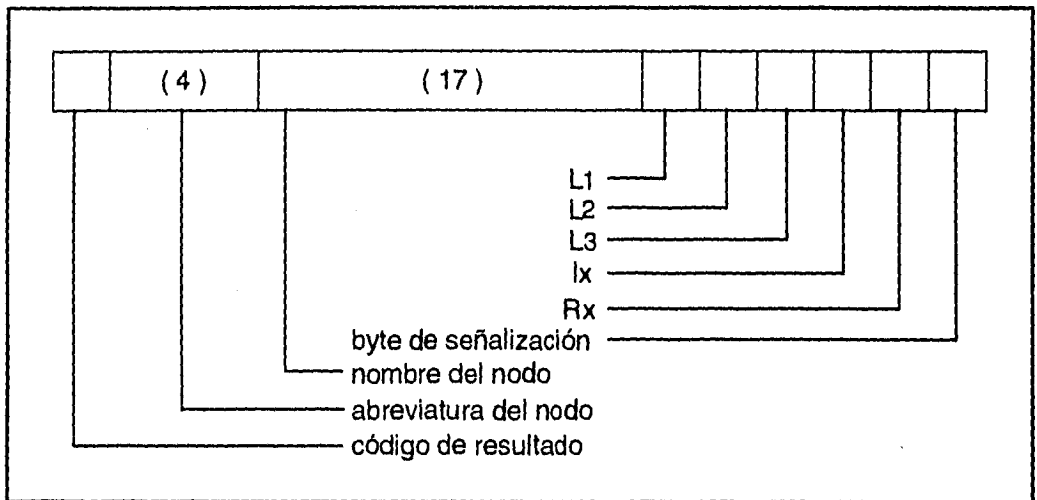


Figura III.6.1 Estructura de un renglón de matriz_panta

Del conjunto de lecturas recibidas de cada nodo remoto, la última se registra en esta matriz. La abreviatura y nombre del nodo son leídos del archivo de inicialización NODOS.DAT y se retienen en memoria mientras el sistema permanezca en operación. Los bytes L1, L2 y L3 contienen la lectura registrada por los sensores de cada nodo, mientras que Lx y Rx se determinan al concluir el enlace con cada computadora remota.

Las definiciones de las estructuras requeridas para el despliegue son:

```
struct despliegue{
    unsigned short int mat_num;
    unsigned short int mat_priori;
};

struct libera{
    time_t fecha;
    unsigned char tipo_alarma;
    unsigned char alarma_bat;
    unsigned char alarma_temp;
    unsigned char alarma_resto;
    unsigned char alarma_txx;
};
```

De acuerdo con el contenido de la información transmitida por las computadoras remotas y el resultado del enlace, el Sistema de Monitoreo asigna una prioridad de atención a cada nodo de conmutación registrado en `matriz_panta`. En la estructura `despliegue`, se almacena esta prioridad, mientras que `libera` registra la última condición anormal.

Asignación de prioridad de atención a las condiciones en el local del conmutador

Los siguientes valores de umbral definidos en el archivo cabecera `rangos.h` permiten clasificar la gravedad de la situación:

```
#define TEMP_OK          20.00
#define TEMP_MAX        23.00
#define VOLT_OK         50.00
#define VOLT_MIN        48.00
```

La atención requerida por parte del usuario para cada condición detectada por los sensores en los nodos de conmutación, se resume en la tablas III.6.1 y III.6.2

Tabla III.6.1 Prioridad de atención para diferentes condiciones en los nodos de la Red

Condición sensada	Estado	Prioridad de atención
temperatura	mayor o igual a TEMP_MAX	máxima
	mayor a TEMP_OK, pero menor a TEMP_MAX	mediana
	menor o igual a TEMP_OK	ninguna
voltaje de baterías	menor o igual a VOLT_MIN	máxima
	mayor a VOLT_MIN, pero menor a VOLT_OK	mediana
	mayor o igual a VOLT_OK	ninguna
relación Rx/lx	cero por ciento	máxima
	cien por ciento	ninguna

Tabla III.6.2 Prioridad de atención de acuerdo a posibles combinaciones del grupo restante de condiciones sensadas por el Sistema de Monitoreo

Alarma Mayor	Alarma Menor	Falla de energía eléctrica	Prioridad
no	no	no	ninguna
no	no	si	máxima
no	si	no	mediana
no	si	si	máxima
si	no	no	máxima
si	no	si	máxima
si	si	no	máxima
si	si	si	máxima

De acuerdo a las tablas mostradas anteriormente y al resultado del enlace de comunicaciones, el campo `mat_priori` de la estructura despliegue puede tomar los siguientes valores:

Tabla III.6.3 Posibles valores asignados al campo `mat_priori` de la estructura despliegue

Mnemónico	Valor numérico	Condición
N_ALRM	0	Se detectó alguna alarma que merece prioridad máxima de atención
N_ALERT	1	Una alarma con prioridad mediana de atención fue detectada por el Sistema
N_NOLECT	2	No se recibió lectura válida al establecer el enlace con el nodo remoto (relación Rx/Ix=0%)
N_NOCOM	3	No se logró establecer comunicación con el nodo remoto
N_NORM	4	No se detectó alarma alguna en la última lectura recibida
N_NUEVO	5	El nodo había sido dado de baja y recientemente fue rehabilitado para ser monitoreado por el Sistema
N_BAJA	6	El nodo está dado de baja

El despliegue de la información en la pantalla de la PC central, se hace realizando el ordenamiento ascendente del campo `mat_priori` en primer lugar y después, de los nodos; de tal manera que los que se presenten al principio, sean los que necesitan una atención mayor. Si existe más de un nodo que tenga la misma prioridad, se desplegarán según el orden en que hayan sido leídos del archivo de inicialización NODOS.DAT.

Asignación de prioridad de atención de acuerdo al resultado del enlace

El campo de *código de resultado* en *matriz_panta* sirve para identificar tanto el estado final del enlace con cada computadora remota así como para saber si el nodo está habilitado para ser monitoreado o no (ver la tabla III.6.4).

Tabla III.6.4 Relación entre el campo *mat_priori* de la estructura despliegue y el código de resultado de la *matriz_panta*

Código de resultado	Condición	Valor en <i>mat_priori</i>
'0'	Nodo dado de baja	N_BAJA
'1'	Nodo habilitado para monitoreo	N_NUEVO si el byte de L3 es igual a 0x0f ⁷ . N_ALRM si se requiere prioridad máxima de atención N_ALERT si se requiere una prioridad mediana de atención N_NORM si se estableció el enlace correctamente y no se detectó ninguna alarma
'2'	No se pudo establecer el enlace de comunicaciones	N_NOCOM
'3'	No se recibió lectura válida al establecer el enlace de comunicaciones	N_NOLECT

⁷ De esta manera se sabe que no es una lectura recibida por la computadora central. No es posible tener un valor de 0x0f en este byte, porque al obtenerlo en el nodo remoto, se hace una operación lógica OR con 0xf0; de tal manera que cuando exista la ausencia de alarmas en este grupo de bits, no se transmitan nulos, como se explicó anteriormente en el punto III.4 (*Adquisición de los datos*)

Registro de la última condición anormal

Si es detectada alguna condición anormal al determinar la prioridad correspondiente a la última lectura recibida de cada nodo, se identifica en la matriz_panta por medio de un '1' en el *byte de señalización* y al mismo tiempo, se almacena la lectura en la estructura libera, si no estaba ocupado anteriormente el campo correspondiente a ese nodo. De esta manera, el Sistema siempre mantendrá el registro de la última condición anormal mientras que no haya sido notificada al usuario.

Por otro lado, en el campo fecha de la estructura libera se almacenan el número de segundos transcurridos desde el primero de enero de 1993 a las 00:00:00 horas, hasta el momento en que se generó la lectura que registró la condición anormal o la hora en que no se pudo establecer el enlace con la PC remota.

El campo tipo_alarma de la misma estructura puede tomar los valores de las siguientes definiciones incluidas en el archivo masks.h:

```
#define MASK_NO_COMUNI      0x12
#define MASK_NO_LECT_VAL    0x11
#define OTRA                 0x80
```

si su valor es MASK_NO_COMUNI, todos los campos de la estructura toman el valor de cero, excepto el de alarma_txx, en el que se registra el modo de marcación en el que está trabajando el Sistema de Monitoreo. Si es MASK_NO_LECT_VAL, el único campo que guarda algún valor es el de alarma_txx, indicando el número de intentos que se realizaron sin éxito para recibir alguna lectura. Por último, si el valor en tipo_alarma es OTRA, los bits L1, L2 y L3 de la lectura recibida son transferidos respectivamente a los campos alarma_bat, alarma_temp y alarma_resto de la estructura, dejando únicamente con un valor de cero el campo alarma_txx.

Cuando el usuario quiere conocer la última condición anormal en algún nodo remoto, el Sistema determina la forma en que será desplegada por medio de esta estructura. Una vez que es liberada la alarma, es decir, reconocida por el usuario, el Sistema asigna un valor de cero a todos los campos de la estructura.

III.7 Almacenamiento de la información en la computadora central

El esquema de almacenamiento al disco de la computadora central, tiene el objetivo fundamental de organizar la información que se produce como resultado de la operación del sistema. El diseño considera también el propósito de disminuir el tiempo de acceso a los datos durante la generación de reportes y adicionalmente, permite el control del espacio reservado para la aplicación en forma tan flexible y sencilla, que la modificación de los parámetros más importantes puede realizarse por el propio usuario.

Fundamentalmente hay dos tipos de archivos utilizados por el sistema; unos son los necesarios para la administración de la información, que se localizan en C:\MONITOR\SYS y otros, los archivos de datos almacenados en la trayectoria C:\MONITOR\DAT. Su propósito y estructura se explican inicialmente en forma independiente y posteriormente se describe su interrelación con las demás componentes, en tanto que sus características principales se muestran en la tabla III.7.1.

Elemento	Directorio	Nombre	Atributos
Contenedor	C:\MONITOR\DAT\	~ DATA ~. _	read only, hidden
Archivo de tamaño	C:\MONITOR\SYS\	~ SIZE ~. _	read only, hidden
Directorio	C:\MONITOR\SYS\	DIR.DAT	-
Archivo de lecturas	C:\MONITOR\DAT\	hhddmmaa.DAT	-
Archivo de número de registros	C:\MONITOR\SYS\	NUMREG.DAT	-
Archivo de nodos	C:\MONITOR\SYS\	NODOS.DAT	-

Tabla III.7.1 Características de los archivos del sistema

III.7.1 El contenedor de datos y el archivo de su tamaño

Este elemento es un área de tamaño dinámico que permite reservar el espacio necesario para la operación del sistema en el disco duro de la computadora central. Asimismo permite al usuario el monitoreo y control del espacio asignado a la información del estado de los nodos de la red.

El archivo contenedor se genera automáticamente cuando el sistema se instala por primera vez, se inicializa con una longitud definida por el programa de instalación y que posteriormente el usuario puede modificar. Al mismo tiempo, se crea otro archivo (archivo de tamaño) que indica el tamaño con el cual el contenedor se creó originalmente. Las características de ambos archivos se pueden consultar en la tabla III.7.1; observe que a fin de buscar mayor seguridad en el sistema, el contenedor y el archivo de tamaño se crean con nombres y atributos muy difíciles de manipular a través del sistema operativo.

Cuando se almacena información, el sistema automáticamente disminuye el tamaño del contenedor y cuando los datos se eliminan, se incrementa su longitud, de modo que siempre se cumple la relación de que *el tamaño instantáneo del contenedor más el espacio ocupado por los archivos de lecturas es una constante igual a la cantidad en megabytes declarada en el archivo de tamaño*.

En la figura III.7.1 se muestra un ejemplo ilustrativo del comportamiento de este elemento durante la operación del sistema. En él, el archivo de tamaño contiene el valor de 10, lo cual indica que el área que se asigna inicialmente al contenedor es de 10 Mbytes, como se muestra en el primer caso (a). Cuando el sistema se encuentra funcionando (representado con el bloque etiquetado como SisMon), se almacenan datos a disco, de modo que el espacio ocupado por lecturas aumenta a 4 Mbytes y el del contenedor disminuye a 6, como se observa en el segundo caso (b). Posteriormente en el tercer caso (c), el sistema elimina los datos más antiguos, por lo que se libera espacio y ahora el área ocupada por la información es de sólo 1 Mbyte dejando 9 libres para el contenedor.

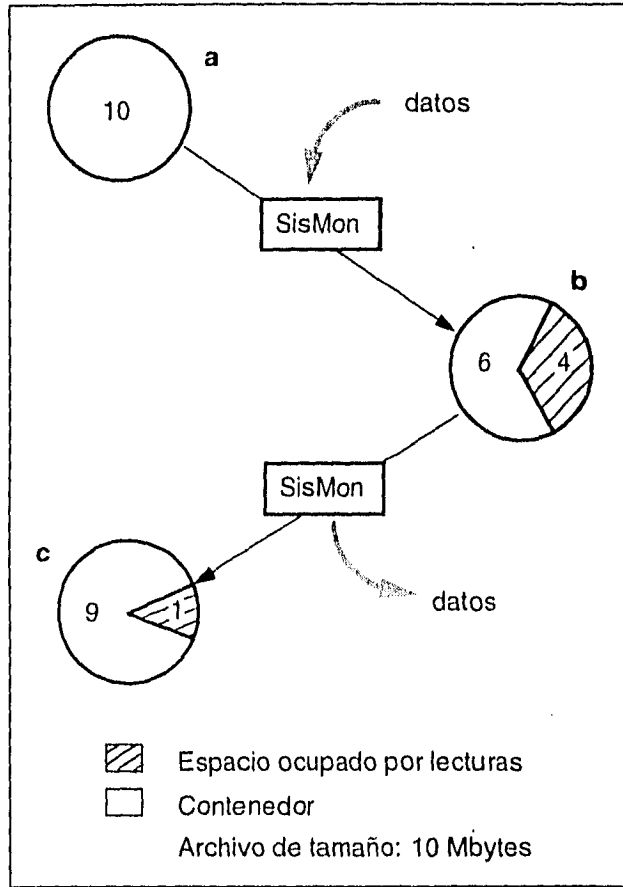


Figura III.7.1 Ejemplo ilustrativo del funcionamiento del contenedor

Las ventajas más importantes de la implantación de este concepto son las siguientes:

- Conociendo el tamaño instantáneo del contenedor y el valor almacenado en el archivo de tamaño, en cualquier instante se puede determinar el porcentaje de utilización del área asignada a los datos.
- Debido a que el contenedor aparta el espacio declarado en el archivo de tamaño desde su creación, se garantiza que siempre hay el necesario para almacenar los datos en el disco.

III.7.2 El directorio

El directorio es uno de los elementos más importantes porque indica los archivos de lecturas que se han generado, y registra los nodos dados de alta durante el monitoreo de la red.

Las características de este archivo se pueden consultar en la tabla III.7.1. Cada registro del directorio tiene la estructura descrita a continuación y mostrada en la figura III.7.2.

- a) En el primer campo se indica el nombre físico de un archivo de lecturas; este nombre se forma con la hora, día, mes y año de la lectura más antigua que contenga y finaliza con la extensión .DAT.
- b) Después del nombre físico, se agrega una bandera de estado a la que se le asigna el valor de '0' cuando el registro apunta a un archivo que ya se ha cerrado. Esta situación generalmente ocurre cuando se ha alcanzado el límite establecido en el número de lecturas que cada archivo puede contener. O bien, la bandera indica que el archivo puede contener más lecturas tomando el valor de '1', como se muestra en el ejemplo de la figura III.7.2. A continuación se inserta un carácter US (*Unit Separator*) que finaliza esta sección.
- c) La longitud de un registro en el directorio es variable ya que se añade un campo para cada uno de los nodos remotos dados de alta en ese momento. Se escribe en un espacio de cuatro caracteres la abreviatura de su nombre, y en otro de siete posiciones se almacena el desplazamiento, en número de registros que se tiene que recorrer, desde el inicio del archivo de lecturas, para llegar a la primera lectura de ese nodo. Cuando aún no se tiene información disponible para un nodo en particular, se escribe un -1 en este campo.

La información de cada uno de los nodos se separa con un carácter de GS (*Group Separator*). Excepto para el último, en cuyo caso en lugar de un carácter GS se agrega un RS (*Record Separator*) el cual indica el fin del registro.

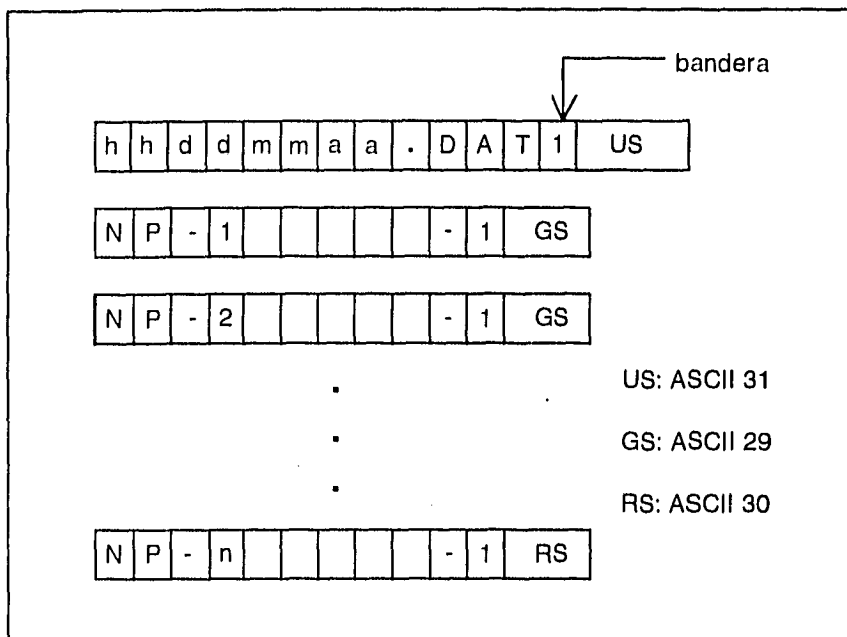


Figura III.7.2 Ejemplo de la estructura de un registro del directorio

En la figura III.7.3 se muestra un ejemplo del archivo directorio, en él se observa que se han generado tres archivos de lecturas durante el funcionamiento del sistema. Del directorio se pueden obtener algunos otros datos de estos archivos tales como la fecha y hora de la información que contienen; en este caso los archivos contienen lecturas de las 12, 14 y 17 horas del 15 de septiembre de 1994 y únicamente el último de ellos tiene capacidad para recibir más información como lo indica el estado de su bandera en 1. Al igual que los dos primeros, este último archivo contiene información de los nodos NP-1, NP-2 y NP-3, y dentro de él, la primera lectura del nodo NP-1 se encuentra al inicio, la primera del nodo NP-2 tres registros después, mientras que la primera lectura del nodo NP-3 se encuentra a seis registros del inicio.

12150994 .DAT0▼NP-1	0↔NP-2	2↔NP-3	4▲
14150994 .DAT0▼NP-1	0↔NP-2	4↔NP-3	7▲
17150994 .DAT1▼NP-1	0↔NP-2	3↔NP-3	6▲

Figura III.7.3 Listado de un archivo directorio

III.7.3 Los archivos de lecturas

La información del estado de los nodos remotos se almacena en archivos de lecturas ordenadas por tiempo, que en condiciones normales de operación contienen al menos 1200 registros (parámetro NUM_MAX_REG) lo cual da un tamaño aproximado de 64kbytes por archivo. La estructura del registro como se muestra en la figura III.7.4, permite establecer relaciones entre los registros a través de *listas ligadas*, con lo cual se obtiene un acceso eficiente a la información durante los posteriores procesos de búsqueda.

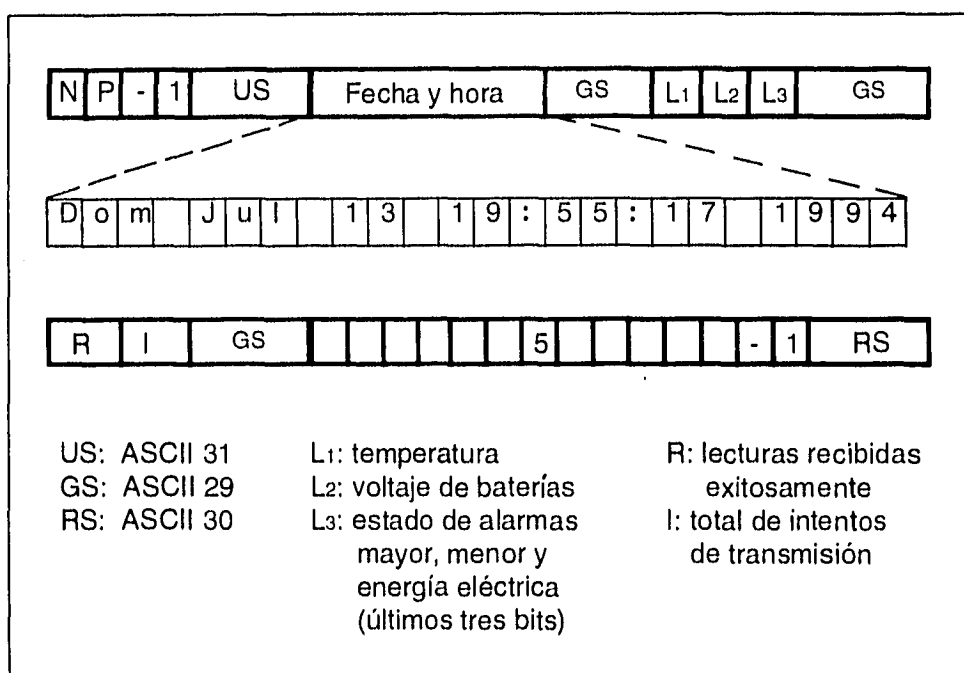


Figura III.7.4 Ejemplo ilustrativo de la estructura de una lectura

Este diseño se explica a continuación:

- a) La abreviatura del nombre del nodo al que corresponde el registro constituye el inicio de la estructura. Esta sección termina con un carácter US.

- b) En el siguiente campo se encuentra la fecha y hora en que se efectuó la lectura de las condiciones del local del conmutador; un carácter GS delimita esta sección.
- c) Los siguientes tres bytes contienen las lecturas a los sensores, los dos primeros son las palabras de ocho bits obtenidas de los convertidores analógico-digitales en los nodos remotos, el primero de ellos corresponde al sensor de temperatura, y el segundo al sensor de baterías; el último de los tres bytes contiene en sus tres bits menos significativos el estado de las alarmas mayor y menor, y de la alimentación de energía eléctrica de AC respectivamente; un carácter GS indica el fin de este campo.
- d) En los siguientes dos bytes se almacenan el número de lecturas recibidas exitosamente y total de intentos de transmisión respectivamente. Estos datos se utilizan posteriormente para calcular la tasa de lecturas recibidas sin error, lo cual permite conocer el estado del canal de comunicaciones; este método resulta más eficiente que almacenar la tasa como un número de punto flotante. La sección finaliza con un carácter GS.
- e) Tomando como referencia la posición del registro, a continuación se escriben en dos campos de siete posiciones, los desplazamientos en número de registros hacia la siguiente y hacia la anterior lectura de ese mismo nodo. Cuando no se conoce la existencia de otro registro relacionado, se escribe un -1 en el campo correspondiente. En el ejemplo mostrado en la figura III.7.4 la siguiente lectura del nodo NP-1 está localizada cinco registros hacia adelante, mientras que no hay una lectura anterior registrada en ese archivo para ese mismo nodo (indicado con -1). Esta sección es parte importante del esquema de acceso a la información, ya que estos desplazamientos se utilizan posteriormente en el módulo de consultas para encontrar rápidamente las lecturas de un mismo nodo. El registro finaliza con un carácter RS.

En la figura III.7.5 se muestra el listado de un archivo de lecturas, en él se pueden observar las características antes mencionadas.

III.7.5 Archivo de nodos

El archivo de nodos contiene todos los nodos remotos que se han registrado e indica los que están dados de alta para monitorearse. Bajo el control del programa, este archivo es modificable por el usuario para incorporar otros nodos al sistema, cambiar la extensión telefónica a los ya registrados o suspender el monitoreo de alguno de ellos. Tiene un límite máximo de 39 registros y cuando el sistema lo crea, tiene ya predefinidos la abreviatura y el nombre de los 27 nodos que actualmente conforman la Red de Telefonía Digital de la UNAM, dejando 12 registros para futuras expansiones de la red. Cada registro de este archivo tiene la estructura mostrada en la figura III.7.6 y sus características se pueden consultar en la tabla III.7.1.

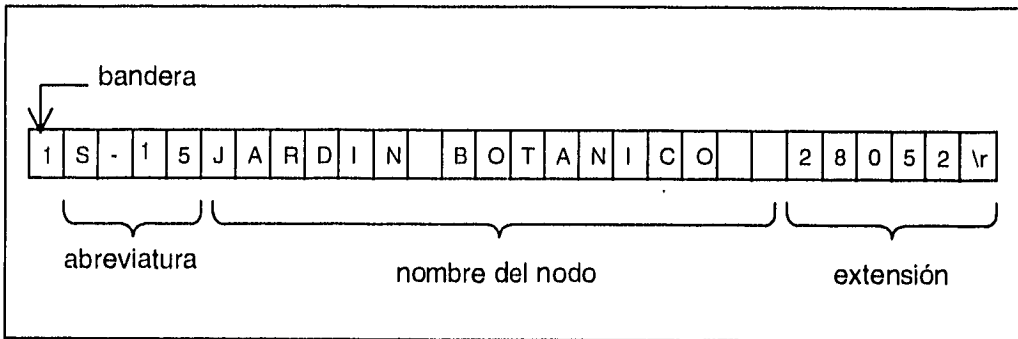


Figura III.7.6 Ejemplo ilustrativo de la estructura de un registro del archivo NODOS.DAT

- La primera posición en el registro está ocupada por una bandera que indica con un '0' que ese nodo no se considera para el proceso de monitoreo y con el valor de '1' el caso contrario.
- El segundo campo de cuatro caracteres, contiene la abreviatura del nombre del nodo. El siguiente campo es el nombre del nodo en 17 posiciones.
- Al final se escribe la extensión telefónica delimitada por un carácter '\r' utilizado para el control del modem.

III.7.6 El proceso de almacenamiento de la información

Durante un ciclo de llamadas a todos y cada uno de los nodos dados de alta en el sistema, la computadora central va reteniendo en el arreglo `matriz_vuelta` la información de las condiciones de los locales de los conmutadores. Al final del ciclo, se efectúa el almacenamiento de la información acumulada para su posterior consulta. En esta etapa, se efectúan algunos procedimientos que tienen el propósito de ordenar y organizar la información antes de su escritura al disco duro. Esta serie de procesos se pueden distinguir en el código de la función `guarda_a_disco` mostrada en la figura III.7.7:

```
#include <string.h>
#include <stdlib.h>

#include "archivos.h"
#include "sist.h"
#include "funcion.h"

extern unsigned long int bytes_lecturas;
extern unsigned short int tam_contenedor;

/*
    Esta función procesa la matriz de información
*/

int guarda_a_disco
    ( unsigned char matriz_vuelta [] [LONG_LECT],
      struct ordena vector [] )
{
    struct ordena *p_vector;
    unsigned char arch_salida [30], nombre [13],
                  matriz_nodos [MAX_NUM_NODOS] [COLUMNAS];
    unsigned int i, j, uso, flag;
    unsigned long int num_reg, num_regs;
    FILE *p_arch2, *p_count;
```

```

lee( matriz_nodos );

/*
Obtiene el equivalente numérico de la fecha para cada
elemento del vector
*/
i = asigna_fecha( vector, matriz_vuelta );
if( i==0 )
    return( 0 );

/* Verifica el espacio en el contenedor */
uso = espacio_conten( );
if( uso>=85 )
    if( uso>=95 )
        do
        {
            flag = borra( );
            bytes_lecturas = busca_lecturas( );
            cambia_size_conten( tam_contenedor );
            actualiza_size_conten( );
            uso = espacio_conten( );
        }
        while( (uso>=70) && (!flag) );
    else
        aviso_back( uso );

/*
Reduce el contenedor 50 kbytes. Esto libera el
espacio necesario para la matriz_vuelta
*/
bytes_lecturas += 50000;
cambia_size_conten( tam_contenedor );

/*
Actualiza la variable que determina el espacio
disponible en disco
*/
actualiza_size_conten( );

```

```

/*
Verifica que el archivo de lecturas aún puede
contener más datos
*/
num_regs = get_num_reg( );
if( num_regs>=NUM_MAX_REG )
{
    cierra_arch( );
    otra_entrada( matriz_nodos );
}

/* Ordena el vector de apuntadores por fecha */
qsort(vector, i, sizeof(struct ordena), compara_vect);

/*
Determina los desplazamientos de cada renglón
de la matriz
*/
asigna_offset( matriz_vuelta, matriz_nodos, vector );

/* Localiza o crea el archivo de datos activo */
arch_activo( nombre, vector, matriz_vuelta );

/* Forma el nombre del archivo de salida */
nombre[12] = '\0';
strcpy( arch_salida, PATH );
strcat( arch_salida, nombre );

/* Realiza la escritura a disco */
p_arch2 = fopen( arch_salida, "a+b" );
p_count = fopen( ARCH_COUNT, "r+b" );
fscanf( p_count, "%7ld", &num_reg );
p_vector = &vector [0];

```

```

for( i=0; i<(MAX_NUM_NODOS*MAXINFO); i++ )
{
    if( p_vector->mat_uso==OCUPADO )
    {
        num_reg++;
        for( j=0; j<LONG_LECT; j++ )
            putc
                (matriz_vuelta [p_vector ->mat_renglon][j],
                 p_arch2);
    }
    else
        break;
    p_vector++;
}
fseek( p_count, 0L, SEEK_SET );
fprintf( p_count, "%7ld", num_reg );
fclose( p_count );
fclose( p_arch2 );

/*
Ajusta el área destinada al almacenamiento de
los datos
*/
bytes_lecturas = busca_lecturas( );
cambia_size_conten( tam_contenedor );
actualiza_size_conten( );

return( 0 );
}

```

Figura III.7.7 Listado de la función guarda_a_disco

Además de `matriz_vuelta`, a la función `guarda_a_disco` se le envía como parámetro el arreglo `vector`. A los elementos de este vector se les utiliza como apuntadores a los renglones de la matriz y describen algunas de sus características, necesarias para un proceso de ordenamiento descrito más adelante. Cada elemento del vector tiene la siguiente estructura:

```
struct ordena {  
    unsigned int mat_renglon;  
    unsigned short int mat_uso;  
    unsigned long int mat_segundos;  
};
```

Cuando el estado de un nodo ha sido estable, algunas veces no transmite la máxima cantidad de lecturas que puede enviar. Por esta razón, al final de un ciclo de monitoreo generalmente no se ocupan todos los renglones de la matriz. Una de las funciones del arreglo `vector` es indicar los renglones de la matriz que contienen información válida, su campo `mat_renglon` contiene el número de renglón de la `matriz_vuelta` y el campo `mat_uso` indica si ese renglón de la matriz se encuentra ocupado con información válida.

- a) La función inicia leyendo del disco duro la matriz de nodos y almacena esta información en `matriz_nodos`. Esta se utilizará para determinar los nodos que se encuentran dados de alta para el monitoreo de la red.
- b) Como un paso previo al ordenamiento de la información por tiempo, la función `asigna_fecha`, obtiene el número de segundos transcurridos desde el primero de enero de 1993, a la fecha y hora indicadas en cada renglón de `matriz_vuelta` y ese valor se asigna al campo `mat_segundos` de `vector`. La función regresa el número de renglones ocupados de `matriz_vuelta`, de modo que cuando el valor retornado es cero, no se tiene que realizar acción alguna y la función regresa inmediatamente al programa que la llamó.
- c) La función `espacio_conten` determina el espacio reservado disponible para el almacenamiento de la información. Esta cantidad se determina en porcentaje con base en la medida del área de datos declarada en el archivo de tamaño y la longitud instantánea del archivo contenedor.

Si el espacio ocupado por información de lecturas aumenta a más del 85%, la función `aviso_back` despliega un mensaje en la pantalla advirtiendo que de no exportar los datos del sistema a través del módulo de respaldos, al llegar este porcentaje al 95%, automáticamente comenzará el proceso de autodepuración. Cuando el espacio disponible disminuye a menos del 5% del total reservado, comienza el proceso de borrado de la información hasta disminuir a 70% el área ocupada por lecturas. Este proceso lo efectúa la función `borra` accedendo al archivo directorio y removiendo los archivos de lecturas ahí referenciados.

Esta característica de **autodepuración** evita que el sistema falle en algún momento por falta de espacio en el disco duro. Conviene hacer notar que modificando el tamaño del contenedor, se puede ajustar el período de tiempo en el que comienza el proceso de borrado. En consecuencia, la modificación del área reservada a los datos constituye uno de los procedimientos de afinación más importantes del sistema, seleccionando un tamaño adecuado para el archivo contenedor, se puede adecuar al sistema a alguna política de respaldos, se puede lograr mayor autonomía y se incrementa la confiabilidad ya que el sistema no sufrirá los efectos de un espacio en disco limitado.

- c) Posteriormente el contenedor disminuye su tamaño 50kbytes, liberando de esta forma el espacio necesario para almacenar en el peor de los casos todas las lecturas. Este procedimiento deja abierta la posibilidad de que la información quede almacenada en forma contigua si se desfragmenta el disco duro antes de instalar el sistema. Las ventajas obtenidas son: reducción en el número de accesos al disco y en consecuencia menor desgaste de los componentes mecánicos e incremento de la velocidad de transferencia de la información del disco duro ya que se disminuye también el tiempo de acceso.
- d) Antes de realizar la escritura al disco, se lee del archivo de número de registros la cantidad de lecturas que se han almacenado en el archivo con bandera de estado en 1 y se compara contra el valor de la constante simbólica `MAX_NUM_REG` (para generar archivos de aproximadamente 64kbytes este parámetro tiene el valor de 1200).

Cuando se sobrepasa este límite, se cambia la bandera de estado del archivo de datos que hasta ese momento almacenaba la información y se genera otro registro en el archivo directorio con las abreviaturas de los nodos dados de alta en ese momento utilizando la información de `matriz_nodos`.

- e) Una vez que el sistema se ha asegurado de que existe el espacio suficiente para el almacenamiento de la información, las lecturas se ordenan por tiempo para garantizar que la información se grave conforme se generó. En este paso, el vector se ordena con la función de biblioteca `qsort` tomando en cuenta el valor en su campo `mat_segundos`, lo cual indirectamente ordena la información contenida en la matriz. Este método de ordenamiento de la matriz a través de un vector de apuntadores resulta más rápido que ordenarla directamente.
- f) A causa de este ordenamiento, ya no se puede garantizar que las lecturas de un mismo nodo queden en bloques contiguos de memoria por lo que para relacionar las lecturas de un mismo nodo, establecimos relaciones entre los registros a través de listas ligadas. La función `asigna_offset` determina para cada uno de los registros, el número de lecturas que se deben saltar para llegar a la siguiente y la anterior del mismo nodo, en este proceso también se hace el ajuste de la información en memoria con la que ya está almacenada en el disco duro.
- g) Finalmente se obtiene del archivo directorio a través de la función `arch_activo` el nombre del archivo de datos con bandera de estado en 1 o bien, si se acaba de crear uno nuevo agregando un registro en el directorio, se le asigna su nombre determinado por la hora, día, mes y año de la primera lectura que se almacenará en él (de este modo el nombre del archivo también indica el conjunto de información que agrupará).
- h) El proceso de escritura a disco por fin puede realizarse, después de grabar cada lectura, el valor contenido en el archivo `num_reg` se incrementa.
- i) Después de haber escrito a disco la información, el tamaño del contenedor se ajusta de acuerdo al nuevo espacio ocupado por las lecturas.

III.8 Consultas a la información almacenada

Para poder consultar los datos generados durante el funcionamiento del sistema, se desarrolló un programa, que utilizando el formato de los registros almacenados permite un acceso eficiente a la información.

El subsistema de consultas es uno enorme en cuanto a la cantidad de código (consta de aproximadamente 100 distintas funciones) pero sencillo en su funcionamiento. Debido al tamaño del programa, fue necesario utilizar *overlays*; esto es, pequeños módulos que se cargan, efectúan algunos procedimientos y posteriormente pueden liberar las direcciones de memoria que ocupaban. El programa está diseñado para un fácil mantenimiento, dejando abiertas posibilidades de actualización y expansión.

Se implantaron dos modalidades; consultas interactivas y reportes direccionados a un archivo de texto. Siendo éste último adecuado en la generación de reportes extensos y el anterior está orientado a una búsqueda manual de información.

El primer módulo que se carga es la interfaz de ventanas que le permite al usuario definir las características del reporte. En esta etapa se definen la fecha, hora y los criterios que se utilizarán para determinar si un registro se incluye en el reporte. En la sección correspondiente a la utilidad de consultas en el manual de usuario, se pueden observar ejemplos de las diferentes pantallas que se presentan. La interfaz es consistente y usa múltiples ventanas. Explicaciones concretas en la ventana de ayuda y la presentación de las teclas válidas según el contexto en la barra de funciones, llevan al usuario a la definición del reporte en una forma sencilla y amigable.

Cuando termina de ejecutarse este primer módulo, se han generado tres objetos. Un vector de caracteres contiene la fecha y hora en que se requiere la información, una matriz que proporciona los nombres y abreviaturas de los nodos seleccionados para generación de su reporte y una estructura indica los criterios de selección que se aplicarán.

La matriz se almacena en un archivo temporal para ser leída por el siguiente módulo. Este método es necesario debido a las limitaciones en el número y longitud de los argumentos que se le pueden enviar a un programa.

Para reportes a pantalla el vector generado contiene la fecha y hora en la que se quiere consultar la información. En el caso de reporte a disco, delimita el intervalo de tiempo, con la fecha y hora iniciales y finales dentro de las cuales se quiere generar el reporte.

La estructura `critero` se utiliza para establecer el criterio de selección durante la búsqueda. Se encuentra definida en el header `matrices.h`:

```
struct critero{  unsigned int ordenacion;
                 unsigned int busqueda;
                 unsigned int temperatura;
                 unsigned int baterias;
                 unsigned int al_mayor;
                 unsigned int al_menor;
                 unsigned int e_electrica;
                 unsigned int destino;
                };
```

El significado de cada uno de los campos de la estructura se muestra en la tabla III.8.1.

Tabla III.8.1 Significado de los campos de la estructura criterio

Campo	Valor	Significado
ordenacion	0	Reporte agrupado por nodo
	1	Reporte ordenado por tiempo
busqueda	0	Reporte de todos los registros disponibles
	1	Reporte sólo de registros seleccionados
temperatura	0	No se reportan temperaturas altas
	> 0	Temperatura de umbral*
baterias	0	No se reporta bajo voltaje en baterías
	> 0	Voltaje de umbral**
al_mayor	0	No se buscan alarmas mayores
	1	Incluir alarmas mayores
e_electrica	0	No se buscan fallas de energía eléctrica
	1	Incluir fallas de energía eléctrica
al_menor	0	No se buscan alarmas menores
	1	Incluir alarmas menores
destino	0	Reporte a pantalla
	1	Reporte a archivo tipo texto

* Cuando la temperatura registrada en la lectura es mayor a la temperatura de umbral indicado en la estructura, el registro se incluye en el reporte.

** Cuando el voltaje registrado en la lectura es menor al voltaje de umbral indicado en la estructura, el registro se incluye en el reporte.

El vector y la estructura se envían como parámetros al siguiente módulo. Una cadena de identificación se agrega al inicio de los argumentos para evitar que el módulo se ejecute directamente desde la línea de comandos con resultados impredecibles.

El primer módulo que se cargó se retiene en memoria mientras (según se haya seleccionado), se cargan y ejecutan los módulos de reportes a disco o a pantalla. Después de su ejecución éstos liberan las direcciones de memoria que ocupaban, dejando espacio libre para otras aplicaciones.

A pesar de que los resultados obtenidos por uno y otro tipo de reporte son completamente diferentes, ambos basan su funcionamiento en las mismas estructuras y características del sistema. Los programas inician accediendo al archivo directorio para determinar la existencia de archivos de lecturas que cumplan con las fechas solicitadas. Recuerde que (como se explicó en la sección III.7.2), en el directorio se encuentra un registro de todos los archivos de lecturas que se tienen almacenados en el disco duro bajo el control del sistema. Cada entrada del directorio contiene el nombre de un archivo de lecturas formado por la hora, mes, día y año de la información más antigua que contiene. De esta forma, examinando únicamente el directorio; se puede saber si se encuentra o no información de las fechas requeridas.

Una vez que se determinó que existen archivos con la información solicitada, se accesa el primero de ellos y se busca la primera lectura que cumpla con el criterio de búsqueda. Cuando se trata de un reporte a disco se leen sucesivamente los registros almacenados hasta que se alcanza la fecha y hora final del reporte o se terminan los archivos de lecturas. En caso de reporte a pantalla el programa se detiene cuando se encuentra la primera lectura válida y se presenta al usuario una interfaz que le permite seguir consultando en la información en forma interactiva. Cada vez que se encuentra un registro de un nodo seleccionado, se envía la lectura a una función que determina si cumple o no con los criterios de búsqueda para desplegarse en la pantalla o escribirse al archivo de salida en el diskette.

Algunas estructuras tales como listas ligadas se crean en memoria al momento de generar el reporte, éstas le permiten al sistema conocer la existencia de lecturas próximas al registro actual, con lo que los accesos son más rápidos y seguros.

III.9 Exportación de la información

Debido a que la capacidad de un disco duro es enorme pero finita, no se puede indefinidamente almacenar información en él. El proceso de autodepuración descrito en el inciso b de la sección III.7.6 evita este problema, pero la consecuente pérdida de información puede no ser una solución óptima. Para solventar esta dificultad, se diseñó un mecanismo alternativo para la liberación del espacio asignado al sistema en el disco duro con el objetivo fundamental de conservar la información que el sistema genera.

A pesar de que el costo por bit de almacenamiento continuamente está bajando, almacenar la información en un disco duro no resulta práctico, por lo que si se desea conservarla, resulta conveniente transferir los datos más antiguos a un medio de menor costo tal como una cinta magnética, un diskette o un disco compacto. Se seleccionó como medio de salida de datos la unidad de diskette, ya que en el momento de desarrollar este trabajo, era aún el periférico estándar en cualquier computadora personal.

Una vez que la información está fuera del sistema, se evita el riesgo de pérdida de datos por falla en el disco duro. Además, estos archivos históricos se pueden usar para otras aplicaciones, tales como estudios estadísticos de las condiciones de la red.

Con el objetivo de exportar la información en algún formato de fácil manejo para el usuario promedio, determinamos que deberían convertirse a uno reconocible por virtualmente cualquier manejador de base de datos. En lugar de un archivo plano o ASCII delimitado, seleccionamos el formato dBASE ya que se ha convertido en un estándar *de facto* en computadoras personales. La mayoría de las hojas de cálculo y manejadores de bases de datos para PC pueden importar datos en este formato.

Para comprender cómo se realiza la interfaz entre la estructura que utiliza internamente el sistema de monitoreo y el formato dBASE, resulta necesario entender cómo se almacena un archivo .DBF.

III.9.1 La estructura de un archivo dBASE

Fundamentalmente, todo archivo de este manejador de base de datos consiste en una cabecera seguida de los datos y una marca de fin de la base.

La cabecera del archivo contiene la información descrita en la tabla III.9.1. La versión dBASE IV no cambia la estructura de la cabecera excepto por el uso de los bytes reservados 12 a 31 (consulte la referencia 4 de la bibliografía para una descripción más detallada).

Tabla III.9.1 Estructura de la cabecera de un archivo .DBF

Byte	Contenido
0	Número de la versión: 0x02 dBASE II 0x03 dBASE III o IV sin campo memo 0x83 dBASE III con campo memo 0x8B dBASE IV con campo memo
1-3	Fecha de la última actualización en hexadecimal: el byte 1 es el año, el 2 el mes y el 3 el día.
4-7	Número de registros de la base de datos en hexadecimal (el byte 7 es el dígito más significativo y el 4 el menos significativo).
8-9	Número de bytes de la cabecera en hexadecimal (el byte 9 es el dígito más significativo y el 8 el menos significativo).
10-11	Longitud de cada registro, expresada en hexadecimal (el byte 11 es el dígito más significativo y el 10 el menos significativo).
12-31	Reservados.
32- hasta el final de la cabecera	Especificaciones de los campos: nombre, tipo, ancho y número de decimales.

La sección de importancia es la correspondiente a las características de los campos la cual empieza en el byte 32. A partir de esta posición, treinta y dos caracteres se reservan para cada campo y describen las características de ellos como se muestra en la tabla III.9.2.

Tabla III.9.2 Significado de los 32 bytes que describen las características de un campo

Bytes	Contenido
0-10	Nombre del campo
11	Tipo: C: Carácter N: Numérico L: Lógico M: Memo D: Fecha F: Número de punto flotante (en dBASE IV)
12-15	Reservado
16	Longitud del campo en hexadecimal
17	Número de posiciones decimales si se trata de un campo numérico
18-19	Reservado
20	Identificador del área de trabajo
20-31	Bytes reservados

A continuación se almacenan los datos en registros de longitud fija, dejando un byte al inicio de cada uno de ellos para la bandera de borrado y las posiciones no utilizadas se rellenan con espacios en blanco. La marca 0x1a indica el fin de la base de datos.

III.9.2 El proceso de exportación de la información

El objetivo de las funciones de exportación de la información, consiste en convertir el archivo de lecturas utilizado por el sistema, a una base de datos en formato dBASE de la siguiente estructura:

Tabla III.9.3 Estructura del archivo de base de datos

Nombre del campo	Tipo	Ancho (bytes)
NODO	Carácter	4
FECHA	Fecha	8
HORA	Carácter	6
TEMPERAT	Numérico	4
BATERIAS	Numérico	4
ALARMA_MAY	Lógico	1
ALARMA_MEN	Lógico	1
FALLA_POW	Lógico	1
TASA_RX_I	Numérico	5

El programa inicia solicitando la unidad en la que se efectuará el respaldo únicamente cuando se haya detectado la existencia de más de una unidad de diskette. Después pide que se inserte un disco en la unidad seleccionada, verifica que se encuentre colocado correctamente, que sea un formato reconocible y que no esté protegido contra escritura. Posteriormente avisa al usuario que el disco que insertó perderá toda la información que tiene almacenada y solicita su confirmación antes de iniciar el proceso.

El sistema restablece algunas áreas reservadas al sistema operativo, lo que se conoce como un formato rápido del disco. Una vez que al disco se le ha borrado la información que contenía en esos sectores, inicia la conversión del archivo de lecturas a una base de datos en formato dBASE.

Del directorio se lee el nombre del archivo de lecturas más antiguo que se tiene en el disco duro, se verifica que exista espacio en el diskette para almacenarlo y comienza a realizar la transferencia. La función `convierte_dbase` listada en la figura III.9.1 muestra en forma general como se realiza la exportación de la información:

```

#include <stdio.h>
#include <conio.h>

#include "bande.h"
#include "sist.h"
#include "funcion.h"

/* Esta función convierte a formato dbase la
   información almacenada en el disco duro,
   para poder ser respaldada en diskettes
*/

int convierte_dbase
(
    FILE *p_archivo_destino,
    FILE *p_archivo_fuente,
    unsigned long int total_registros )
(
    unsigned char  lectura [LONG_LECT+1],
                  lectura_uno [LONG_LECT+1];
    int i, primera;
    unsigned long int registros;

    put_header( p_archivo_destino );
    registros = 0L;
    primera = TRUE;
    do
    {
        for( i=0; i<LONG_LECT; i++ )
        {
            lectura [i] = getc( p_archivo_fuente );
            if( primera )
                lectura_uno [i] = lectura [i];
        }
        primera = FALSE;
        escribe_dbase( lectura, p_archivo_destino );
        registros++;
        hora_sistema( INVISIBLE );
        flecha_backup( registros, total_registros );
        porcentaje_conten( );
    }
    while( !no_fin( p_archivo_fuente ) );

```

```
fseek( p_archivo_destino, -1L, SEEK_CUR );
putc( 0x1A, p_archivo_destino );

fecha_dbase( p_archivo_destino, lectura_uno );
numero_regs_dbase( p_archivo_destino, registros );
}
```

Figura III.9.1 Secuencia de conversión de un archivo de lecturas a formato dBASE

- a) A la función se le envían tres parámetros, el primero es el apuntador a el archivo .DBF que se generará: *p_archivo_destino. Otro apuntador *p_archivo_fuente, controla el acceso al archivo de lecturas. El número total de registros en el parámetro total_registros es utilizado por las funciones que despliegan el porcentaje de avance en la pantalla.
- b) El proceso inicia colocando la cabecera de una base de datos dBASE a el archivo de salida a través de la función invocada por la siguiente instrucción.

```
put_header( p_archivo_destino );
```

La cabecera que se le coloca al archivo de base de datos se encuentra en el código de la función put_header y se muestra en la figura III.9.2. A esta rutina se le envía como parámetro el apuntador al archivo en donde debe colocar la información.

En la cabecera se indica el número de la versión de dBASE en la que se exportará la información, el número de bytes de la cabecera, la longitud de cada registro y las especificaciones de cada campo, según se indicó en las tablas III.9.1 y III.9.2. El número de registros y la fecha de la última actualización se insertan posteriormente.

```
0X03, 0X00, 0X00, 0X00, 0X00, 0X00, 0X00, 0X00,
0X41, 0X01, 0X23, 0X00, 0X00, 0X00, 0X00, 0X00,
0X00, 0X00, 0X00, 0X00, 0X00, 0X00, 0X00, 0X00,
0X00, 0X00, 0X00, 0X00, 0X00, 0X00, 0X00, 0X00,
0X4E, 0X4F, 0X44, 0X4F, 0X00, 0X64, 0XAD, 0X04,
0XDE, 0X12, 0X00, 0X43, 0X10, 0X59, 0X0B, 0X00,
0X04, 0X00, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X46, 0X45, 0X43, 0X48, 0X41, 0X00, 0X00, 0X04,
0XDE, 0X12, 0X00, 0X44, 0X10, 0X59, 0X0B, 0X00,
0X08, 0X00, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X48, 0X4F, 0X52, 0X41, 0X00, 0X00, 0X00, 0X04,
0XDE, 0X12, 0X00, 0X43, 0X10, 0X59, 0X0B, 0X00,
0X06, 0X00, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X54, 0X45, 0X4D, 0X50, 0X45, 0X52, 0X41, 0X54,
0X00, 0X12, 0X00, 0X4E, 0X10, 0X59, 0X0B, 0X00,
0X04, 0X01, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X42, 0X41, 0X54, 0X45, 0X52, 0X49, 0X41, 0X53,
0X00, 0X12, 0X00, 0X4E, 0X10, 0X59, 0X0B, 0X00,
0X04, 0X01, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X41, 0X4C, 0X41, 0X52, 0X4D, 0X41, 0X5F, 0X4D,
0X41, 0X59, 0X00, 0X4C, 0X10, 0X59, 0X0B, 0X00,
0X01, 0X01, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X41, 0X4C, 0X41, 0X52, 0X4D, 0X41, 0X5F, 0X4D,
0X45, 0X4E, 0X00, 0X4C, 0X10, 0X59, 0X0B, 0X00,
0X01, 0X01, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X46, 0X41, 0X4C, 0X4C, 0X41, 0X5F, 0X50, 0X4F,
0X57, 0X00, 0X00, 0X4C, 0X10, 0X59, 0X0B, 0X00,
0X01, 0X01, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X54, 0X41, 0X53, 0X41, 0X5F, 0X52, 0X58, 0X5F,
0X49, 0X00, 0X00, 0X4E, 0X10, 0X59, 0X0B, 0X00,
0X05, 0X01, 0XB4, 0X6A, 0X6E, 0X64, 0XAA, 0X21,
0XCE, 0X47, 0X0C, 0X00, 0XB4, 0X6A, 0X7E, 0X64,
0X0D, 0X20, 0X1A
```

Figura III.9.2 Cabecera que se le coloca al archivo de salida a través de la función put_header

- c) Después de la cabecera, la siguiente sección de un archivo .DBF es propiamente el área de datos. En la función `convierte_dbase` el ciclo

```
do
  {
    .
    .
    .
  }
while (!no_fin (p_archivo_fuente));
```

permite la lectura de registros del archivo de lecturas mientras no se alcance la marca de fin de archivo.

En el cuerpo del ciclo se lee un registro del archivo de lecturas y se transforma a uno en el de salida a través de la función `escribe_dbase` mostrada en la figura III.9.3. Observe que al final de cada registro se escribe un carácter 0x20 que se utiliza como bandera de borrado para el siguiente registro que se escribirá al archivo.

- d) Finalmente como se muestra en la función `convierte_dbase`, para concluir el proceso de conversión de un archivo de lecturas, con las instrucciones

```
putc( 0x1A, p_archivo_destino );
fecha_dbase( p_archivo_destino, lectura_uno );
numero_regs_dbase( p_archivo_destino, registros );
```

se agrega la marca 0x1a al final del archivo, indicando el final de la base de datos. La fecha de la primera lectura se agrega como fecha de última actualización y se escribe la cantidad de registros de la base.

Al final el archivo de lecturas se remueve del disco y la entrada correspondiente en el directorio se borra. Este proceso se repite mientras haya espacio disponible en el diskette y existan aún más archivos por respaldar.

```

#include <stdio.h>

#include "funcion.h"

/*
    Esta función escribe la información almacenada
    al disco en formato dBASE
*/

int escribe_dbase( unsigned char lectura[],
                  FILE *p_archivo_destino )
{
    int i;
    unsigned char fecha [8];

    /* Escribe la abreviatura del nodo */
    for( i=0; i<4; i++ )
        putc( lectura[i], p_archivo_destino );

    /* Escribe la fecha */
    asigna_nombre( fecha, lectura );

    putc( lectura [25], p_archivo_destino ); /* Año */
    putc( lectura [26], p_archivo_destino );
    putc( lectura [27], p_archivo_destino );
    putc( lectura [28], p_archivo_destino );

    putc( fecha [4], p_archivo_destino );    /*Mes*/
    putc( fecha [5], p_archivo_destino );

    putc( fecha [2], p_archivo_destino );    /*Dia*/
    putc( fecha [3], p_archivo_destino );

    /* Escribe la hora */
    putc( lectura [16], p_archivo_destino ); /*Hora*/
    putc( lectura [17], p_archivo_destino );
    putc( lectura [19], p_archivo_destino ); /*Min.*/
    putc( lectura [20], p_archivo_destino );
    putc( lectura [22], p_archivo_destino ); /*Seg.*/
    putc( lectura [23], p_archivo_destino );

    /* Escribe la lectura de temperatura */
    fprintf( p_archivo_destino, "%4.1f",
            ((float)lectura[30]*50.0)/256.0 );
}

```

```

/* Escribe el voltaje de baterías */
fprintf( p_archivo_destino, "%4.1f",
        ((float)lectura[31]*100.0)/256.0 );

/* Escribe el estado de la alarma mayor */
if( lectura[32] & 0x4 )
    putc( 'T', p_archivo_destino );
else
    putc( 'F', p_archivo_destino );

/* Escribe el estado de la alarma menor */
if( lectura[32] & 0x2 )
    putc( 'T', p_archivo_destino );
else
    putc( 'F', p_archivo_destino );

/* Estado de la energía eléctrica */
if( lectura[32] & 0x1 )
    putc( 'T', p_archivo_destino );
else
    putc( 'F', p_archivo_destino );

/* Tasa de recibidos contra intentados en % */
fprintf( p_archivo_destino, "%5.1f", (float)
        (lectura[34]/lectura[35])*100.0 );

/*Bandera de borrado del siguiente registro */
putc( 0x20, p_archivo_destino );
}

```

Figura III.9.3 Función que realiza la conversión de una lectura a un registro dBASE

III.10 Inicialización

El Sistema de Monitoreo de la Red de Telefonía Digital de la UNAM realiza, al inicio de su operación, una prueba general del hardware y software de las microcomputadoras en que será ejecutado; de acuerdo al siguiente esquema:

- Verificación del hardware con que cuenta la PC
- Creación de las estructuras de administración de los datos
- Configuración de los elementos de comunicaciones

En las siguientes páginas se describirán los que, desde nuestro punto de vista, son los principales aspectos del proceso de inicialización.

III.10.1 Verificación del hardware de la PC

El Sistema hace un reconocimiento del hardware que tendrá cada una de las computadoras que utilice y muestra en pantalla algunas de sus características: el tipo de máquina en que se está ejecutando, el modo de video actual y la existencia de coprocesador matemático. Además, se verifican especialmente:

- El número de puertos paralelos y seriales disponibles. Es indispensable contar con un puerto serial, ya que sin él no es posible realizar el monitoreo; además, en las computadoras remotas es obligatoria la disponibilidad de un puerto paralelo en la dirección 0x378, ya que por este conducto se obtienen los datos de los dispositivos sensores
- El tamaño de la memoria RAM base. Se verifica que cada computadora remota cuente con por lo menos 512 kbytes, mientras que la computadora central deberá tener 640.
- El número de unidades de disco instaladas. Este valor se utiliza posteriormente. Los módulos de mantenimiento y consultas del Sistema permiten la opción de seleccionar la unidad a la que se dirigirá la información, si es que se ha detectado más de una unidad de diskette.

Por otra parte, el Sistema verifica *adicionalmente* en la computadora central, la existencia de la unidad de disco duro; obtiene y muestra en pantalla su capacidad total.

III.10.2 Creación de las estructuras de administración de los datos

El software de administración de la información está instalado en el disco duro de la computadora central, por lo que es en este equipo *únicamente* en el que el Sistema crea los siguientes elementos, después de haber verificado el hardware de la PC:

- Las rutas C:\MONITOR\SYS y C:\MONITOR\DAT;
- El contenedor de datos y el archivo de su tamaño (cf. III.7.1);
- Los archivos del directorio (cf. III.7.2), del número de registros (cf. III.7.4), de nodos (cf. III.7.5) y del número de nodos registrados en el Sistema;

Cuando el Sistema se ejecuta por primera vez, los crea y posteriormente, los utiliza cada vez que se reanuda su operación después de alguna interrupción.

III.10.3 Configuración de los elementos de comunicaciones

Las comunicaciones asincronas tienen parámetros que deben ser ajustados para permitir el establecimiento de un enlace. En el Sistema de Monitoreo las computadoras que se intentan comunicar deben configurar sus puertos seriales con los mismos valores.

Los parámetros que se deben configurar son:

- El número de *bits por dato o longitud de palabra*. Pueden ser desde cinco hasta ocho bits. Generalmente se utiliza una longitud de palabra de siete bits, ya que con ella se puede transmitir el conjunto de caracteres ASCII, sin embargo, para transmitir el conjunto de caracteres *ASCII extendido* que maneja la PC, son necesarios ocho bits por dato.

- El número de *bits de paro*. Este puede ser uno, uno y medio o dos. Es poco común utilizar uno y medio o dos bits de paro; la mayoría de las comunicaciones seriales utilizan un bit de paro.
- La *paridad*. Puede ser *par, non, marca, espacio* o *no tener paridad* cuando la longitud de palabra transmitida es de ocho bits.
- *Velocidad de transmisión*. Puede ser desde 110 hasta 19,200 bps, dependiendo de la versión del BIOS de la PC propietaria del puerto. En la actualidad la mayoría de las microcomputadoras pueden manejar desde 300 hasta 9,600 bps.

Cada una de las lecturas transmitidas por los nodos remotos incluye los datos originados por los sensores, que constan de ocho bits. Por esta razón, el Sistema de Monitoreo programa los puertos seriales de todas las computadoras con ocho bits por dato, sin paridad y, para incrementar su confiabilidad, dos bits de paro.

Una vez programado el puerto serial, se habilita una prueba interna. Al poner un uno lógico en el bit 4 del registro de control de salidas RS-232, se interconectan y prueban las señales CTS, RTS, DSR, DTR, DCD, GPO-2, RI y GPO-1 (cf. III.5.1). Una vez terminada la prueba, el Sistema levanta en el puerto serial utilizado, la señal de que indica que la PC está preparada (DTR), indicándole así al modem que está habilitada para la transferencia de información.

Posteriormente los modems utilizados por el Sistema se configuran con el conjunto de comandos AT. El usuario debe crear un archivo que contenga las instrucciones que configurarán el modem a utilizar; de esta manera, el Sistema es flexible y permite la sustitución de este equipo con pocas modificaciones. Para una explicación detallada de las consideraciones necesarias durante la creación de estos archivos, consulte al apéndice A.

Finalmente y a través del modem, el programa descuelga la línea y verifica la disponibilidad del tono de marcación. En este punto del proceso de inicialización se tiene ya la seguridad de contar con todos los elementos para el correcto funcionamiento del Sistema y comienza el monitoreo de la Red.

Capítulo IV

Manual del usuario

IV.1 Requerimientos del sistema

En esta sección se exponen las características del equipo necesario para la operación del Sistema de Monitoreo.

En todos los nodos se requiere:

- Una línea telefónica; puede ser una extensión de la Red Digital o un número directo independiente que comience con 62.
- Un modem compatible Hayes con una velocidad de transmisión de 300 a 9600 bps y sus cables de conexión necesarios. La siguiente sección explica detalladamente las características requeridas. Tome en cuenta que el modem con la velocidad de transmisión mas baja determina la velocidad máxima en la red ya que todo el sistema se debe configurar a ese valor. El uso de un modem externo permite su instalación y reemplazo más fácil y rápidamente, aunque un interno simplifica el cableado de la interfaz de comunicación y de la alimentación de energía.
- Sistema operativo MS-DOS versión 3.3 o superior.

En cada nodo remoto se requiere una computadora personal IBM compatible para efectuar las lecturas de los sensores, así como para controlar la transmisión de datos. La configuración necesaria de cada una de estas computadoras es:

- Memoria RAM base de al menos 512 kbytes.
- Un puerto paralelo (en 0x378) y uno de comunicación serial (en 0x3f8 o 0x2f8).
- Una unidad de diskette para la carga del programa.
- Durante el proceso de instalación de un nodo, un monitor a color resulta muy útil, aunque un monocromático es suficiente para verificar las lecturas obtenidas de los sensores y la transmisión de datos. La autonomía del sistema permite la ausencia de operadores en los nodos remotos por lo que el monitor puede desconectarse posteriormente.

El equipo periférico externo adicional en cada nodo remoto es:

- La tarjeta de sensores. La tarjeta se conecta con el puerto paralelo a través de un cable de conductores calibre AWG28 o mayor, longitud máxima de 2 m, conectores DB-25 macho en cada extremo y conexiones *uno a uno* entre sus terminales.
- Fuente de alimentación para la tarjeta de sensores y los cables de conexión que sean necesarios (ver el punto *Instalación del sistema*). La fuente debe proporcionar 5, 12 y -12V y se podría conectar además a un UPS (fuente de energía no interrumpible), para que en caso de falla de energía, el sistema no quedara fuera de funcionamiento.

En el nodo central las características de la computadora que realiza la función de monitoreo son:

- Se recomienda un equipo basado en un microprocesador 80286 o un modelo de mayor capacidad ya que durante los accesos a disco se genera una gran carga de trabajo. Sin embargo, el sistema central puede funcionar incluso en una máquina 8088.
- Memoria RAM base de 640 kbytes.
- Espacio disponible de al menos 6 Mbytes en el disco duro.
- Un puerto de comunicación serial (en 0x3f8 o 0x2f8).
- El sistema está diseñado para operar correctamente en un monitor monocromático aunque un monitor a color proporciona una mejor visualización de la situación mostrada en la pantalla.

IV.1.1 Características del modem utilizado por el Sistema

En la actualidad existe una gran variedad de modems en el mercado. Sin embargo, deben observar algunas características requeridas por el Sistema de Monitoreo:

1. Hacer uso del conjunto de *comandos Hayes* o *comandos AT*⁸ para iniciar y finalizar una comunicación, así como para configurar sus parámetros de operación.

⁸ Llamados así porque generalmente comienzan con el prefijo AT (solicitud de **at**ención del modem). Esta forma de control fue utilizada inicialmente por la marca *Hayes Microcomputer Company*, al lanzar al mercado su línea de *Smartmodem Hayes*, convirtiéndose en un estándar *de facto*.

2. Utilizar *códigos resultantes*.

El modem seleccionado debe responder, a cada uno de los comandos que se le indique ejecutar, con un mensaje o un código numérico. El Sistema de Monitoreo reconoce algunos *códigos resultantes* en forma de mensajes. Estos tienen la forma:

\r\nCODIGO RESULTANTE\r\n

Los caracteres \r (ASCII 13) y \n (ASCII 10) sirven como delimitadores y son utilizados por el Sistema para el control del modem.

3. Ser capaz de modificar sus parámetros de configuración y operación por medio de los *registros internos "s"*.
4. Apegarse a alguno de los estándares de velocidad de transmisión mostrados en la tabla III.10.1. Si no se utiliza el mismo estándar en todos los modems del Sistema de Monitoreo, podrían no ser capaces de establecer una comunicación entre sí.

Tabla III.10.1 Estándares en la velocidad de comunicaciones

Velocidad (en bps)	Estándar de Estados Unidos	Estándar internacional
110 y 300	Bell 103	V.21
1200	Bell 212A	V.22
2400	V.22 bis	V.22 bis
4800, 9600 y 19200	V.29	V.29
	V.32	V.32
	V.32 bis	V.32 bis
	V.42	V.42
	V.42 bis	V.42 bis

IV.2 Instalación del Sistema de Monitoreo

En esta sección se detallan los pasos necesarios para instalar el Sistema de Monitoreo tanto en la computadora central como en los nodos remotos.

IV.2.1 Instalación del nodo central

1. Conecte el modem a la microcomputadora y a la línea telefónica según lo indique el manual de instalación que viene con el modem. *En este caso se recomienda utilizar un modem interno para simplificar las conexiones.*
2. Deberá crear en el disco duro los directorios correspondientes a las siguientes trayectorias:

C:\MONITOR\EXE\

C:\MONITOR\MDM\

3. Del diskette de programas fuente, copie los siguientes archivos ejecutables contenidos en el directorio \EXE, al directorio C:\MONITOR\EXE\ que acaba de crear:

A_DISCO.EXE,	CONSULTA.EXE,	MONITOR.EXE,
A_PANTA.EXE,	CONTLONG.EXE,	PPTO.EXE,
BKUP.EXE,	DIALMODE.EXE,	PRUEBA.EXE.
CENTRAL.EXE,	MANTE.EXE,	

Si requiere generar alguno de estos programas, tendrá que compilarlo siguiendo las instrucciones proporcionadas en el apéndice B.

4. Deberá crear el archivo de configuración correspondiente al modem con que cuente (extensión .MDM requerida). Ubíquelo en la siguiente ruta:

C:\MONITOR\MDM\

Puede tomar como referencia los ejemplos provistos en el apéndice A.

5. Ejecute el programa del sistema central de acuerdo a las indicaciones de la sección IV.3 de este manual.

6. Durante la inicialización del Sistema deberá seguir las instrucciones presentadas en pantalla. En este paso deberá indicar los nodos remotos que se encuentran ya instalados así como sus números telefónicos. También deberá seleccionar el archivo de extensión .MDM que contenga los comandos necesarios para la configuración correcta del modem.

IV.2.2 Instalación de un nodo remoto

1. Durante la instalación de la tarjeta de sensores se debe tener cuidado en su manejo porque tiene componentes sensibles a descargas de electricidad estática. Al seleccionar el lugar más adecuado para colocar los sensores tome en cuenta las siguientes indicaciones:
 - A fin de mantener un estándar en la posición de la tarjeta, recomendamos colocarla en la parte posterior del conmutador.
 - Al instalar la tarjeta en una posición vertical deberá cuidar que el sensor de temperatura quede en la parte inferior a fin de que no le afecte la circulación por convección del aire caliente generado por los demás circuitos.
2. Conecte la alimentación de la tarjeta de acuerdo a la polarización indicada en ella; las características de la fuente de alimentación requerida se describen en el punto IV.1. Las conexiones a los sensores también deberán respetar las polaridades indicadas en la tarjeta.
3. Identifique en la tarjeta las terminales correspondientes al sensor del nivel de voltaje de las baterías. Conecte estas terminales a las del banco de baterías que proporcionan -48V. Una vez que el programa de la computadora se encuentre funcionando deberá calibrar este sensor .
4. Identifique en la tarjeta las terminales del sensor de energía eléctrica de AC, sus terminales se deberán conectar a las de alimentación del equipo rectificador de voltaje. Con el selector de número de fases en la tarjeta, deberá elegir dos o tres fases según corresponda.

5. Identifique en la tarjeta las terminales de los detectores de alarmas mayor y menor, estas se deberán conectar a las del conmutador que van a las lámparas indicadoras en los locales.
6. Conecte el modem a la microcomputadora y a la línea telefónica según lo indique el manual de instalación que viene con el modem. *En este caso se recomienda utilizar un modem externo para reducir el tiempo de instalación y reemplazo.*
7. Ejecute el programa del sistema remoto siguiendo las indicaciones provistas en la sección IV.3. Si es necesario recompilar el programa siga las indicaciones del apéndice B hasta obtener el archivo ejecutable REMOTA.EXE.

Es importante modificar los archivos de configuración de la PC utilizada, para que este programa se ejecute automáticamente al arrancar la computadora.

8. Después de ejecutar el programa, conecte la tarjeta de sensores al primer puerto paralelo de la microcomputadora. La configuración y características del cable de interfaz entre la tarjeta y la computadora se proporcionan en la sección IV.1.
9. El sensor de nivel de voltaje de baterías se deberá calibrar ajustando los potenciómetros de la tarjeta hasta que se despliegue en pantalla un valor que coincida aproximadamente con el valor medido directamente a través de un multímetro entre las terminales del banco de baterías. Para una calibración más precisa, con el mismo procedimiento ajuste el voltaje en la terminal $V_{ref}/2$ del convertidor AD a 2.5V (cf. sección III.3.2).
10. Una vez realizados los pasos anteriores será necesario dar de alta el nodo remoto en el sistema central como se indica en la sección IV.6.1.

IV.3 Ejecución del sistema desde la línea de comandos del MS-DOS

Aún cuando el sistema central puede ejecutarse con sólo teclear:

```
C:\MONITOR\EXE\MONITOR.EXE
```

es conveniente analizar los parámetros opcionales que controlan la configuración del programa y que se pueden proporcionar desde la línea de comandos.

La sintaxis completa para la ejecución del sistema es:

```
MONITOR [/puerto] [/S:velocidad] [/marcación] [/validación]
```

Las variables en las opciones son:

puerto

Es el puerto de comunicaciones que se empleará. Puede tener los valores COM1 o COM2, dependiendo del que se tenga disponible en la máquina. Si se omite, el sistema tomará por defecto COM1.

velocidad

Determina la velocidad de transmisión de información. Los valores que puede tomar son 300, 600, 1200, 2400, 4800 y 9600 bps. Cuando no se especifica una velocidad, tomará la de 2400 bps.

marcación

- I Al proporcionar esta opción se emplea el modo de marcación interno (de extensión a extensión con cinco dígitos). Es el modo usado por defecto.
- X Se emplea cuando se requiere que las llamadas se hagan saliendo de la Red de Telefonía Digital de la UNAM. El sistema utilizará el modo de marcación externo (esto es, de la forma 9 + 62 + número de extensión).
- D Se utiliza si se necesita que el monitoreo sea realizado con una línea telefónica independiente de la red telefónica de la Universidad. Permite la marcación con un número directo que inicie con 62.

validación

- V Habilita la validación de extensiones telefónicas de acuerdo al plan de numeración para cada nodo. Es la opción por defecto.
- NV Deshabilita la validación del número de extensión telefónica.

El programa MONITOR es el principal del sistema, el cual a su vez, llama a una serie de módulos secundarios. Sin embargo, es posible ejecutar algunos de ellos en forma independiente, cuando el usuario así lo precise y sin que el sistema efectúe el monitoreo de la red.

A continuación se listan los programas secundarios que se pueden ejecutar, los parámetros que requieren, y una breve descripción de su función. Todos ellos se encuentran en el directorio:

C:\MONITOR\EXE\

Inicialización del sistema:

PRUEBA [/puerto] [/v|/nv] [/S:velocidad]

Este programa realiza un conteo de los recursos de la computadora central para evaluar si éstos son los mínimos requeridos por el sistema. También efectúa la configuración del modem conectado y verifica que la línea telefónica proporcione tono de marcación. El significado de los parámetros que utiliza es el mismo que para el programa MONITOR.

Respaldos:

BKUP

Efectúa la transferencia de la información acumulada a un diskette, liberando espacio en el disco duro para el almacenamiento de nuevos datos.

Prueba de los puertos de comunicaciones:

PPTO

La ejecución de este programa permite verificar el funcionamiento de los puertos de comunicaciones.

Consultas a la información almacenada:

CONSULTAS

Genera reportes de las condiciones de los locales de los conmutadores en las fechas que el usuario indique.

Para una referencia más detallada del empleo de cualquiera de estos módulos, consulte su sección correspondiente en este manual de usuario.

Asimismo, en el sistema remoto también es necesario que considere los parámetros requeridos y los opcionales que se pueden agregar en la línea de comandos para la configuración del sistema. La sintaxis es:

```
A:\REMOTA ARCHIVO.MDM [/puerto] ' [/S:velocidad]
```

ARCHIVO.MDM

Especifica el nombre y la trayectoria del archivo de configuración del modem. Es necesario que indique su trayectoria completa y la extensión .MDM.

Los parámetros puerto y velocidad siguen las mismas consideraciones que para el programa MONITOR explicado anteriormente.

Al proporcionar los parámetros descritos para los sistemas central y remoto, deben tomarse en cuenta las siguientes consideraciones generales:

- Los parámetros son excluyentes: Si se proporcionan dos o más valores diferentes para un mismo parámetro, el programa finalizará desplegando un mensaje de error.
- No importa el orden en que se proporcionen los parámetros al teclearlos en la línea de comandos.
- Si se indica la opción:

/?

los programas proporcionan ayuda en línea y los demás parámetros se ignoran.

IV.4 Visualización del estado de la Red

El Sistema de Monitoreo de la Red de Telefonía Digital muestra en pantalla, de forma *instantánea*, los valores de los parámetros que determinan el entorno de funcionamiento de los equipos de conmutación, que son:

- La *temperatura* sensada en el local del equipo de conmutación, en grados centígrados, con una precisión de un décimo de grado. Representada por la columna *Temp.* ($^{\circ}\text{C}$).
- El *voltaje de las baterías* de respaldo para el equipo de conmutación (columna *Bate.* (V)), en volts y también con una precisión de una décima de volt.
- La presencia de *las alarmas mayor y menor*, en las columnas *May.* y *Men.* respectivamente.
- La ausencia del suministro *energía eléctrica* para cada equipo de conmutación, en la columna cuyo título es *Pot.*

Se representa además, el estado en que se encuentran los enlaces entre el nodo de DGSCA y los demás que conforman la Red, obteniendo un indicador relacionando el número de *lecturas recibidas* de cada nodo remoto (R_x) y el total de *intentos* realizados por recibirlas (I_x). Esta relación se muestra en forma porcentual, con una precisión de una décima; en la columna R_x/I_x .

Para la visualización del estado de la Red, el Sistema utiliza tres pantallas, cada una de las cuales puede mostrar en forma jerárquica un máximo de trece nodos, agrupados de acuerdo a los siguientes criterios:

- Los nodos se muestran de acuerdo con la prioridad de atención a la condición reportada, de tal manera que *dentro de una misma pantalla, el primer nodo mostrado será el que requiera mayor atención por parte de los operadores de la Red*. Si algún nodo presenta un parámetro de los referidos anteriormente con un valor que pueda ser catalogado como alarma crítica, se le asigna la *mayor prioridad de atención*.

- En caso de que se tengan nodos con la misma prioridad de atención, se mostrarán según el orden en que hayan sido leídos de los archivos de inicialización del Sistema.

Después de que el Sistema termina de establecer un enlace de comunicaciones con algún nodo remoto, muestra *automáticamente* la primera pantalla, que es la que contiene los nodos que requieren una mayor atención; a menos que se haya presionado previamente alguna de las teclas de desplazamiento vertical. Si es éste el caso, el Sistema indicará a la derecha de la pantalla en una gráfica con una barra deslizante, la porción del conjunto de nodos que se estará observando.

En caso de que no haya nodos que mostrar en la pantalla que se seleccionó, se mostrará la leyenda:

No hay nodos que puedan desplegarse en esta pantalla

IV.4.1 Despliegue en pantalla

Para determinar la prioridad de atención que deberá asignarle a cada nodo, y por lo tanto, en qué pantalla será mostrado, el Sistema hace una clasificación según los valores obtenidos para cada uno de los parámetros monitoreados y considera como *alarma* un valor fuera de las expectativas.

Aquí se presenta la clasificación de alarmas y sus atributos durante el despliegue cuando está disponible un monitor a color.

Críticas. Merecen la mayor atención por parte de los operadores de la Red porque pueden ser la causa de que el equipo de conmutación quede fuera de operación o son un indicativo de que el enlace con el nodo remoto se ha interrumpido. El Sistema considera como alarmas críticas las siguientes condiciones:

- El valor de temperatura obtenido es mayor o igual a 23.0 °C.
- El voltaje leído del banco de baterías del equipo de conmutación es menor a 48.0 V.
- La alarma mayor del conmutador está presente.
- Se detecta la ausencia del suministro de energía eléctrica para el equipo.

Todas ellas se muestran en pantalla con texto en color amarillo sobre fondo rojo, mientras que cuando:

- La relación Rx/Ix sea igual a cero por ciento, en la pantalla se mostrará con atributos normales (texto gris con fondo azul) la leyenda:

No se recibió lectura válida en xxx intentos

Precautorias. Indican la posibilidad de un problema de muchas mayores consecuencias si no se les atiende oportunamente. Las siguientes condiciones son consideradas como alarmas precautorias:

- Un valor de temperatura que sea mayor o igual a los 20.0 °C, pero menor a los 23.0 °C.
- El voltaje del banco de baterías es mayor a 48.0 V, pero menor a los 50.0V.
- La alarma menor del conmutador está presente.

Estas alarmas son desplegadas con texto en color amarillo sobre el fondo normal del área de decodificación de las lecturas.

De operación. Pueden presentarse debido a la avería de los sensores de temperatura y voltaje del banco de baterías de respaldo que se encuentran en el local de cada equipo de conmutación y tienen la misma prioridad y atributos de despliegue que una alarma crítica. Se indican con la cadena de caracteres:

--.-

para poder diferenciarla de cualquier otro valor sensado por el Sistema.

De comunicaciones. Si el enlace entre el modem del nodo central y el del remoto no resultó exitoso, entonces se considera que existe un problema de comunicaciones y aparece la leyenda:

No se estableció comunicación con este nodo

Además de las alarmas críticas, precautorias, de operación y comunicaciones, se considera también el caso en el que un nodo está habilitado para el monitoreo, pero que sin embargo, todavía no se ha intentado establecer comunicación con él. La totalidad de los casos considerados durante el despliegue de la información de los nodos queda resumida por la tabla IV.4.1. La figura IV.4.1 muestra un ejemplo de las pantallas de visualización del estado de la Red.

Tabla IV.4.1 Prioridad con que son mostrados los nodos en la pantalla de visualización del estado de la Red

Prioridad	Qué es desplegado en pantalla:
la más alta	las alarmas críticas que fueron detectadas las alarmas precautorias registradas
↑ 	"No se recibió lectura válida en xxx intentos" "No se estableció comunicación con este nodo" el último estado registrado del nodo remoto en atributos normales de despliegue (No existieron alarmas) "No se han recibido lecturas desde la alta"
la más baja	Nada. No se despliega este nodo ya que está dado de baja y por lo tanto inhabilitado su monitoreo.

Sistema de Monitoreo de la Red						
Nodo	Temp.(°C)	Bate.(U)	May.	Men.	Pot.	Tx/Rx
NP-1 ARQUITECTURA	23.6	50.3	NO	NO	SI	100.0
NP-2 TORRE II	21.9	50.3	NO	NO	NO	100.0
NP-3 I.I.M.A.S.	21.9	50.3	SI	NO	NO	100.0
NP-4 D.G.S.C.A.	23.6	50.3	SI	SI	SI	100.0
NP-5 ZONA CULTURAL	21.9	50.3	NO	SI	NO	100.0
S-1 RECTORIA	21.9	50.3	NO	SI	NO	100.0
S-2 ECONOMIA	No se recibió lectura válida en 15 intentos					
S-3 PERSONAL	No se recibió lectura válida en 10 intentos					
S-4 INGENIERIA	No se estableció comunicación con este nodo					
S-6 MEDICINA	No se estableció comunicación con este nodo					
S-7 VETERINARIA	20.0	50.0	NO	NO	NO	100.0
S-8 GEOGRAFIA	No se han recibido lecturas desde la alta					
S-9 QUIMICA	No se han recibido lecturas desde la alta					
Canal de Comunicaciones						
Contactando con: VETERINARIA						

Figura IV.4.1 Ejemplo de las pantallas de visualización del estado de la Red

IV.4.2 Registro y liberación de alarmas

Debido a que es necesario hacer del conocimiento de los operadores de la Red de Telefonía, las situaciones anormales que se han registrado, el Sistema de Monitoreo señala los nodos en los que se ha presentado alguna alarma de interés, anteponiendo un asterisco a su abreviatura y reteniendo en memoria los valores de los parámetros sensados en ese momento. Esta señalización se puede eliminar únicamente cuando se consultan las condiciones que generaron la alarma a través de la utilería *Libera Estado de Alarmas* (LEA).

Para acceder dicha utilería, se presiona la tecla <F10> en cualquier pantalla de visualización del estado de la Red. En una ventana aparecerán los nombres de los nodos en los que se detectó una situación extraordinaria. Para consultar la información disponible, deberá posicionarse en el nodo deseado utilizando las teclas de desplazamiento vertical y oprimir la tecla <Enter>. En la figura IV.4.2 se muestra un ejemplo de la pantalla desplegada.

Si desea eliminar la señalización algún nodo alarmado, conteste afirmativamente a la pregunta mostrada en la ventana: ¿Se libera esta alarma?. Una vez consultada la información requerida, oprima la tecla <F3> para continuar con el monitoreo del Sistema. En las pantallas de visualización aparecerán los nodos consultados sin el asterisco al principio de su abreviatura.

Nodo	Temp.(°C)	Date.(U)	May.	Men.	Pot.	Tx/Rx
*NP-1 ARQUITECTURA					SI	100.0
*NP-2 TORRE II						
NP-3 I.I.M.A.S.						
NP-4 D.G.S.C.A.						
*NP-5 ZONA CULTU						
S-1 RECTORIA						
S-2 ECONOMIA						
S-3 PERSONAL						
*S-4 INGENIERIA						
*S-6 MEDICINA						
S-7 VETERINARIA						
S-8 GEOGRAFIA						
S-9 QUIMICA						

Nodos en los que se observó alarma

ARQUITECTURA
TORRE II
ZONA CULTURAL
INGENIERIA
MEDICINA

F3 SALIR

No se han recibido

No se estableció comunicación con este nodo en modo interno de marcación

El: Sáb Dic 14 12:52:43 '94

¿Se libera esta alarma?
<S/N>

Contactando con: VETERINARIA

Salir

Figura IV.4.2 Ejemplo de una pantalla de la utilería de liberación de alarmas

IV.5 Interfaz con el usuario en los módulos del sistema

En los módulos de mantenimiento y consultas se presenta una interfaz consistente que le permite al usuario en forma sencilla y amigable efectuar los procedimientos descritos en este manual de usuario.

La figura IV.5.1 ejemplifica las diferentes partes que generalmente conforman una pantalla:

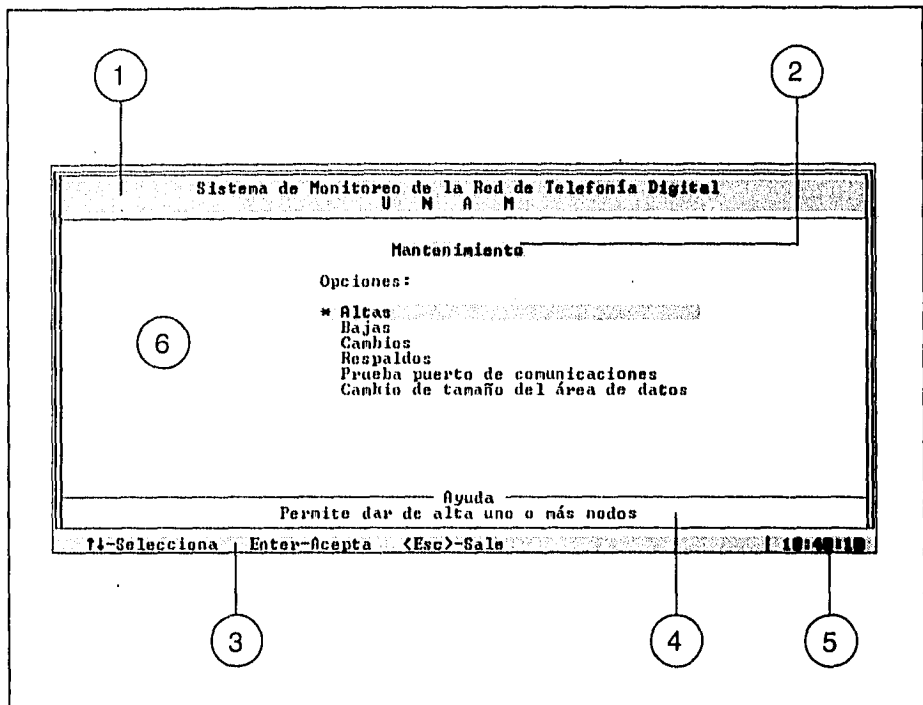


Figura IV.5.1 Ejemplo ilustrativo de la interfaz con el usuario en las utilerías del sistema

Como se observa en la figura, se presentan las siguientes partes:

- En cualquier pantalla aparecerá como encabezado el nombre del programa: Sistema de Monitoreo de la Red de Telefonía Digital, UNAM (punto 1 en la figura).

- El nombre del módulo generalmente aparece en la posición indicada por el punto 2. En este caso, la pantalla mostrada corresponde a un menú del subsistema de mantenimiento.
- En la parte inferior izquierda aparece una barra de funciones (punto 3) que muestra las teclas válidas según el contexto en el que el usuario se encuentre.
- En la parte inferior aparece la ventana de ayuda (punto 4), utilizada para desplegar mensajes de error, textos explicatorios de las opciones que se pueden seleccionar y en algunos casos la situación actual del proceso que se está ejecutando.
- La hora del sistema aparece en la esquina inferior derecha, indica que el programa está ejecutándose (punto 5).
- Al centro de la pantalla se presenta el área de trabajo, en ella se solicitan opciones y se muestran los resultados. En la figura se indica con el número 6. La posición del cursor generalmente se indica con un asterisco a la izquierda de la opción, lo que permite visualizar correctamente la opción seleccionada en caso de que se ejecute el programa en un monitor monocromático.

A fin de proporcionar una referencia rápida de los pasos requeridos para realizar alguna tarea, a lo largo de este manual especificaremos las opciones que se deben elegir escribiéndolas separadas entre sí por una diagonal.

Ejemplo: Para incorporar un nodo al sistema hay que entrar al módulo de mantenimiento y elegir la opción de altas. Estos pasos los representaremos con:

Mantenimiento / Altas

IV.6 Módulo de mantenimiento

A través de este módulo se pueden especificar los nodos que se consideran para el monitoreo de la Red, así como sus números telefónicos. También aquí se controla el espacio reservado en el disco duro para los datos del Sistema, permite convertir esta información del formato interno a un archivo dBASE y comprueba el funcionamiento de los puertos de comunicaciones disponibles. Las opciones principales del módulo se muestran en el siguiente diagrama.

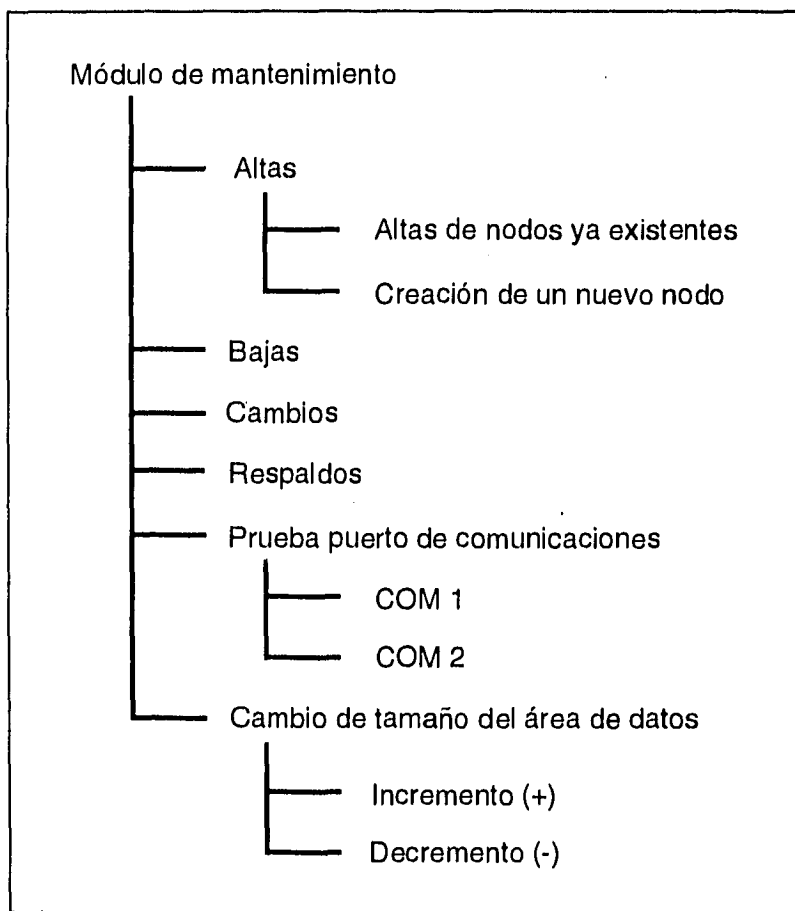


Figura IV.6.1 Estructura del módulo de mantenimiento

El módulo se puede ejecutar presionando la tecla de función <F1> desde la pantalla principal del Sistema de Monitoreo. Un momento después aparecerá el menú principal del módulo de mantenimiento:

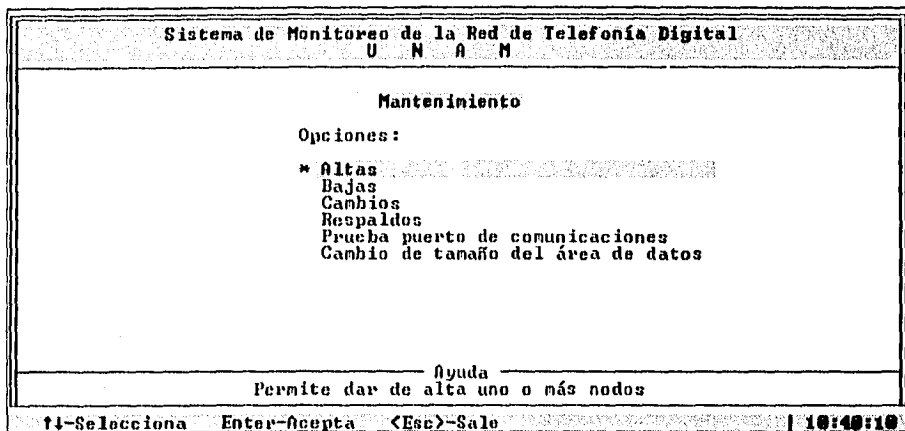


Figura IV.6.2 Menú principal del módulo de mantenimiento

Presentará las siguientes opciones:

Altas

Para incorporar nodos al Sistema de Monitoreo.

Bajas

Permite suspender el monitoreo de nodos ya registrados en el Sistema.

Cambios

Realiza cambios únicamente en los números telefónicos asociados a cada uno de los nodos dados de alta.

Respaldos

Al seleccionar esta opción se puede transferir la información del disco duro a un diskette.

Prueba del puerto de comunicaciones

Permite verificar el correcto funcionamiento de los puertos de comunicaciones disponibles para el Sistema.

IV.6.1 Altas

Con referencia a la figura IV.6.1, para entrar a este menú efectúe las siguientes selecciones:

Mantenimiento / Altas

La pantalla que aparecerá (figura IV.6.3) mostrará dos opciones

Altas de nodos ya existentes

Creación de un nuevo nodo

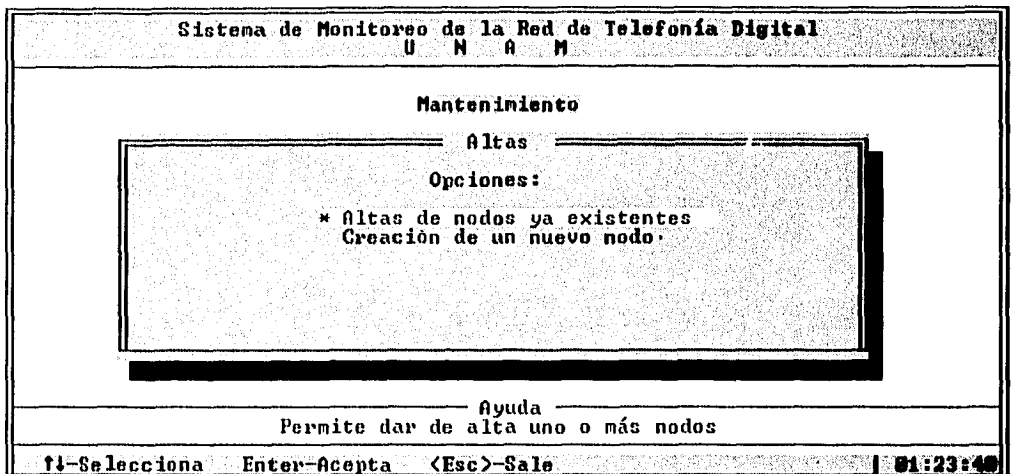


Figura IV.6.3 Opciones del menú de altas

IV.6.1.1 Altas de nodos ya existentes

Esta opción permite el monitoreo de nodos que ya han sido registrados en el Sistema pero que se encuentran dados de baja. Se elige esta opción al seguir la siguiente ruta (cf. figura IV.6.1):

Mantenimiento / Altas / Altas de nodos ya existentes

Los nodos que se encuentran dados de baja se despliegan en una ventana como se muestra en la figura IV.6.4:

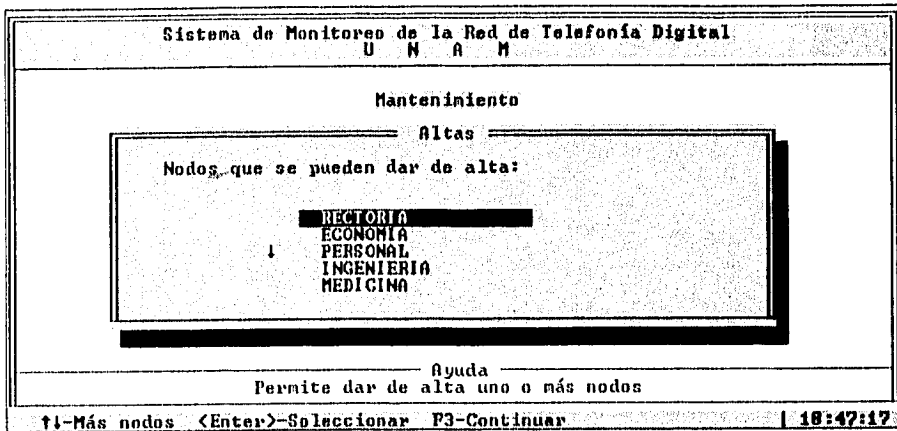


Figura IV.6.4 Nodos disponibles para darse de alta

Con las flechas de desplazamiento vertical posicione el cursor en los nodos que requiere dar de alta, y márkuelos con <Enter>. Si esta tecla se oprime nuevamente en un nodo ya seleccionado (indicado con un asterisco), se cancelará la elección. Si alcanza el final de la lista y presiona la tecla <↓>, se escuchará un sonido indicando que se ha alcanzado el último nodo registrado, y aparecerá en la *ventana de ayuda* la pregunta:

¿Desea dar de alta más nodos?

Si su respuesta es afirmativa, se le pedirá el nombre de un nuevo nodo, su abreviatura y extensión telefónica. Las consideraciones necesarias para dar de alta un nuevo nodo se presentan la siguiente sección *Creación de un nuevo nodo*.

Si su respuesta es negativa el Sistema regresará a la lista de nodos ya registrados, posicionándose en el último de ellos.

Una vez que se han escogido los nodos, o se han dado de alta nuevos, con la tecla de función <F3> se le indica al programa que continúe. Brevemente aparecerá una lista de los nodos que han sido seleccionados para incorporarse al Sistema. Finalmente, el programa regresará a la pantalla principal del módulo.

IV.6.1.2 Creación de un nuevo nodo

Es posible crear varios nodos nuevos hasta llegar a la capacidad del Sistema de treinta y nueve. Con referencia a la figura IV.6.1, con las opciones:

Mantenimiento / Altas / Creación de un nuevo nodo

para crear un nuevo nodo, la pantalla que aparecerá es la siguiente:

Sistema de Monitoreo de la Red de Telefonía Digital
U M A M

Mantenimiento

Altas

Nodos que se pueden dar de alta:

F.E.S. CUAUITLAN
PITAGORAS
NODO 27
NODO 28
NUEVO NODO

Nombre: nodo 29
Abreviatura: np29
Extensión telefónica: 55555

Ayuda
Ese número corresponde a otro nodo en el Sistema

11-Más nodos <Enter>-Seleccionar F3-Continuar | 13:03:36

Figura IV.6.5 Campos para dar de alta un nuevo nodo

En el campo que indica Nombre se deberá escribir el nombre del nodo que se pretende crear, éste puede tener un máximo de diecisiete caracteres. Una vez proporcionado el nombre del nuevo nodo, al oprimir la tecla <Enter> aparecerá el campo Abreviatura en donde deberá escribir la abreviatura con que se conocerá al nuevo nodo, ésta puede tener un máximo de cuatro caracteres. Por último aparecerá el campo Extensión telefónica, para capturar en cinco posiciones el número de extensión telefónica a la que se marcará durante el monitoreo de la red.

Al proporcionar la información solicitada en los campos se deben tomar en cuenta los siguientes puntos:

- Los caracteres permitidos son las letras A-Z, a-z, los números 0 a 9, guión, subguión, espacio en blanco, diagonal y punto.
- El nombre y la abreviatura de los nodos deben ser únicos.
- De acuerdo a los parámetros dados al inicio de la ejecución del Sistema, éste validará que el número se encuentre dentro del plan original de numeración que se tiene para cada nodo de conmutación de la Universidad (parámetro /v), o que ignore esta validación (parámetro /nv).
- El número que se va a proporcionar para el nuevo nodo no debe estar asignado a algún otro ya existente.
- No se aceptan campos vacíos.

Estas consideraciones también deben hacerse cuando el nuevo nodo se crea en la opción de ALTAS, de la sección anterior.

IV.6.2 Bajas

Esta opción permite la suspensión del monitoreo a los nodos que se seleccionen, pero no se eliminan completamente del Sistema. En consecuencia, si se requiere monitorear algún nodo anteriormente dado de baja, no necesita crearse sino sólo darse de alta (sección IV.6.1.1 *Altas de nodos ya existentes*).

Con respecto a la figura IV.6.1, para entrar a esta opción elija:

Mantenimiento / Bajas

La pantalla que se desplegará se muestra en la figura IV.6.6:

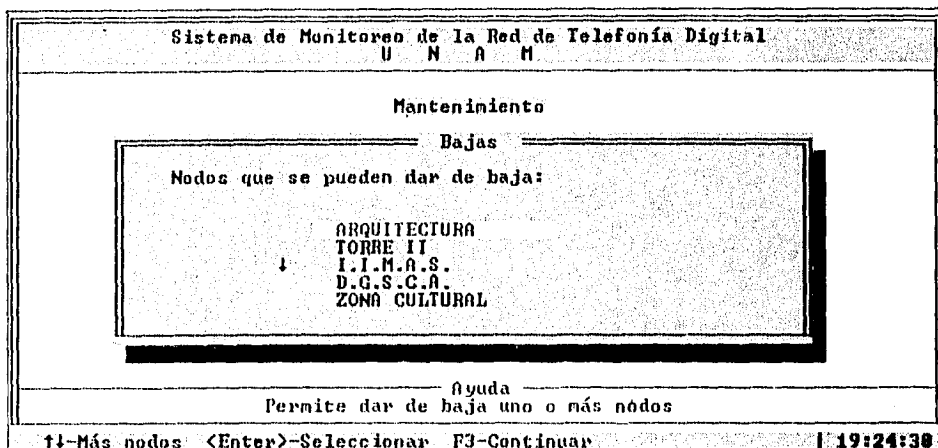


Figura IV.6.6 Nodos que se pueden dar de baja

En la pantalla se muestran los nodos que se monitorean por el Sistema. Para inhibir la marcación hacia alguno de ellos, será necesario posicionarse con las flechas de desplazamiento en su nombre y marcarlo con la tecla <Enter>. Si oprime nuevamente esta tecla en un nodo ya escogido, se cancelará la selección.

Cuando haya finalizado la elección de los nodos, oprima la tecla de función <F3> para indicarle al Sistema que continúe. Una vez realizado, se mostrará brevemente la lista de los nodos seleccionados y automáticamente regresará al menú principal del módulo de mantenimiento.

IV.6.3 Cambios

Este menú permite realizar modificaciones al número telefónico asignado a los nodos dados de alta dentro del Sistema.

Con respecto a la figura IV.6.1, el acceso a esta opción se logra seleccionando:
Mantenimiento / Cambios

La pantalla que se despliega se muestra a continuación:

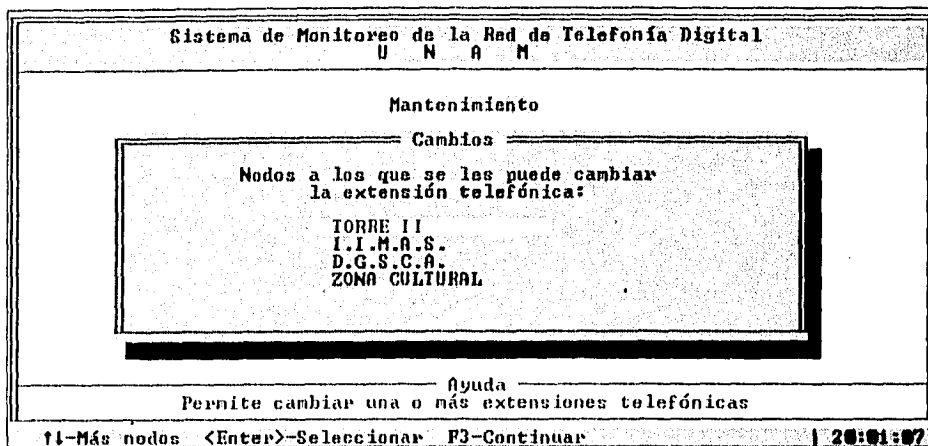


Figura IV.6.7 Nodos que se pueden modificar en su extensión telefónica

Con las flechas de desplazamiento vertical se posiciona el cursor en el nodo que requiere el cambio, y se escoge con la tecla <Enter>. Inmediatamente se desplegará una pantalla mostrando el número actual de extensión telefónica y un campo en el cual se captura la nueva extensión que se asignará al nodo.

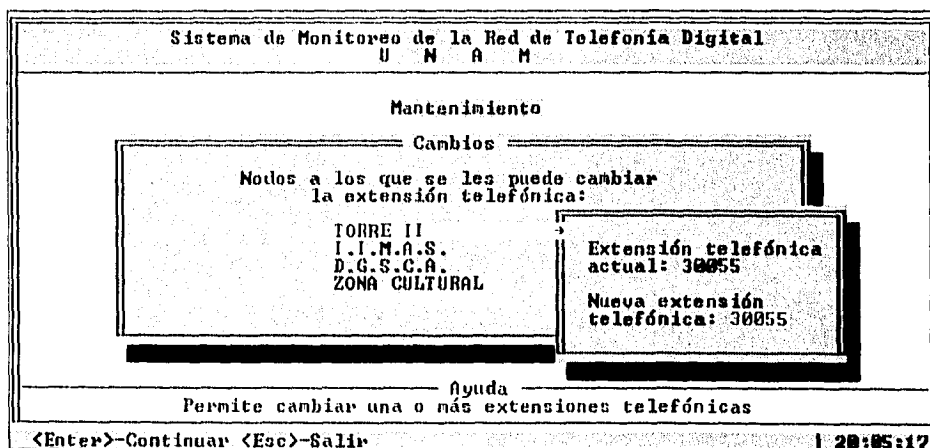


Figura IV.6.8 Cambio de extensión telefónica del nodo

Al cambiar la extensión telefónica se deben de tomar en cuenta las siguientes consideraciones:

- Si al ser ejecutado el Sistema, se utiliza el parámetro /v (que es el utilizado por defecto), validará las extensiones telefónicas de acuerdo al rango de numeración de cada nodo. Si, en cambio, se ejecuta con el parámetro /nv, el número de extensión proporcionado debe coincidir con el plan de numeración de la Universidad, que incluye números de reserva para un crecimiento futuro.
- El número de extensión no debe repetirse con los ya existentes (cf. figura IV.6.8b).

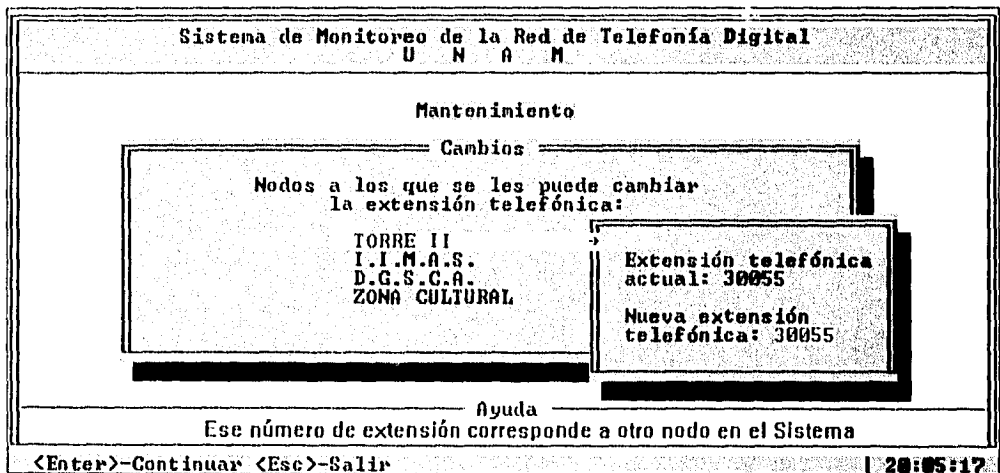


Figura IV.6.8b Mensaje de error en el cambio de extensión telefónica de un nodo

Puede indicar al Sistema que se han finalizado los cambios presionando la tecla de función <F3>. Se mostrará brevemente la lista de los nodos que se han modificado y el Sistema regresará automáticamente al menú principal del módulo de mantenimiento.

IV.6.4 Respaldos

Esta utilidad le permite transferir la información acumulada por la operación del sistema a un diskette cuando se sabe que se va a saturar el espacio asignado en el disco duro, o bien cuando lo requiera como parte de su plan de respaldos. Este proceso libera espacio de almacenamiento, de modo que el programa puede continuar en forma segura con el monitoreo y almacenamiento de los datos del estado de los nodos de la red.

Con referencia a la figura IV.6.1, el acceso a este módulo desde el Sistema de Monitoreo se logra a través de:

Mantenimiento / Respaldos

Puede también ejecutar independientemente esta utilidad tecleando su nombre desde la línea de comandos del MS-DOS:

C:\MONITOR\EXE\BKUP

En caso de que exista más de una unidad de diskette, el programa inicia solicitando aquélla en la que se efectuará el respaldo (fig. IV.6.9). Después de pedir que se inserte un diskette ya formateado (fig. IV.6.10) y esperar a que usted presione la tecla de función <F3>, verifica que se encuentre colocado correctamente, que tenga un formato adecuado y que no esté protegido contra escritura.

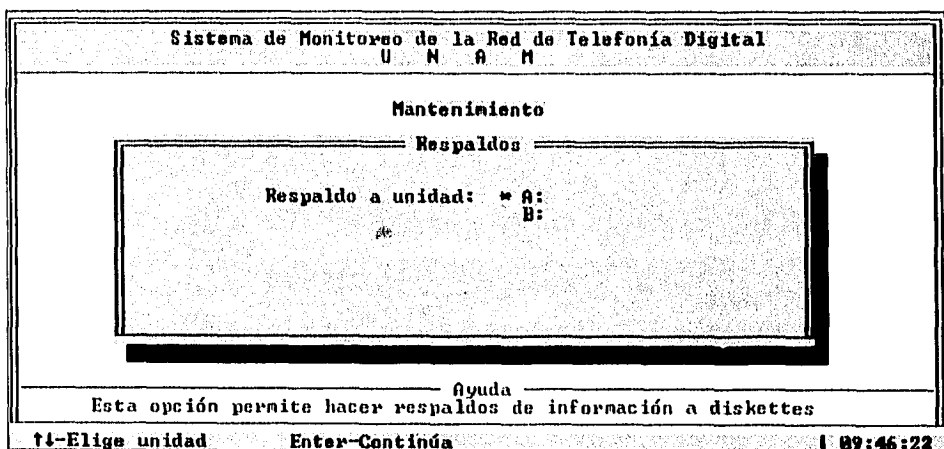


Figura IV.6.9 Solicitud de la unidad para efectuar el respaldo

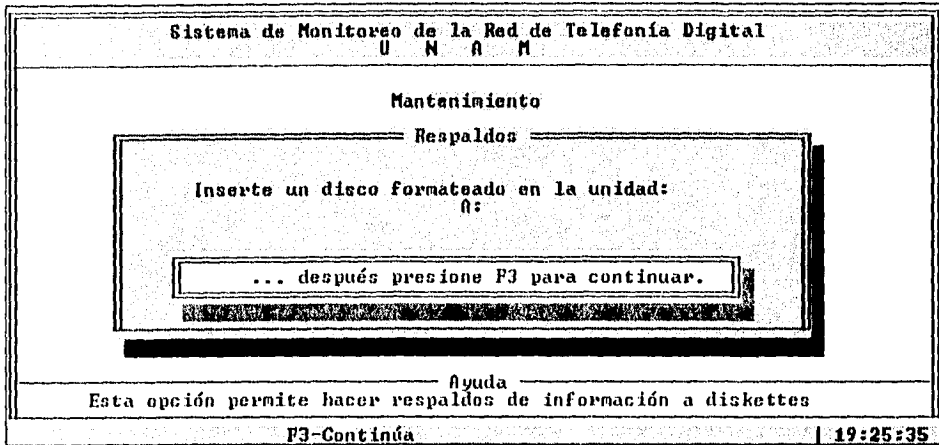


Figura IV.6.10 Solicitud de inserción del disco de respaldo en la unidad seleccionada

Posteriormente realiza un *formato rápido*¹ al diskette en el que se efectuará la transferencia de la información. Debe tomarse en cuenta que el disco que se inserte perderá toda la información que contenga, es por esta razón que el programa despliega un aviso que indica que toda la información contenida en él se perderá (fig. IV.6.11), al mismo tiempo que pide la confirmación antes del inicio del proceso. Podrá indicar al sistema que inicie el respaldo, si responde afirmativamente a la pregunta ¿Continúa?, que aparece en la pantalla.

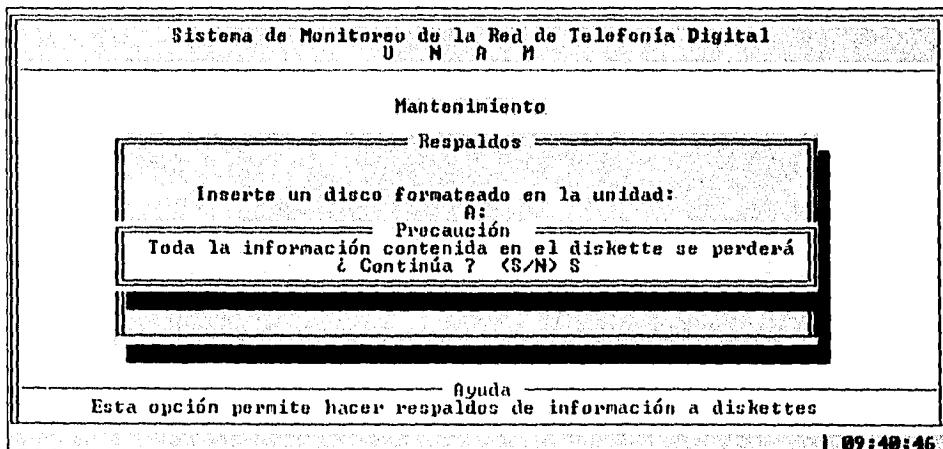


Figura IV.6.11 Solicitud de confirmación de inicio del proceso de respaldo

¹ Este tipo de formato hace uso de la estructura de un disco previamente formateado para acelerar el proceso. Únicamente se reconstruyen el directorio y las dos tablas de FAT.

A pesar de que en este paso el programa hace una verificación del estado de la superficie del disco, recomendamos previamente darle formato al diskette aunque sea nuevo y verificar que no contenga ningún sector dañado, ya que si se almacenara información sobre alguno de ellos, se perdería completamente. La siguiente imagen se mostrará, en el momento en que se inicia el proceso:

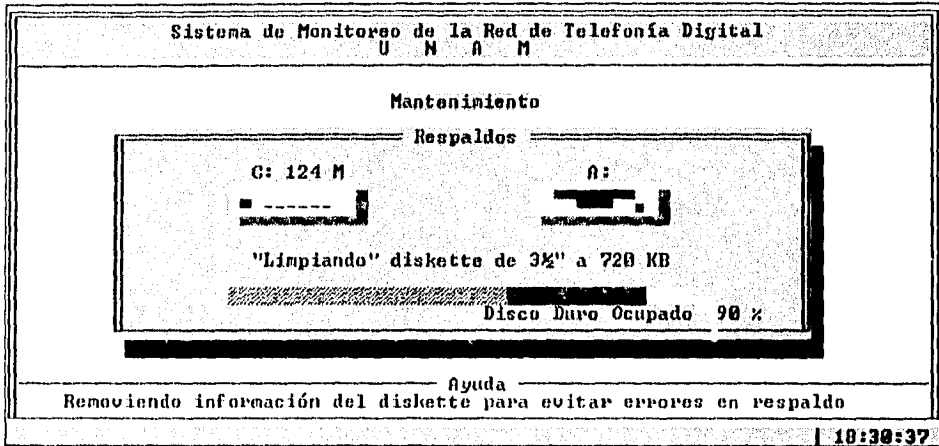


Figura IV.6.12 Formato rápido del diskette de respaldo

Una vez que al disco se le ha borrado la información que contenía, se realiza el proceso de conversión de un archivo de lecturas a una base de datos en formato dBASE (fig. IV.6.13).

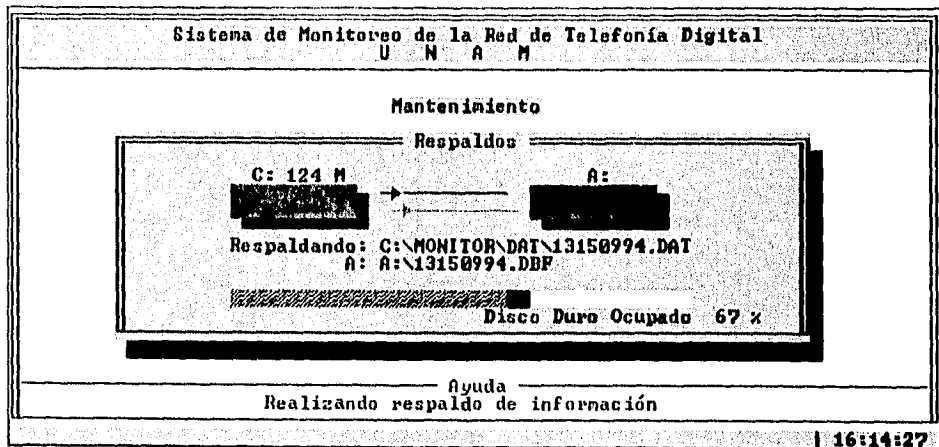


Figura IV.6.13 Conversión de un archivo de lecturas a una base de datos dBASE

El archivo dBASE generado contiene los siguientes campos:

Nombre del campo	Tipo	Ancho (bytes)	Contenido
NODO	Carácter	4	Abreviatura del nombre del nodo al que pertenece la información del registro.
FECHA	Fecha	8	La fecha en la que se generó el registro, en el formato mm/dd/aa.
HORA	Carácter	6	La hora de la lectura en el formato hhmmss.
TEMPERAT	Numérico	4	Temperatura del local del conmutador en grados centígrados y con precisión a un décimo de unidad.
BATERIAS	Numérico	4	Nivel de voltaje de las baterías, en voltios y con precisión a un décimo de unidad.
ALARMA_MAY	Lógico	1	La existencia de la alarma mayor durante la lectura se indica con T y el caso contrario con F.
ALARMA_MEN	Lógico	1	La existencia de la alarma menor durante la lectura se indica con T y el caso contrario con F.
FALLA_POW	Lógico	1	La falla en el suministro de energía eléctrica de AC se indica con T y el caso contrario con F.
TASA_RX_I	Numérico	5	Tasa de lecturas recibidas sin error, contra el total de intentos de transmisión, en porcentaje.

Una vez que se ha agotado el espacio disponible en el diskette y existen otros archivos que no han sido respaldados, el sistema le preguntará si desea seguir exportando información, repitiéndose nuevamente todo el proceso. Cuando no hay más archivos por transferir, se enviará un mensaje indicándolo (fig. IV.6.14), y el programa regresará al menú principal del módulo de mantenimiento.

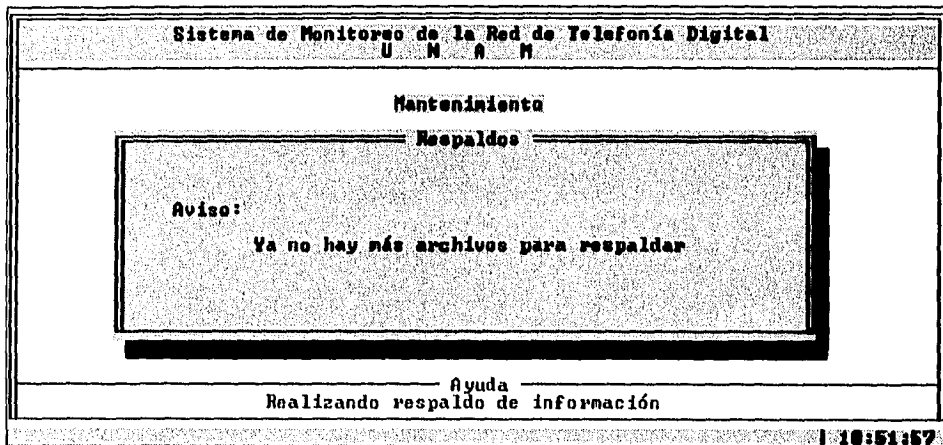


Figura IV.6.14 Aviso, no existen más archivos de lecturas

IV.6.5 Prueba de los puertos de comunicaciones

A pesar de que durante la inicialización del sistema se efectúa una prueba interna del puerto al cual se tiene conectado el modem, algunos problemas podrían no detectarse². Cuando se sospecha que pueden estar ocurriendo errores durante la comunicación de los datos, resulta conveniente realizar una prueba completa del puerto serial. Para este propósito se desarrolló un programa que determina la confiabilidad de los puertos que el sistema de monitoreo puede utilizar (puertos COM1 en 0x3f8 y COM2 en 0x2f8).

La utilería puede ejecutarse desde el módulo de mantenimiento del Sistema de Monitoreo a través de la ruta (ver figura IV.6.1) :

Mantenimiento / Prueba puerto comunicaciones

También se puede ejecutar independientemente tecleando su nombre desde la línea de comandos del MSDOS:

C:\MONITOR\EXE\PPTO

Si la máquina tiene los dos puertos seriales se presentará la siguiente pantalla:

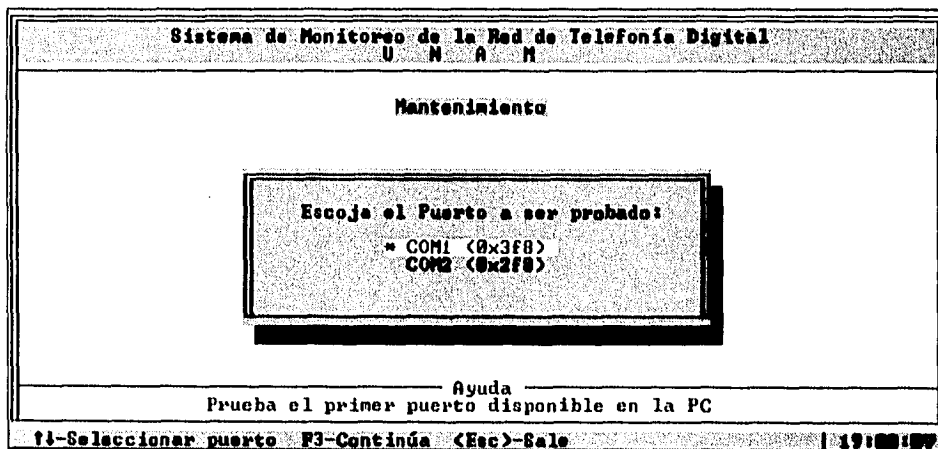


Figura IV.6.15 Pantalla de selección del puerto a probar

² Puede suceder que el controlador de comunicaciones se encuentre en buen estado pero los manejadores de línea estén dañados. La prueba interna no detectará el error, siendo necesaria una prueba con una conexión externa.

Las flechas de desplazamiento vertical le permiten seleccionar el puerto que se quiere probar, y con la tecla de función <F3> se le indica al programa que continúe.

En caso de que la máquina cuente únicamente con un puerto serial, no se mostrará la pantalla anterior, pasará directamente a la pantalla que se muestra a continuación:

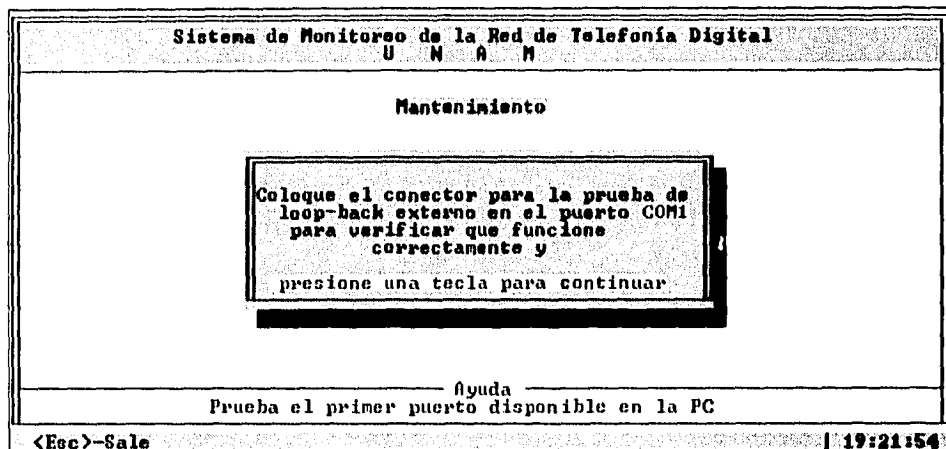


Figura IV.6.16 Solicitud de inserción del conector externo

El aviso mostrado le pide colocar un conector para la prueba de *loop-back* externa en el puerto seleccionado. Este conector puede ser DB-25 o DB-9, del tipo hembra y deberá tener la configuración NULL MODEM mostrada en la figura IV.6.17.

Después de colocar el conector y presionar cualquier tecla, se inicia el proceso. El desarrollo de la prueba se mostrará en la pantalla como se muestra en la figura IV.6.18. Se verifica el estado de las líneas CTS, RTS, DSR, DCD y DTR; el resultado se indica en la parte superior. La sección de transmisión y recepción se prueba a las velocidades que el sistema puede utilizar: 300, 600, 1200, 2400, 4800 y 9600 bps. Durante esta prueba en la parte inferior se despliegan en dos ventanas los caracteres transmitidos y recibidos. Si se detecta algún funcionamiento anormal el programa intentará la operación dos veces más. Finalmente se mostrará el diagnóstico del estado del puerto.

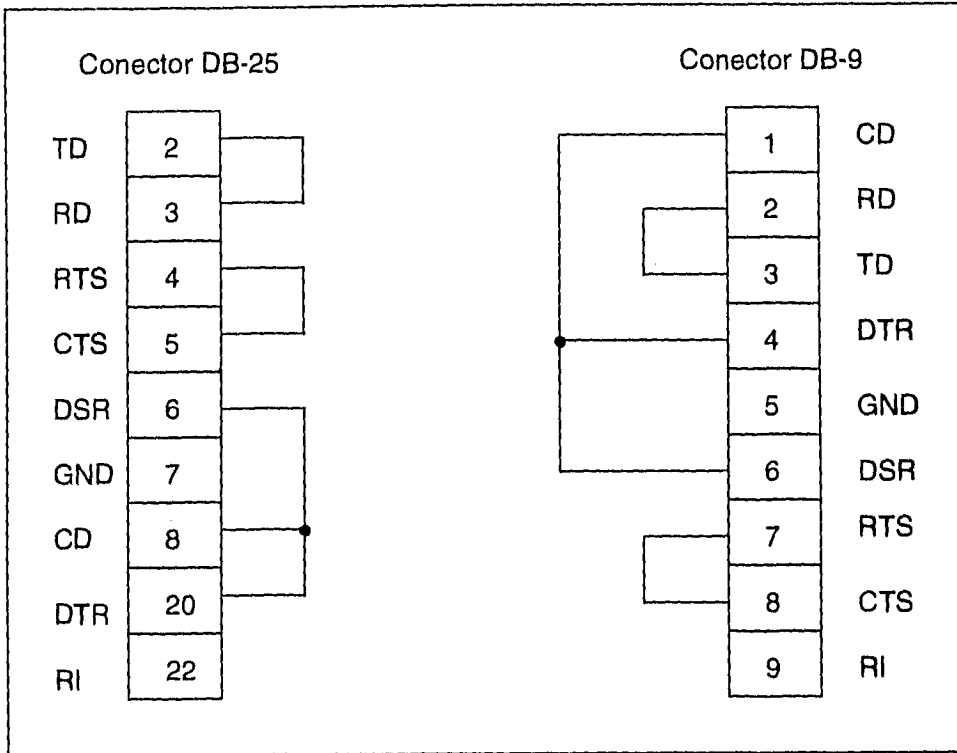


Figura IV.6.17 Configuración del conector para la prueba externa del puerto

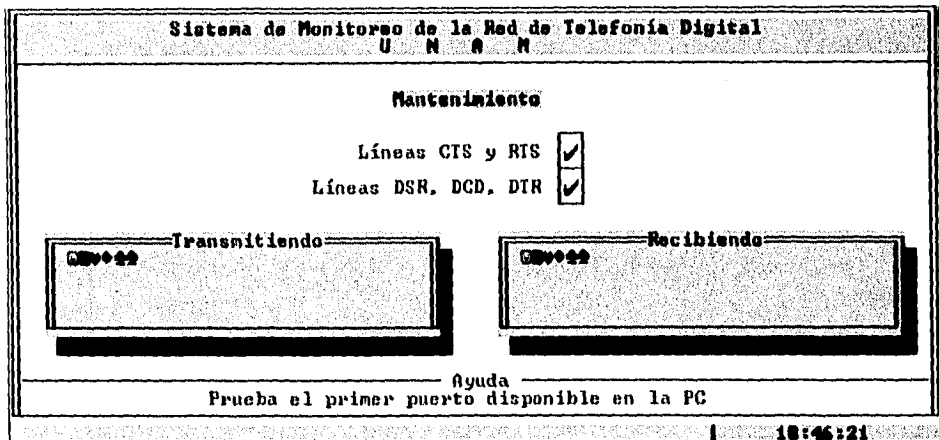


Figura IV.6.18 Desarrollo de la prueba del puerto de comunicaciones seleccionado

IV.6.6 Cambio del tamaño del área de datos

Este módulo le permite realizar cambios en el tamaño del área reservada para el almacenamiento de los datos del Sistema. De esta forma, se le puede adecuar a las condiciones de espacio en disco disponible.

Con referencia a la figura IV.6.19, para ejecutar este programa se debe seguir la siguiente ruta:

Mantenimiento / Cambio del tamaño del área de datos

Cuando el programa inicia se despliega la siguiente pantalla:

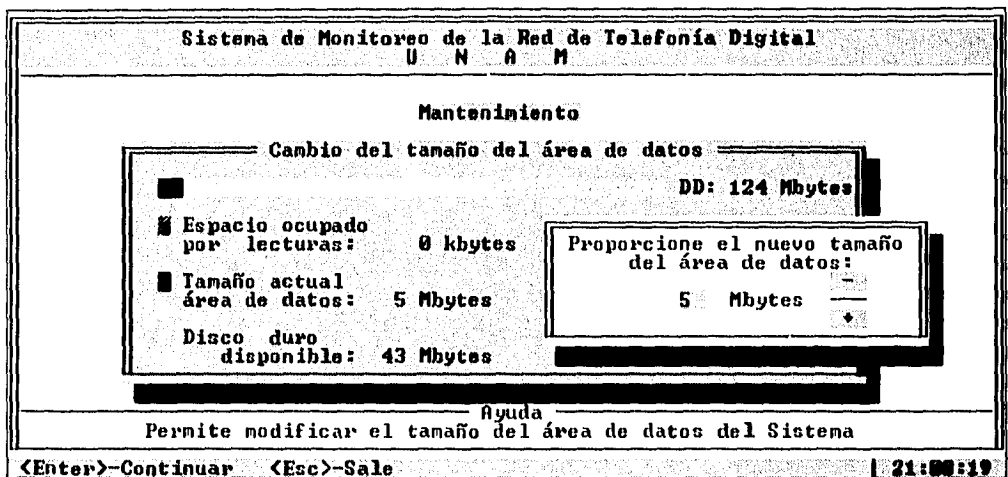


Figura IV.6.19 Pantalla de modificación del espacio asignado a datos

Como se muestra en la figura, se indica en una gráfica el espacio ocupado por las lecturas, el disponible dentro de la asignada al almacenamiento de datos, el ocupado por cualquier otra información y el libre en el disco. También se pueden observar los valores numéricos de los parámetros más importantes. A la derecha aparece una ventana que permite modificar el tamaño del área de datos.

El cambio se realiza presionando las teclas <+> o <-> para aumentarla o disminuirla, respectivamente. Los valores máximo y mínimo permitidos son de cinco y veinte Mbytes.

Una vez alcanzado el valor correcto, presione la tecla <Enter> para guardar el cambio y terminar. La tecla <Esc> le permite salir del programa descartando las modificaciones.

Este procedimiento le permitirá definir el área de datos más adecuada y además ajustar el período que debe transcurrir entre cada ciclo de respaldos a diskette.

IV.7 Módulo de consultas

El módulo de consultas le permite conocer el comportamiento de las condiciones de los locales de los conmutadores durante un período de tiempo. El programa genera los reportes en base a la consulta de la información generada durante la operación del sistema y almacenada en el disco duro de la computadora central. Se pueden realizar consultas de dos tipos: unos son reportes cuya salida se direcciona a un archivo tipo texto, otros son reportes con la salida de información a pantalla. Los primeros son adecuados para la generación de reportes extensos y los segundos permiten una consulta interactiva. Adicionalmente se pueden elegir las características del reporte lo que permite una búsqueda selectiva de información.

El programa se puede ejecutar desde la pantalla principal del Sistema de Monitoreo pulsando la tecla <F2> o desde la línea de comandos del MS-DOS con el comando:

```
C:\MONITOR\EXE\CONSULTA
```

Al entrar a este módulo, en una ventana se mostrarán los nodos que se encuentran disponibles para su consulta, como lo muestra la figura IV.7.1.

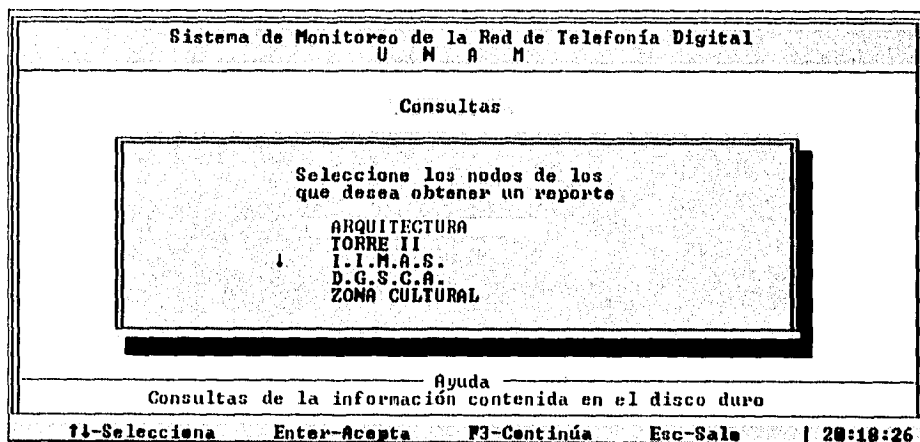


Figura IV.7.1 Selección de nodos para incluir en el reporte

Como se indica en la barra de funciones, las flechas de desplazamiento vertical permiten posicionarse en algún nodo. En la ventana se muestran los primeros cinco nodos, cuando existen otros que no se están desplegando, aparece una pequeña flecha a la izquierda de la lista que indica el sentido de la tecla de desplazamiento vertical que se debe presionar para mostrar los otros nodos ocultos. Después de posicionarse sobre algún nodo que se desea incluir en el reporte, se debe marcar con la tecla <Enter>; cada nodo que ha sido marcado se desplegará con un asterisco a la izquierda, pudiendo eliminar la selección si se posiciona el cursor nuevamente sobre él y se tecléa <Enter>. Una vez que se han seleccionado los nodos, se puede continuar presionando la tecla de función <F3>.

Existe la opción de abandonar el módulo desde esta pantalla al presionar la tecla <Esc>. Sin embargo, antes salir, se le pedirá que confirme su selección.

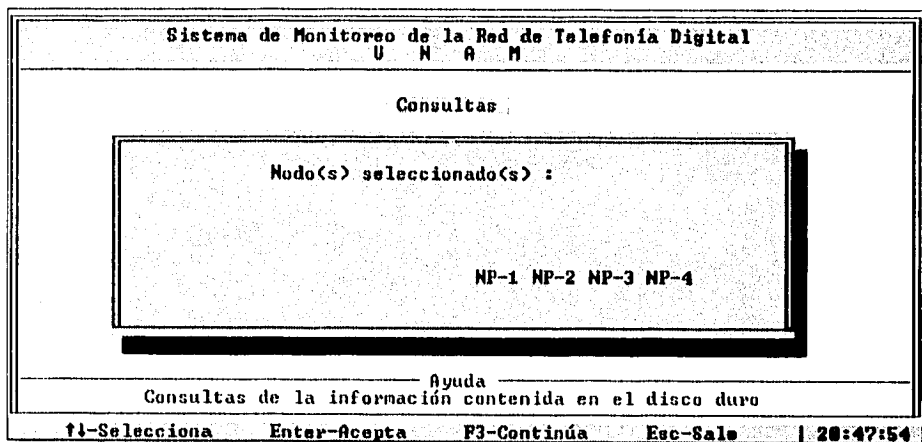


Figura IV.7.2 Despliegue de los nodos seleccionados

Después de teclear <F3> el programa mostrará brevemente las abreviaturas de los nombres de los nodos que se seleccionaron (figura IV.7.2) y se desplegará la ventana de la figura IV.7.3. Esta pantalla le permitirá determinar si el reporte se realizará de todos los registros disponibles o únicamente de aquellos con características especiales.

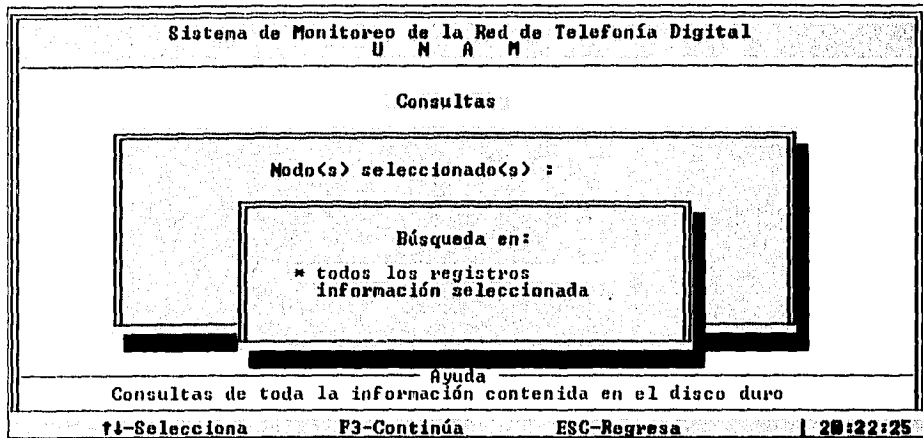


Figura IV.7.3 Selección del tipo de búsqueda

Con las teclas de desplazamiento vertical posiciónese en la opción requerida y presione la tecla de función <F3> para continuar. La tecla <Esc> le permite regresar a la pantalla anterior. Si usted seleccionó búsqueda sobre información seleccionada, aparecerá otra pantalla (figura IV.7.4) en la que puede establecer los criterios que determinan si un registro se incluye en el reporte.

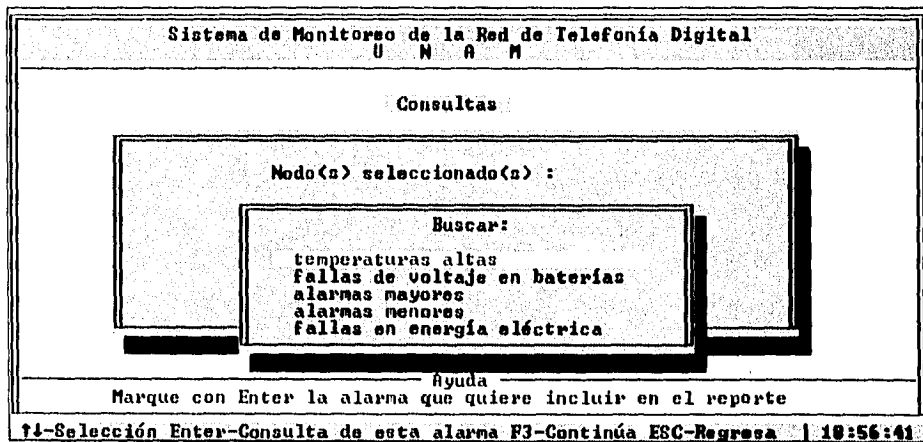


Figura IV.7.4 Selección de criterios para el reporte

Usando las flechas de desplazamiento vertical posiciónese en el parámetro que desea que se considere al generar el reporte y márkelo con <Enter>. Para continuar presione la tecla de función <F3>.

Si usted seleccionó fallas de voltaje en baterías o temperaturas altas, se le preguntará el voltaje y la temperatura de umbral a través de una pequeña ventana que aparecerá a la derecha como se muestra en la figura IV.7.5.

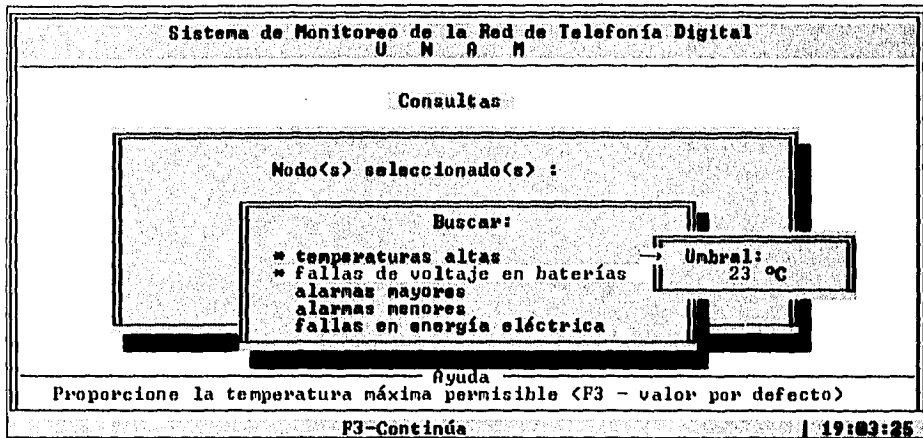


Figura IV.7.5 Valores de umbral en los criterios de selección

Si la lectura al voltaje de las baterías del conmutador es menor al voltaje de umbral proporcionado, o bien, si la temperatura detectada es mayor al valor proporcionado como temperatura de umbral, el registro se incluirá en el reporte. En esta pantalla con la tecla de función <F3> puede aceptar el valor por defecto mostrado en la ventana o puede teclear un valor diferente. Es importante tener en cuenta que basta con que se cumpla uno de los criterios de búsqueda seleccionados para que el registro correspondiente aparezca en el reporte.

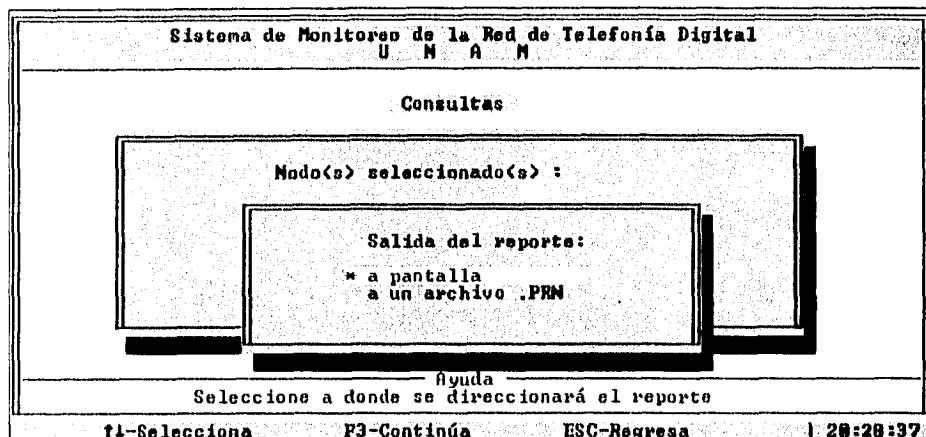


Figura IV.7.6 Selección del medio de salida del reporte

La siguiente ventana (figura IV.7.6) le permite establecer el medio de salida de la información, éste puede ser la pantalla o un archivo para impresión. Posiciónese en el medio de salida deseado y presione la tecla de función <F3>. Si seleccionó como medio de salida un archivo, se le preguntará el formato con el que desea que aparezca ordenada la información, por nombre de nodo o cronológicamente conforme se fueron generando los registros (figura IV.7.7). Posiciónese en el formato de salida correcto y presione la tecla de función <F3>.

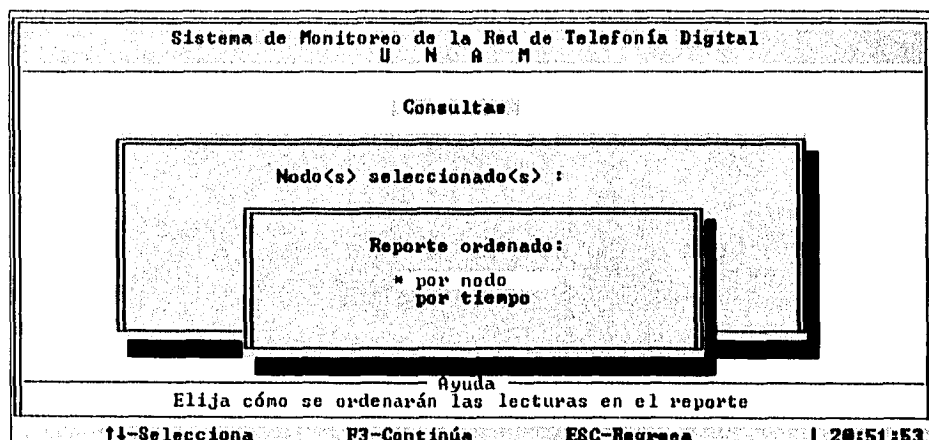


Figura IV.7.7 Selección de la forma de ordenación

En esta serie de ventanas, puede regresar a la inmediata anterior tecleando <Esc>, excepto en la primera de ellas en cuyo caso como ya se mencionó, al presionar esta tecla puede salir del programa.

Después de haber proporcionado el formato del reporte, aparecerán en pantalla los nodos y características que han sido seleccionadas. Con un formato similar al de la figura IV.7.8 se le preguntará si los datos mostrados son correctos. Si detecta algún error conteste N y el programa le volverá a solicitar las características hasta definir correctamente su reporte.

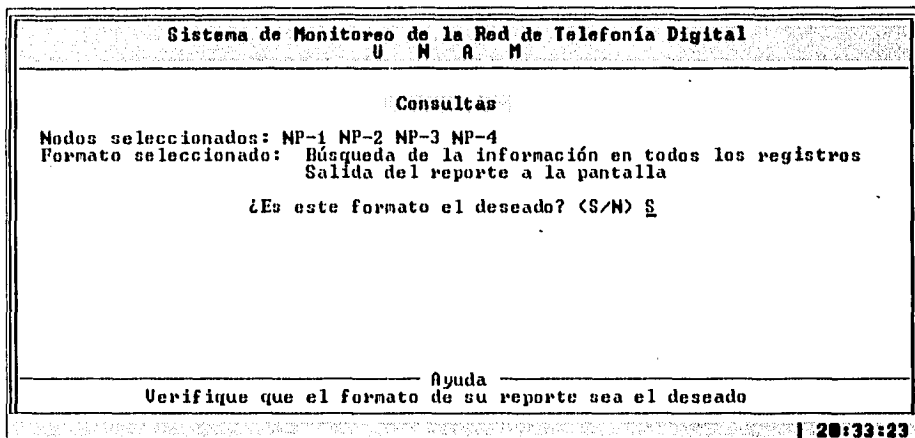


Figura IV.7.8 Nodos y características seleccionadas

Al responder afirmativamente con S a la pregunta de si el formato presentado es el correcto, en la siguiente pantalla el programa le solicitará la fecha y hora de inicio en el formato mostrado en la figura IV.7.9. En caso de que haya seleccionado reporte a un archivo, también se requerirán la fecha y hora límite. Al final de cada dato no es necesario teclear <Enter>, el programa automáticamente pasará al siguiente campo. Si es necesario, puede corregir algún dato con la tecla <Backspace>.

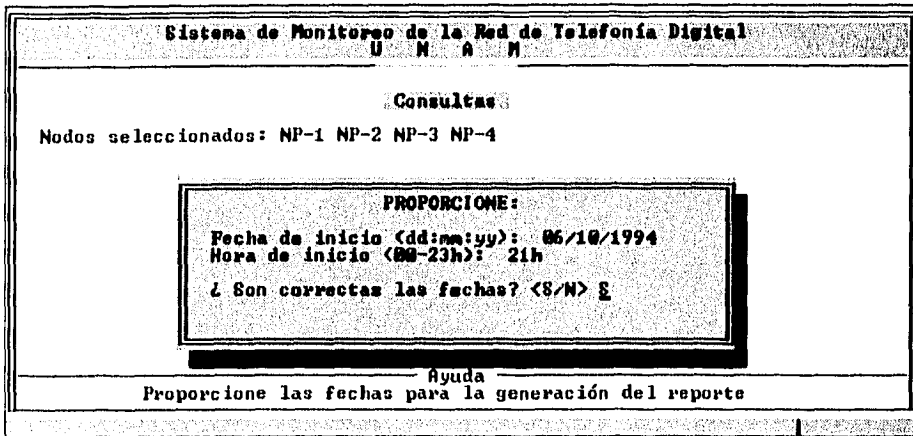


Figura IV.7.9 Fecha y hora de reporte requerido

Si seleccionó la salida del reporte a pantalla, inmediatamente el programa empezará a acceder y desplegar la información almacenada en el disco duro (figura IV.7.10). Usted puede desplazarse verticalmente y así observar otras lecturas no mostradas inicialmente. Con la tecla <Esc> podrá finalizar la consulta y salir del módulo.

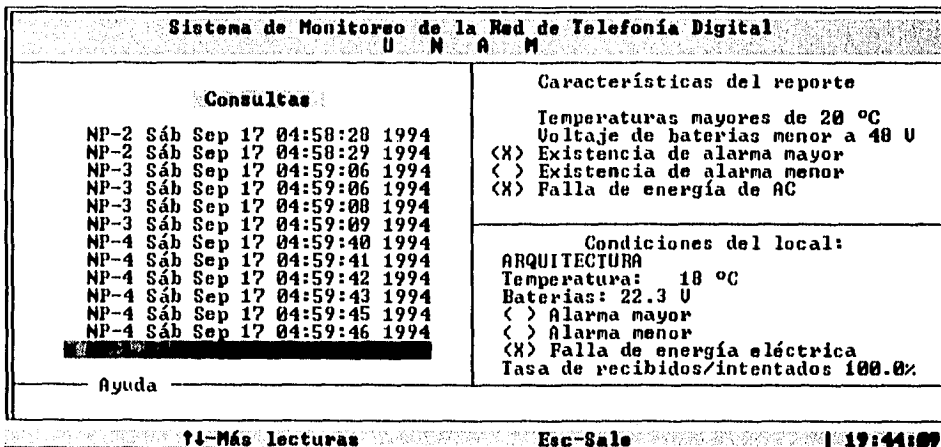


Figura IV.7.10 Ejemplo de un reporte a pantalla

Si seleccionó como medio de salida un archivo de impresión y tiene más de una unidad de diskette, se le pedirá que indique en cuál de ellas se colocará el diskette para generar el reporte (figura IV.7.11).

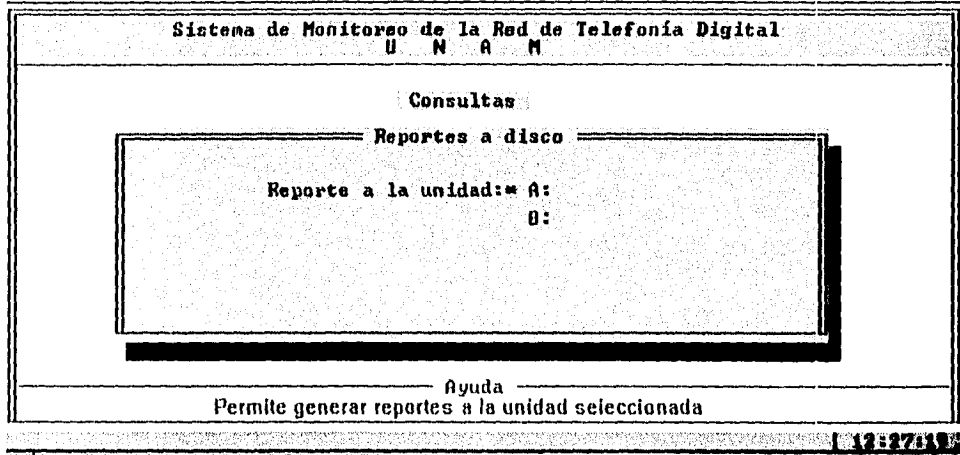


Figura IV.7.11 Solicitud de unidad para efectuar el reporte

A continuación se le solicitará que inserte un diskette en la unidad seleccionada (figura IV.7.12). En caso de que tenga sólo una unidad de diskette, después de proporcionar las fechas para la generación del reporte, el programa pasará directamente a esta pantalla saltando la anterior.

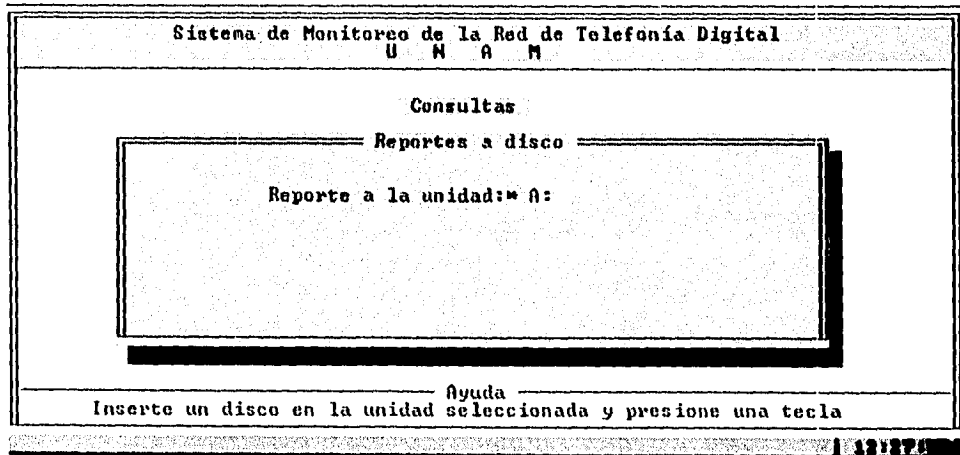


Figura IV.7.12 Solicitud de inserción de un disco en la unidad seleccionada o disponible

Finalmente deberá proporcionar el nombre del archivo para almacenar el reporte. Como se muestra en la figura IV.7.13, debe teclear el nombre completo del archivo sin la unidad. Si omite la extensión, se tomará como valor por defecto .PRN. Utilice la tecla <Backspace> para corregir algún error y <Enter> para continuar.

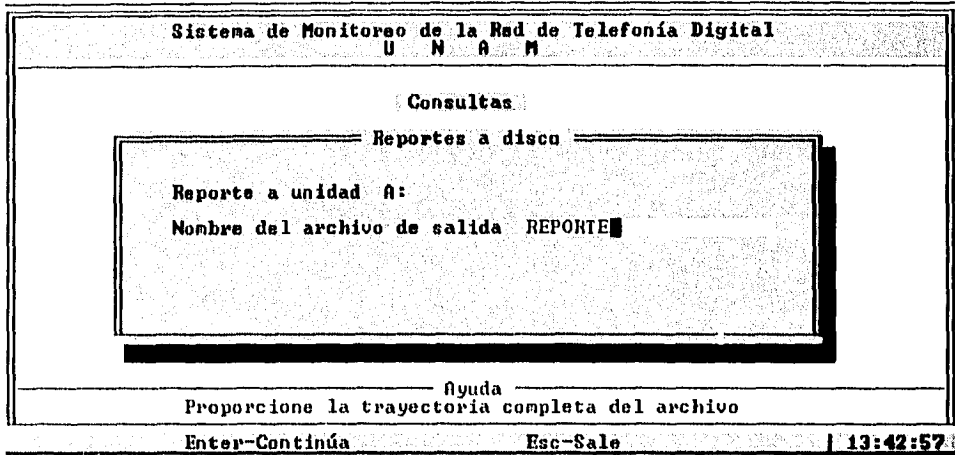


Figura IV.7.13 Pantalla para solicitar el nombre del archivo de salida

Una vez que se ha proporcionado el archivo de salida, el programa comenzará a acceder y mostrar en pantalla los archivos que sean necesarios hasta finalizar el reporte (figura IV.7.14).

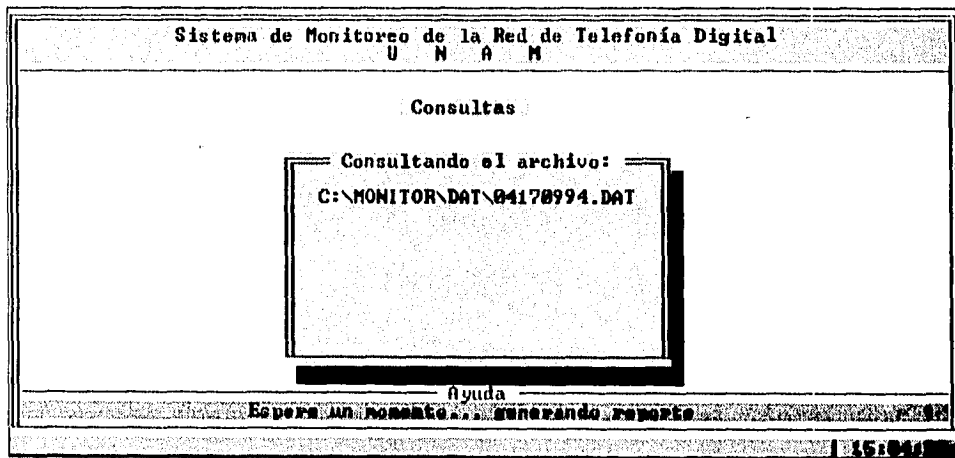


Figura IV.7.14 Pantalla que indica los archivos que están siendo consultados

Cuando aparezca el mensaje de Reporte finalizado, puede retirar el diskette. El archivo incluirá los caracteres de control necesarios, haciendo que sea posible la impresión inmediata del reporte.

Es importante observar la primera página del reporte generado ya que ahí se indica alguna eventual terminación anormal del proceso, además de las características y parámetros considerados su generación. La figura IV.8.15 muestra un ejemplo de esta primera página:

```
Sistema de Monitoreo de la Red de Telefonía Digital

                                U N A M

Reporte del estado de los nodos de la Red:
NP-1  ARQUITECTURA
NP-2  TORRE II
NP-3  I.I.M.A.S.
NP-4  D.G.S.C.A.

Reporte del período comprendido de :
04:00 h del 17/09/1994  a  06:00 h del 17/09/1994

Lecturas ordenadas por tiempo

Se reportan sólo:
Temperaturas mayores de  20 °C
Voltaje de baterías menor de  48 V
Falla de alimentación de energía de CA
Presencia de alarma mayor en el conmutador
```

Figura IV.8.15 Ejemplo de la primera página de un reporte a disco

IV.8 Otras opciones

El Sistema de Monitoreo utiliza tres diferentes tipos de marcación:

- directa.- es la utilizada por una línea directa al marcar hacia una extensión telefónica de la Red Digital de la UNAM:

62 + número de la extensión telefónica dentro de la Red

- externa.- es de la forma:

9 + 62 + número de la extensión telefónica

y es la que una extensión puede utilizar para simular una llamada del exterior de la Red hacia otro número de extensión telefónica.

- interna.- Esta forma de marcación es la que utiliza por defecto el Sistema de Monitoreo y está formado solamente por los cinco dígitos de la extensión telefónica

Al utilizar la combinación de teclas <ALT>+<D> desde cualquier pantalla de visualización de la información, se muestra la forma de marcación actual y por medio de las teclas de desplazamiento vertical y horizontal, se puede seleccionar cualquiera de las otras dos disponibles. Al finalizar la selección, el Sistema muestra en la ventana inferior el resultado de la operación. Esta opción también puede determinarse cambiando los parámetros opcionales, al iniciar la operación del Sistema desde la línea de comandos del MS-DOS (cf. IV.3).

Por otro lado, el Sistema de Monitoreo ofrece la posibilidad de activar o desactivar la bocina del modem utilizado. Por medio de la combinación de teclas <ALT>+<S> en cualquier pantalla de visualización del estado de la Red, se presenta una ventana en la cual se muestra la condición prevaleciente y, con las teclas <+> y <->, puede cambiarse. Por defecto, el Sistema mantiene la bocina del modem encendida.

Conclusiones y comentarios

Durante el desarrollo de este proyecto obtuvimos algunas conclusiones particulares al proyecto y otras aplicables al desarrollo de cualquier sistema de monitoreo, independientes incluso al tipo de red y que pueden aportar consideraciones útiles a trabajos posteriores.

Generalidades

La Universidad ha hecho inversiones significativas para la instalación, operación y expansión de la red. Ella debe asegurar que la gente siempre tenga acceso al servicio telefónico que necesita para realizar su trabajo, pero una red no manejable no es útil. La confiabilidad es muy importante ya que aún cortas interrupciones pueden tener un efecto medible en productividad. En una red telefónica de esta extensión geográfica no es factible realizar una inspección personal y rutinaria en los dispositivos de hardware para localizar eventuales fallas. Se requiere conocer remotamente el estado de los enlaces de RDI, satelitales, microondas y líneas de fibra óptica. Por esta razón, creemos necesario el desarrollo de herramientas tales como el Sistema de Monitoreo aquí presentado, que puedan notificar cuando ocurren condiciones de falla y de bajo rendimiento. Del proceso de diseño e implantación de este sistema, se describen a continuación los puntos que consideramos más importantes.

Del diseño e implantación de la tarjeta de sensores

Al evitar el uso de adaptadores instalados en las ranuras de expansión se redujo el tiempo de instalación y reemplazo. Sin necesidad de recurrir a dispositivos sofisticados, el diseño de la tarjeta se logró empleando circuitos ampliamente disponibles comercialmente y de bajo costo. Asimismo, en la selección de los componentes en el circuito impreso se tomó en cuenta la facilidad de calibración por el usuario. En el diseño se consideraron también los aspectos de confiabilidad, detección de fallas y mínimo número de componentes. Estas características siguen las líneas directrices expresadas en el punto II.1 (*Requerimientos de la solución*).

Del diseño del protocolo de comunicaciones

El aspecto de la confiabilidad fue la preocupación fundamental en el esquema de transmisión de la información. Por ello, durante el proceso de inicialización del sistema, se efectúa una verificación de las condiciones necesarias para crear el canal de comunicaciones. Las rutinas de comunicación controlan la mayoría de los errores detectados por el UART y actúan ante las situaciones que pueden presentarse durante el establecimiento de una conexión. El protocolo de comunicaciones es sencillo y en consecuencia veloz, fija tiempos límite para la ocurrencia de algún evento y verifica la integridad de los datos calculando una suma de verificación. Aunque con este método algunos errores multibit podrían no detectarse, resulta muy efectivo en la mayoría de las situaciones. Todos ellos son aspectos que deben tomarse en cuenta si se desea obtener un sistema de comunicación tolerante a fallas.

De la visualización y generación de reportes del estado de la red

Durante el desarrollo del sistema concluimos que en el más alto nivel de administración de la red, se debe supervisar el rendimiento de los enlaces; por ello, además de conocer el estado de las condiciones físicas de los locales de los conmutadores, añadimos en nuestro sistema un indicador del estado general de los canales de comunicaciones, que originalmente no estaba considerado en el proyecto.

Los programas que presentan la información del estado de la red se escribieron pensando en implantar una interfaz consistente, intuitiva y amigable lo que facilitó el aprendizaje, operación e interpretación para el usuario.

Del sistema de almacenamiento de los datos a disco

La implantación de la idea de un área reservada a la información obtenida durante el monitoreo a la red, es una de las propuestas principales de este trabajo ya que no conocemos aplicaciones para MS-DOS que utilicen un mecanismo similar. Este *contenedor de datos* permite la administración sencilla y eficiente del espacio asignado al sistema.

Alrededor de esta estructura de control se creó un proceso de autodepuración que, al remover la información más antigua garantiza el espacio necesario para que los datos más recientes se conserven en el disco, incrementando al mismo tiempo la confiabilidad del sistema al evitar fallas por falta de espacio en disco.

El diseño de las estructuras de archivos de un sistema (en este caso el contenedor de datos, el directorio y los archivos de lecturas) debe tomar en cuenta el manejo del disco duro por parte del sistema operativo, para disponer de los datos de manera que se minimicen los costos de acceso. Fue así que a fin de mantener los tiempos de acceso razonablemente cortos, el diseño contempla la agrupación y organización de la información en archivos relativamente pequeños, pero que el usuario accesa a través del Sistema como si fueran un archivo único.

Del subsistema de altas, bajas y cambios

Conforme la red crece, los puntos de interés y de control como área de supervisión de la red también se incrementan. El manejo de fallas llega a ser especialmente difícil cuando la red crece en extensión geográfica, por lo que un sistema de monitoreo debe soportar el crecimiento de la red, no inhibirlo. En consecuencia, la posibilidad de incorporar más nodos a la red, suspender el monitoreo de alguno de ellos y permitir el cambio de su extensión telefónica permite al sistema ajustarse a las condiciones cambiantes durante la operación de la red.

De la exportación de la información

La idea de continuidad e integración a los trabajos de investigación o desarrollo dentro de la Dirección de Telecomunicaciones Digitales es la línea directriz del mecanismo de exportación de la información que nuestro sistema proporciona. Permite a futuros trabajos estudiar el comportamiento histórico de las condiciones de los conmutadores de la red, sin obstaculizarse o detenerse por las convenciones y limitaciones que internamente nuestro sistema establece.

Apéndices

Apéndice A. Archivos de configuración de modem

Para la creación de un archivo de configuración puede utilizar un editor de textos y debe respetar las siguientes indicaciones:

- Cada comando de configuración debe empezar con el prefijo AT y tener una longitud máxima de 40 caracteres.
- Debe utilizarse una sola línea para cada comando de configuración y terminarla con un retorno de carro o *Enter*.
- Las primeras cuatro líneas deben contener obligatoriamente, sin importar su orden, los comandos del modem para su inicialización y restablecimiento, la secuencia de escape y la etiqueta de identificación que se le quiera otorgar. Cada una de estas líneas utilizan etiquetas para poder ser identificadas y se muestran resumidas en la tabla III.10.2.

Tabla III.10.2

Etiqueta	Parámetro
ID=	Este no es un comando, es la etiqueta de identificación que le quiera dar al modem. *
RESET=	Comando de restablecimiento.
SETUP=	Comando de inicialización.
ESCAPE=	Secuencia de escape. Si se requiere un retorno al final del comando, deberá indicarlo con \r.

- * Aunque este identificador del modem es opcional, debe incluirse esta línea en el archivo de configuración. El no incluirla ocasiona la invalidación del archivo.

Todos los comandos AT y los valores de los registros internos "s" se proporcionarán basándose en la sintaxis indicada en el manual del modem que desee utilizar. Es posible utilizar prácticamente cualquier marca de modem *Hayes compatible* si se toman en cuenta las siguientes indicaciones sobre los algunos comandos:

ATS0=1 Este comando habilita la capacidad de contestación automática. Gracias a esta configuración, los modems remotos son capaces de establecer el enlace automáticamente al detectar la llamada de la computadora central.

ATS2=43 El registro **S2** determina cuál es el carácter que conforma la secuencia de escape.

Durante la transferencia de información, los modems se encuentran en *modo línea*. Para salir de este modo y entrar al *modo comando*, en el cual los caracteres enviados al puerto serial no son transmitidos, se le debe dar una secuencia de escape antecedida de una pausa (o *tiempo de guarda*), para diferenciarla de cualquier otra cadena de caracteres.

El Sistema de Monitoreo considera el signo más (+) para esta secuencia. Si no se utiliza el mismo valor ASCII para configurar el modem a través del registro S2, el Sistema no será capaz de terminar adecuadamente una conexión.

ATS3=13 **S3** determina el valor ASCII para el carácter de retorno de carro (<CR>). El valor comúnmente utilizado es un ASCII 13 y así se considera en el Sistema de Monitoreo.

ATS7=5 Este registro tiene dos funciones. La primera de ellas, es determinar el tiempo máximo de espera para tono de marcación después de haberle enviado al modem el modificador **W** en la cadena de marcación.

Durante su proceso de inicialización, el Sistema de Monitoreo, descuelga la línea telefónica y determina la existencia de tono de marcación. Sin embargo, los conmutadores de la Red de Telefonía Digital de la UNAM no proporcionan este tono indefinidamente. Si detectan una línea descolgada por más de cierto tiempo, interrumpen el servicio y envían un tono agudo. Esta característica obliga a programar los modems utilizados por el Sistema de Monitoreo con un tiempo de espera máximo para la detección del tono de marcación. Consideramos que un valor de cinco segundos en este parámetro es adecuado para este propósito.

La segunda función del registro **S7** es la siguiente: durante el establecimiento de un enlace, el valor en este registro determina el tiempo máximo que el modem puede esperar entre la marcación de un número y la detección de una señal portadora. Si el modem no la detecta dentro de este tiempo, cuelga y da el código resultante equivalente a NO CARRIER.

Es importante considerar que **este valor no permite el establecimiento de un enlace**. Al realizar llamadas a los nodos remotos, el Sistema de Monitoreo incrementa el valor asignado al registro **S7**, lo que permite al modem del equipo central detectar la portadora y efectuar la conexión.

ATS12=50 S12 almacena el valor del tiempo de guarda de la secuencia de escape en unidades de veinte milisegundos. El Sistema de Monitoreo considera este tiempo como un segundo. Si no se programa con este mismo tiempo el modem, podría no ser capaz de terminar correctamente un enlace.

ATQ0 El comando **Q** habilita (**Q0**) o deshabilita (**Q1**) los códigos resultantes. El Sistema de Monitoreo requiere que el resultado de cada comando sea reportado por el modem, por lo cual se debe incluir este comando.

ATV1 El comando **V** determina el formato de los códigos resultantes, que pueden ser en forma numérica (**V0**) o como mensajes (**V1**); El Sistema de Monitoreo identifica los mensajes.

ATXn El comando **X** determina las posibles respuestas del modem; variando el valor **n** asignado al modem con este comando, se pueden obtener diferentes conjuntos de códigos resultantes.

El Sistema de Monitoreo reconoce los mensajes: OK, NO CARRIER, ERROR, CONNECT (en sus diferentes variaciones), RING (o RINGING) y NO DIAL (permite reconocer los mensajes que así comienzan tales como NO DIAL TONE o NO DIALTONE) y se basa en ellos para determinar las acciones a efectuar en su verificación de la comunicación con el modem y al establecer una conexión con otro; por lo cual se debe asignar el valor adecuado a **n** para habilitar el conjunto de códigos resultantes que incluyan estos mensajes.

A continuación se proporcionan los archivos de configuración de dos modems que se probaron con el sistema. El primero de ellos es para el Hayes Smart Modem 1200, un modem full duplex asíncrono de 1200 bps. Recuerde que las primeras cuatro líneas son requeridas y que el identificador del modem puede seleccionarse libremente.

```
ID=SMART MODEM HAYES 1200
RESET=ATZ
SETUP=ATE0Q0V1
ESCAPE=+++
ATB1
ATC1
ATF1
ATL2
ATM1
ATX4
ATY1
ATS0=1
ATS1=0
ATS2=43
ATS3=13
ATS4=10
ATS5=8
ATS6=2
ATS7=5
ATS8=2
ATS9=6
ATS10=10
ATS11=70
ATS12=50
```

Otro modem que se utilizó es el MultiTech MT1432ZDX, éste es full duplex, asíncrono y capaz de operar hasta 14,400 bps. El siguiente archivo de configuración lo programa para operar a 2400 bps:

```
ID=MT1432ZDX MULTI TECH
RESET=ATZ
SETUP=ATE0Q0V1
ESCAPE=+++AT\r
AT&F
AT&Q1
AT$MB2400
AT$SB2400
ATX2
ATM1
ATRO
AT&E1
AT&E3
AT&E6
AT&E8
AT&E10
AT&E13
AT&E14
AT$E0
AT$BA0
AT$A0
AT#A0
AT&B0
AT&BS1
AT&C1
AT&D2
AT$EB0
AT%E1
AT%E4
AT#F2
AT#F1
AT&C0
AT#L0
AT&P0
AT$R0
AT&R1
AT&RF1
AT&S1
```

AT&SF0
AT&T5
AT#T1
ATY0
ATS0=1
ATS2=43
ATS3=13
ATS4=10
ATS5=8
ATS6=2
ATS7=5
ATS8=2
ATS9=6
ATS10=7
ATS11=70
ATS12=37
ATS24=20
ATS25=1
AT&W0

Apéndice B Compilando el proyecto

En este apéndice se proporcionan las instrucciones y consideraciones necesarias para compilar los programas del Sistema de Monitoreo desde el Turbo C Integrated Development Environment, versión 2.0.

1. En su disco duro cree el directorio `C:\MONITOR\C`
2. Copie del directorio `\C` del diskette de programas fuente, todos los archivos al directorio creado en el punto 1.
3. Copie los archivos del directorio `\INCLUDE` del diskette de programas fuente, al directorio que contiene los *headers* de Turbo C, generalmente `TC\INCLUDE`
4. Cámbiese al directorio de Turbo C y ejecute `TC.EXE`, entrará al ambiente turbo de Borland. Explicaciones más detalladas acerca de la utilización de este programa se pueden encontrar en la referencia 10 de la bibliografía.
5. Asegúrese de que el compilador tiene seleccionado el modelo `SMALL` a través de `Options/Compiler/Model`
6. Especifique el nombre de uno de los archivos de proyecto (`.PRJ`) a través de `Project/Project name`. Seleccione por ejemplo:
`REMOTA.PRJ`
7. Presione la tecla de función `<F9>` (`Make`), el Turbo C empezará a accesar compilar y ligar los archivos que forman parte de ese proyecto y generará un programa ejecutable `REMOTA.EXE` colocándolo en donde se especifique por la opción `/Options/Directories/Output directory`
8. Repita los pasos 6 y 7 para cada uno de los archivos de proyecto, finalmente obtendrá los siguientes programas ejecutables:
`A_DISCO.EXE, A_PANTA.EXE, BKUP.EXE, CENTRAL.EXE,`
`CONSULTA.EXE, CONTLONG.EXE, DIALMODE.EXE, MANTE.EXE,`
`MONITOR.EXE, PPTO.EXE, PRUEBA.EXE y REMOTA.EXE`

Bibliografía

- [1] *Advanced C Programming*
Steve Ovalline
The Peter Norton Computing Group, Brady Publishing

- [2] *C Programmer's Guide to Serial Communications*
Joe Campbell
Howard W. Sams and Company, U.S.A.1987

- [3] *Circuitos Integrados Lineales y Amplificadores Operacionales,*
segunda edición revisada
Robert F. Coughlin, Frederick F. Driscoll
Prentice Hall Hispanoamericana, México 1987

- [4] *dBASE for Professionals with dBASE IV*
Neil Dunlop
Van Nostrand Reinhold, 1990

- [5] *Digital Integrated Electronics*
Herbert Taub, Donald Schilling
Mc Graw Hill, International Editions

- [6] *El Lenguaje de Programación C,* segunda edición
Brian W. Kernighan, Dennis M. Ritchie
Prentice Hall Hispanoamericana, México 1991

- [7] *Inside the IBM PC and PS/2,* third edition
Peter Norton
Brady Publishing, New York 1990

- [8] *The Peter Norton Programmer's Guide to the IBM PC*
Peter Norton
Microsoft, Washington 1985

- [9] *The Winn L. Rosh Hardware Bible*, second edition
Winn L. Rosh
Brady Publishing, New York 1992

- [10] *Turbo C, User's Guide*, first edition
Borland International
U.S.A. 1988

- [11] *Turbo C, Reference Manual*, first edition
Borland International
U.S.A. 1988