

26
2ej



Universidad Nacional Autónoma de México

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
A R A G O N

" Desarrollo de un módulo de generación de reportes
para el banco de pruebas de pequeños sistemas foto-
voltaicos del Instituto de Investigaciones Eléctricas,
utilizando el programa FIX DMACS "

FALLA DE ORIGEN

TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A :
LETICIA HERNANDEZ RAMIREZ

San Juan de Aragón, Edo. de México

1995



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Ser joven es tener ideales y luchar hasta lograrlos, es soñar con el futuro por el que se trabaja en el presente, es tener siempre algo que hacer, algo que crear, algo que conocer, algo que dar...

Dedicatoria

A Dios

Por alimentar mi espíritu y fortalecer con ello la confianza en mí misma para salir siempre adelante, hasta lograr uno de mis más caros anhelos.

A la memoria de Carmen Dimas

A Guadalupe Noé

A tí, por tu gran sentido de responsabilidad que has mostrado siempre y que has sabido inculcarme. Gracias padre por impulsar mis sueños y gozar con mis realizaciones, por tu confianza, firmeza, cariño y respeto para conmigo, pero antes que todo esto gracias por ser mi amigo; y por enseñarme que después de una gran tormenta siempre sale el sol, un sol cuya luz nos permite ver que el mundo es todavía hermoso.

A Micaela

A tí, mujer extraordinaria que me diste la vida. Gracias madre por permitirme caminar tomada de tu mano por la vida, por tu ternura, por dedicarme gran parte de tu tiempo, por tus cuidados y desvelos y por mostrarme con hechos que detrás de un gran hombre siempre hay una gran mujer. Ten la satisfacción de ser la mejor mamá.

A Sofia, Ma. Guadalupe y Antonio

A ustedes, por los momentos gratos y amargos que he pasado a su lado, momentos que nos han enseñado a madurar y a crecer juntos, unidos y felices; características esenciales que muestran el verdadero valor de la palabra *hermanos*. Gracias por alegrar mi vida con sus ocurrencias y cariños. Y sólo un recordatorio: ahora es su turno.

Agradecimientos

Al Instituto de Investigaciones Eléctricas por el apoyo tanto técnico como económico que me otorgó para la realización de esta tesis.

A la Universidad Nacional Autónoma de México por brindarme la oportunidad de realizar mis estudios a nivel licenciatura, a través de sus instalaciones en la **Escuela Nacional de Estudios Profesionales Aragón**.

Al Dr. Francisco Villaseñor Rivera y a la **Ing. Silvia Vega Muytay** por su asesoría, ayuda, consejos y comentarios en el desarrollo del presente trabajo.

Al Ing. Javier Legunas M., al Ing. Jaime Agredano D. y al Lic. Víctor Cortés por su gran disponibilidad para ayudarme en cualquier momento.

Al Ing. Narciso Acevedo H., a la Profa. Irene Godínez G., a la Lic. Flor Morales R. y al Dr. Sergio Cuevas G. por la bonita y sincera amistad que me han brindado. Gracias por sus consejos.

A mis compañeros de generación (B9-B3); en especial a **Leura F. Falcón S. y M. Alejandro Mández G.,** por compartir conmigo tantas cosas bellas que dejaron una huella imborrable en nuestras vidas. Hago extensivo este agradecimiento a los Sres. **Lidia y Víctor Falcón** por el apoyo y comprensión que nos brindaron durante nuestra formación profesional.

A mis compañeros becarios del IIE; en especial a **César Gtz. S., Francisco Morales C., Javier Ramírez G., Mayolo Salinas C., Jorge A. Mtz. P., Jorge Alvarado, Ms. Elena Calixto O. Juan G. Vandeke R., Fco. J. Rodríguez V., M. Miguel Pérez R., Carlos A. Delgado D., Juño Cruz M., Enrique E. Rojo R., Gonzalo Mtz. H.** por acompañarme durante esta bonita etapa de mi vida.

Al personal del Departamento de Fuentes No Convencionales de Energía.

A la familia Avilés Sotelo por abrirme las puertas de su hogar y hacerme sentir como parte de él durante mi estancia en Cuernavaca.

Hago extensivo este agradecimiento a todas aquellas personas que no menciono y que de alguna forma contribuyeron en la elaboración de este trabajo.

Indice general

INTRODUCCION, /

1. SISTEMAS FOTOVOLTAICOS, 1

1.1 GENERALIDADES, 1

1.2 CONFIGURACION BASICA DE UN SISTEMA FOTOVOLTAICO, 3

1.3 OPERACION DE LOS COMPONENTES, 5

1.3.1 SUBSISTEMA DE GENERACION ELECTRICA, 5

1.3.1.1 LA CELDA FOTOVOLTAICA, 6

A) Funcionamiento, 6

B) Tipos de celdas solares, 7

a) Silicio de cristal simple, 7

b) Silicio policristalino, 8

c) Cinta de silicio, 8

d) Silicio amorfo, 8

1.3.1.2 MODULOS FOTOVOLTAICOS, 9

A) Características físicas, 9

B) Características eléctricas, 9

a) Corriente, 11

b) Voltaje, 12

1.3.1.3 ARREGLOS FOTOVOLTAICOS, 12

A) Arreglos de plato fijo, 14

B) Arreglos portátiles, 14

C) Arreglos móviles, 14

1.3.1.4 SUBSISTEMA DE ALMACENAMIENTO DE ENERGIA, 15

A) Capacidad amp-hora, 15

B) Profundidad de descarga, 16

1.3.1.5 SUBSISTEMA DE CONTROL, 17

A) Controlador Shunt, 17

1.4 VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS FOTOVOLTAICOS, 18

1.4.1 VENTAJAS, 18

1.4.2 DESVENTAJAS, 18

1.5 CARACTERISTICAS DE LOS MODULOS FOTOVOLTAICOS ESTUDIADAS EN EL IIE, 19

1.5.1 CARACTERISTICAS DEL SISTEMA FOTOVOLTAICO, 19

1.5.2 CARACTERISTICAS DEL SISTEMA DE ADQUISICION DE DATOS, 21

- 2. EL PROGRAMA FIX DMACS, 23**
 - 2.1 CONSIDERACIONES BASICAS, 23**
 - 2.1.1 CONSIDERACIONES DE *HARDWARE*, 24
 - 2.1.2 CONSIDERACIONES DE *SOFTWARE*, 24
 - 2.1.3 INTERFAZ DE USUARIO, 25
 - 2.1.4 ARQUITECTURA, 26
 - 2.2 PROGRAMAS QUE LO CONSTITUYEN, 28**
 - 2.2.1 PROGRAMAS DE ALTO NIVEL, 28**
 - 2.2.1.1 EMS, 28
 - 2.2.1.2 SHELL, 28
 - 2.2.1.3 SAC, 29
 - 2.2.2 PROGRAMAS DE BAJO NIVEL, 29**
 - 2.2.2.1 DATABASE, 29
 - 2.2.2.2 HISTORICAL TRENDING, 35
 - 2.2.2.3 GRAPHICS, 36
 - 2.2.2.4 REPORT, 37
 - 2.2.2.5 RECIPES, 39
 - 2.2.2.6 SCHEDULER, 39
 - 2.2.2.7 DOS WINDOW, 40
 - 2.2.2.8 SYSTEM CONFIG UTILITY, 40
- 3. ANALISIS DEL PROGRAMA FIX DMACS PARA LA GENERACION DE REPORTES, 42**
 - 3.1 ALMACENAMIENTO DE DATOS, 43**
 - 3.2 ANALISIS DE REQUERIMIENTOS, 43**
 - 3.2.1 PROPOSITO GENERAL DEL GENERADOR DE REPORTES, 49
 - 3.2.2 REPORTES DIARIOS, 49
 - 3.2.3 REPORTES SEMANALES Y REPORTES MENSUALES, 53
 - 3.2.4 REPORTES ANUALES, 53
 - 3.3 FIX DMACS COMO HERRAMIENTA PARA REALIZAR EL MODULO DE REPORTES, 54**
 - 3.4 LENGUAJE C COMO OPCION AUXILIAR PARA REALIZAR EL MODULO DE REPORTES, 55**
 - 3.5 ELECCION DE LA ALTERNATIVA MAS VIABLE, 56**
- 4. DESARROLLO DEL SOFTWARE PARA UTILIZAR FIX DMACS EN LA GENERACION DE REPORTES, 59**
 - 4.1 GENERALIDADES, 59**
 - 4.1.1 DEFINICION DE *SOFTWARE*, 59
 - 4.1.2 QUE ES UNA INTERFAZ DE USUARIO, 58
 - 4.2 ESPECIFICACIONES DE DISEÑO PARA EL MODULO REPORTES, 62**
 - 4.2.1 CONSIDERACIONES INICIALES, 62
 - 4.2.2 DESCOMPOSICION MODULAR DEL PROGRAMA *REPORTES*, 63
 - 4.2.2.1 MODULO *FORMATOS FIX DMACS*, 65
 - 4.2.2.2 MODULO *ALTA*, 67
 - 4.2.2.3 MODULO *DIARIOS*, 68
 - A) Submódulo *GRAFICAS DIARIAS*, 70
 - B) Submódulo *TABLAS*, 73
 - 4.2.2.4 MODULO *SEMANALES*, 76
 - A) Submódulo *TABLA RESUMEN*, 78

- B) Submódulo *FALLAS*, 80
- 4.2.2.6 MODULO *MENSUALES*, 82
- A) Submódulo *TABLA RESUMEN*, 84
- B) Submódulo *FALLAS*, 85
- C) Submódulo *MENSUAL*, 86
- 4.2.2.8 MODULO *ANUALES*, 88
- 4.2.2.7 MODULO *IMPRIMIR*, 88
- 4.2.2.8 MODULO *ARCHIVAR*, 89
- 4.2.2.9 MODULO *INTERFAZ*, 90

5. PRUEBAS E IMPLEMENTACION, 93

5.1 PROCEDIMIENTO DE CODIFICACION DEL PROGRAMA *REPORTES*, 93

5.2 PRUEBAS A LAS FUNCIONES PRINCIPALES, 96

- 5.2.1 FUNCION DE VALIDACION DE PANTALLA, 97
 - 5.2.2 FUNCIONES DE ACCESO A LA BASE DE DATOS *FOT*, 98
 - 5.2.3 FUNCION PARA DAR DE ALTA UN SISTEMA FOTOVOLTAICO, 99
 - 5.2.4 FUNCIONES PARA CREAR LOS REPORTES DIARIOS, 100
 - 5.2.4.1 GRAFICA, 100
 - 5.2.4.2 TABLA RESUMEN, 102
 - 5.2.4.3 TABLA DE FALLAS, 104
 - 5.2.4.3 LISTADO DE DATOS, 105
 - 5.2.5 FUNCIONES PARA CREAR LOS REPORTES SEMANALES Y MENSUALES, 108
 - 5.2.5.1 TABLA RESUMEN, 108
 - 5.2.5.2 TABLA DE FALLAS, 109
 - 5.2.6 FUNCION PARA CREAR EL ARCHIVO MENSUAL, 112
 - 5.2.7 FUNCION DE DESPLIEGUE EN PANTALLA DE LOS REPORTES GENERADOS, 112
 - 5.2.8 FUNCIONES DE IMPRESION Y ARCHIVADO DE LOS REPORTES GENERADOS, 113
 - 5.2.9 FUNCIONES DE ALARMA, 113
- #### **5.3 REQUERIMIENTOS DE *SOFTWARE* Y *HARDWARE* PARA IMPLEMENTAR EL PROGRAMA *REPORTES*, 114**
- 5.3.1 REQUERIMIENTOS DE *SOFTWARE*, 114
 - 5.3.2 REQUERIMIENTOS DE *HARDWARE*, 115

6. GUIA DE USUARIO, 116

- 6.1 QUE ES UNA GUIA DE USUARIO, 116
- 6.2 DESCRIPCION DE LOS PROGRAMAS COMPONENTES DEL MODULO *REPORTES*, 117
- 6.3 DESCRIPCION DEL FUNCIONAMIENTO DEL MODULO DE REPORTES, 119
 - 6.3.1 COMO ENTRAR, COMO SELECCIONAR Y COMO SALIR DEL MODULO *REPORTES*, 119
 - 6.3.2 COMO GENERAR REPORTES, 121
 - 6.3.2.1 *VENTANA DE SISTEMAS*, 122
 - 6.3.2.2 COMO GENERAR REPORTES DIARIOS, 124
 - A) *Ventana de FECHA* para los reportes diarios, 125
 - B) *Ventana OTRA FECHA* para los reportes diarios, 129
 - 6.3.2.3 COMO GENERAR REPORTES SEMANALES, 130
 - 6.3.2.4 COMO GENERAR REPORTES MENSUALES, 132
 - 6.3.2.5 COMO GENERAR REPORTES ANUALES, 137
 - 6.3.3 COMO IMPRIMIR Y COMO ARCHIVAR TODO TIPO DE REPORTES, 138
 - 6.3.3.1 COMO IMPRIMIR, 139

- 6.3.3.2 COMO ARCHIVAR, 142
- 6.3.4 COMO DAR DE ALTA UN SISTEMA DENTRO DEL GENERADOR DE REPORTES, 143

CONCLUSIONES Y RECOMENDACIONES, 147

BIBLIOGRAFIA, 150

APENDICES

- A. COMO CREAR LOS FORMATOS DE DATOS PROVENIENTES DE FIX DMACS PARA EL GENERADOR DE REPORTES, 153
- B. COMO CREAR EL ARCHIVO ALTA.DAT, 157
- C. RUTINAS DE PROGRAMACION DEL MODULO REPORTES, 159
 - C.1 PROGRAMA PRINCIPA.C, 159
 - C.2 PROGRAMA VENTANAS.C, 179
 - C.3 PROGRAMA SEMYMEN.C, 185
 - C.4 PROGRAMA MENSUAL.C, 193
 - C.5 PROGRAMA GRAFICA.C, 194
 - C.6 PROGRAMA DIARIOS. C, 199
- D. GLOSARIO, 206

Indice de figuras

- Figura 1.1** Diagrama a bloques de un sistema fotovoltaico, 5
Figura 1.2 Proceso fotovoltaico de una celda solar, 7
Figura 1.3 Curva típica de corriente-voltaje, 10
Figura 1.4 Tres módulos conectados en paralelo, 13
Figura 1.5 Baterías conectadas en serie, paralelo y paralelo-serie, 16
Figura 1.6 Fotografía de un pequeño sistema fotovoltaico, puesto a prueba para electrificación rural por el Instituto de Investigaciones Eléctricas, 20
- Figura 2.1** Menú principal de FIX DMACS (*MASTER MENU*), 25
Figura 2.2 Menú de entrada_salida para FIX DMACS, 26
Figura 2.3 Panorama general de la arquitectura de FIX DMACS, 27
Figura 2.4 Entrada y salida de datos en una base de datos creada en FIX DMACS, 31
Figura 2.5 Adicionando un bloque a una base de datos en FIX DMACS, 32
Figura 2.6 Ejemplo de una cadena para adquisición y control de datos, 34
Figura 2.7 Interfaz gráfica para despliegue de datos en tiempo real, provenientes de los sistemas fotovoltaicos, 37
- Figura 3.1** Menú de selección para el programa *Historical Trend Display*, 44
Figura 3.2 Gráfica producida por el programa *Historical Trend Display*, 45
Figura 3.3 Formato del archivo.ASCII que se genera mediante el programa *Historical Trend Display*, 46
Figura 3.4 Gráfica elaborada con los datos obtenidos en FIX DMACS, mediante Excel, 51
- Figura 4.1** Diagrama modular del programa *REPORTES*, 64
Figura 4.2 Diagrama de flujo de información para el programa *REPORTES*, 64
Figura 4.3 Descomposición del módulo *DIARIOS*, 70
Figura 4.4 Descomposición del módulo *SEMANALES*, 78
Figura 4.5 Descomposición del módulo *MENSUALES*, 84
- Figura 5.1** Gráfica generada sin validación de datos (con valores completos), 101
Figura 5.2 Gráfica generada validando datos perdidos por interrupciones de usuario o falla de energía eléctrica, 102
- Figura 6.1** Selección del programa *REPORTES*, 119
Figura 6.2 Pantalla inicial del módulo *REPORTES*, 120
Figura 6.3 Ventana de salida para el módulo *REPORTES*, 121
Figura 6.4 Ventana de *SISTEMAS*, 122
Figura 6.5 Ventana que indica que el sistema no se encuentra dado de alta, 124
Figura 6.6 Menú *DIARIOS*, 125

- Figura 6.7** Ventana de FECHA, 126
- Figura 6.8** Ventana que indica que no se generó ningún archivo de salida, 128
- Figura 6.9** Ventana que indica la generación incompleta de los archivos de salida para la selección de la opción TODOS en la ventana de SISTEMAS, 128
- Figura 6.10** Ventana OTRA FECHA, 129
- Figura 6.11** Menú SEMANALES, 136
- Figura 6.12** Menú MENSUALES, 133
- Figura 6.13** Ventana de FECHA para archivos mensuales, 136
- Figura 6.14** Ventana de IMPRESION y ARCHIVADO para los reportes diarios, 138
- Figura 6.15** Aviso de impresión para una gráfica, 141
- Figura 6.16** Aviso de impresión para cualquier tipo de reporte, excepto una gráfica, 141
- Figura 6.17** Ventana de UNIDADES, 142
- Figura 6.18** Ventana de ALTA, 144
- Figura 6.19** Ventana que permite introducir el identificador para cada sistema fotovoltaico, 145

Introducción

En la actualidad, uno de los aspectos más importantes dentro la vida de una nación es la producción de energía eléctrica, siendo ésta un aspecto fundamental para su desarrollo socio-económico.

Con toda seguridad la demanda mundial de energía eléctrica continuará ascendiendo en el transcurso de los próximos años. Es un hecho que la humanidad se enfrenta a la creciente escasez de combustibles fósiles y al correspondiente impacto sobre el medio ambiente de los contaminantes emitidos al procesar éstos.

Por lo anterior, se ha empezado a desarrollar tecnología adecuada, precisa y suficiente que permita utilizar otras fuentes energéticas que complementen y/o sustituyan las ya existentes.

Para el propósito de esta tesis, cuando hablamos de fuentes complementarias de energía nos referimos a aquellas que son inagotables y que han estado presentes desde que la humanidad existe; tales como el sol, el viento, la biomasa y la fuerza del oleaje marítimo. Si agrupamos estas fuentes podemos denominarlas ***Fuentes No Convencionales de Energía***.

Puede decirse que la principal fuente no convencional de energía es el sol, el cual suministra a la tierra cerca de 173 billones de kilowatts, es decir, 30 000 veces el consumo total de energía en los países industrializados.

Para realizar el presente trabajo sólo nos será útil (por el momento) conocer con más profundidad la energía proveniente del sol, también llamada ***energía solar***. En la búsqueda por la utilización de ésta se han desarrollado tecnologías en países como Estados Unidos de Norteamérica y Japón (entre otros), quienes han puesto a prueba proyectos de alimentación eléctrica por medio de captación solar en casas habitación, comunidades rurales, industrias y cuyos resultados han sido muy favorables y eficientes, lo que contribuye a mejorar día con día la tecnología de cada país con respecto a este campo.

Hablando de energía solar, nuestro país no es la excepción, ya que desde hace algunos años científicos mexicanos se han preocupado por conocer y aplicar los principios fundamentales de captación solar en proyectos de investigación que han sido diseñados de acuerdo a las características geográficas, climatológicas y sociales de nuestro país.

De entre los proyectos propuestos, surge el de la captación solar por medio de pequeños paneles (o módulos) fotovoltaicos que permitan iluminar durante la noche, con la ayuda de baterías, zonas rurales (con población menor a los 500 habitantes), así como también proporcionar electricidad para radio, comunicación, refrigeración, bombeo de agua; entre otras aplicaciones.

Antes de instalar los módulos fotovoltaicos en la zona rural a la que son designados, éstos se prueban para asegurar que su funcionamiento futuro sea óptimo y adecuado. Dichas pruebas consisten en realizar mediciones de varios tipos: medición de voltajes y corrientes producidos por cada panel y consumidos por la carga, medición de temperatura en el sistema, humedad relativa, temperatura ambiente, cantidad de insolación recibida, velocidad del viento y presión atmosférica, entre otras.

Estos parámetros están siendo medidos por sensores conectados a un adquirente de datos que interactúa con una computadora personal para almacenar en archivos los datos que resulten de las mediciones realizadas. Para facilitar esta tarea se utiliza un conjunto de programas para computadora denominado FIX DMACS (FIX = *Fully Integrated Control System*, DMACS = *Distributed Manufacturing, Automation and Control Software*), que permite el manejo de datos en tiempo real y el almacenamiento de datos históricos.

La realización de este trabajo surge con la necesidad de organizar y procesar esos datos almacenados en un formato de reportes computarizado que nos permita visualizarlos en pantalla e imprimirlos en forma ordenada mediante gráficas de comportamiento y tablas diarias, semanales y mensuales. Todo lo anterior, con la finalidad de predecir el comportamiento de los módulos en prueba antes de llevar a cabo su instalación real.

El proceso de dichos datos anteriormente era manual y consistía de varios pasos, tales como grabar los datos provenientes del monitoreo de cada uno de los sistemas en discos flexibles y después procesarlos a través de un paquete comercial para generar las gráficas y tablas requeridas.

El objetivo principal de esta tesis es realizar en forma automática las tareas antes mencionadas y dentro del ambiente de trabajo del programa FIX DMACS.

A continuación se proporciona una breve descripción de las partes integrantes del presente trabajo.

En el *capítulo 1* se describen, de una manera generalizada, las características básicas de un sistema fotovoltaico.

El *capítulo 2* nos resume el funcionamiento de los módulos de *software* por los que está integrado el programa FIX DMACS, así como los requerimientos indispensables para su instalación.

El *capítulo 3* responde a un estudio de los módulos de FIX DMACS para determinar si es posible realizar con este conjunto de programas el módulo de reportes que cumpla con los requerimientos especificados por el usuario o si es necesario utilizar otra herramienta adicional de *software* que permita programar dicho módulo. Además se incluye la especificación de requerimientos para elaborar el programa computacional que permitirá generar los reportes (módulo de reportes).

El *capítulo 4* corresponde al diseño de las rutinas de programación que permitieron llegar a la elaboración del módulo de reportes para evaluación de pequeños sistemas fotovoltaicos.

En el *capítulo 5* se describen las pruebas, implementación e instalación del módulo de reportes para evaluación de pequeños sistemas fotovoltaicos.

El *capítulo 6* es una guía de usuario que permitirá consultar cualquier duda respecto al funcionamiento y operación del módulo de reportes realizado.

Finalmente se dan las conclusiones de este trabajo, así como algunas recomendaciones para mejoras futuras del mismo.

CAPITULO I

Sistemas fotovoltaicos

1.1 GENERALIDADES

La conversión directa de luz solar en electricidad, es una tecnología emergente que empieza a cobrar gran fuerza por la simplicidad de su instalación y por sus estudios económicos favorables. Para llevar a cabo dicha conversión es necesaria la utilización de varios elementos que permiten captar y almacenar la energía proveniente del sol. A estos elementos en su conjunto se le denomina *sistemas fotovoltaicos*.

Puede abreviarse la palabra fotovoltaico como FV (en inglés PV, de *photovoltaic*), cuyo significado se deriva de las dos palabras siguientes:
photo - (fot'o) Producido por luz
voltalc - (váltaic) Producción de corriente y voltaje eléctricos

Los sistemas fotovoltaicos convierten la luz solar directamente en electricidad mediante pequeños dispositivos llamados *celdas solares*. Las celdas solares (o fotovoltaicas como también se les conoce) se elaboran a partir de delgadas capas de material semiconductor, generalmente cristal de silicio.

La luz, en teoría corpuscular y ondulatoria, se describe como una partícula conocida con el nombre de **fotón**. El efecto fotovoltaico en que se fundamenta el mecanismo de conversión energética de la celda solar, consiste esencialmente en la transformación de la energía proveniente de los fotones contenidos en la radiación solar a energía de tipo continuo (corriente directa).

Una sola celda fotovoltaica tiene una producción eléctrica muy pequeña, por lo que para lograr la corriente y el voltaje requeridos en una aplicación dada es necesario conectarlas en serie o en paralelo. La electricidad producida por este medio puede ser enviada directamente a la carga, almacenada en baterías o alimentada a sistemas de red eléctrica.

Debido a que las celdas fotovoltaicas en conjunto tienen un amplio rango de aplicaciones, ya sea para uso doméstico o productivo (alimentación a red eléctrica), se clasifican por su tamaño en varias formas que se explican a continuación:

Módulo fotovoltaico considerado como la unidad más pequeña dentro de la clasificación de equipo generador de electricidad producida por el sol y está integrado por un conjunto básico de celdas, desde menos de una docena hasta cien celdas.

Arreglo que puede definirse como una agrupación de módulos interconectados en serie y/o paralelo, ya sea para incrementar el voltaje y/o la corriente necesarios para alimentar una gran carga.

El tipo de electricidad generada por una celda fotovoltaica es **Corriente Directa (C.D.)**. La corriente directa tiene una polaridad definida esto es, que circula en una sola dirección.

Un corto circuito en un sistema fotovoltaico puede producir corriente hasta un cierto límite, a diferencia de los sistemas de corriente alterna que tienden a generar un flujo infinito de corriente, por lo que al producirse un fenómeno de este tipo no se daña ningún componente del sistema. Sin embargo, si el corto circuito se produce en algún componente ajeno al módulo, por ejemplo en la carga, podrían dañarse otros componentes como las baterías.

Es importante y muy recomendable calcular con la mayor exactitud posible la carga que va a soportar el sistema para evitar una caída sustancial de voltaje. Por ejemplo, podría haber una caída de voltaje si se utilizará cableado innecesario o inadecuado en la aplicación.

Como en los sistemas de corriente alterna, para los sistemas fotovoltaicos la secuencia de conexión y desconexión también es importante para algunos componentes. Tal es el caso de los controladores de carga que deben ser conectados en una secuencia determinada para prevenir posibles daños.

1.2 CONFIGURACION BASICA DE UN SISTEMA FOTOVOLTAICO

El sistema fotovoltaico más simple es aquel que está compuesto por un arreglo de celdas conectado directamente a la carga (sistema directo). Las aplicaciones que requieren energía sólo durante el día son ideales para este tipo de sistemas. El bombeo del agua para irrigación o para almacenamiento en un tanque o tinaco, son ejemplos de estas aplicaciones.

La configuración de un sistema directo puede incrementarse adicionando un *interruptor de carga* que permita desactivar la carga del sistema si se llegara a producir alguna falla.

Es posible implementar al sistema una *batería de almacenamiento*. La batería tiene la función básica de proporcionar energía durante la noche o los días nublados. Una batería actúa como un búffer entre el arreglo solar y la carga, suministrándole energía a la carga durante períodos de insuficiencia solar. Esta energía se almacena en la batería durante los períodos de intensidad solar.

La utilización de una batería de almacenamiento implica la necesidad de adicionar un dispositivo que prevenga la sobrecarga en la misma. La manera más sencilla de resolver este problema es utilizar *módulos de autorregulación*, que son dispositivos diseñados para liberar un voltaje muy bajo que no sobrecarga a la batería, cuando ésta ya está totalmente cargada.

Dado que la mayoría de los sistemas no poseen módulos de autorregulación es posible utilizar un **controlador de carga** (o regulador de carga) que evite la sobrecarga y regule la carga en las baterías. Algunos controladores de carga tienen como característica opcional un **desactivador de carga**, el cual desactiva algunas o todas las cargas que pudiera tener el sistema de acuerdo al estado de carga de las baterías. Es recomendable que un controlador de carga posea un desactivador de carga ya que éste permite extender la vida de las baterías.

Utilizando energía solar es posible generar también corriente alterna para algunas aplicaciones que así lo requieran. Para convertir corriente directa a corriente alterna se utiliza un **inversor**. El inversor debe ser protegido contra temperaturas extremas e inclemencias del tiempo. Su costo es un poco elevado y reduce la eficiencia de sistema, si no es empleado adecuadamente.

Si la carga de corriente alterna opera durante períodos en que el arreglo fotovoltaico no suministra energía se deben incluir una o más baterías y un controlador de carga al sistema fotovoltaico.

De las consideraciones anteriores se deduce que un sistema fotovoltaico se compone básicamente de tres subsistemas:

- 1.- SUBSISTEMA DE GENERACIÓN ELECTRICA
- 2.- SUBSISTEMA DE CONTROL
- 3.- SUBSISTEMA DE ALMACENAMIENTO DE ENERGÍA

A esta clasificación podría agregarse un subsistema adicional si fuera necesario hacer uso de un inversor: el de **condicionamiento de potencia**.

La **carga** no puede considerarse como un subsistema más dentro del sistema fotovoltaico, sin embargo es de suma importancia tomarla en cuenta ya que determina el grado de complejidad del sistema mismo.

En la figura 1.1 se muestra un diagrama a bloques de un sistema fotovoltaico en general.

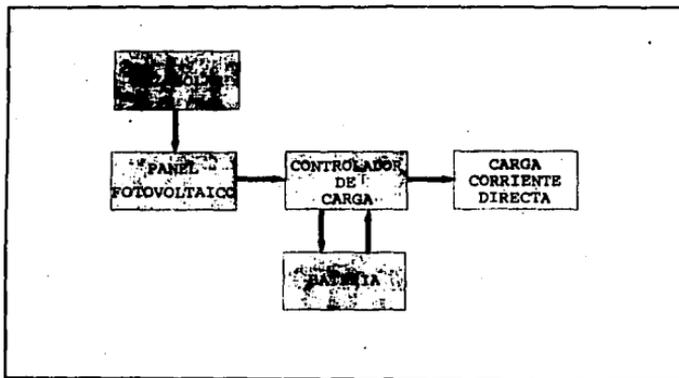


Figura 1.1 Diagrama a bloques de un sistema fotovoltaico

1.3 OPERACION DE LOS COMPONENTES

A continuación se explica con mayor detalle cada subsistema que se especificó en el inciso anterior.

1.3.1 SUBSISTEMA DE GENERACION ELECTRICA

Este subsistema puede estar integrado por un módulo o por un arreglo de módulos fotovoltaicos. Debido a que la unidad primordial de ambos es la celda fotovoltaica, se consideró necesario dar una explicación más amplia, a continuación, de la misma.

1.3.1.1 LA CELDA FOTOVOLTAICA

Actualmente, la mayoría de las celdas fotovoltaicas son hechas a base de silicio (Si). El silicio tiene como características principales su extremada pureza, semiconductividad y abundancia (ocupa el segundo lugar dentro de la clasificación de los más abundantes sobre la corteza terrestre). Posee valencia cuatro en la tabla periódica.

A) Funcionamiento

Las celdas fotovoltaicas, consisten de una unión "P-N" en un semiconductor, entre una capa positiva "P" que contiene cargas positivas móviles o "huecos" y una capa negativa "N" con cargas móviles negativas (electrones).

Cada átomo de silicio tiene cuatro electrones en la última órbita, con los que se liga en la estructura molecular. Si se introduce una impureza tal como el fósforo o el arsénico, con 5 electrones de valencia en la estructura molecular, el material resultante tendrá un exceso de electrones, mismos que se desplazarán dentro de la estructura como portadores de corriente. Este material con exceso de electrones es un semiconductor tipo "N" (negativo). Por otra parte, si al silicio puro se le introduce un elemento como boro o galio con tres electrones de valencia, resulta un material con deficiencia de electrones, esto es, hay un "hueco" por llenar con un electrón por cada átomo de boro o galio. El semiconductor resultante se denomina de tipo "P" (positivo). Los huecos funcionan también como portadores de carga, ya que al llenarse resulta otro hueco en un átomo cercano.

El material de las celdas fotovoltaicas resulta del enlace entre semiconductores "P" y "N". Dicho enlace se logra impregnando impurezas de tipo "N" sobre estructuras de tipo "P" o viceversa.

El proceso fotovoltaico en una celda solar se muestra en la figura 1.2

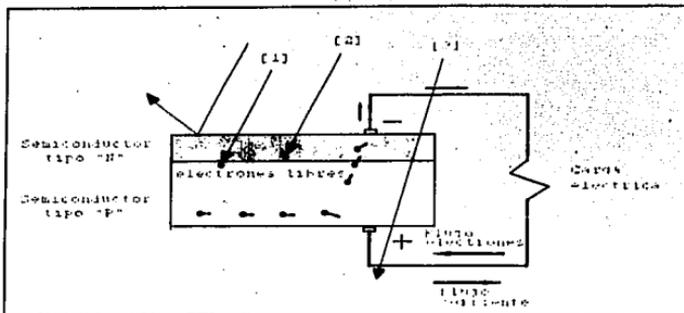


Figura 1.2 Proceso fotovoltaico de una celda solar

Cuando un fotón incide sobre un electrón con la energía apropiada [1] lo libera, creando un hueco en los átomos cercanos al enlace. Los electrones liberados se desplazan hacia el electrodo en la capa "N" [2] y los huecos hacia el electrodo en la capa "P", al cerrar el circuito exterior se crea un flujo de electrones del electrodo "N" al "P" y, por lo tanto, una corriente. Fotones con mayor energía, además de liberar un electrón generan calor, mientras que los de menor energía se absorben como calor o pasan a través de la fotocelda [3].

En esta operación no se consume material alguno y la celda puede operar por tiempo indefinido.

B) Tipos de celdas solares

Existen cuatro tipos generales de celdas fotovoltaicas:

a) Silicio de cristal simple

En la actualidad, la mayor parte de celdas fotovoltaicas son elaboradas a base de este tipo de cristal. Para llevar a cabo su fabricación el silicio es purificado, fundido y cristalizado en lingotes que son cortados en placas muy delgadas. Estas placas tienen un color uniforme que puede ser azul o negro.

Las celdas tienen una ligera carga eléctrica positiva. Una delgada capa en la parte superior tiene la carga negativa. La celda se sujeta a una base denominada respaldo que por lo regular es una capa de metal que refuerza físicamente la celda y proporciona un contacto eléctrico en la parte inferior.

Debido a que la parte superior de la celda debe estar abierta a los rayos solares, se aplica una delgada rejilla de metal en la misma. Esta rejilla debe ser lo suficientemente delgada como para admitir la cantidad adecuada de rayos solares.

Una celda típica produce potencias pequeñas, entre 1 y 2 watts, con un voltaje aproximado de 0.5 volts y 2 amperes de corriente. El tamaño puede variar pero, entre más área tenga la celda, mayor será la cantidad de corriente que produzca.

b) Silicio policristalino

El silicio policristalino está integrado por una gran cantidad de gránulos provenientes del cristal de silicio simple. El tamaño y la calidad de los gránulos determinan su efectos en la configuración de la celda solar.

Las celdas policristalinas se elaboran de una manera similar a las celdas de cristal de silicio simple, sólo que el costo del silicio utilizado en este tipo de celdas es menor, tanto como su eficiencia.

c) Cinta de silicio

Estas celdas son hechas a base de silicio fundido y operan de manera similar a las celdas policristalinas.

d) Silicio amorfo

Los tres tipos de silicio previamente descritos poseen una estructura cristalina diferente al silicio amorfo. Se puede decir que el silicio amorfo no tiene estructura. Se fabrica depositando gran cantidad de placas delgadas de silicio vaporizado en un recipiente de vidrio, plástico o metal. El silicio amorfo no tiene un color definido, ya que puede ser de diferentes colores. Es conocido

también como película de silicio.

Las celdas de este tipo se dimensionan en yardas cuadradas, que pueden ser conectadas en serie para configurar un módulo.

La eficiencia de los módulos fotovoltaicos de silicio amorfo es aproximadamente menor en un 50%, comparados con las tres tecnologías anteriormente descritas.

Existen otros posibles materiales para la elaboración de celdas que están siendo probados, como el arseniato de galio (GaAs) que inicia su auge en este campo.

1.3.1.2 MODULOS FOTOVOLTAICOS

A) Características físicas

En la mayoría de las aplicaciones, el medio volt producido por una sola celda es inadecuado e insuficiente. Es por esta razón que las celdas se conectan en serie para incrementar el voltaje. Si se requiriera incrementar también la corriente, las cadenas conectadas en serie pueden ser conectadas a su vez en paralelo.

Este tipo de conexiones se intercalan entre dos placas: una placa superior de vidrio o plástico transparente y una placa inferior de plástico también o una combinación de plástico y metal. A este paquete se le denomina *módulo* o *pánel*. Típicamente un módulo es el bloque principal en la construcción de un sistema fotovoltaico.

Grupos de módulos pueden ser interconectados en serie y/o paralelo para formar un *arreglo*. Al adicionar a este arreglo baterías de almacenamiento, controladores de carga y dispositivos que condicionen la energía generada se configura lo que se llama *sistema fotovoltaico*.

B) Características eléctricas

Para asegurar la compatibilidad con las baterías de almacenamiento o con la carga es necesario conocer las características eléctricas de los módulos fotovoltaicos.

Dichas características se refieren básicamente al voltaje y a la corriente producidos por el módulo. Como sabemos, la "I" mayúscula es la abreviación de **corriente** expresada en **ámperes**, "V" es utilizada para representar **voltaje** en **volts**.

Un módulo fotovoltaico produce su corriente máxima cuando no hay esencialmente resistencia en el circuito. Esta condición se cumple cuando existe un corto circuito de corriente entre las terminales positivas y negativas.

El máximo de corriente producido es denominado **corriente de corto circuito** (I_{cc}). Cuando el módulo es cortocircuitado, el voltaje en el circuito es cero.

En forma inversa a lo anterior, la producción máxima de voltaje se genera cuando existe una ruptura o apertura en el circuito, llamada comúnmente **voltaje de circuito abierto** (V_{ca}). Bajo esta condición la resistencia es infinitamente alta y no hay corriente. La figura 1.3 muestra una curva típica de corriente-voltaje.

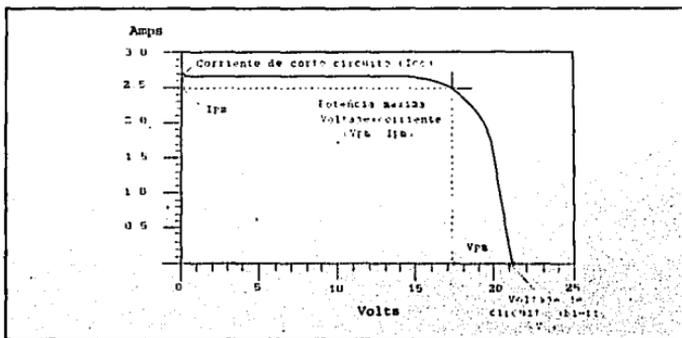


Figura 1.3 Curva típica de corriente-voltaje

FALLA DE ORIGEN

FALLA DE ORIGEN

Como se puede observar en la gráfica, la corriente de corto circuito ocurre en un punto en la curva cuando el voltaje es cero. El voltaje de circuito abierto ocurre cuando la corriente es cero.

Dos elementos muy importantes en un sistema fotovoltaico son la **cantidad de luz disponible** (la cantidad de luz existente en el medio ambiente) que incide sobre él y la **temperatura en la celda**. La cantidad de luz disponible afecta la producción de corriente (a mayor cantidad de luz, mayor producción de corriente). La temperatura de la celda afecta la producción de voltaje (a mayor temperatura de la celda menor voltaje).

La cantidad de **potencia** proveniente de un módulo fotovoltaico en cualquier punto a lo largo de la curva es expresado en **watts**. La potencia en watts se calcula multiplicando la corriente con el voltaje ($W = V \times A$).

La curva (I-V) descrita en la gráfica anterior se basa en un módulo que se encuentra operando en condiciones estándar de radiación solar y temperatura modular.

a) Corriente

Las **condiciones estándar de radiación solar** se asumen como 1000 watts de energía solar por metro cuadrado (1000 W/m^2), a lo que comúnmente se llama **un sol** o un **sol pico**. Si la cantidad de radiación solar se disminuyera a la mitad de un sol, la cantidad de corriente de salida se disminuye también a la mitad.

Si se orienta la parte superior de un módulo fotovoltaico de tal forma que la mayor parte de los rayos solares incidan sobre él, se producirá la potencia máxima de salida.

Debido a que las celdas fotovoltaicas son semiconductores eléctricos, un oscurecimiento parcial del módulo puede ocasionar que las celdas a las cuales afecta este ensombrecimiento se calienten y actúen como conductores ineficientes en lugar de generadores eléctricos, lo que daña seriamente a las celdas. Una sola celda con sombra podría reducir la potencia de salida hasta en un 80%. Si un módulo que se encuentra operando en condiciones estándar se le oscurece momentáneamente, la producción de electricidad se detiene.

b) Voltaje

La temperatura del módulo afecta el voltaje de salida en forma inversa. Una temperatura alta en el módulo puede reducir el voltaje de 0.1 volts a 0.04 volts por cada grado centígrado. Es por ello que se recomienda que los módulos se instalen separados de la superficie donde se van a sujetar para proporcionarles ventilación (a un espacio de 4 a 6 pulgadas).

El voltaje de salida de un módulo depende también del número de celdas conectadas en serie. Los módulos típicos utilizan 30, 32, 33, 36 o 44 celdas conectadas en serie.

Los módulos con 30-32 celdas son considerados módulos de autorregulación.

Los módulos de 36 celdas son los más industriales. Son su ligereza y su alto voltaje registrado (16.7 volts), lo que les permite librar la reducción de voltaje de salida cuando los módulos están operando a altas temperaturas. Sin embargo, requieren que se les adicione un controlador de carga.

Finalmente, los módulos de 44 celdas proporcionan un voltaje estimado en 20.3 volts y se utilizan cuando son necesarios voltajes altos y un extenso cableado en conexiones.

1.3.1.3 ARREGLOS FOTOVOLTAICOS

En la mayoría de las aplicaciones, la potencia de salida de un módulo es inadecuado o insuficiente para la carga. Si esto sucede, los módulos individuales pueden ser interconectados en serie y/o paralelo para incrementar el voltaje y la corriente de salida. Dichas combinaciones sólo se utilizan en los *arreglos fotovoltaicos*.

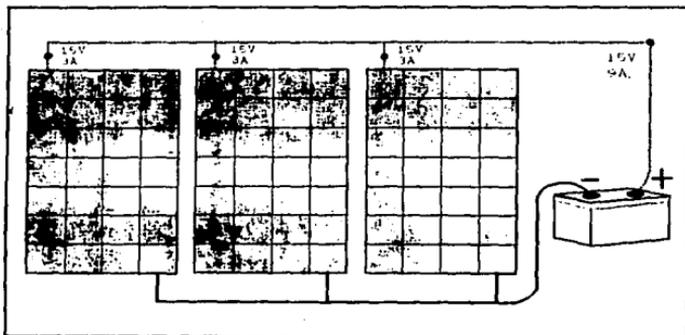


Figura 1.4 Tres módulos conectados en paralelo

Cuando los módulos se conectan en paralelo, la corriente se incrementa y si se conectan en serie es el voltaje quien se incrementa.

Si el sistema utiliza una batería de almacenamiento, puede ocurrir que el flujo de corriente se invierta, esto es, que circule de la batería hacia el arreglo, lo que dañaría la batería. Para evitar la inversión del flujo de corriente se utiliza un **diodo de bloqueo**. Los diodos son dispositivos eléctricos que permiten el flujo de corriente en una sola dirección. Una desventaja del diodo de bloqueo es que provoca una caída de voltaje en el arreglo. Utilizando diodos *schottky* es posible disminuir la caída de voltaje (de 0.3 volts en lugar 0.7 volts).

Algunos sistemas utilizan un controlador de carga para abrir el circuito cuando se detecta una inversión de corriente, en lugar del diodo de bloqueo.

Al conectar varios módulos en serie, es necesario utilizar **diodos "bypass"** que permitan la desactivación del(los) módulo(s) que falle(n) en algún instante, sin perjudicar el funcionamiento de los demás módulos. El diodo *bypass* permite que la corriente proveniente de otros módulos circule a través del lado contrario de donde se encuentra el diodo, eliminando así al módulo dañado.

Los **diodos de aislamiento** son utilizados para aislar una cadena de módulos en serie cuyo flujo de corriente pueda dañar a las demás cadenas o alterar la potencia de salida del arreglo. Se utilizan en las cadenas de módulos conectados en serie y son diferentes a los diodos *bypass*.

A) Arreglos de plato fijo

Los arreglos estacionarios o de plato fijo son los más comunes. Algunos permiten ajustar su ángulo de inclinación en forma horizontal. Este ajuste puede realizarse en varias ocasiones a lo largo del año, pero por lo regular este cambio se hace sólo dos veces al año. Este tipo de arreglo no captura tanta energía como los arreglos móviles y su utilización se limita a lugares remotos o peligrosos.

B) Arreglos portátiles

Los arreglos portátiles son de dimensiones pequeñas (de hasta un pie cuadrado) y pueden ser llevados fácilmente por una persona para realizar diferentes funciones, por ejemplo recargar baterías para comunicaciones. En ocasiones, estos arreglos son montados en vehículos para mantener cargada su batería en largos períodos de inactividad. Algunos pueden instalarse en camiones camper para proporcionar energía durante las salidas campestres.

C) Arreglos móviles

Este tipo de arreglos se nombran así debido a que tienen la capacidad de seguir (o rastrear) al sol en su paso por el cielo. Su seguimiento puede ser en uno o dos ejes. Es su capacidad de rastreo solar lo que les permite captar grandes cantidades de energía, ya que la luz solar incide casi perpendicularmente sobre ellos. Funcionan mejor en lugares limpios y despejados.

Normalmente, los arreglos de un solo eje siguen al sol de este a oeste. El ángulo entre los módulos y la tierra no cambia. Los arreglos de dos ejes cambian en dirección este-oeste y a su vez el ángulo de inclinación con respecto a la Tierra, siendo su única desventaja su complejidad en comparación con los anteriores.

1.3.1.4 SUBSISTEMA DE ALMACENAMIENTO DE ENERGÍA

Este subsistema está representado por las baterías que son dispositivos de almacenamiento de energía eléctrica generada por los módulos fotovoltaicos durante el período de exposición solar. Dicha energía es liberada durante las noches o cuando hay días nublados. Sin embargo, si la energía requerida por la carga es mayor que la suministrada por el arreglo durante el día, las baterías pueden complementar a la energía proporcionada por el sistema fotovoltaico.

Al intervalo en el cual se incluyen un período de recarga y un período de descarga de las baterías se le denomina *ciclo*. Idealmente, las baterías son recargadas al 100% de su capacidad durante la fase de recarga, pero no pueden ser descargadas totalmente en cada ciclo.

Si el sistema no incluye un controlador de carga, las baterías pueden dañarse seriamente por las sobrecargas, al grado de tener la necesidad de sustituirlas.

El desempeño de las baterías de almacenamiento se puede describir de dos maneras:

A) Capacidad amp-hora

La capacidad amp-hora es simplemente el número de amperes que pueden ser liberados por una batería en el transcurso de una hora.

Los diseñadores de sistemas utilizan esta especificación para determinar en qué medida pueden operar los sistemas sin recibir energía proveniente del sol que recargue las baterías. Esta medición de *días de autonomía* es parte importante dentro del diseño de un sistema.

En teoría, una batería de 200 amp-hora es capaz de liberar 200 amperes en una hora, 50 amperes por cuatro horas y un amper por 200 horas. Esto no es totalmente cierto ya que algunas baterías son diseñadas para realizar descargas rápidas en cortos períodos de tiempo (como las baterías de los automóviles) sin dañarse y otras son diseñadas para realizar descargas lentas en largos períodos de tiempo. De la reflexión anterior, se puede decir que las baterías automotrices no deben ser empleadas en sistemas fotovoltaicos.

Si la batería es cargada y descargada en un índice diferente al especificado, la capacidad amp-hora puede incrementarse o decrementarse.

La temperatura es otro factor que afecta esta capacidad. Una temperatura baja reduce su capacidad.

B) Profundidad de descarga

Esta describe el porcentaje del total de la capacidad amp-hora en la batería que es utilizado durante un ciclo de carga-descarga de la misma.

Existen diferentes tipos de baterías entre las que destacan las baterías de plomo-ácido y las de níquel-cadmio.

Las baterías, como las celdas fotovoltaicas, pueden ser conectadas en serie para incrementar el voltaje y conectadas en paralelo para incrementar la corriente. A un grupo de baterías interconectadas de esta manera se le denomina banco de baterías. La figura 1.5, muestra este tipo de conexiones.

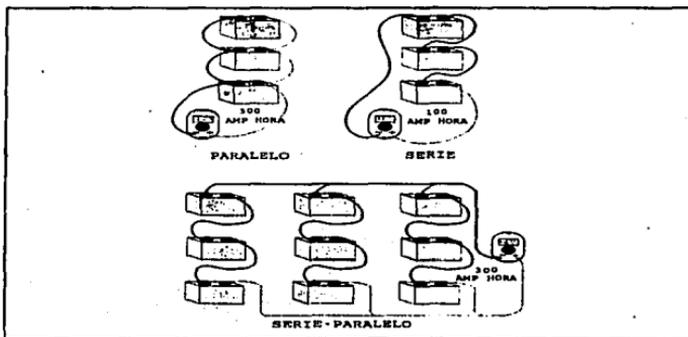


Figura 1.5 Baterías conectadas en serie, paralelo y paralelo_serie

1.3.1.5 SUBSISTEMA DE CONTROL

El subsistema de control está representado por el controlador de carga. La función primaria de un controlador de carga es la de proteger a la batería de una sobrecarga o sobredescarga. Cualquier sistema que tenga una carga impredecible y una batería de almacenamiento o alguna característica que sobrecargue o sobredescargue excesivamente a la batería necesita de un controlador de carga y/o un desconector de carga a bajo voltaje. Los sistemas con una carga pequeña predecible y continua pueden ser diseñados para operar sin un controlador de carga.

Los datos que el fabricante proporciona permiten establecer los límites de utilización del controlador tales como corriente de carga, temperatura de operación, pérdidas, etc.

Existen varios métodos básicos para regular la carga de la batería en un módulo o en arreglo. El método que describiremos como ejemplo es el llamado *regulación Shunt*. Cualquier método puede utilizarse efectivamente, ya que cada uno puede incorporar cierto número de variaciones que podrían alterar el comportamiento básico y la aplicabilidad de un controlador de carga.

A) Controlador Shunt

Un controlador Shunt regula la carga de una batería interrumpiendo la corriente fotovoltaica del sistema mediante un corto circuito. Un diodo de bloqueo es utilizado en serie entre la batería y el elemento del sistema que se está cortocircuitando.

Si la interconexión de las baterías se realiza tanto en serie como en paralelo, se incrementa tanto el voltaje como la capacidad amp-hora del banco de baterías. Las conexiones en buen estado y el cableado mínimo son indispensables para alargar la vida útil de éstas.

1.4 VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS FOTOVOLTAICOS

1.4.1 VENTAJAS

- No tienen partes móviles que se desgasten fácilmente.
- No contienen fluidos o gases que puedan derramarse o fugarse lo que los hace altamente seguros y no contaminantes.
- Tienen respuesta rápida, alcanzando un nivel de producción alto casi inmediatamente.
- Requieren poco mantenimiento si están bien fabricados e instalados correctamente.
- Se pueden fabricar de silicio, el segundo elemento más abundante sobre la corteza terrestre.
- Son modulares, es decir permiten un amplio rango de aplicaciones solar-eléctricas como: *pequeña escala*, para aplicaciones remotas o de uso residencial; *escala intermedia*, como potencia suplementaria para comercios y comunidades y; *gran escala*, para estaciones de potencia centralizadas.
- Tienen amplio rango de capacidad de generación, de microwatts a megawatts.
- Se prestan para instalaciones locales, esto es, potencia descentralizada y dispersa (por ejemplo zonas rurales de nuestro país).

1.4.2 DESVENTAJAS

- Actualmente no son económicamente competitivos para la mayoría de las aplicaciones, principalmente en las escalas intermedia y grande.
- Su fabricación involucra el manejo de sustancias que pueden ser nocivas para el ambiente en caso de descargas accidentales.

- Las tecnologías de producción se encuentran controladas por los países industrializados.

1.5 CARACTERISTICAS DE LOS MODULOS FOTOVOLTAICOS ESTUDIADAS EN EL IIE

1.5.1 CARACTERISTICAS DEL SISTEMA FOTOVOLTAICO

Existen varios sistemas fotovoltaicos que constituyen el banco de pruebas del Instituto de Investigaciones Eléctricas. La marca de cada componente varía, ya que se busca cual es el mejor en cuanto a rendimiento y eficiencia, así como vida útil.

Cada sistema está compuesto por un panel o módulo único, una batería, un controlador de carga y una carga pequeña (lámparas fluorescentes). Puede decirse que estos sistemas son a pequeña escala, ya que se pretende con ellos (como se mencionó en la introducción) dar servicio de iluminación a zonas rurales.

A continuación se describe el modelo de un módulo típico en el banco de pruebas de los sistemas fotovoltaicos:

Módulo M75. Este módulo es fabricado por la empresa Siemens. Utiliza celdas fotovoltaicas de silicio monocristalino. Consta de 33 celdas solares conectadas en serie y tiene una superficie total de 0.40 metros cuadrados. Cuenta con dos cajas en la parte posterior: una positiva y una negativa las cuales contienen un diodo de paso y son de cierre hermético.

Características eléctricas módulo Siemens modelo M75	
Potencia nominal	48 watts a 25 °C
Corriente nominal	3.02 amperes
Voltaje nominal	15.9 volts
Corriente de circuito abierto	3.4 amperes a 25 °C
Voltaje de circuito abierto	19.8 volts a 25 °C

- **Batería.**- Se están probando también baterías de diferentes marcas, pero la más sobresaliente es la de marca CALE que es una batería de plomo-ácido con una capacidad de 90 amperes-hora a 12 volts.

- **Controlador de carga.**- Los controladores de carga son de la marca CONDUMEX para trabajar a 12 volts con un manejo de corriente de 15 amperes de módulo a batería y de 20 amperes de batería a carga. .

- **Carga.**- La carga está representada por 2 lámparas fluorescentes de 9 y 13 watts de la marca OSRAM.

La figura 1.6 nos muestra la fotografía de uno de los módulos en prueba.

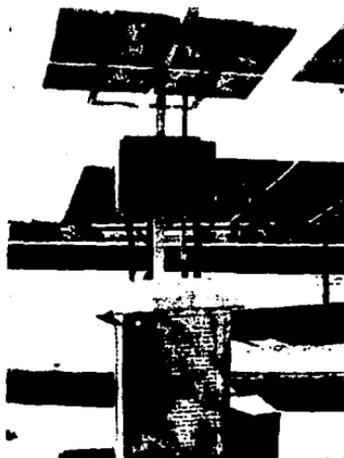


Figura 1.6 Fotografía de un pequeño sistema fotovoltaico puesto a prueba para electrificación rural por el Instituto de Investigaciones Eléctricas

1.5.2 CARACTERISTICAS DEL SISTEMA DE ADQUISICION DE DATOS

El sistema de adquisición de datos está integrado por un módulo electrónico de medición de variables, por una computadora personal y dispositivos de medición.

El módulo electrónico de medición de variables es un equipo de la línea Nectpac modelo X7225 de la marca norteamericana Kaye Instruments con capacidad de 20 canales analógicos. Todos los sistemas electrónicos de este tipo que se encuentran en uso se comunican con la computadora personal por medio de una línea de comunicación RS-485. Los módulos de medición se configuraron de acuerdo al tipo de medición por medio del programa manejador de entrada/salida de FIX DMACS. Se utiliza un convertidor de puerto RS-232 a RS-485.

Se realiza la medición variables por medio de sensores que son transductores que permiten medir el valor de las siguientes variables: voltaje y corriente en el módulo, voltaje y corriente en la batería, temperatura tanto en la batería como en el módulo fotovoltaico, radiación solar, temperatura ambiente, velocidad del viento, presión atmosférica y precipitación pluvial.

La computadora personal utilizada tiene las siguientes características:

- Marca Olivetti
- Microprocesador 486
- Coprocesador matemático
- Velocidad 66 MHz
- Memoria RAM de 4 Mbytes
- Disco duro de 100 Mbytes
- 1 puerto serie
- 1 puerto paralelo
- 1 manejador de disco flexible de 5.25 pulgadas
- 1 manejador de disco flexible de 3.5 pulgadas
- 1 *ratón* (del inglés *mouse*)

En ella se encuentra instalado el programa multitareas FIX DMACS mediante el cual se comunica el sistema electrónico de adquisición de datos con la computadora personal. Dentro de este programa multitareas se dispone de un programa que adquiere los datos a partir de la información transferida por el sistema electrónico de adquisición de datos. Dichos datos son enviados a diferentes archivos que son almacenados en el disco duro.

Resumiendo, el sol es una fuente inagotable de energía, energía que puede ser captada y convertida en *corriente directa* por medio de celdas solares. La *celdas solares o fotovoltaicas* se elaboran con material semiconductor, generalmente cristal de silicio (en sus diferentes formas).

Debido a su amplio rango de aplicaciones, las celdas solares se agrupan en 2 formas: módulos y arreglos fotovoltaicos. Los *módulos* están compuestos desde menos de una docena hasta cien celdas. Los *arreglos* son grupos de módulos conectados en serie y/o paralelo.

Un *sistema fotovoltaico básico* está integrado por un *módulo o panel* y la *carga*. Esta configuración básica puede incrementarse si se agregan otros elementos como: *batería* para almacenar energía que será utilizada por la carga durante las noches o cuando haya días nublados, *controlador de carga* que sirve para controlar el voltaje y la corriente tanto en la carga como en las baterías y un *inversor* para convertir la corriente directa en corriente alterna para aplicaciones que necesiten este tipo de corriente.

Debido a que el servicio de energía eléctrica no llega a a todas la comunidades de nuestro país se han elaborado planes para generar energía eléctrica por captación solar. El Instituto de Investigaciones Eléctricas (IIE) participa en estos planes realizando pruebas para caracterizar pequeños sistemas fotovoltaicos integrados por un módulo, un controlador de carga, una batería y una carga que consta de 2 lámparas fluorescentes.

Hasta aquí, hemos dado algunos conceptos que nos serán de gran utilidad para la elaboración de un generador de reportes que nos permita caracterizar a los sistemas fotovoltaicos del banco de pruebas.

CAPITULO 2

El programa FIX DMACS

2.1 CONSIDERACIONES BASICAS

FIX *Fully Integrated Control System*

DMACS *Distributed Manufacturing, Automation and Control Software*

Es un paquete de *software* registrado por Intellution que consta de varios programas diseñados para proporcionar el monitoreo y control completo de procesos automatizados. Estos procesos pueden ser de varios tipos: elaboración de productos químicos, elaboración de alimentos enlatados, ensamblaje de automóviles, generación de energía eléctrica, etc.

Este *software* consta de dos tipos de programas definidos como programas de *bajo nivel* y programas de *alto nivel*. Los programas de *bajo nivel* interactúan con el operador y el despliegue de datos, además de otras funciones. Los programas de *alto nivel*, una vez inicializados, se ejecutan sin intervención del operador.

Este conjunto de programas está diseñado para instalarse de dos maneras: en red de área local o en computadora personal única. La *versión distribuida* se instala en la configuración tipo red y la *versión para nodo único* se utiliza para instalarse en una computadora personal.

La *versión distribuida* puede ser instalada para trabajar con un grupo de computadoras personales, donde una(s) de ellas toma(n) el papel de nodo(s) *SCADA (Supervisory, Control and Data Acquisition)* y las demás actúan como nodos *View* o de despliegue que permiten monitorear el proceso, enviar y almacenar los datos obtenidos al o a los nodo(s) *SCADA*, sin intervenir en el proceso de estos últimos. En caso de tener una sola computadora, es posible instalar la *versión para nodo único* donde la computadora personal actúa como nodo *SCADA* y como nodo *View* a la vez.

La función principal que realiza el nodo *SCADA* consiste en recibir datos provenientes de los demás nodos y ajustar el proceso que se está ejecutando de acuerdo a los datos recibidos. La información proveniente de un nodo *SCADA* puede ser enviada a programas de aplicación en los nodos *View* o a otros nodos *SCADA* (sólo en caso de tener configuración tipo red).

2.1.1 CONSIDERACIONES DE *HARDWARE*

- Computadora personal 386 ó 486 compatible 100% con IBM
- 2 Mb en RAM (como mínimo)
- Coprocesador matemático
- Una unidad de disco flexible (3.5 ó 5.25 pulgadas)
- Disco duro con 20 Mb libres (como mínimo)
- Un puerto paralelo
- Monitor a color EGA o VGA
- Un *ratón* de dos botones
- Llave para protección de *software*, que debe configurarse para permitir el acceso a determinados programas de FIX DMACS

2.1.2 CONSIDERACIONES DE *SOFTWARE*

- Sistema operativo MS-DOS o PC-DOS versión 5.0
- Manejador de memoria expandida
- Manejador de entradas y salidas para nodos *SCADA*

2.1.3 INTERFAZ DE USUARIO

Los menús, pantallas y mensajes que se observan en FIX DMACS constituyen su interfaz de usuario. Esta interfaz proporciona al usuario suficientes facilidades de operación, siendo una de ellas la ayuda en línea. Además emplea menús que son puntos o pantallas de entrada a otras aplicaciones.

El principal punto de entrada es el **MASTER MENU** que se muestra en la siguiente figura:

MASTER MENU 3 0	
FOREGROUND TASKS	BACKGROUND TASKS
PVIEW - PIXEL OPERATOR DISPLAY	*SACH - SAC FOR EHS 4 0
PDRAW - PIXEL DISPLAY BUILDER	*NET - NET POLLING TASK
DBC - DATABASE CREATE	*SCHD - SCHEDULER INTELEC
DRB - DATABASE BUILDER	ALH - ALARM SERVICES
DWS - DOS WINDOW	*HTC - HISTORICAL COLLECTION
SUN - ALARM SUMMARY DISPLAY	SCHG - SCHEDULER
TGE - TAG GROUP EDITOR	
KHE - KEY HACKO EDITOR	
SCU - SYSTEM CONFIG UTILITY	
LOG - SECURITY LOGIN	
HTA - HISTORICAL ASSIGNMENT	
HTD - HISTORICAL DISPLAY	
RCP - EDITOR DE RECETAS	
SC - PROGRAMACION HORARIA	
REP - REPORT CREATOR	
REPORTES	

Figure 2.1 Menú principal de FIX DMACS (MASTER MENU)

La configuración de esta pantalla depende de los programas de *bajo* y *alto nivel* que el usuario haya seleccionado para su instalación.

Para entrar y salir de los menús de FIX DMACS se utiliza otro menú de entrada y salida que se muestra a continuación:

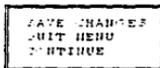


Figura 2.2 Menú de entrada-salida para FIX DMACS

Save Changes. Salva los cambios y sale de la aplicación.

Quit menu. Sale de la aplicación sin salvar los cambios.

Continue. Permite que se siga ejecutando la aplicación aunque se haya intentado salir de la misma.

Se accesa a los programas de dos maneras: La primera consiste en posicionarse en la opción deseada y presionar la tecla <ENTER>. La segunda consiste en posicionarse con el *ratón* en la opción deseada y pulsar el botón izquierdo.

2.1.4 ARQUITECTURA

La siguiente figura muestra un panorama general de la arquitectura del programa FIX DMACS.

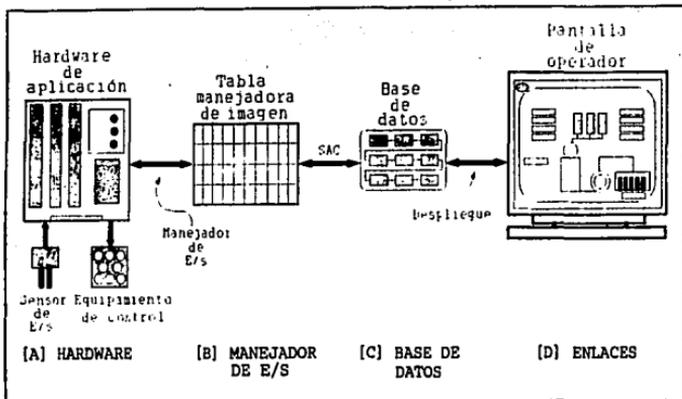


Figura 2.3 Panorama general de la arquitectura de FIX DMACS

A) El **HARDWARE** enlaza por un lado a los sensores de entrada y salida y por otro lado al puerto serial o puerto *com* de la computadora personal. Los sensores de entrada y salida pueden compararse con el sistema nervioso que transmite los impulsos del cuerpo humano (el hardware de aplicación) al cerebro (el PLC del inglés *Programmable Logic Controlator* = Controlador Lógico Programable).

B) El **MANEJADOR DE ENTRADA Y SALIDA** obtiene los datos del hardware de aplicación y los libera en la *tabla manejadora de imagen*. Tanto la *tabla manejadora de imagen* como la base de datos residen en la computadora personal. El programa de *búsqueda, alarma y control (SAC)* accesa a la *tabla manejadora de imagen* en intervalos definidos por el usuario, regresa esos datos y los respalda en la base de datos. La tabla manejadora de datos puede compararse con un correo, donde la información es liberada y enviada a la dirección que se indica.

C) La **BASE DE DATOS** está integrada por conjuntos de bloques que realizan funciones especiales con los datos procesados. Esos datos son respaldados por la **SAC** con valores provenientes de la *tabla manejadora de imagen*. Si los bloques son encadenados en forma adecuada, esos bloques pueden analizar los datos procesados o permitir el control de determinada aplicación.

D) Los **ENLACES** permiten monitorear los datos procesados en tiempo real por los bloques. Esto es, actúan como ventanas de entrada a la base de datos. Si los valores cambian en los bloques, los enlaces reflejan este cambio en la pantalla de operador. Utilizando una ventana de operador, es posible monitorear la aplicación y responder a las condiciones de alarma o ajustar el proceso en cualquier sitio.

2.2 PROGRAMAS QUE LO CONSTITUYEN

2.2.1 PROGRAMAS DE ALTO NIVEL

Proporcionan la capacidad de ejecución simultánea de otros programas (de alto y bajo nivel), mientras se están adquiriendo datos en tiempo real. Los programas responsables de esta capacidad son los siguientes:

2.2.1.1 EMS

Está representado por el *software manejador de memoria expandida* (del inglés *Expanded Memory Support*) y proporciona la memoria necesaria para que varios programas se ejecuten simultáneamente.

2.2.1.2 SHELL

Este programa permite a la computadora realizar operaciones multitarea. Es por esta característica que es posible monitorear un proceso desde la pantalla de operador, mientras se construyen interfaces gráficas y se almacenan datos. FIX DMACS corre bajo dos versiones del Shell: el estándar y el ampliado. La versión correcta se ejecuta automáticamente cuando el FIX DMACS es cargado.

2.2.1.3 SAC

El programa de *búsqueda, alarma y control* (del inglés *Search, Alarm and Control*) es el responsable de actualizar continuamente los valores de los bloques de la base de datos con valores provenientes de la *tabla manejadora de imagen*. Sin la SAC, los bloques de la base de datos nunca se comunicarían con el proceso de *hardware*. Una de las responsabilidades de la SAC es procesar las instrucciones en los bloques de la base de datos. La SAC realiza esta tarea de acuerdo a los intervalos de tiempo que se definen en el menú de configuración de cada bloque.

2.2.2 PROGRAMAS DE BAJO NIVEL

Estos programas proporcionan las facilidades necesarias para trabajar con FIX DMACS. Por ejemplo, permiten crear una base de datos, realizar una interfaz gráfica, determinar los intervalos de tiempo en que van a procesarse los datos, etc. Cabe mencionar que estos programas se encuentran ubicados en la parte izquierda del *MASTER MENU* o menú principal de FIX DMACS.

A continuación se describirá con más detalle cómo funciona cada uno de ellos:

2.2.2.1 DATABASE

La fuente principal de toda información es el proceso mismo (controladores, sensores, motores, entre otros). Dicha información es procesada por la base de datos.

La base de datos juega un papel muy importante dentro del proceso de automatización.

Para diseñar una base de datos en FIX DMACS es necesario analizar con detenimiento la aplicación que se quiere automatizar y que va ser la fuente de todos los datos.

Para realizar un proceso de automatización es necesario:

- Analizar el proceso a automatizar y hacer el registro de cómo opera, incluyendo:
 - Entradas y salidas del dispositivo controlador.
 - Tipos de dispositivos a emplear.
 - Unidades de dimensionamiento para los valores a obtener o procesar.
- Diseñar una estrategia de automatización que explique cómo se quiere automatizar el proceso.
- Crear un algoritmo que combine el análisis del proceso y la estrategia de automatización.
- Dibujar un diagrama de flujo listando los tipos de bloques que realizar cada paso en la automatización y las instrucciones específicas para cada bloque.

Como se sabe, el corazón del nodo *SCADA* es su propia base de datos. Dicha base de datos tiene como propósitos generales:

- La recepción de valores de entrada.
- La manipulación de esos datos de acuerdo a la estrategia de control utilizada, es decir si cumple con el objetivo propuesto (generación de otras salidas, operaciones matemáticas, etc).
- Checar los valores de entrada y salida de acuerdo a los límites de los mismos.
- Ajustar los valores anteriores si es preciso.
- Enviar señales de alarma a pantalla de operador, impresoras, archivos y otros.

La siguiente figura ilustra cómo son introducidos los valores a una base de datos, cómo pasan a través de una secuencia de bloques y cómo se obtiene la salida de la base de datos.



Figura 2.4 Entrada y salida de datos en una base de datos creada en FIX DMACS

Como ya se ha mencionado, la base de datos está integrada por bloques. Los **bloques** son unidades de instrucciones capaces de realizar alguna o todas las siguientes tareas:

- Recibir valores de otros bloques o directamente de la *tabla manejadora de imagen*.
- Manipular los valores de acuerdo a sus instrucciones.
- Realizar cálculos basándose en algún valor.
- Regresar valores procesados a la *tabla manejadora de imagen*.

FIX DMACS proporciona diferentes tipos de bloques, cada uno capaz de realizar una función única.

Cuando FIX DMACS es instalado en el nodo *SCADA*, se crea una base de datos vacía con espacio reservado para cada grupo de bloques. Los bloques aún no han sido configurados, pero la memoria ha sido reservada para ellos.

Cuando se quiere utilizar un bloque para automatizar un paso específico del proceso, se adiciona uno a la base de datos utilizando el programa *Database Builder (DBB)*, como se muestra en la siguiente figura:

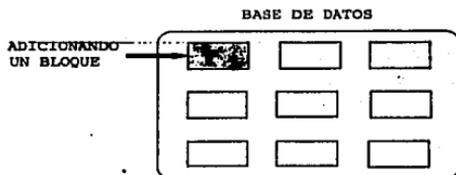


Figura 2.5 Adicionando un bloque a una base de datos en FIX DMACS

Siguiendo el paso anterior, el usuario define las instrucciones específicas a través del menú que DBB despliega. Esta información incluye:

- El identificador del bloque o *tag name*.
- Si los valores o información van a ser recibidos de la *tabla manejadora de imagen* o de otro bloque.
- Cuándo debe ser enviada la información.
- Si los valores van a ser manipulados.
- Si los valores van a ser escalados para la pantalla de operador.

Si los bloques son encadenados adecuadamente, éstos pueden chequear, automatizar, adquirir datos y mantener un proceso. Un proceso de automatización muy largo, involucra un gran número de entradas y salidas lo que implica una gran cantidad de cadenas de bloques que procesen esos valores.

Una vez construida la cadena de bloques, FIX DMACS procesa cada instrucción del bloque en el intervalo de tiempo que el usuario define. Como ya mencionamos anteriormente, la SAC es la responsable de procesar adecuadamente la base de datos.

Existen diferentes tipos de bloques para bases de datos en FIX DMACS, los cuales han sido divididos en bloques primarios, de salida, de control, secundarios, por lotes, para control de procesos estadísticos y bloques SQL.

Los *bloques primarios* reciben datos de la *tabla manejadora de imagen*, entre otros procesos. Por lo regular estos bloques se asocian con una o más piezas de *hardware*.

Los *bloques de salida* regresan los valores a la *tabla manejadora de imagen*.

Los *bloques de control* proporcionan la capacidad de control continuo, directo y digital.

Los *bloques secundarios* manipulan los datos de acuerdo a sus instrucciones. Deben recibir su entrada de un bloque primario o de un bloque anterior y realizar una función específica para la salida. Los bloques secundarios no pueden estar al principio de una cadena.

Los *bloques por lotes* son especialmente diseñados para operaciones de control discontinuo.

Los *bloques de control de procesos estadísticos* permiten el análisis estadístico de datos, así como cálculos, control supervisorio y despliegue de los datos estadísticos.

Los *bloques SQL* leen y escriben datos a una base de datos relacional o aun servidor remoto (cuando se tiene la *versión distribuida* de FIX DMACS).

El diseño de cada cadena varía de acuerdo con la tarea que vaya a realizar. Las cadenas con funciones como la adquisición y control de datos, reciben entradas provenientes de la *tabla manejadora de imagen* a través de un bloque de entrada, manipulan los datos con los bloques secundarios y retornan un valor ajustado a la *tabla manejadora de imagen* a través de los bloques de salida. La siguiente figura muestra el diseño de una cadena para adquisición y control de datos.

Existen diferentes tipos de bloques para bases de datos en FIX DMACS, los cuales han sido divididos en bloques primarios, de salida, de control, secundarios, por lotes, para control de procesos estadísticos y bloques SQL.

Los *bloques primarios* reciben datos de la *tabla manejadora de imagen*, entre otros procesos. Por lo regular estos bloques se asocian con una o más piezas de *hardware*.

Los *bloques de salida* regresan los valores a la *tabla manejadora de imagen*.

Los *bloques de control* proporcionan la capacidad de control continuo, directo y digital.

Los *bloques secundarios* manipulan los datos de acuerdo a sus instrucciones. Deben recibir su entrada de un bloque primario o de un bloque anterior y realizar una función específica para la salida. Los bloques secundarios no pueden estar al principio de una cadena.

Los *bloques por lotes* son especialmente diseñados para operaciones de control discontinuo.

Los *bloques de control de procesos estadísticos* permiten el análisis estadístico de datos, así como cálculos, control supervisorio y despliegue de los datos estadísticos.

Los *bloques SQL* leen y escriben datos a una base de datos relacional o aun servidor remoto (cuando se tiene la *versión distribuida* de FIX DMACS).

El diseño de cada cadena varía de acuerdo con la tarea que vaya a realizar. Las cadenas con funciones como la adquisición y control de datos, reciben entradas provenientes de la *tabla manejadora de imagen* a través de un bloque de entrada, manipulan los datos con los bloques secundarios y retornan un valor ajustado a la *tabla manejadora de imagen* a través de los bloques de salida. La siguiente figura muestra el diseño de una cadena para adquisición y control de datos.



Figura 2.6 Ejemplo de una cadena para adquisición y control de datos

El análisis, diseño del esqueleto e implementación de la base de datos que guarda los valores que se monitorean de los sistemas fotovoltaicos se elaboró de la siguiente forma:

■ Se determinó qué mediciones eran necesarias para caracterizar y evaluar un sistema fotovoltaico. Las variables fundamentales que debían medirse son las siguientes:

- Voltaje
- Corriente
- Temperatura del panel fotovoltaico
- Temperatura de batería
- Irradiación solar
- Dirección del viento
- Velocidad del viento
- Temperatura ambiente
- Precipitación pluvial
- Presión atmosférica

■ Una vez determinadas las variables a medir y monitorear se construyó la base de datos mediante la opción *Database Builder* (este programa permite llenar el esqueleto de la base de datos creado por el programa complementario *Database Create*). La base de datos para este proceso se constituyó de acuerdo a lo siguiente:

■ Creación de bloques de entrada, definiendo:

- Tipo de medición (analógica, digital, etc.).
- Dirección del canal donde se medirá (este canal se encuentra en el módulo electrónico que se utiliza para tomar las mediciones que se especificaron anteriormente).

- Definición de etiquetas asignables a cada variable de medición y/o control (p.ej. VFV_1 para voltaje fotovoltaico en el sistema uno).
 - Intervalo de muestreo para cada variable.
- Creación de bloques de cálculo aplicables a cada variable medible y/o al resultado de otros bloques de cálculo en forma encadenada.
- Creación de bloques de salida que muestran la manipulación de los datos obtenidos de las mediciones realizadas.

Los pasos anteriores se realizan para todas las variables del proceso, en función de la información que se requiere monitorear, los datos a adquirir y las acciones de control a efectuar.

2.2.2.2 HISTORICAL TRENDING

Proporciona un campo de aplicación muy amplio para muestreo, despliegue y almacenamiento de datos en proceso. Una vez instalado este programa se puede acceder automáticamente a la colección de datos.

Cuando los datos son colectados por el nodo SCADA y son almacenados en los nodos designados como nodos de colección (en caso de tener una configuración tipo red) o en la computadora personal que actúa como nodo SCADA y colector de datos a la vez, es posible utilizar el programa *Historical Display* para mostrar esos datos en una gráfica multipuntos. Los datos que se despliegan en las gráficas pueden ser guardados en archivos ASCII.

El *HISTORICAL TRENDING* está compuesto por tres programas, que aparecen en el menú principal:

- *Historical Trend Assignment*. Mediante este programa se determinan cuáles serán los bloques de donde se van a coleccionar datos históricos y los parámetros en común de los mismos, considerándolos como grupo de colección. Esta determinación se lleva a cabo basándose en la estrategia de colección de datos que debe ser elaborada antes de utilizar dicho programa.

- **Historical Trend Collection.** Se utiliza para indicar que los datos históricos deben empezar a colectarse en el momento de la activación de dicho programa.

- **Historical Trend Display.** Este programa transfiere la colección de datos históricos a despliegues gráficos. Las gráficas pueden ser enviadas a impresión o exportadas a archivos ASCII que guardan únicamente los datos graficados en ese momento. Dichos archivos toman el nombre que les asigna el usuario.

Historical Trend Collection puede activarse en cualquier nodo (sólo en configuración tipo red) o desde la computadora personal que actúa como nodo SCADA y nodo colector de datos a la vez; sin embargo, los otros dos programas necesitan que la llave de *software* esté configurada para reconocerlos.

Este conjunto de programas permite dos tipos de almacenamiento histórico de datos: automático y manual.

- Los archivos de datos históricos automáticos que se almacenan en forma automática se encuentran en el subdirectorio C:\DMACS\HTR, dichos archivos reciben por nombre la hora y fecha con las cuales están siendo adquiridos.

- Los archivos de datos históricos manuales son almacenados en el subdirectorio C:\DMACS\HTRDATA, y llevan el nombre que les proporciona el usuario.

2.2.2.3 GRAPHICS

Este programa ofrece la ventaja de hacer más reales y detalladas las pantallas de usuario. Consta de dos programas: *Pixel Display Builder (PDRAW)* y *Pixel Operator Display (PVIEW)*.

El programa *Pixel Display Builder* proporciona al usuario algunas facilidades de un paquete gráfico, es decir, permite realizar líneas, círculos, elipses, rectángulos. Además permite colorear y enlazar otras pantallas con gráficos creadas por el mismo paquete. Pero no sólo eso, sino también permite enlazar los datos que se están adquiriendo en tiempo real a la pantalla que se dibuja.

Se puede decir que el programa anterior es sólo el instrumento para enlazar los datos que el usuario requiere con el dibujo que se realiza, sin embargo no permite que el usuario vea la interfaz completa activa. Esta tarea corresponde al programa *Pixel Operator Display* que permite mostrar todas las interfaz de usuario activas que fueron creadas mediante el primer programa. A continuación se muestra la interfaz realizada mediante este paquete para mostrar los datos monitoreados en tiempo real de cada uno de los sistemas fotovoltaicos a prueba.

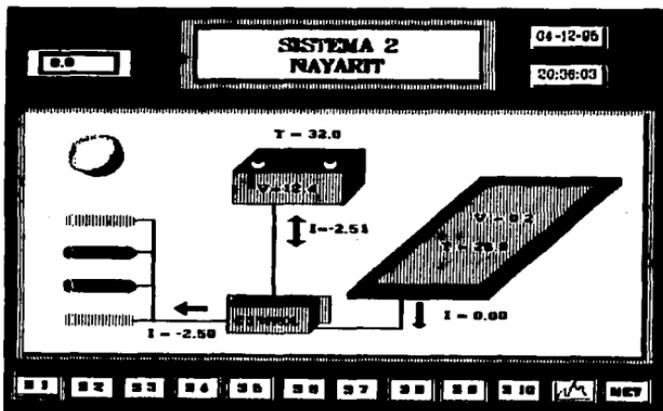


Figura 2.7 Interfaz gráfica para despliegue de datos en tiempo real, provenientes de los sistemas fotovoltaicos

2.2.2.4 REPORT

Mediante este paquete es posible generar formatos para despliegue y almacenamiento de archivos de datos de acuerdo a la fecha y hora en que el usuario los necesite.

Este módulo consta de dos programas. Estos dos programas son **Report Creator** y **Report Generator**.

- a) Se utiliza el programa **Report Creator** para crear un formato fijo de reportes, dicho formato enlaza los valores de la base de datos que el usuario especifique. Los formatos que son creados de esta manera toman por nombre el proporcionado por el usuario y tienen la extensión .DEF.
- b) Para que el formato creado genere los datos automáticamente en un tiempo y fecha determinados se utiliza el programa **Scheduler Config**, que permite al usuario hacer una programación horaria del o de los formato(s) creado(s) mediante **Report Creator**.
- c) Una vez realizada la programación horaria para la generación de reportes, el programa **Scheduler** inicializa el programa **Report Generator** y carga el archivo con el formato definido para la generación de datos. Una vez cargado, el reporte es compilado y enviado a su destinación (archivo, pantalla o impresora).
- d) Es posible compilar un reporte sin auxiliarse del **Scheduler** corriendo el **Report Generator** desde el indicador de sistema operativo. La instrucción correspondiente para compilar y ejecutar un reporte se explica en el siguiente capítulo, ya que es muy útil para el mismo.

El directorio en el cual se graban los archivos para reportes es C:\DMACS\REP.

Este programa tiene las siguientes características.

- Ayuda en línea acerca de los comandos utilizados en el paquete.
- El programa **Report Creator** notifica inmediatamente el formato a utilizar y los errores existentes en el mismo.
- Si al ejecutar el reporte se encuentran errores, éstos se imprimen en forma de mensajes al final de la generación del archivo.
- Todos los archivos principales son automáticamente respaldados. Es posible recuperar un archivo que se ha modificado.

Se tiene acceso a este programa por medio del **MASTER MENU**.

2.2.2.5 RECIPES

Son otro subgrupo de programas de FIX DMACS utilizados para definir, cargar y controlar unidades de proceso (recetas) en la aplicación que se está controlando.

Estos programas permiten definir un conjunto de parámetros de bloques como un simple archivo y, un tiempo más tarde, cargar en el proceso un nuevo grupo de parámetros de bloque.

La definición de unidades de proceso es completamente irrestrictible y dichas unidades pueden ser estructuradas como se quiera en la aplicación. Las recetas pueden ser cargadas automáticamente mediante el programa *Scheduler*.

Este subgrupo consta de tres programas:

RCP. Programa para construir las recetas.

RCLOAD. Programa para cargar recetas.

RCPSAVE. Programa para salvar las recetas (respaldar).

2.2.2.6 SCHEDULER

Este programa se utiliza para predefinir el tiempo de ejecución de una tarea. Con él es posible realizar las siguientes funciones:

- Imprimir reportes.
- Cargar y salvar recetas.
- Correr programas realizados en lenguaje C, mediante el subpaquete *C Database Access program*. Se hace necesario recalcar que dichos programas no deben utilizar interrupciones, ni desplegar datos en pantalla.

Este módulo está integrado por dos programas: *Scheduler Config* y el *Scheduler Background task*.

El primero permite crear una lista de las tareas que necesitan ser realizadas en determinadas circunstancias y/u horarios. El segundo nos permite buscar y monitorear todos los procedimientos necesarios para ejecutar las tareas listadas mediante el primer programa. Este programa ejecuta las tareas basándose en la fecha y hora o cuando se lleva a cabo un evento con ciertas características que se han especificado en la base de datos.

Mediante este módulo es posible especificar el día o días en que se quiera llevar a cabo la tarea, su duración y la frecuencia a la que ejecutará la tarea listada.

2.2.2.7 DOS WINDOW

Este es un programa pequeño que permite salir de FIX DMACS temporalmente para ejecutar programas propios, del sistema operativo y otros programas.

2.2.2.8 SYSTEM CONFIG UTILITY

Este programa permite configurar y declarar a los programas que deben aparecer en la pantalla principal de FIX DMACS (*MASTER MENU*). Mediante éste es posible cargar un programa ejecutable propio de FIX DMACS y definir sus características de ejecución, tales como el directorio donde se encuentra el programa a ser ejecutado, la leyenda que permitirá seleccionarlo, etc.

Existen otros programas integrantes de FIX DMACS, sin embargo no es posible proporcionar más información sobre ellos porque no se cuenta con los manuales de usuario correspondientes.

Resumiendo, FIX DMACS es un programa computacional multitareas que proporciona un conjunto completo de *software* para: monitoreo, adquisición de datos y control de procesos (industriales, químicos, etc.). Los programas integrantes se clasifican en **programas de bajo nivel y alto nivel**.⁸ Los programas de *bajo nivel* sólo se ejecutan si el usuario así lo quiere; los de *alto nivel* son inicializados cada vez que se inicia la ejecución del programa principal. Son programas de *alto nivel*: el **SHELL, SAC y EMS**. Son programas de *bajo nivel*: **DATABASE, SCU, REPORT, GRAPHICS**, etc. Para realizar el análisis que se explica en el capítulo siguiente era necesario explicar con un poco de detalle cada módulo de FIX DMACS. Después de esta descripción es posible decir que los módulos que más nos interesan son: **DATABASE, REPORT e HISTORICAL TREND**.

CAPITULO 3

Análisis del programa FIX DMACS para la generación de reportes

Originalmente existían algunos problemas básicos que debían ser resueltos antes de iniciar con el diseño y la programación del módulo generador de reportes; por lo cual nos apoyamos en las siguientes preguntas para solucionarlos con mayor facilidad:

- 1.- Cómo almacena FIX DMACS los datos y cómo es posible acceder a ellos.
- 2.- Cuáles son los requerimientos que deberá satisfacer el generador de reportes.
- 3.- Es posible realizar la interfaz usuario-computadora mediante el *software* propio de FIX DMACS o es necesario utilizar otro tipo de *software* que reúna las características que satisfagan los requerimientos especificados.

3.1 ALMACENAMIENTO DE DATOS

De acuerdo con el párrafo descriptivo correspondiente a la figura 2.3, se sabe que FIX DMACS manipula los datos que se están adquiriendo mediante la *tabla manejadora de imagen* que los recibe y respalda en una base de datos formada por cadenas de bloques que procesan a su vez sólo los datos especificados por el usuario y retorna los datos ya procesados a la *tabla manejadora de imagen*.

Se entiende como *tabla manejadora de imagen* a un conjunto de direcciones de memoria reservadas en la computadora, donde es posible almacenar temporalmente los datos (procesados o no por la base de datos). Las posiciones de memoria reservada que actúan como *tabla manejadora de imagen* no se encuentran especificadas en los manuales correspondientes al conjunto de programas de FIX DMACS, por lo que no es posible conocer la posición exacta de esas direcciones de memoria y, en consecuencia, acceder los datos directamente.

Ahora bien, se sabe que los datos una vez procesados se almacenan en forma automática en *archivos históricos* con intervalos exactos de tiempo (ocho horas). Este tiempo es especificado por el usuario en el *Historical Trend Collection*. El nombre de cada archivo está formado por la fecha y el intervalo de tiempo correspondiente y es codificado de la siguiente forma:

AAMDDHH.H##

donde **AA** es el año, **MM** el mes, **DD** el día y **HH** la hora de inicio en que el archivo fue abierto. La extensión **.H##** indica la longitud del archivo en horas. Por ejemplo, un archivo con longitud de ocho horas, abierto a las dieciséis horas del 4 de octubre de 1994 tendrá el siguiente nombre:

94100416.H08

y será almacenado en el subdirectorio C:\DMACS\HTRDATA. Todos los datos anteriormente mencionados están definidos dentro de FIX DMACS y no es posible cambiarlos (excepto, la longitud del archivo en horas).

Estos archivos históricos de datos tienen un formato especial (formato binario), lo que complica el acceso directo mediante cualquier editor común, por lo que se buscó otra forma de acceso.

El usuario que hasta el momento trabaja con FIX DMACS encontró la forma de guardar manualmente los datos requeridos en archivos con formato ASCII. El proceso a realizar por el usuario es el siguiente e involucra la generación de una gráfica antes de archivar los datos que en ella se muestran.

Mediante el acceso al programa *Historical Trend Display* es posible generar en pantalla la gráfica de datos históricos que se mencionó anteriormente. Para la generación de gráficas es necesario llenar el formato que se muestra a continuación.

HTD SELECTION MENU						
GROUP						
GROUP DESCRIPTION						
START TIME 10.00.00	PREVIOUS 1 HOUR	DISPLAY MODE 1				
START DATE 04.30.92	PREVIOUS 2 HOUR	SAVE DEF.UT 1				
DURATION 00.02.00.00	PREVIOUS 3 HOUR					
COLOR	MODE	TAG	FIELD	MODE	LOW END	HIGH END
WHITE	F.CV		SAMPLE		DEFAULT	DEFAULT
RED	F.CV		SAMPLE		DEFAULT	DEFAULT
BLUE	F.CV		SAMPLE		DEFAULT	DEFAULT
GREEN	F.CV		SAMPLE		DEFAULT	DEFAULT
YELLOW	F.CV		SAMPLE		DEFAULT	DEFAULT
CYAN	F.CV		SAMPLE		DEFAULT	DEFAULT
MAGENTA	F.CV		SAMPLE		DEFAULT	DEFAULT
BROWN	F.CV		SAMPLE		DEFAULT	DEFAULT
PATH C:\DMACSH\PLATA						
F1 - HELP F5 - DISPLAY F10 - EXIT						

Figura 3.1 Menú de selección para el programa *Historical Trend Display*

Como se observa, este formato permite dar un nombre y una descripción al grupo de datos que serán seleccionados una vez que se cumpla con este requisito. También proporciona la opción de cambiar la fecha y hora de inicio de la gráfica, así como el intervalo de muestreo de la misma. La fecha y hora *por default* que aparecen en el formato corresponden al día y hora en que se está ejecutando la opción. Permite también darle un color diferente a cada grupo de datos a graficar. Pulsando la tecla [F5] se despliega una gráfica similar a la siguiente:

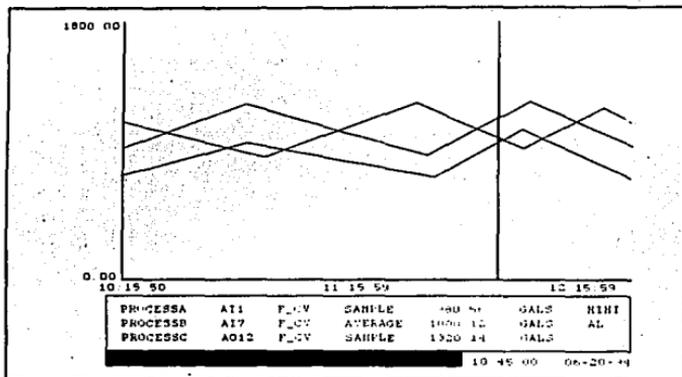


Figura 3.2 Gráfica producida por el programa *Historical Trend Display*

Una vez que se ha generado la gráfica, los valores que se muestran pueden ser almacenados en archivos ASCII presionando las teclas <Alt> y <A> simultáneamente.

Después de esta acción aparece el siguiente mensaje.

Enter filename:

que permite dar un nombre al archivo que guarda los datos de la gráfica.

Los datos guardados en este archivo tienen el siguiente formato:

FECHA A	FECHA B	FECHA C	FECHA D					
NODO A	NODO B	NODO C	NODO D					
BLOQUE A	BLOQUE B	BLOQUE C	BLOQUE D					
CAMPO A	CAMPO B	CAMPO C	CAMPO D					
MODO A	MODO B	MODO C	MODO D					
HORA COLECCION	VALOR1	ESTADO1	VALOR2	ESTADO2	VALOR3	ESTADO3	VALOR4	ESTADO4
HORA COLECCION	VALOR1	ESTADO1	VALOR2	ESTADO2	VALOR3	ESTADO3	VALOR4	ESTADO4
HORA COLECCION	VALOR1	ESTADO1	VALOR2	ESTADO2	VALOR3	ESTADO3	VALOR4	ESTADO4
HORA COLECCION	VALOR1	ESTADO1	VALOR2	ESTADO2	VALOR3	ESTADO3	VALOR4	ESTADO4

Figura 3.3 Formato del archivo ASCII que se genera mediante el programa *Historical Trend Display*

Este formato contiene los campos:

Fecha.- Es la fecha en que se colectaron los datos.

Nodo.- Es el nodo de FIX DMACS donde se están colectando los datos.

Bloque.- Es el nombre del bloque de donde se están accedendo los datos.

Campo.- Es el tipo de campo con que se representan los datos obtenidos (enteros, punto flotante, etc.).

Modo.- Es el modo de muestreo que se utiliza para desplegar los datos.

Hora Colección.- Es la hora en que fueron colectados los datos de los bloques.

Valor.- Es el valor numérico de esa variable en el tiempo que se especifica al principio del renglón.

Estado.- Indica si el valor numérico obtenido es válido o no.

Para obtener valores de muestreo cada 5 minutos era necesario especificar un tiempo de duración de la gráfica de 22:30 horas ya que este intervalo de tiempo es dividido entre 270 que es el número total de datos que el *Historical Trend Display* muestra en cada gráfica. Como se puede observar, no es posible obtener de esta forma los 288 datos que se adquieren en un día completo.

Es importante mencionar que desde que este *software* empezó a ser utilizado para mostrar en pantalla el comportamiento general de los módulos fotovoltaicos en cuestión, se utiliza el procedimiento anteriormente descrito para almacenar los datos que se están midiendo de ellos; lo que implica un desperdicio de tiempo considerable si se compara con el tiempo que se llevaría realizar el mismo procedimiento en forma automática.

El obtener los archivos de datos en forma manual representaba una limitante muy importante para realizar el módulo generador de reportes; ya que dichos archivos debían ser creados automáticamente en el ambiente del programa FIX DMACS y dentro de la interfaz de usuario-computadora que se realizara.

Dados los antecedentes anteriores era preciso encontrar una forma de automatizar la generación de archivos de datos con FIX DMACS y en un formato accesible.

Como se describió en el capítulo anterior, FIX DMACS cuenta con un programa denominado *Report Creator*, que permite crear formatos para obtener los valores que se respaldan en los archivos en binario que crea FIX DMACS para guardar los valores que recibe del monitoreo de los sistemas fotovoltaicos.

Utilizando la función EDIT del menú correspondiente a este programa, es posible crear un formato que contenga los nombres de las variables que guardan los datos que el usuario necesita.

Una vez realizadas las especificaciones correspondientes (tales como nombres de las variables que contienen los datos, dimensión del intervalo de tiempo en que se deben reportar los datos, nodo del que se van a adquirir, longitud máxima del campo a ser ocupado por la variable y otros (Ver Apéndice A), el formato es archivado de acuerdo al nombre que el usuario especifica y con la extensión .DEF. Este formato es una especie de esqueleto, el cual es llenado cuando se ejecuta el programa que realiza la acción de llenado.

Las dimensiones de cada página del formato son: 78 columnas por 22 renglones. Hay que tener cuidado con estas especificaciones para evitar un desbordamiento en el formato.

Para elaborar este formato existen dos condiciones: las variables que se especifiquen dentro del mismo deben estar previamente declaradas en la base de datos en uso y también en el *Historical Trend Assignment*.

En el manual de FIX DMACS se describe una forma de generar automáticamente estos reportes, esto es mediante la utilización del programa *Scheduler* que permite compilar y ejecutar este formato en el tiempo y en la fecha que se defina en el mismo. Sin embargo, el programa *Scheduler* por lo que se encuentra incluido dentro del paquete con el que se está trabajando, no lo que no es posible realizar la ejecución automática mediante esta opción.

Existe otra forma de ejecutar un reporte. En la pantalla principal de FIX DMACS hay una opción que permite salir temporalmente del menú principal para ejecutar otras funciones ajenas al ambiente del programa. Esta opción es *DOS WINDOW*. Una vez fuera del ambiente FIX DMACS se utiliza el programa de *Report Generator (GEN.EXE)* desde el indicador de sistema operativo para llenar el formato del reporte que se ha elaborado mediante el programa *Report Creator* (dentro del ambiente FIX DMACS). Los datos que se obtienen corresponden a los monitoreados en la fecha que se especifique de la siguiente forma:

C:\DMACS\GEN /R[Nombre del formato de reporte] /D[Fecha] /F

donde C:\ corresponde a la unidad de disco duro de la computadora, **DMACS** es el nombre del directorio en que se encuentran grabados los archivos ejecutables propios de FIX DMACS y **GEN** es el programa generador de reportes (*GEN.EXE*) que se ejecuta con los argumentos: **/R** que corresponde al nombre del formato de reportes que se va a llenar, **/D** que es la fecha en que fueron monitoreados los datos y **/F** es la especificación que omite todo carácter diferente a los números que describen a los valores de las variables.

Esta opción se consideró como válida porque de esta forma es posible acceder los datos históricos requeridos en forma automática y con un formato entendible, mediante un programa adicional (ya sea elaborado en FIX DMACS o en algún lenguaje de programación) que permitiera ejecutar el comando descrito anteriormente y dentro del ambiente FIX DMACS. Además de que los datos podrían obtenerse de un día completo (24 horas).

Para considerar óptima esta opción era indispensable conocer las razones por las cuales es necesario realizar un módulo de reportes adicional. Lo que se especifica a continuación es una descripción de esas razones.

3.2 ANALISIS DE REQUERIMIENTOS

La especificación de requerimientos juega un papel muy importante para el desarrollo de cualquier programa de *software*. Es de esta especificación de quien depende que un programa de *software* tenga éxito o fracase. Roger S. Pressman [PRE93] en su libro sobre Ingeniería del *software* expresa al respecto:

Para que un esfuerzo de desarrollo de *software* tenga éxito, es esencial comprender perfectamente los requisitos del *software*. Independientemente de lo bien diseñado o codificado que esté el programa, si se ha analizado y especificado pobremente, decepcionará al usuario y desprestigiará al que lo ha desarrollado...

Los siguientes puntos definen los requerimientos básicos del generador de reportes que se va a desarrollar.

3.2.1 PROPOSITO GENERAL DEL GENERADOR DE REPORTES

Este deberá permitir el acceso automático a los archivos de datos históricos generados mediante el programa FIX DMACS, además deberá ser capaz de manipular y procesar dichos datos de tal forma que se satisfagan los requerimientos que se especifican a continuación.

El generador de reportes deberá contar con cuatro opciones básicas: tipos de *reportes diarios, semanales, mensuales y anuales*. De cada opción se derivarán otras que permitirán, de acuerdo a la fecha en que sea necesario el reporte, realizar tareas tales como graficación de diversos datos así como generación de tablas que reflejen en sus resultados la manipulación de los datos accedidos por FIX DMACS, sin la intervención del usuario.

3.2.2 REPORTES DIARIOS

Se pretende que este tipo de reporte contemple cuatro opciones: *gráficas, tabla resumen diaria, tabla de fallas y listado de datos*.

Es necesaria la elaboración de *gráficas* que permitan de alguna manera predecir tanto el comportamiento futuro como el tiempo de vida útil de cada sistema una vez instalado en la zona rural a la que es designado, mediante la observación completa del comportamiento diario del mismo. Es recomendable la graficación de los valores que se obtienen a lo largo de día con la finalidad de comparar la gráfica resultante con las gráficas obtenidas del mismo sistema en otros días y en otros niveles de insolación. Los valores van a obtenerse en intervalos de 5 minutos para detallar con mayor exactitud el comportamiento de cada sistema.

Hasta el momento, este tipo de gráficas se realiza utilizando los datos obtenidos manualmente por el operador de FIX DMACS, mediante un laborioso proceso que implica los siguientes pasos:

- Generar los datos que el usuario necesita por cada sistema en forma manual desde el *Historical Trend Display* de FIX DMACS.
- Concatenar los archivos obtenidos y después separarlos por día. Cada archivo sólo contiene la información de veintidos horas con treinta minutos de un día y el resto se encuentra contenido en otro archivo. Hasta ese momento se había encontrado que sólo utilizando ese intervalo de tiempo era posible obtener los datos cada 5 minutos.
- Después de separarlos por día, los archivos se accesan mediante un paquete de *software* diferente a FIX DMACS que permite graficar e imprimir tanto los valores como las gráficas obtenidos de este proceso.

Un ejemplo de la gráfica diaria resultante se muestra en la figura 3.4.

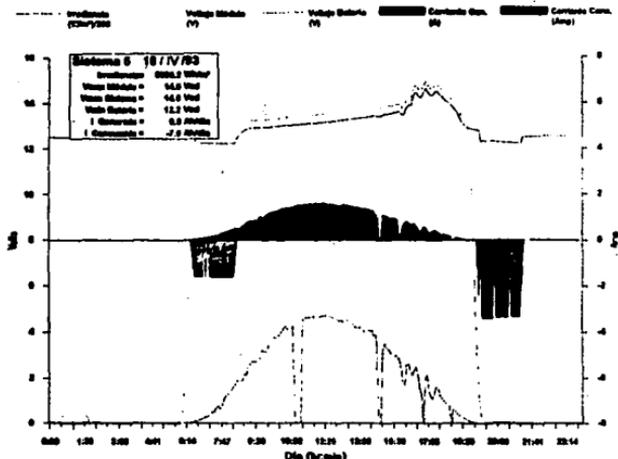


Figura 3.4 Gráfica elaborada con los datos obtenidos manualmente en FIX DMACS, mediante Excel

Además de graficar los valores obtenidos contra el tiempo durante un día, es preciso conocer otros datos que nos permitan evaluar el comportamiento de cada sistema fotovoltaico; por lo que se hace necesaria la elaboración de una *tabla resumen* que contenga el resultado de determinados cálculos realizados con base en los datos obtenidos. Tales datos son el nivel de insolación promedio al que se encuentra expuesto el sistema durante un día determinado, así como:

- Voltaje máximo generado por el módulo.
- Voltaje máximo del sistema (voltaje máximo en la batería).
- Voltaje mínimo en la batería.
- Corriente generada por el módulo.
- Corriente consumida por la carga.
- Corriente consumida por el controlador de carga.

La cantidad promedio diaria de cada una de las corrientes (excepto para la corriente consumida por el controlador) así como la insolación promedio (o irradiancia) se calculan por la sumatoria de 288 valores dividida entre 12. Los 288 valores se obtienen al muestrear cada variable en intervalos regulares de 5 minutos durante un día completo; el divisor 12 representa al número de muestras por hora. Esto es:

$$\frac{1}{12} \sum_{i=1}^{i=288} [VALOR_{VARIABLE}]$$

Para el cálculo de la corriente consumida por el controlador de carga es necesario realizar la siguiente operación por cada intervalo de 5 minutos.

$$I_{\text{controlador}} = I_{\text{pánal}} + I_{\text{carga}} - I_{\text{batería}}$$

Una vez obtenidos los 288 valores se realiza una sumatoria de ellos y se divide entre 12.

Las unidades de dimensionamiento dependen de la variable que se promedie. Así, por ejemplo, la cantidad de corriente generada por el módulo se dimensiona en amperes-hora/día.

Como podemos observar en la figura 3.4, esta tabla resumen se incluye dentro de la gráfica.

La *tabla de fallos* no interviene directamente en la predicción del comportamiento futuro de un módulo. Es necesaria, para saber en qué momento se produce una falla en la adquisición y almacenamiento de los datos que se están monitoreando. Se dice que no interviene directamente en la predicción del comportamiento futuro del módulo porque la falla producida puede que no se derive del módulo fotovoltaico en sí, sino que se deba a la falla de otros elementos relacionados con éste tales como la computadora que está adquiriendo los datos, el sistema de entrada-salida a la computadora, la energía eléctrica que alimenta la computadora personal, etc.

El *listado de datos* es necesario para corroborar que los resultados obtenidos en las tres opciones anteriores correspondan a los valores que se muestran en el listado.

3.2.3 REPORTES SEMANALES Y REPORTES MENSUALES

Se pretende que estos tipos de reportes contemplen las opciones *tabla resumen* y *tabla de fallas*. Al tipo de reportes mensual se le deberá agregar una opción adicional: *archivo mensual*.

La *tabla resumen* deberá contener exactamente los mismos datos que la tabla resumen diaria. Sólo que los datos obtenidos en estas tablas serán los valores máximos, excepto para el voltaje mínimo en batería del cual se obtendrá el valor más bajo registrado en una semana y en un mes, respectivamente.

Al igual que en los reportes diarios, la *tabla de fallas* deberá contener las fallas que ocurran en una semana y en un mes, respectivamente; así como el porcentaje de datos perdidos cada semana y cada mes.

La fecha para cada reporte semanal y mensual será seleccionada por el usuario.

El *archivo mensual* deberá contener los siguientes datos:

- Temperatura ambiente
- Temperatura en el módulo
- Temperatura en la batería
- Corriente generada
- Corriente consumida
- Voltaje en batería
- Radiación solar

Este archivo sólo se generará eventualmente, ya que algunos de los datos que se incluyen sólo son necesarios para observar en qué condiciones ambientales y de temperatura operan los sistemas fotovoltaicos.

3.2.4 REPORTES ANUALES

Los requerimientos para el formato de reportes anual no se han especificado hasta el momento. Sin embargo, esta opción deberá ser incluida dentro del módulo de reportes aunque no sea implementada.

Otro punto importante para la elaboración de la interfaz correspondiente al módulo de reportes es que al seleccionar una de las opciones, además de ser desplegada en pantalla, se genere a la vez un archivo con nombre predeterminado por el programador. Además de desplegar en pantalla el archivo generado debe ser posible enviarlo a impresión, sin necesidad de utilizar alguna otra herramienta de *software* diferente a la que se está empleando.

Los archivos de salida que se generen, deberán tener un encabezado (excepto el archivo mensual), que identifique el sistema fotovoltaico del que provienen, así como la fecha en que se produce y el tipo de reporte al que corresponden.

3.3 FIX DMACS COMO HERRAMIENTA PARA REALIZAR EL MÓDULO DE REPORTES

Una vez encontrada una forma de acceder los datos provenientes de FIX DMACS en forma automática y definidos los requerimientos de usuario, es necesario determinar si es posible realizar la programación del módulo de reportes únicamente utilizando los programas que constituyen a este *software*. Si no es posible, hay que determinar qué otro *software* nos sería de utilidad.

FIX DMACS proporciona la opción de realizar interfaces de usuario mediante un programa llamado *Pixel Display Builder*. Este programa proporciona una amplia gama de opciones: tiene la flexibilidad de un editor gráfico (con un editor gráfico es posible generar líneas y figuras geométricas de diferentes colores y tamaños, además de adicionar texto al dibujo realizado), permite enlazar la interfaz dibujada previamente con diferentes tipos de datos como los que se están adquiriendo en tiempo real, así como desplegar la fecha y hora actuales dentro de la interfaz misma. Además, también es posible realizar enlaces que permitan mostrar otras pantallas o ejecutar algunos otros programas integrados en el ambiente de FIX DMACS.

Como se especificó anteriormente, los reportes deberían generarse de acuerdo a una fecha. He aquí la primera limitante. Una vez creada una interfaz, no existe una opción diferente a las especificadas en los manuales, que permita leer una cadena de caracteres desde teclado y, mucho menos, enlazar esa cadena de caracteres con otra para completar una orden que pudiera ejecutarse dentro de la interfaz que se ha creado. Dicha orden correspondería a la especificada anteriormente para llenar un formato de reportes, fuera del

ambiente FIX DMACS.

Otro factor importante en este análisis es la necesidad de crear la gráfica de comportamiento de cada sistema fotovoltaico con los datos generados cada 24 horas y con las especificaciones ya descritas. La única forma de crear una gráfica similar es mediante el programa *Historical Trend Display*, que nos permite desplegar en pantalla los valores que se han declarado previamente como históricos en el programa *Historical Trend Assignment*, además de ser declarados para graficarse en el mismo *Historical Trend Display*. La gráfica resultante sólo muestra los valores que se han declarado como se explicó anteriormente, además de la fecha y la hora en que es creada. No es posible realizar operaciones con los datos que allí se grafican y tampoco es posible desplegar el resultado de las operaciones en la gráfica que se encuentra en pantalla.

3.4 LENGUAJE C COMO OPCION AUXILIAR PARA REALIZAR EL MODULO DE REPORTES

Dadas las limitaciones anteriores, era necesario determinar una herramienta alternativa de *software* (diferente a FIX DMACS) que nos permitiera cumplir con los requerimientos especificados.

En el manual de usuario de FIX DMACS (en la parte correspondiente al programa *Scheduler* del capítulo anterior) se menciona que es posible la ejecución de otros programas dentro del ambiente FIX DMACS, cuyo código fuente estuviera escrito y compilado en lenguaje C. Sin embargo, la realización de estas aplicaciones dependerá de las especificaciones de otro módulo de FIX DMACS, denominado *C Database Access program*. El *software* y la documentación correspondientes a éste último no se encuentran disponibles, por lo que no fue posible determinar qué características podrían o deberían tener los programas escritos en lenguaje C. En algunas partes de los manuales de FIX DMACS sólo se especifica que los programas elaborados en lenguaje C no deberían utilizar interrupciones.

Pasando por alto esta limitante, se pensó en realizar un programa en lenguaje C a manera de prueba, que nos permitiera determinar hasta qué grado se afectaba el funcionamiento general de FIX DMACS.

Al realizar y compilar un programa en lenguaje C, se genera un archivo ejecutable que podría ser definido como un módulo adicional dentro del ambiente FIX DMACS. Para poder ejecutar un programa dentro de este ambiente es necesario declararlo dentro de otro módulo de FIX DMACS denominado *System Config Utility (SCU)*. En este módulo se especifica el directorio donde se encuentra el programa ejecutable correspondiente al módulo que se va a declarar, así como el nombre del mismo y el identificador que permitirá seleccionarlo para ser ejecutado dentro del menú principal. Por ejemplo: el archivo ejecutable del módulo *Report Creator* se encuentra en el directorio de C:\DMACS y responde al nombre de *REP.EXE*. Como se habrá visto en la figura correspondiente al menú principal de FIX DMACS del capítulo anterior, *Report Creator* tiene por identificador a *REP - REPORT CREATOR*. Todas estas características fueron definidas en el *SCU*.

3.5 ELECCION DE LA ALTERNATIVA MAS VIABLE

Para definir con claridad la alternativa más viable era necesario realizar la prueba que se especificó en el punto anterior.

Esta prueba consistió de los siguientes pasos:

- Elaboración de un formato de datos que permitiera acceder los valores que se monitorean del sistema fotovoltaico 1. Dichos valores son:
 - Voltaje en el módulo
 - Corriente en el módulo
 - Voltaje en la batería
 - Corriente en la batería
 - Corriente de carga
 - Radiación solar
- Elaboración del programa en lenguaje C que permitiera llenar el formato de datos anterior y elaborar una gráfica con esos datos.
- Declaración de este programa en el *SCU* como módulo *REPORTES*.
- Ejecución del módulo *REPORTES*.

Los resultados de la prueba fueron aceptables, ya que no se produjo ningún problema que alterara el funcionamiento del programa FIX DMACS. Habría que determinar si se podría producir alguna interrupción en la adquisición de datos en tiempo real cuando la longitud del programa ejecutable se incrementara, interrupción que podría derivarse del manejo de memoria.

Hasta el momento se ha hablado del lenguaje C, como una herramienta para realizar el módulo generador de reportes, pero es necesario exponer cuáles son los fundamentos que nos llevan a proponer este lenguaje.

La primera implementación del lenguaje C la realizó Dennis Ritchie en un computador DEC PDP-11 utilizando el sistema operativo UNIX. El lenguaje C es el resultado del proceso de desarrollo que empezó con un lenguaje llamado BCPL que influyó en un lenguaje llamado B, que inventó Ken Thompson y condujo al desarrollo de C. Durante muchos años, el estándar para C fue el proporcionado con el sistema operativo UNIX. Con el incremento de la popularidad de las microcomputadoras se crearon muchas implementaciones de C. La mayoría de ellas eran muy compatibles con las otras a nivel de código fuente. Sin embargo, al no existir un estándar, existían discrepancias.

Para corregir esta situación se creó un comité en 1983 que empezó el trabajo para la creación de un estándar, que finalmente definía el lenguaje C.

A continuación se da una breve descripción de algunas características del lenguaje C.

- C es un *lenguaje de alto nivel*. Los lenguajes de alto nivel permiten al programador y al programa independizarse de la máquina.
- Normalmente se considera a C como un lenguaje estructurado. Aunque el término de lenguaje estructurado por bloques no se aplica estrictamente a C en el sentido académico, C es informalmente parte de ese grupo de lenguajes. Un *lenguaje estructurado* es aquel que puede seccionar y esconder del resto del programa toda la información e instrucciones que realizan una tarea específica.

- **C es un lenguaje de programador.** Al contrario de la mayoría de los lenguajes de alto nivel, C impone pocas restricciones a lo que puede hacerse con él. Un programador puede crear y mantener una biblioteca única de funciones que se ha personalizado para su uso.

- **C permite compilación separada,** lo que hace que los grandes proyectos sean fáciles de mantener y de manejar.

De todas las consideraciones anteriormente expuestas, podemos deducir que si utilizamos únicamente los módulos de FIX DMACS no es posible realizar una interfaz de usuario que nos permita manipular los datos históricos de tal forma que se satisfagan las necesidades del usuario. Sin embargo, nos podemos auxiliar del programa *Report Creator* para crear un formato, con los datos que se necesitan, que nos permita acceder, en forma automática, los datos históricos de los archivos con formato binario que crea FIX DMACS. Como ya se dijo, dicho formato puede ser llenado con los datos de una fecha determinada si se utiliza el programa ejecutable de *Report Generator* (*GEN.EXE*) con todos sus parámetros y se ejecuta desde el indicador (*prompt*) del sistema operativo mediante otro programa. Este otro programa, podría ser realizado en un lenguaje de programación como C que es un lenguaje estructurado que nos permite realizar una programación modular, característica importante en el sistema que se va a elaborar ya que es necesario realizar varias tareas completas por separado, pero que estén ligadas mediante una interfaz de usuario. La ejecución de dicho programa debe permitirnos llenar los formatos de datos que se definan, manipular dichos datos y generar archivos que cumplan con los requerimientos definidos al principio del capítulo, sin afectar la ejecución del conjunto de programas de FIX DMACS. Una vez realizada la programación de esta herramienta de computadora, se compilaría y se generaría un archivo ejecutable que podría ser declarado en el *SCU* y ejecutado dentro del ambiente FIX DMACS.

Hasta el momento, la propuesta se considera aceptable. Es preciso realizar pruebas que nos lleven a aprobar en forma definitiva esta decisión, lo cual se definirá más adelante.

REFERENCIAS

[PRE93] PRESSMAN, S. ROGER. Ingeniería del software: un enfoque práctico. (México, McGraw Hill, tercera edición, 1993). p.181.

CAPITULO 4

Desarrollo del software para utilizar FIX DMACS en la generación de reportes

4.1 GENERALIDADES

Hasta el momento, en los capítulos anteriores se han mencionado términos como ***software*** e ***interfaz de usuario***, pero no se han definido con claridad, por lo que se da una explicación breve de cada uno de ellos; ya que son factores clave en el desarrollo de este capítulo.

4.1.1 DEFINICION DE SOFTWARE

Es un conjunto de instrucciones (programas de computadora) que cuando se ejecutan proporcionan la función y el comportamiento deseado. Las instrucciones del ***software*** se construyen mediante un lenguaje de programación que tiene un vocabulario limitado, una gramática definida explícitamente y reglas bien formadas de sintaxis y semántica. Estos atributos son esenciales para la traducción de dichas instrucciones por la computadora.

Las clases de lenguajes que se utilizan actualmente son los *lenguajes máquina*, los *lenguajes de alto nivel* y los *lenguajes no procedimentales*.

Los *lenguajes máquina* son una representación simbólica del conjunto de instrucciones de la UCP (Unidad Central de Proceso) de una computadora que permite realizar programas mantenibles y bien documentados si y sólo si el programa está bien diseñado y tiene poca documentación.

Los *lenguajes de alto nivel* permiten al programador y al programa independizarse de la máquina. Cuando se utiliza un traductor sofisticado, el vocabulario, la gramática, la sintaxis y la semántica de un lenguaje de alto nivel pueden ser más sofisticados que los lenguajes máquina. De hecho los compiladores e intérpretes de los lenguajes de alto nivel producen lenguaje máquina como salida. Lenguajes de programación modernos tales como C, Pascal o ADA se utilizan ampliamente, dado que tienen un gran soporte de programación por procedimientos y estructuración de datos.

En la década pasada, apareció un grupo de *lenguajes no procedimentales* o de cuarta generación. En este tipo de lenguajes el programador no especifica los detalles procedimentales. Un programa en un lenguaje procedimental es "la especificación del resultado deseado, en vez de la especificación requerida para conseguir el resultado". En la actualidad, estos lenguajes se han utilizado en aplicaciones de bases de datos y en otras áreas de procesamiento de datos para negocios.

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales (es decir, un algoritmo).

4.1.2 QUE ES UNA INTERFAZ DE USUARIO

La *interfaz de usuario* es el mecanismo a través del cual se establece un diálogo entre el programa y el usuario.

Puede decirse que una buena parte del éxito de un programa de software se debe a la interfaz de usuario que lo "envuelve". Si la interfaz de usuario es fácil de aprender, simple de utilizar, directa y no muy estricta, el usuario podrá hacer buen uso de lo que hay dentro. Si, por el contrario, no tiene ninguna de esas características, habrá problemas invariablemente. Para

evitar estos problemas hay que contemplar varios aspectos importantes para realizar una buena interfaz de usuario. Tales aspectos son:

- **Consistencia.** Se debe utilizar un formato consistente para la selección de menús, la entrada de órdenes, la visualización de datos y otras funciones que necesite la interfaz de usuario.
- **Ofrecimiento de una realimentación significativa.** Se debe proporcionar al usuario una realimentación visual y auditiva para asegurar que se establece una comunicación bidireccional.
- **Reducir la cantidad de información que debe ser memorizada entre acciones.** No se debe esperar que el usuario recuerde una lista de números y nombres para que pueda reutilizarlos de entrada a la siguiente función.
- **Perdonar errores.** El sistema debe protegerse de los errores del usuario que pudieran afectarle causándole una falla.
- **Utilizar ventanas (si están disponibles) para modularizar los diferentes tipos de información.** Las ventanas mantienen accesibles al usuario muchos tipos diferentes de información.
- **Buscar la eficiencia en el diálogo, el movimiento y el pensamiento.** El número de pulsaciones debe ser minimizado, la distancia que un *ratón* debe recorrer entre dos pulsaciones también debe ser tenida en cuenta al diseñar el formato de presentación, y el usuario debe encontrarse pocas veces una situación en la que se tenga que preguntar, "¿Qué significa esto?".

Hasta aquí hemos descrito dos aspectos que son parte fundamental en el desarrollo del presente trabajo. El generador de reportes que se va a realizar es en sí un *software* que va a interactuar con otro programa de computadora para generar los reportes necesarios que permitirán evaluar el comportamiento de pequeños sistemas fotovoltaicos. Esta interacción humano-computadora se va a realizar mediante una interfaz de usuario, interfaz que deberá contemplar en su estructura, si no todas, al menos una buena parte de las características que se han descrito anteriormente.

4.2 ESPECIFICACIONES DE DISEÑO PARA EL MODULO **REPORTES**

4.2.1 CONSIDERACIONES INICIALES

Para programar el módulo generador de reportes se utilizarán:

- El módulo *REPORT* de FIX DMACS con sus dos programas. El programa de *Report Creator (GEN.EXE)* se utilizará para generar los formatos necesarios que accesarán los datos provenientes de la base de datos *FOT* que permite la manipulación de los valores provenientes del monitoreo de los sistemas fotovoltaicos. El programa *Report Generator* se utilizará para llenar los formatos anteriores mediante un programa de enlace.

- Para realizar el programa de enlace, así como todas las rutinas de graficación, acceso y manejo de los datos, cálculos, despliegue de información y otros se utilizará un compilador de lenguaje C. Las rutinas para desarrollar la interfaz de usuario correspondiente se elaborarán también en lenguaje C, elección que se hizo por las facilidades que nos proporciona este lenguaje (ver capítulo 3).

La base de datos *FOT* está configurada para recibir y procesar datos de diez sistemas fotovoltaicos. En concordancia con esta configuración se determinó que el programa *REPORTES* contemplará diez sistemas fotovoltaicos en su estructura, de los cuales será posible generar todos los tipos de reportes que integren al generador.

El programa *REPORTES* deberá ser ejecutable dentro del ambiente de trabajo del programa FIX DMACS, de manera independiente y sin interrumpir el proceso de adquisición y almacenamiento de datos que realiza el mismo.

La mayor parte de las funciones del programa *REPORTES* se activarán mediante el *ratón* (del inglés *mouse*) de la computadora.

Se utilizará el teclado para las funciones que así lo requieran.

Cada vez que se realice la selección de una opción del generador de reportes, el identificador de la opción deberá cambiar de color para indicar al usuario que se ha seleccionado dicha opción.

4.2.2 DESCOMPOSICION MODULAR DEL PROGRAMA *REPORTES*

El programa *REPORTES* se dividió para su programación en nueve partes básicas:

- **Elaboración de los formatos de datos para cada sistema fotovoltaico (Módulo *FORMATOS FIX DMACS*).**
- **Generación de un archivo de datos que contenga el identificador de cada uno de los sistemas fotovoltaicos (Módulo *ALTA*).**
- **Desarrollo de las rutinas para generar reportes diarios (Módulo *DIARIOS*).**
- **Desarrollo de las rutinas para generar reportes semanales (Módulo *SEMANALES*).**
- **Desarrollo de las rutinas para generar reportes mensuales (Módulo *MENSUALES*).**
- **Desarrollo de las rutinas para generar reportes anuales (Módulo *ANUALES*).**
- **Desarrollo de las rutinas para impresión de reportes (Módulo *IMPRIMIR*).**
- **Desarrollo de las rutinas para archivar reportes (Módulo *ARCHIVAR*).**
- **Desarrollo de las rutinas para elaborar la interfaz de usuario y enlazar dicha interfaz con las demás partes en que se ha dividido el desarrollo del módulo de reportes (Módulo *INTERFAZ*).**

Esta división se muestra en la figura 4.1

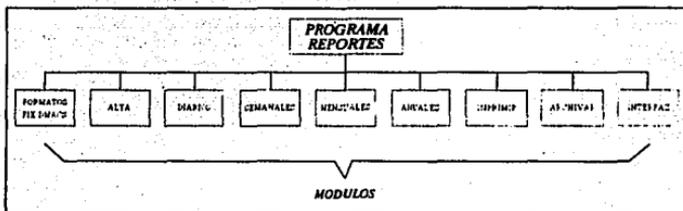


Figura 4.1 Diagrama modular del programa REPORTES

Como se especificó en los requerimientos, cada uno de los reportes deberá ser desplegado en pantalla (excepto el archivo mensual, por su tamaño), guardado en disco disco duro o flexible e imprimible, por lo que se elaboró el siguiente diagrama de flujo de información.

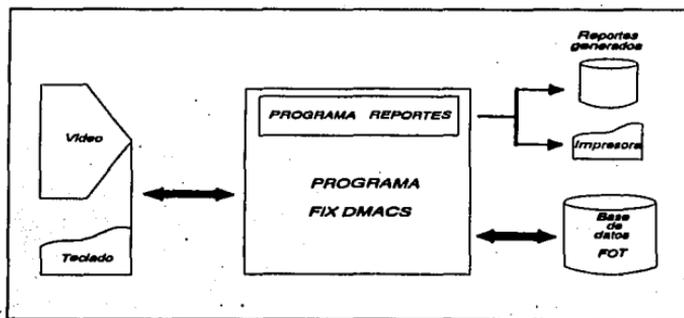


Figura 4.2 Diagrama de flujo de información para el programa REPORTES

4.2.2.1 MODULO FORMATOS FIX DMACS

Este módulo permitirá crear los formatos necesarios para acceder los archivos de datos históricos en los cuales FIX DMACS guarda los valores que se monitorean de los sistemas fotovoltaicos.

Estos formatos se realizarán mediante el programa *Report Creator* de FIX DMACS y permitirán acceder la base de datos *FOT*.

Se necesitan elaborar dos formatos de datos por cada uno de los diez sistemas fotovoltaicos: uno para obtener los datos que nos permitan generar reportes diarios y el otro para obtener los datos necesarios para crear un archivo mensual especial. Los grupos de datos para cada tipo de reporte son diferentes, por lo que es necesario elaborar dos formatos.

Los formatos para generar reportes diarios deberán contener los valores:

- Voltaje del módulo fotovoltaico
- Corriente del módulo fotovoltaico
- Voltaje de batería
- Corriente de carga
- Corriente de batería
- Radiación solar

Estos valores se obtienen de los bloques en la base de datos *FOT* cuyos identificadores respectivos son los siguientes:

- VFV [Número de sistema fotovoltaico]
- IFV [Número de sistema fotovoltaico]
- VBB [Número de sistema fotovoltaico]
- IC [Número de sistema fotovoltaico]
- IBB [Número de sistema fotovoltaico]
- IBB [Número de sistema fotovoltaico]
- RSP

Los formatos para generar el archivo mensual deberán contener los siguientes valores:

- Temperatura ambiente
- Temperatura del módulo fotovoltaico

- Temperatura de batería
- Corriente del módulo fotovoltaico
- Corriente de carga
- Voltaje de batería
- Radiación solar

Los identificadores respectivos en la base de datos *FOT* son los siguientes:

- TA
- TFV_[Número de sistema fotovoltaico]
- TBB_[Número de sistema fotovoltaico]
- IBB_[Número de sistema fotovoltaico]
- IFV_[Número de sistema fotovoltaico]
- IC_[Número de sistema fotovoltaico]
- VBB_[Número de sistema fotovoltaico]
- RSP

El acceso a los archivos de datos que FIX DMACS guarda automáticamente se hará mediante estos formatos y el programa *GEN.EXE*, mediante las siguientes líneas de órdenes:

GEN /RGRPAN[Número de sistema fotovoltaico] /D[Fecha] /F
GEN /RMENS[Número de sistema fotovoltaico] /D[Fecha] /F

donde *GEN* es el nombre del programa generador de reportes de FIX DMACS; */R*, */D* y */F* son los parámetros necesarios para enlazar los datos históricos con el formato de reportes (ver capítulo 3), *GRPAN[Número de sistema fotovoltaico]* y *MENS[Número de sistema fotovoltaico]* corresponden al nombre del formato para generar los reportes diarios y mensuales, respectivamente y *Fecha* esta formada por el mes, día y año en que fueron monitoreados y almacenados los datos que se especifican en el formato respectivo.

La fecha para estos formatos será leída desde teclado dentro de la interfaz de usuario del programa *REPORTES*. La línea de órdenes correspondiente será generada y ejecutada dentro del mismo programa.

4.2.2.2 MODULO ALTA

Como ya se ha mencionado anteriormente, el programa *REPORTES* funcionará dentro del ambiente de FIX DMACS. Sin embargo, la única interacción que tendrá con FIX DMACS será a través de su módulo para generar reportes.

En teoría, se sabe que los datos que se almacenan provienen del monitoreo de diez sistemas fotovoltaicos, pero en la práctica el monitoreo depende de las conexiones de los sensores al adquirente de datos que el operador realice.

Debido a que la única interacción que se tiene con FIX DMACS es a través de su generador de reportes y que no siempre se están adquiriendo datos para los diez sistemas fotovoltaicos, se determinó crear el archivo *ALTA.DAT* que contenga los números e identificadores de los sistemas fotovoltaicos de los cuales se están adquiriendo datos.

Las rutinas del módulo *ALTA*, permitirán al usuario dar de baja o de alta en el archivo *ALTA.DAT* a los sistemas fotovoltaicos de los cuales se estén adquiriendo y monitoreando datos.

La generación de todos los reportes dependerá del archivo *ALTA.DAT*, ya que cada vez que se pretenda generar un reporte para un determinado sistema fotovoltaico, se buscará su existencia en este archivo.

El proceso para dar de baja o alta un sistema fotovoltaico será el siguiente:

El usuario deberá seleccionar la opción para dar de alta o de baja un sistema fotovoltaico. Una vez realizada esta acción se desplegará una ventana que contenga los identificadores de los diez sistemas fotovoltaicos. El usuario seleccionará uno de esos identificadores. El programa buscará en el archivo *ALTA.DAT* el número del sistema. Si lo encuentra, lo dará de baja. Si no lo encuentra lo dará de alta en el mismo archivo y pedirá al usuario que introduzca el identificador para el sistema que dará de alta. Las rutinas del módulo *ALTA*, deberán permitir dar de baja o de alta a un mismo sistema las veces que el usuario desee.

4.2.2.2 MODULO ALTA

Como ya se ha mencionado anteriormente, el programa *REPORTES* funcionará dentro del ambiente de FIX DMACS. Sin embargo, la única interacción que tendrá con FIX DMACS será a través de su módulo para generar reportes.

En teoría, se sabe que los datos que se almacenan provienen del monitoreo de diez sistemas fotovoltaicos, pero en la práctica el monitoreo depende de las conexiones de los sensores al adquirente de datos que el operador realice.

Debido a que la única interacción que se tiene con FIX DMACS es a través de su generador de reportes y que no siempre se están adquiriendo datos para los diez sistemas fotovoltaicos, se determinó crear el archivo *ALTA.DAT* que contenga los números e identificadores de los sistemas fotovoltaicos de los cuales se están adquiriendo datos.

Las rutinas del módulo *ALTA*, permitirán al usuario dar de baja o de alta en el archivo *ALTA.DAT* a los sistemas fotovoltaicos de los cuales se estén adquiriendo y monitoreando datos.

La generación de todos los reportes dependerá del archivo *ALTA.DAT*, ya que cada vez que se pretenda generar un reporte para un determinado sistema fotovoltaico, se buscará su existencia en este archivo.

El proceso para dar de baja o alta un sistema fotovoltaico será el siguiente:

El usuario deberá seleccionar la opción para dar de alta o de baja un sistema fotovoltaico. Una vez realizada esta acción se desplegará una ventana que contenga los identificadores de los diez sistemas fotovoltaicos. El usuario seleccionará uno de esos identificadores. El programa buscará en el archivo *ALTA.DAT* el número del sistema. Si lo encuentra, lo dará de baja. Si no lo encuentra lo dará de alta en el mismo archivo y pedirá al usuario que introduzca el identificador para el sistema que dará de alta. Las rutinas del módulo *ALTA*, deberán permitir dar de baja o de alta a un mismo sistema las veces que el usuario desee.

Cada vez que el usuario seleccione un tipo de reporte (diarios, semanales o mensuales), el programa deberá verificar la existencia del archivo **ALTA.DAT**. Este archivo se creará en forma inicial con un editor de textos (ver Apéndice B) y se modificará con las rutinas de este módulo.

El módulo **ALTA** tendrá como *entradas*:

- El número del sistema fotovoltaico que se pretende dar de alta o de baja.
- El identificador del sistema fotovoltaico si se desea dar de alta dentro del programa **REPORTES**.

El módulo **ALTA** tendrá como *salida*:

- La orden para dar de baja o de alta el sistema fotovoltaico que seleccionó el usuario.

4.2.2.3 MODULO DIARIOS

Este módulo será capaz de generar cuatro tipos de reportes: gráficas, tabla resumen, tabla de fallas o pérdidas y listado de datos para cada sistema fotovoltaico.

Los reportes diarios se generaran en forma individual o colectiva; es decir, para un solo sistema o para todos los sistemas.

Para tener acceso a este módulo, el usuario deberá seleccionar la opción reportes **DIARIOS** de la interfaz de usuario que corresponde a este programa. Una vez seleccionada esta opción se activará una ventana que permita al usuario seleccionar el o los sistema(s) fotovoltaico(s) de los cuales necesite generar reportes diarios.

La generación de estos reportes dependerá del módulo **ALTA**. Dado que el programa **REPORTES** contemplará a diez sistemas fotovoltaicos en su estructura, éste permitirá al usuario seleccionar un solo sistema o todos los sistemas que se encuentren dados de alta en el módulo **ALTA**.

Un vez seleccionado el sistema fotovoltaico, el programa *REPORTES* deberá verificar si éste se encuentra dado de alta o no.

Si el sistema fotovoltaico no se encuentra dado de alta, entonces se desplegará un mensaje que indique al usuario que el sistema no se encuentra. Si esto sucede, el programa deberá permitir al usuario hacer una nueva selección.

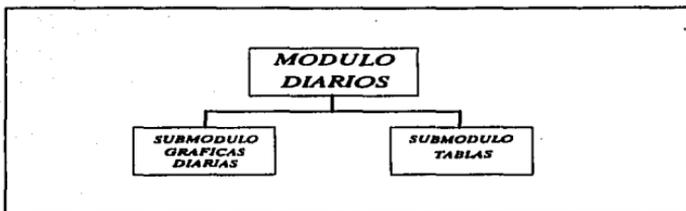
Si la selección de usuario es correcta, entonces se desplegará una ventana que contemple las cuatro opciones del generador de reportes diarios.

El usuario deberá seleccionar una de las cuatro opciones. Una vez seleccionado un reporte diario se desplegará una ventana para que el usuario introduzca la fecha en que necesita generar el reporte. Dependiendo de la selección hecha por el usuario, el programa enlazará uno de los dos submódulos que integran el módulo *DIARIOS*.

El módulo *DIARIOS* deberá proporcionar al usuario la facilidad de generar los cuatro tipos de reportes con sólo introducir una vez la fecha en que necesita el reporte, o con fechas distintas si así lo quiere.

Si el o los reporte(s) de salida no se generara(n) correctamente, se desplegará una ventana que así lo indique y el sistema regresará a la ventana para reportes diarios. Si la generación es correcta, entonces se activará una ventana que permita enlazar los módulos *ARCHIVAR* o *IMPRIMIR* que permitirán, como su nombre lo indica, archivar y/o imprimir el o los reporte(s) generado(s).

El módulo *DIARIOS* deberá ser dividido para su programación en dos submódulos, como se muestra en la siguiente figura:

Figura 4.3 Descomposición del módulo *DIARIOS*

La programación de los dos submódulos de *DIARIOS* se hará utilizando el editor y compilador de QuickC versión 2.5, que se auxiliará de las herramientas de Quinn Curtis. El programa *REPORTES* (editado y compilado con BorlandC versión 3.0) creará internamente la instrucción necesaria para enlazar y ejecutar los dos programas que integrarán el módulo *DIARIOS*.

A) Submódulo *GRAFICAS DIARIAS*

La gráfica a elaborar será de dos dimensiones. En el eje X, se graficará el tiempo en intervalos de 5 minutos y de las 00:00 a las 23:55 horas. En el eje Y se graficarán los valores que se generarán con el formato de datos que se creó con el generador de reportes de FIX DMACS. Cada grupo de datos a desplegar será de un color distinto. Esta gráfica deberá llevar un recuadro dentro, similar al de la figura 3.4. Dicho recuadro deberá contener los siguientes datos:

- Número del sistema fotovoltaico
- Identificador del sistema¹
- Fecha en que los datos fueron adquiridos
- Irradiancia
- Voltaje máximo del módulo
- Voltaje máximo del sistema

¹ Este identificador corresponde al nombre del lugar donde será instalado el sistema fotovoltaico a prueba.

- Voltaje mínimo en batería
- Corriente generada por el módulo
- Corriente consumida por la carga
- Corriente consumida por el controlador de carga

Los valores calculados tendrán cuatro dígitos como parte entera y tres dígitos para la parte decimal.

Para imprimir la gráfica es necesario un *manejador de impresión*. Los manejadores de impresión son archivos que convierten los comandos de una computadora en el lenguaje que utilice la impresora o el plotter en la cual se quiere imprimir.

Dado que el número de datos a graficar es bastante grande, es necesario hacer una correcta asignación de memoria para no interferir en el proceso normal de ejecución de FIX DMACS.

Para realizar estas rutinas de graficación, conversión y asignación de memoria se emplearan las herramientas para ciencias e ingeniería (del inglés *Science and Engineering Tools*) de Quinn Curtis. Estas herramientas son una colección de procedimientos y funciones computacionales de propósito general las cuales resuelven los problemas más comunes de graficación y análisis de datos encontrados en las aplicaciones de ciencia e ingeniería.

Algunos de los procedimientos de Quinn Curtis permiten hacer una correcta asignación de memoria, además de que proporcionan manejadores de impresión, de entre los cuales destacan dos: uno para impresión en plotter (lenguaje HPGL que es el lenguaje gráfico para la marca Hewlett Packard -del inglés *Hewlett Packard Graphics Language*-) y otro para una impresora láser, la HP Laser Jet III.

Las herramientas que se han mencionado anteriormente son programas y librerías que pueden combinarse y compilarse con otras rutinas elaboradas por el programador para obtener los resultados deseados. La compilación de estas rutinas sólo se puede realizar mediante el compilador de QuickC versión 2.0 (o una versión superior).

De la edición y compilación de las rutinas para generar las gráficas diarias, se creará un programa ejecutable con argumentos. El programa **REPORTES** creará la línea de órdenes con los argumentos necesarios para ejecutar el programa **GRAFICA** dentro de la interfaz de usuario del programa mismo. Las **entradas** para este submódulo son los argumentos que servirán para ejecutar el programa **GRAFICA** y las proporcionará el mismo programa de las elecciones que el usuario haya realizado para generar la(s) gráfica(s).

El submódulo **GRAFICAS DIARIAS** tendrá como **entradas**:

- El número del sistema fotovoltaico seleccionado en la interfaz de usuario.
- El identificador del sistema fotovoltaico en cuestión. Para generar gráficas diarias será condición que el sistema fotovoltaico se encuentre en el archivo **ALTA.DAT**, por lo que el identificador correspondiente será leído de este archivo.
- La fecha que corresponde al día en que fueron monitoreados y almacenados los datos para el sistema fotovoltaico. Esta fecha será introducida por el usuario después de seleccionar la opción para generar gráficas de la ventana para reportes diarios.
- El nombre del archivo de salida que deberá llevar lo siguiente: día (dos dígitos), mes (dos letras), año (los dos últimos dígitos), número de sistema (dos dígitos) y la extensión **.PLT**.
- Una bandera que indique si sólo se va a generar una gráfica o si se van a generar gráficas para todos los sistemas fotovoltaicos. Esta bandera se activará al seleccionar la opción que permitirá al submódulo **GRAFICAS DIARIAS** generar gráficas de todos los sistemas fotovoltaicos encontrados en el archivo **ALTA.DAT**. Si el usuario selecciona un solo sistema, el submódulo generará la gráfica, la desplegará en pantalla y esperará a que éste pulse una tecla para continuar. Si el usuario eligiera generar gráficas para todos los sistemas dados de alta, entonces las gráficas deberán generarse y desplegarse en forma continua y en orden ascendente (por número de sistema fotovoltaico).

Las rutinas de este submódulo realizarán todas las validaciones de datos necesarias para que la gráfica generada aparezca sin cortes ni picos al ser desplegada. Además, se determinó que el archivo de salida que se obtenga estará configurado para imprimir dicha gráfica en una impresora HP Laser Jet III, por lo que el manejador de impresora se asignará automáticamente dentro de estas rutinas.

El submódulo *GRAFICAS DIARIAS* tendrá como *salidas*:

- La gráfica generada, lista para imprimirse o archivar.
- Si la gráfica se genera correctamente, se desplegará un mensaje en pantalla que así lo indique.

Si el reporte de salida se genera correctamente, se desplegará una ventana que permita al usuario imprimirlo y/o archivarlo. Para realizar esta tarea se crearon los módulos de propósito general *ARCHIVAR* e *IMPRIMIR*, cuyo diseño se describe en las secciones 4.2.2.7 y 4.2.2.8.

B) Submódulo *TABLAS*

Dentro de este submódulo se incluirán las tres opciones restantes del generador de reportes diarios. Estas opciones corresponden a la tabla resumen, a la tabla de fallas y a la tabla de datos. Se agruparon estas tres opciones debido a que el archivo de datos a procesar es el mismo para cada fecha introducida, desde teclado mediante la interfaz de usuario para el programa *REPORTES*.

Debido a que el número de datos a manejar es muy grande es necesaria una cantidad considerable de memoria para manipular dichos datos. Como los valores que se emplean son los mismos que para la generación de gráficas se utilizarán también las herramientas de Quinn Curtis.

■ Tabla resumen

Esta tabla llevará un encabezado que indicará el tipo de reporte, el número e identificador del sistema fotovoltaico y la fecha del reporte al que

corresponde. Todos los datos que se ubican en el recuadro de la gráfica serán incluidos después del encabezado de la tabla resumen.

Los valores calculados tendrán cuatro dígitos como parte entera y tres dígitos para la parte decimal.

■ **Tabla de fallas**

Las rutinas para generar esta tabla buscarán todas las pérdidas de datos que se hayan producido el día que especifique el usuario en la interfaz correspondiente y realizarán el cálculo del porcentaje de datos perdidos en ese mismo día.

Esta tabla llevará un encabezado que indicará el tipo de reporte, el número, el identificador del sistema y la fecha del reporte al que corresponde. Después del encabezado se incluirán el listado de los datos perdidos, la hora en que hayan ocurrido y el porcentaje de datos perdidos durante el día.

Si no se llegara a detectar pérdida alguna de datos durante el día especificado, el reporte de salida que se genere llevará el mismo encabezado que se especifica en el párrafo anterior y un mensaje que indique que no hubo pérdidas en el sistema durante ese día.

■ **Listado de datos**

Este listado corresponde a los datos que se generaron durante el día que especifique el usuario. Las rutinas para generar este reporte crearán un encabezado para los datos del archivo que se obtiene al acceder la base de datos *FOT* con los formatos diarios elaborados con el generador de reportes de FIX DMACS.

El encabezado para esta tabla de datos indicará el tipo de reporte, el número, el identificador del sistema y la fecha del reporte al que corresponde. Después del encabezado se incluirán el listado de datos tal y como se obtuvo del archivo fuente que proporciona FIX DMACS.

De la edición y compilación de las rutinas para generar las tres tablas diarias, se creará un programa ejecutable con argumentos. El programa

REPORTES creará la línea de órdenes con los argumentos necesarios para ejecutar el programa **DIARIOS** dentro de la interfaz de usuario del programa mismo. Las **entradas** para este submódulo son los argumentos que servirán para ejecutar el programa **DIARIOS** y las proporcionará el programa **REPORTES** de las elecciones que el usuario haya realizado para generar la(s) tabla(s).

El submódulo **TABLAS** tendrá como **entradas**:

- El número del sistema fotovoltaico seleccionado en la interfaz de usuario.
- El identificador del sistema fotovoltaico en cuestión. Para generar cada una de las tablas diarias será condición que el sistema se encuentre en el archivo **ALTA.DAT**, por lo que el identificador correspondiente será leído de este archivo.
- La fecha que corresponde al día en que fueron monitoreados y almacenados los datos para el sistema fotovoltaico. Esta fecha será introducida por el usuario después de seleccionar una de las opciones para generar tablas de la ventana para reportes diarios.
- La opción que indique el tipo de tabla a generar.
- Una bandera que indique si sólo se va a generar una tabla o si se van a generar tablas para todos los sistemas fotovoltaicos. Esta bandera se activará cuando el usuario seleccione la opción para generar alguna de las tablas para todos los sistemas fotovoltaicos dados de alta mediante el módulo **ALTA**. Si el usuario selecciona un solo sistema fotovoltaico el programa enlazará el submódulo **TABLAS**, generará la tabla, la desplegará en pantalla y esperará una respuesta del usuario para continuar. Si el usuario eligiera generar tablas de todos los sistemas, entonces las tablas deberán generarse y desplegarse en orden ascendente (por número de sistema fotovoltaico) y en forma continua.

Las rutinas del submódulo **TABLAS** realizarán todas las validaciones de datos necesarias para generar la tabla resumen. Las otras dos tablas no necesitan validaciones de datos.

El submódulo *TABLAS* tendrá como *salidas*:

- El reporte generado, listo para imprimirse o archivar. El nombre del archivo de salida para el reporte generado deberá llevar lo siguiente: día (dos dígitos), mes (dos letras), año (los dos últimos dígitos), número de sistema (dos dígitos) y la extensión que corresponde a cada tabla. Dicha extensión la asignará el submódulo *TABLAS* de la siguiente manera: *.TAB* para la tabla resumen, *.ERR* para la tabla de fallas y *.DAT* para la tabla de datos.

- Si la tabla de salida no se generara correctamente se deberá desplegar un mensaje en pantalla que así lo indique.

Si la tabla de salida se genera correctamente, se desplegará una ventana que permita al usuario imprimirla y/o archivarla. Para realizar esta tarea se crearon los módulos de propósito general, cuyo diseño se describe en las secciones 4.2.2.7 y 4.2.2.8.

4.2.2.4 MODULO SEMANALES

Este módulo será capaz de generar dos tipos de reportes: tabla resumen y tabla de fallas, o pérdidas de datos, para cada sistema fotovoltaico.

Los reportes semanales se generarán en forma individual o colectiva; es decir, para un solo sistema o para todos los sistemas.

La generación de los reportes semanales dependerá del archivo *ALTA.DAT*. Dado que el programa *REPORTES* contemplará a diez sistemas fotovoltaicos en su estructura, el programa permitirá al usuario seleccionar un solo sistema o todos los sistemas que se encuentren dados de alta en el archivo *ALTA.DAT*.

Para tener acceso a este módulo, el usuario deberá seleccionar la opción reportes *SEMANALES* de la interfaz de usuario que corresponde a este programa. Una vez seleccionada esta opción deberá activarse una ventana que permita al usuario seleccionar el o los sistema(s) fotovoltaico(s) de los cuales necesite generar reportes semanales.

Una vez seleccionado el sistema fotovoltaico, el programa *REPORTES* deberá verificar si está dado de alta o no.

Si el sistema fotovoltaico no se encuentra dado de alta, entonces se desplegará un mensaje que así lo indique al usuario. Si esto sucede el programa deberá permitir al usuario hacer una nueva selección.

Si la selección de usuario es correcta, entonces se desplegará una ventana que contemple las dos opciones del generador de reportes semanales.

El usuario deberá seleccionar una de las opciones. Una vez seleccionado un reporte semanal se desplegará una ventana para que el usuario introduzca la fecha en que quiere el reporte. Dependiendo de la selección hecha por el usuario antes de introducir la fecha, el programa ejecutará uno de los dos submódulos que integran el módulo *SEMANALES*.

No será posible seleccionar más de un tipo de reporte a la vez.

El módulo *SEMANALES* deberá proporcionar al usuario la facilidad de generar los dos tipos de reportes con sólo introducir una vez la fecha en que necesita el reporte, o con fechas distintas si así lo quiere.

Si el o los reporte(s) de salida no se generara(n) correctamente, se desplegará una ventana que así lo indique y el sistema regresará a la ventana para reportes semanales. Si la generación es correcta, entonces se activará una ventana que permita enlazar los módulos *ARCHIVAR* e *IMPRIMIR* para imprimir y/o archivar el o los reporte(s) generado(s).

El módulo *SEMANALES* deberá ser dividido para su programación en dos submódulos, como se muestra en la siguiente figura:

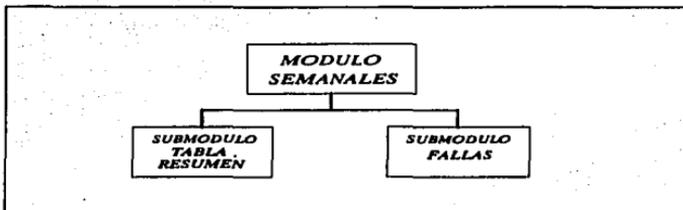


Figura 4.4 Descomposición del módulo *SEMANALES*

La programación del módulo *SEMANALES* se hará utilizando el editor y compilador de BorlandC versión 3.0.

A) Submódulo *TABLA RESUMEN*

Este submódulo servirá para generar la tabla resumen semanal que obtendrá los valores máximos y mínimos (para el voltaje mínimo en batería) de las tablas resumen diarias de siete días (una semana). Las tablas resumen diarias debieron ser creadas con anterioridad mediante el módulo *DIARIOS*.

Esta tabla llevará un encabezado que indicará el tipo de reporte, el número, el identificador del sistema fotovoltaico y las fechas inicial y final de una semana. Todos los valores máximos y mínimos (para el voltaje mínimo en batería), obtenidos en la semana serán incluidos después del encabezado de la tabla resumen semanal.

Los valores calculados tendrán cuatro dígitos como parte entera y tres dígitos para la parte decimal.

El proceso para generar la tabla resumen semanal será el siguiente:

El usuario deberá seleccionar la opción *SEMANALES* y un sistema fotovoltaico (o la opción para generar reportes de todos los sistemas que se encuentren dados de alta). Si el sistema fotovoltaico se encuentra dado de alta, entonces se desplegará una ventana que contenga las opciones tabla

resumen y tabla de fallas. Al seleccionar la opción tabla resumen, el programa **REPORTES** desplegará una ventana que pregunte al usuario la fecha del primer día de la semana en que quiera generar la tabla resumen semanal. Una vez introducida esta fecha por el usuario, el programa verificará la existencia de los siete reportes que contengan la tabla resumen diaria correspondiente a cada día de la semana. Esta búsqueda se hará en el directorio de disco duro de la computadora que el programador designe. Si no existiera alguno de los reportes diarios, no será posible generar la tabla resumen semanal. Si el submódulo **TABLA RESUMEN**, encuentra todos los archivos necesarios, entonces procederá a obtener los valores máximos durante esa semana y guardará, en forma temporal, estos valores en un archivo con la extensión **.SET**. Una vez generado el reporte, lo desplegará en pantalla y esperará que el usuario pulse una tecla para continuar (si el usuario seleccionó un solo sistema) o creará y desplegará, en forma continua, la tabla resumen semanal para cada uno de los sistemas encontrados en el archivo **ALTA.DAT**. Después de este despliegue de información aparecerá la ventana que permita enlazar los módulos para imprimir y/o archivar el o los reporte(s) generado(s).

Este submódulo será utilizado también para generar la tabla resumen mensual, ya que el procedimiento de búsqueda y cálculo es el mismo.

El submódulo **TABLA RESUMEN** tendrá como **entradas**:

- El número del sistema fotovoltaico seleccionado en la interfaz de usuario.
- El identificador del sistema fotovoltaico en cuestión. Para generar la tabla resumen semanal o mensual será condición que el sistema fotovoltaico se encuentre dentro del archivo **ALTA.DAT**, por lo que el identificador correspondiente será leído de este archivo.
- La fecha que corresponde al primer día de la semana o del mes (según el tipo de reporte). Esta fecha será introducida por el usuario después de seleccionar la opción que identifica a la tabla resumen de la ventana para reportes semanales o mensuales.
- La opción que indique el reporte a generar (semanal o mensual).
- Una bandera que indique si sólo se va a generar una tabla o si se van a generar tablas para todos los sistemas fotovoltaicos. Esta bandera se activará

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

cuando el usuario seleccione la opción para generar la tabla resumen de todos los sistemas fotovoltaicos encontrados en el archivo *ALTA.DAT*. Si el usuario selecciona un solo sistema fotovoltaico, el programa enlazará el submódulo *TABLA RESUMEN*, generará la tabla, la desplegará en pantalla y esperará a que el usuario pulse cualquier tecla para continuar. Si el usuario eligiera generar tablas de todos los sistemas, entonces las tablas deberán generarse y desplegarse en orden ascendente (por número de sistema fotovoltaico) y en forma continua.

Estas entradas las proporcionará el programa *REPORTES* de las elecciones que realice el usuario.

El submódulo *TABLA RESUMEN* tendrá como *salidas*:

- La tabla resumen generada, lista para imprimirse o archivar. El nombre del archivo de salida para la tabla resumen deberá llevar lo siguiente: día (dos dígitos), mes (dos letras), año (los dos últimos dígitos), número de sistema (dos dígitos) y la extensión *.SET* o *.MET* que corresponde a la tabla resumen semanal o mensual, respectivamente.
- Si la tabla resumen semanal o mensual no se generara correctamente el programa deberá desplegar un mensaje que así lo indique.

Si la tabla resumen se genera correctamente, se desplegará una ventana que permita al usuario imprimirla y/o archivarla. Para realizar esta tarea se crearon los módulos de propósito general, cuyo diseño se describe en las secciones 4.2.2.7 y 4.2.2.8.

B) Submódulo *FALLAS*

Las rutinas de este submódulo generarán, en este caso, la(s) tabla(s) de fallas semanal(es) para uno o todos los sistemas fotovoltaicos. Estas rutinas buscarán todas las pérdidas de datos que se hayan producido durante una semana y realizarán el cálculo del porcentaje semanal de los datos perdidos. La búsqueda se hará en los siete reportes de la tabla de fallas diaria que debieron ser creados previamente mediante el módulo *DIARIOS*. El primer día de la semana lo indicará el usuario una vez que teclee la fecha en la ventana destinada para ello.

La tabla de fallas semanal llevará un encabezado que indicará el tipo de reporte, el número, el identificador del sistema y la fechas inicial y final de la semana en que se genera el reporte. Después del encabezado se incluirán el listado de datos perdidos, la hora en que haya ocurrido tal pérdida y el porcentaje de datos perdidos durante esa semana.

Si no se llegara a detectar pérdida alguna de datos durante la semana, el archivo de salida que se genere llevará el mismo encabezado que se especifica en el párrafo anterior y después un mensaje que indique que no hubo pérdidas de datos en el sistema durante esa semana.

El proceso para generar la tabla de fallas semanal es similar al que se describió en el submódulo *TABLA RESUMEN* con dos únicas variantes:

- El usuario deberá seleccionar la opción tabla de fallas.
- El archivo de salida tendrá la extensión *.SEE*.

Este submódulo será utilizado también para generar la tabla de fallas mensual, ya que el procedimiento de búsqueda y cálculo es el mismo.

El submódulo *FALLAS* tendrá como *entradas*:

- El número del sistema fotovoltaico seleccionado en la interfaz de usuario.
- El identificador del sistema fotovoltaico en cuestión. Para generar la tabla de fallas será condición que el sistema fotovoltaico se encuentre en el archivo *ALTA.DAT*, por lo que el identificador correspondiente será leído de este archivo.
- La fecha que corresponde al primer día de la semana (o al primer día de las cuatro semanas en caso de que el reporte sea mensual). Esta fecha será introducida por el usuario después de seleccionar la opción que identifica a la tabla de fallas de la ventana para reportes semanales (o mensuales).
- La opción que indique el reporte a generar (semanal o mensual).
- Una bandera que indique si sólo se va a generar una tabla o si se van a generar tablas para todos los sistemas fotovoltaicos. Esta bandera se activará cuando el usuario seleccione la opción para generar la tabla de fallas para

todos los sistemas fotovoltaicos que se encuentren en el archivo **ALTA.DAT**. Si el usuario selecciona un solo sistema fotovoltaico el programa enlazará el submódulo **FALLAS**, generará la tabla, la desplegará en pantalla y esperará a que el usuario pulse cualquier tecla para continuar. Si el usuario eligiera generar tabla de fallas para cada uno los sistemas que se encuentren en el archivo **ALTA.DAT**, entonces las tablas deberán generarse y desplegarse en orden ascendente (por número de sistema fotovoltaico) y en forma continua.

Estas entradas las proporcionará el programa **REPORTES** de las elecciones que realice el usuario.

El submódulo **FALLAS** tendrá como *salidas*:

- La tabla de fallas generada, lista para imprimirse o archivarla. El nombre del archivo que guardará la tabla de fallas deberá llevar lo siguiente: día (dos dígitos), mes (dos letras), año (los dos últimos dígitos), número de sistema (dos dígitos) y la extensión **.SEE** o **.MEE** que corresponde a la tabla de fallas semanal o mensual, respectivamente.
- Si la tabla de fallas no se genera correctamente el programa deberá desplegar un mensaje que así lo indique.

Si la tabla de fallas se genera con éxito, se desplegará una ventana que permita al usuario imprimirla y/o archivarla y además enlazar los módulos **IMPRIMIR** y/o **ARCHIVAR**.

4.2.2.5 MODULO MENSUALES

Este módulo será capaz de generar tres tipos de reportes: tabla resumen, tabla de fallas, o pérdidas de datos, y un archivo mensual especial para cada sistema fotovoltaico.

Los reportes mensuales podrán generarse en forma individual o colectiva; es decir, para un solo sistema o para todos los sistemas.

La generación de los reportes mensuales dependerá del archivo **ALTA.DAT**. Dado que el programa **REPORTES** contemplará a diez sistemas fotovoltaicos en su estructura, el programa permitirá al usuario seleccionar un

solo sistema o todos los sistemas que se encuentren dados de alta mediante el módulo *ALTA*.

Para tener acceso a este módulo, el usuario deberá seleccionar la opción reportes *MENSUALES* de la interfaz de usuario que corresponde a este programa. Una vez seleccionada esta opción deberá activarse una ventana que permita al usuario seleccionar el o los sistema(s) fotovoltaico(s) de los cuales necesite generar reportes mensuales.

Una vez seleccionado el sistema fotovoltaico, el programa *REPORTES* deberá verificar si está dado de alta o no.

Si el sistema fotovoltaico no se encuentra dado de alta, entonces se desplegará un mensaje que así lo indique. Si esto sucede el programa deberá permitir al usuario hacer una nueva selección.

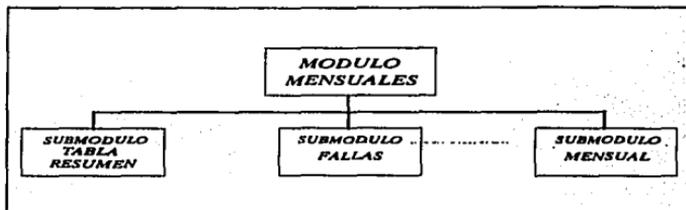
Si la selección de usuario es correcta, entonces se desplegará una ventana que contemple las tres opciones del generador de reportes mensuales.

El usuario deberá seleccionar una de las tres opciones. Una vez seleccionado un tipo de reporte mensual se desplegará una ventana para introducir la fecha en que se quiere el reporte. Dependiendo de la selección hecha por el usuario antes de introducir la fecha, el programa enlazará uno de los tres submódulos que integran el módulo *MENSUALES*.

El módulo *MENSUALES* deberá proporcionar al usuario la facilidad de generar las tablas resumen y de fallas mensuales con sólo introducir una vez la fecha en que necesita el reporte, o con fechas distintas si así lo quiere. Para generar el archivo mensual, el programa pedirá al usuario que introduzca sólo el mes y el año, ya que el reporte empezará a generarse a partir del primero hasta el último día del mes, por especificaciones del mismo usuario.

Si el o los reporte(s) mensual(es) no se generara(n) correctamente, se desplegará una ventana que así lo indique. Si la generación es correcta, entonces se activará una ventana que permita enlazar los módulos *ARCHIVAR* e *IMPRIMIR* para imprimir y/o archivar el o los reportes generados.

El módulo *MENSUALES* deberá ser dividido para su programación en tres submódulos, como se muestra en la siguiente figura:

Figura 4.5 Descomposición del módulo *MENSUALES*

La programación del módulo *MENSUALES* se hará utilizando el editor y compilador de BorlandC versión 3.0.

A) Submódulo *TABLA RESUMEN*

Este submódulo servirá para generar la tabla resumen mensual que obtendrá los valores máximos y mínimos (para el voltaje en batería) de las tablas resumen semanales de cuatro semanas, a partir de la fecha que declare el usuario en la interfaz del programa *REPORTES*. Los reportes semanales deberán ser creados con anterioridad con el módulo *SEMANALES*.

Esta tabla llevará un encabezado que indicará el tipo de reporte, el número, el identificador del sistema fotovoltaico, la fecha de inicio de la primera semana y la fecha final de la cuarta semana (del día primero al día veintiocho). Todos los valores máximos y mínimos (para el voltaje mínimo en batería), obtenidos en las cuatro semanas serán incluidos después del encabezado de la tabla resumen mensual.

Los valores calculados tendrán cuatro dígitos como parte entera y tres dígitos para la parte decimal.

El proceso para generar la tabla resumen mensual será el siguiente:

El usuario deberá seleccionar la opción *MENSUALES* y un sistema fotovoltaico (o la opción para generar reportes de todos los sistemas que se encuentren en el archivo *ALTA.DAT*). Si el sistema fotovoltaico se encuentra dado de alta, entonces se desplegará una ventana para reportes mensuales. Al seleccionar la opción tabla resumen, el programa *REPORTES* desplegará una ventana que pregunte la fecha de inicio para el reporte mensual (que es el primer día de las cuatro semanas). Una vez introducida esta fecha por el usuario, el programa verificará la existencia de los cuatro reportes que contengan la tabla resumen semanal a partir de la fecha de inicio. Esta búsqueda se hará en el directorio de disco duro de la computadora que el programador designe. Si no existiera alguno de los reportes semanales, no será posible generar la tabla resumen mensual. Si el submódulo *TABLA RESUMEN*, encuentra todos los reportes necesarios, entonces procederá a obtener los valores máximos durante las cuatro semanas y guardará, en forma temporal, estos valores en un archivo con la extensión *.MET*. Una vez generada la tabla resumen mensual, la desplegará en pantalla y esperará que el usuario pulse una tecla para continuar (si el usuario sólo seleccionó un sistema) o creará y desplegará, en forma continua, la tabla resumen mensual para todos los sistemas dados de alta. Después de este despliegue de información aparecerá la ventana que permita enlazar los módulos para imprimir y/o archivar el o los reporte(s) generado(s).

Este submódulo es el mismo que se utilizará para generar la tabla resumen semanal, ya que el procedimiento de búsqueda y cálculo es el mismo.

Las *entradas y salidas* para el submódulo *TABLA RESUMEN* son las mismas que se especificaron en el inciso A) de la sección 4.2.2.4.

B) Submódulo *FALLAS*

Las rutinas para generar esta tabla buscarán todas las fallas o pérdidas de datos que se hayan producido durante veintiocho días (cuatro semanas) y realizarán el cálculo del porcentaje mensual de los datos perdidos. El primer día de la semana lo indicará el usuario una vez que teclee la fecha en la ventana destinada para ello.

La tabla de fallas mensual llevará un encabezado que indicará el tipo de reporte, el número, el identificador del sistema fotovoltaico, la fecha de inicio de la primera semana y la fecha final de la cuarta semana (el día primero y el

día veintiocho). Después del encabezado se incluirá el listado de todos los valores perdidos, la hora en que hayan ocurrido y el porcentaje de valores perdidos durante esas cuatro semanas.

Si no se llegara a detectar pérdida de datos durante las cuatro semanas, el archivo de salida que se genere llevará el mismo encabezado que se especifica en el párrafo anterior y después un mensaje que indique que no hubo pérdidas en el sistema durante esas cuatro semanas.

El proceso para generar la tabla de fallas mensual es similar al que se describió en el submódulo **TABLA RESUMEN** con dos únicas variantes:

- El usuario deberá seleccionar la opción tabla de fallas.
- El archivo de salida tendrá la extensión **.MEE**.

El submódulo **FALLAS** tendrá como **entradas** y **salidas** las mismas que se indican en el inciso B) de la sección 4.2.2.4.

C) Submódulo *MENSUAL*

Este submódulo servirá para generar, por cada uno de los diez sistemas fotovoltaicos, un archivo mensual que contenga el valor de las variables especificadas en el formato mensual de datos que se describe en la sección 4.2.2.1, en intervalos de cada cinco minutos.

Se le llama formato mensual, no porque permita obtener los datos de un mes completo, sino porque se utilizará para obtener los datos por cada día del mes, del primero al último día de acuerdo al calendario.

El archivo mensual no llevará encabezado alguno.

El proceso para generar el archivo mensual será el siguiente:

El usuario deberá seleccionar la opción **MENSUALES** y un sistema fotovoltaico (o la opción para generar reportes de todos los sistemas que se encuentren en el archivo **ALTA.DAT**). Si el sistema fotovoltaico se encuentra dado de alta, entonces se desplegará una ventana para reportes mensuales. Al seleccionar la opción archivo mensual, el programa **REPORTES** desplegará una ventana que pregunte el mes y el año para el archivo mensual. Una vez

introducidos estos dos datos por el usuario, el programa creará la orden de enlace para generar un archivo diario de datos (de las 00:00 a las 23:55 hrs.) en intervalos de cada cinco minutos con el formato mensual de datos declarado en FIX DMACS. Se generará un archivo por cada día del mes a partir del día primero del mismo. A la vez que se va generando, se irá concatenando en el archivo mensual.

Si alguno de los archivos provenientes de FIX DMACS no se generara correctamente, el programa deberá detectar esta falla e indicar al usuario lo que está sucediendo y no generará ningún archivo mensual.

Dado que el archivo mensual será muy largo, el programa no desplegará en pantalla el contenido del mismo, sólo lo creará. Una vez generado el reporte, el programa detendrá su ejecución hasta que el usuario pulse una tecla para continuar (si el usuario sólo seleccionó un sistema) o creará en forma continua los archivos mensuales para todos los sistemas dados de alta.

El submódulo *MENSUAL* tendrá como *entradas*:

- El número del sistema fotovoltaico seleccionado en la interfaz de usuario.
- El mes y el año para generar el archivo mensual. Esta fecha será introducida por el usuario después de seleccionar la opción para generar el archivo mensual de la ventana para reportes mensuales
- Una bandera que indique si sólo se va a generar un archivo mensual o si se van a generar archivos mensuales para todos los sistemas fotovoltaicos. Esta bandera se activará cuando el usuario seleccione la opción para generar la tabla de fallas para todos los sistemas fotovoltaicos encontrados en el archivo *ALTA.DAT*. Si el usuario selecciona un solo sistema fotovoltaico el programa enlazará el submódulo *MENSUAL*, generará el archivo mensual y esperará a que el usuario pulse una tecla para continuar. Si el usuario eligiera generar el archivo mensual para cada uno de los sistemas, entonces los archivos mensuales se generarán en forma continua y en orden ascendente (por número de sistema fotovoltaico).

Estas entradas las proporcionará el programa *REPORTES* de las elecciones que realice el usuario.

El submódulo *MENSUAL* tendrá como *salidas*:

- El archivo mensual generado, listo para imprimirse o archivar. El nombre del archivo que guarde el archivo mensual deberá llevar lo siguiente: día (dos dígitos), mes (dos letras), año (los dos últimos dígitos), número de sistema (dos dígitos) y la extensión **.MEN**.

- Si este reporte no se genera correctamente, el programa deberá desplegar un mensaje que así lo indique.

Si el archivo mensual se genera correctamente, se desplegará una ventana que permita al usuario imprimirlo y/o archivarlo mediante el enlace de los módulos *ARCHIVAR* e *IMPRIMIR*.

4.2.2.6 MODULO ANUALES

Este módulo se incluyó, ya que a futuro se pretenden generar también reportes anuales, por lo que el diseño del mismo queda pendiente hasta que se definan los requerimientos.

4.2.2.7 MODULO IMPRIMIR

Este módulo será de propósito general, es decir que se utilizará para enviar a impresión todos los reportes que se generen mediante el programa *REPORTES*.

La opción de imprimir reportes se deberá ubicar junto con la opción de archivar reportes para que una vez generado el reporte sea posible imprimirlo y/o archivarlo.

Una vez seleccionada dicha opción, se deberá desplegar un aviso que pregunte si se quiere continuar con la impresión o si se quiere abortar la impresión.

Para la opción *GRAFICAS DIARIAS*, la interfaz de usuario correspondiente deberá desplegar un mensaje que indique al usuario que es necesaria la utilización de una impresora HP Laser Jet III o HP Laser Jet 4 y que le pregunte si desea continuar con la impresión.

Para las opciones restantes, el programa deberá desplegar un aviso que pregunte al usuario si desea continuar con la impresión, antes de enviar el reporte de salida a impresora.

Si el usuario introduce una respuesta afirmativa entonces el programa enviará a impresión la gráfica o el reporte generado, por el puerto paralelo LPT1 de la computadora al cual deberá estar conectada la impresora. Una vez enviado a impresión el reporte o reporte(s) generado(s), el programa regresará a la ventana que contenga las opciones para imprimir y/o archivar reportes.

Si la respuesta por parte del usuario es negativa entonces el programa regresará también a la ventana que contenga las opciones para imprimir y/o archivar reportes, sin realizar ninguna acción.

El módulo *IMPRIMIR* tendrá como *entradas*:

- El tipo de reporte que el usuario seleccionó.
- El nombre o nombres de los archivos para el o los reportes generados.
- La respuesta del usuario antes de imprimir.

El módulo *IMPRIMIR* tendrá como *salida* la orden para enviar a impresión el archivo generado.

4.2.2.8 MODULO ARCHIVAR

Este módulo será de propósito general, es decir que se utilizará para archivar en unidades de disco flexible o en unidad de disco duro todos los reportes que se generen por medio del programa *REPORTES*.

La opción de archivar se deberá ubicar junto con la opción de imprimir para que una vez generado el reporte sea posible imprimirlo y/o archivarlo.

Una vez seleccionada la opción *ARCHIVAR*, se deberá desplegar una ventana que indique las unidades en que el usuario puede guardar el reporte de salida. Estas unidades serán A, B ó C. Para almacenar los archivos en disco

duro, el programa asignará automáticamente un directorio para guardar dichos archivos (este directorio será definido por el programador).

El módulo *ARCHIVAR* tendrá como *entradas*:

- El o los nombre(s) de los archivo(s) para el o los reporte(s) generado(s).
- La unidad en que se desea guardar la información.

El módulo *ARCHIVAR* tendrá como *salida* la orden para archivar el o los reporte(s) de salida en la unidad seleccionada por el usuario.

Si el usuario no quisiera archivar los reportes generados, el programa le permitirá no hacerlo.

Independientemente de la decisión del usuario, el programa *REPORTES* regresará a la ventana que contiene las opciones para imprimir y/o archivar reportes.

4.2.2.9 MODULO INTERFAZ

Hasta aquí, se han estructurado las rutinas para generar todos los reportes que se especificaron en los requerimientos, sin embargo falta por diseñar la interfaz de usuario que permita enlazar en un solo plano, cada uno de los módulos que se han detallado. Al igual que los módulos *SEMANALES* y *MENSUALES*, las rutinas para crear la interfaz de usuario se editarán y compilarán en BorlandC versión 3.0.

La interfaz de usuario se estructura de la siguiente manera:

- El acceso para la mayor parte de las funciones se hará mediante la validación de pantalla con *retón*. Sólo se utilizará el teclado para funciones que así lo requieran, tales como introducir la fecha para generar un reporte. La validación de pantalla se realizará a través de la interrupción 86.
- Se creará una pantalla principal que contemple las siguientes opciones: DIARIOS, SEMANALES, MENSUALES, ANUALES y ALTA.

- Si el usuario selecciona una de las cuatro primeras opciones se activará una ventana que permita seleccionar un solo sistema fotovoltaico del cual se desee generar reportes o todos los sistemas fotovoltaicos que se encuentren en el archivo *ALTA.DAT*. Para tal fin se creará una ventana que contemple a los diez sistemas para seleccionarlos en forma individual y una opción adicional que permita seleccionar a todos los sistemas.

- Por cada una de las cuatro primeras opciones se creará una ventana para que el usuario elija el tipo de reporte a generar. Para los reportes diarios, la ventana correspondiente deberá contener las opciones: *GRAFICAS*, *TABLA RESUMEN*, *TABLA DE FALLAS* y *LISTADO DE DATOS*. Para los reportes semanales las opciones serán: *TABLA RESUMEN* y *TABLA DE FALLAS*. Para los reportes mensuales, las opciones serán *TABLA RESUMEN*, *TABLA DE FALLAS* y *ARCHIVO MENSUAL*. Las opciones para los reportes anuales no se han definido, por lo que deberá crearse una ventana que indique que esta función no se ha implementado (en forma temporal).

- Se crearán además dos ventanas para introducir la fecha en que se quiera el reporte. La primera de ellas se utilizará para leer desde teclado el día, mes y año que forman parte de la fecha para todos los tipos de reportes a generar, excepto para la opción *ARCHIVO MENSUAL*. Para introducir la fecha a partir de la cual se generará el archivo mensual, se creará la otra ventana que permitirá al usuario introducir por teclado al programa, sólo el mes y el año.

- Se creará una ventana para imprimir y/o archivar cualquier tipo de reporte generado. Esta ventana deberá contener dos opciones: *IMPRIMIR* y *ARCHIVAR*.

Para la opción *IMPRIMIR* se crearán dos ventanas para preguntar al usuario si desea continuar con la impresión. Una de esas ventanas se utilizará para imprimir las gráficas diarias generadas y advertirá al usuario que sólo podrá enviar a impresión la(s) gráfica(s) creada(s) si se encuentra conectada una impresora HP Laser Jet III al puerto paralelo de la computadora (ver la sección 4.2.2.7). La otra ventana servirá para enviar a impresión los reportes que se generen a partir de las opciones restantes.

Para la opción *ARCHIVAR* se creará una ventana que indique las unidades de disco en las cuales sea posible archivar el o los reporte(s) generado(s). Estas unidades serán A, B ó C. Para la unidad C (disco duro) se definirá un directorio especial donde serán archivados todos los reportes de

salida siempre y cuando el usuario elija la unidad C para tal efecto.

- Para que el usuario pueda dar de alta un sistema se creará una ventana que contenga los diez sistemas fotovoltaicos y que indique cuáles están dados de alta y cuáles no. Para que el usuario pueda dar de alta un sistema, se creará otra ventana que permita introducir el identificador del sistema fotovoltaico. Este identificador tendrá como máximo doce caracteres y/o dígitos.

- Se crearán los mensajes de alarma necesarios para evitar que el usuario pueda truncar el funcionamiento normal del programa *REPORTES*.

Hasta aquí, se ha diseñado la estructura del programa *REPORTES*. Este programa se fragmentó en nueve módulos básicos: *FORMATOS FIX DMACS*, *ALTA*, *DIARIOS*, *SEMANALES*, *MENSUALES*, *ANUALES*, *IMPRIMIR*, *ARCHIVAR* e *INTERFAZ*, cada uno de los cuales pretende satisfacer al máximo los requerimientos especificados en el capítulo tres. Las rutinas de programación se listan en el apéndice C y las pruebas para la implementación del mismo se detallan en el capítulo siguiente.

CAPITULO 5

Pruebas e implementación

En este capítulo se describen las pruebas que se aplicaron al programa *REPORTES* antes de su instalación formal y definitiva. También se listan los requerimientos de *software* y *hardware* para la instalación del mismo.

La codificación del programa *REPORTES* se hizo por partes. Cada vez que se codificaba una de esas partes, ésta se compilaba y se probaba en forma individual dentro del ambiente del programa *FIX DMACS* para observar si no se afectaba el funcionamiento del mismo y después se adicionaba a la interfaz de usuario correspondiente para ser probada en forma integrada.

5.1 PROCEDIMIENTO DE CODIFICACION DEL PROGRAMA *REPORTES*

La programación y compilación del programa *REPORTES* se realizó de la siguiente manera:

- Creación de la pantalla principal del programa en cuestión con las funciones: *DIARIOS*, *SEMANALES*, *MENSUALES*, *ANUALES* y *ALTA*.

- Edición del archivo *ALTA.DAT*.
- Creación de la ventana que contiene los identificadores de los diez sistemas fotovoltaicos, para la opción ALTA.
- Creación de la ventana que muestra el identificador de cada uno de los diez sistemas fotovoltaicos (S1,...S10) y la opción TODOS. Esta ventana se despliega cada vez que el usuario seleccione una de las funciones (u opciones) principales del programa *REPORTES*, excepto para la opción ALTA.
- Creación de las ventanas que permiten seleccionar cada una de las opciones para los reportes diarios, semanales y mensuales.
- Validación de las posiciones en pantalla, mediante el *ratón*, para cada una de las funciones principales.
- Validación de las posiciones en pantalla, mediante el *ratón*, de los identificadores de cada uno de los diez sistemas fotovoltaicos para la función ALTA.
- Codificación de las rutinas para dar de baja o de alta un sistema fotovoltaico dentro de la interfaz de usuario del programa *REPORTES*.
- Validación de las posiciones en pantalla, mediante el *ratón*, de cada uno de los identificadores de los diez sistemas fotovoltaicos y para la opción TODOS de las funciones DIARIOS, SEMANALES y MENSUALES.
- Creación de los enlaces entre las opciones principales, los sistemas fotovoltaicos, sus identificadores y sus ventanas respectivas (despliegue de menús para cada función principal).
- Validación de las posiciones en pantalla, mediante el *ratón*, de cada una de las funciones que se derivan de la opción DIARIOS.
- Creación de la ventana que permite leer la fecha para crear reportes diarios, semanales y mensuales (excepto para la opción ARCHIVO MENSUAL) y validación de esa fecha que se introduce por teclado en dicha ventana.
- Creación de los formatos de datos de FIX DMACS para reportes diarios.

- Creación del enlace para acceder a la base de datos de FIX DMACS y generar los archivos de datos diarios de acuerdo a la fecha introducida por el usuario y con los formatos creados en el punto anterior.
- Codificación de las rutinas para generar el programa GRAFICA.EXE (para la creación de gráficas diarias).
- Codificación de las rutinas para generar el programa DIARIOS.EXE (para la creación de tablas diarias).
- Creación de los enlaces necesarios para ejecutar los dos programas anteriores de acuerdo a la selección de una de las opciones, mediante el *ratón*, de la ventana para reportes diarios.
- Creación de los enlaces necesarios para desplegar en pantalla los reportes generados con los dos programas anteriores.
- Validación de las posiciones en pantalla, mediante el *ratón*, de cada una de las funciones que se derivan de la opción SEMANALES.
- Codificación de las rutinas para generar las tablas resumen y tablas de fallas semanales y mensuales.
- Enlace de estas rutinas con los identificadores respectivos de la interfaz de usuario del programa *REPORTES*. Estos identificadores corresponden a TABLA RESUMEN y TABLA DE FALLAS de las ventanas para reportes semanales y mensuales.
- Creación de los enlaces necesarios para desplegar en pantalla las tablas anteriores.
- Creación de la ventana que permite leer la fecha para la opción ARCHIVO MENSUAL (sólo el mes y el año) y validación de esa fecha que se introduce por teclado en dicha ventana.
- Creación de los formatos de datos de FIX DMACS para la opción ARCHIVO MENSUAL.

- Creación de las rutinas para acceder la base de datos de FIX DMACS y generar los archivos mensuales de datos a partir de la unión de archivos diarios. Los archivos diarios se crean con los formatos del punto anterior, a partir del primer día del mes que especifica el usuario en la ventana para la opción ARCHIVO MENSUAL.
- Creación de los enlaces necesarios para ejecutar las rutinas de la opción ARCHIVO MENSUAL desde la interfaz de usuario del programa *REPORTES*.
- Validación de las posiciones en pantalla, mediante el *ratón*, de cada una de las funciones que se derivan de la opción DIARIOS.
- Creación de la ventana que contiene las opciones para imprimir y archivar todo tipo de reportes, así como los enlaces necesarios para enviar a impresión y archivar los reportes que se generen.
- Creación de los mensajes de alarma para cada una de las funciones del generador de reportes.

El código de programación correspondiente al programa *REPORTES* se lista en el Apéndice C.

5.2 PRUEBAS A LAS FUNCIONES PRINCIPALES

Como se mencionó anteriormente, antes de implementar en forma definitiva el programa *REPORTES* se realizaron pruebas de funcionalidad y estabilidad por función y después para el sistema integrado.

Las funciones que se probaron son las siguientes:

- ◆ Función de validación de pantalla.
- ◆ Funciones de acceso a la base de datos *FOT*.
- ◆ Función para dar de alta un sistema fotovoltaico.

- Creación de las rutinas para acceder la base de datos de FIX DMACS y generar los archivos mensuales de datos a partir de la unión de archivos diarios. Los archivos diarios se crean con los formatos del punto anterior, a partir del primer día del mes que especifica el usuario en la ventana para la opción ARCHIVO MENSUAL.
- Creación de los enlaces necesarios para ejecutar las rutinas de la opción ARCHIVO MENSUAL desde la interfaz de usuario del programa *REPORTES*.
- Validación de las posiciones en pantalla, mediante el *ratón*, de cada una de las funciones que se derivan de la opción DIARIOS.
- Creación de la ventana que contiene las opciones para imprimir y archivar todo tipo de reportes, así como los enlaces necesarios para enviar a impresión y archivar los reportes que se generen.
- Creación de los mensajes de alarma para cada una de las funciones del generador de reportes.

El código de programación correspondiente al programa *REPORTES* se lista en el Apéndice C.

5.2 PRUEBAS A LAS FUNCIONES PRINCIPALES

Como se mencionó anteriormente, antes de implementar en forma definitiva el programa *REPORTES* se realizaron pruebas de funcionalidad y estabilidad por función y después para el sistema integrado.

Las funciones que se probaron son las siguientes:

- ◆ Función de validación de pantalla.
- ◆ Funciones de acceso a la base de datos *FOT*.
- ◆ Función para dar de alta un sistema fotovoltaico.

◆ Funciones para crear los reportes diarios:

GRAFICA
TABLA RESUMEN
TABLA DE FALLAS
LISTADO DE DATOS

◆ Funciones para crear los reportes semanales y mensuales:

TABLA RESUMEN
TABLA DE FALLAS

◆ Función para crear el archivo mensual.

◆ Función de despliegue en pantalla de los reportes generados.

◆ Funciones de impresión y archivado de los reportes creados.

◆ Funciones de alarma para indicar al usuario que ha producido alguna falla, ya sea por parte de usuario o del programa.

5.2.1 FUNCION DE VALIDACIÓN DE PANTALLA

La validación se realizó de acuerdo a la posición en pantalla de los identificadores que corresponden a cada una de las funciones. Se tomó en cuenta que la pantalla en modo texto normal consta de 25 renglones y 80 columnas (del renglón 0 al 24 y de la columna 0 a la 79).

El mecanismo de validación es el siguiente:

- Establecimiento de las coordenadas inicial y final de cada identificador¹ (por ejemplo para la opción DIARIOS las coordenadas inicial y final son 16,3 y 22,3). Estas coordenadas se establecieron dentro del programa *REPORTES*.
- Obtención de las coordenadas en pantalla en las cuales el usuario posiciona el cursor del *ratón*, por medio de la función *Mousexy* (ver Apéndice C), cada vez que selecciona uno de los identificadores.

¹ Se denomina identificador a cualquier palabra que haga referencia a una función; DIARIOS, IMPRIMIR, S1, S10, GRAFICA, etc.

- Si las coordenadas del *ratón* están dentro del rango de coordenadas inicial y final (rango de selección) definidas dentro del programa *REPORTES*, entonces se procederá a realizar la función que se asocia al identificador seleccionado. Esta función puede ser activación o desactivación de una ventana para seleccionar otro identificador, la creación de algún reporte o bien la impresión o archivado de el o los reporte(s) creado(s).

Siguiendo este mecanismo se verificó que el rango de selección, por medio del *ratón*, para cada identificador fuera el correcto y que no hubiese traslape de funciones. Este traslape podría generarse si la posición de un identificador estuviera dentro o fuera parte del rango de selección de otro identificador. Para evitar este traslape se crearon banderas para cada función. Para cada uno de los identificadores se verificó que la activación y desactivación de cada bandera fuera la correcta.

5.2.2 FUNCIONES DE ACCESO A LA BASE DE DATOS *FOT*

Como se ha mencionado anteriormente, para acceder la base de datos *FOT* y obtener los datos necesarios para la creación de reportes que permitieran observar el comportamiento de los sistemas fotovoltaicos a prueba, se utilizó el programa *GEN.EXE*. Se verificó si la cadena de caracteres para ejecutar dicho programa dentro del programa *REPORTES* se creaba correctamente y de acuerdo con el número de sistema fotovoltaico y la fecha que el usuario había elegido dentro de la interfaz de usuario correspondiente. Hubo algunos problemas para ejecutar la línea de órdenes (cadena de caracteres), ya que el programa *REPORTES* generaba *basura* (caracteres adicionales que no se definieron dentro de dicho programa) al final de dicha línea, lo que impedía la creación de los archivos de datos. El problema se solucionó adicionando un carácter nulo al final de cada cadena generada.

El enlace entre la base de datos *FOT* y el programa *REPORTES* se llevó a cabo sin ningún inconveniente, una vez solucionado el problema que se explicó anteriormente.

Otro problema que surgió en este proceso de enlace y obtención de datos fue que el programa *GEN.EXE* crea el archivo de datos requerido con el mismo nombre que el formato de datos, solo que en lugar de la extensión *.GEN* coloca la extensión *.RPT*. Este nombre es siempre el mismo, aunque la fecha sea distinta. Además, si el programa *GEN.EXE* se ejecuta por segunda

ocasión, los datos no se sobrescriben, sino que se agregan al final del archivo creado.

Para eliminar este inconveniente lo que se hizo fue borrar, desde el programa *REPORTES*, el archivo obtenido de FIX DMACS cada vez que por alguna circunstancia fuera necesario obtener datos para una fecha distinta a la introducida antes que ésta.

Se realizaron pruebas de acceso a la base de datos *FOT* por medio de los formatos diarios y mensuales creados por el programa *Report Creator* de FIX DMACS para cada uno de los diez sistemas fotovoltaicos. Estas pruebas de acceso no interfirieron el funcionamiento normal de FIX DMACS.

5.2.3 FUNCION PARA DAR DE ALTA UN SISTEMA FOTOVOLTAICO

Esta es una de las funciones más importantes dentro del programa *REPORTES*, debido a que éste sólo creará los reportes de los sistemas fotovoltaicos que se den de alta mediante la opción ALTA de la interfaz de usuario de dicho programa.

Se verificó si el sistema fotovoltaico que el usuario selecciona se da de baja en realidad una vez que el programa detecta la existencia de éste dentro del archivo *ALTA.DAT*.

Se verificó también que el sistema fuera dado de alta si el programa detecta que no se encuentra contenido dentro del archivo *ALTA.DAT*. El identificador de dicho sistema es convertido a letras mayúsculas y es insertado a este archivo en el lugar que le corresponde. Este lugar lo asigna el programa realizando una ordenación ascendente de los sistemas que se encuentren en el archivo correspondiente. De esta ordenación ascendente depende que la creación y el despliegue en pantalla de los reportes creados se haga también en orden ascendente cada vez que se selecciona la opción que permite crear cualquier tipo de reporte para todos los sistema fotovoltaicos dados de alta.

Se verificó también que el usuario no diera de baja todos los sistemas. Por lo menos uno de ellos debía estar en el archivo *ALTA.DAT*.

5.2.4 FUNCIONES PARA CREAR LOS REPORTES DIARIOS

Cada una de las funciones para generar los reportes diarios se probó en forma individual y fuera del ambiente de FIX DMACS, simulando el enlace y obtención de valores de la base de datos *FOT*; esto es, se creó un archivo de datos de FIX DMACS en forma manual (hay que recordar que el archivo siempre se genera con el mismo nombre, independientemente de la fecha en que fueron guardados los datos que contiene). Con ese archivo se probaron en forma inicial todas las funciones del generador de reportes diarios.

5.2.4.1 GRAFICA

Cuando la adquisición de datos provenientes del monitoreo de cualquiera de los sistemas fotovoltaicos se realiza sin interrupciones, la gráfica se genera completa. Sin embargo, si hubo interrupciones (del usuario o de energía eléctrica) durante ese día, había que hacer una validación de datos para evitar una alteración en la gráfica que se crea.

Esta validación se hizo graficando el valor anterior en lugar del valor perdido, siempre y cuando la hora u horas de los datos perdidos se encontrara dentro de cierto rango. El rango se determinó observando los datos de días completos para ese mismo sistema fotovoltaico. Este rango es diferente para cada grupo de datos a graficar.

Por ejemplo, el voltaje en el pánel fotovoltaico es casi nulo de las 00:00:00 a las 05:30:00 horas, el valor del mismo se va incrementando conforme va amaneciendo y tiende a ser nulo nuevamente de las 18:30:00 a las 23:55:00 horas. Si hubiera una interrupción eléctrica entre las 00:00:00 y las 05:30:00 horas, la gráfica no se afecta si se sustituye el valor perdido por cero, sin embargo, si la interrupción ocurre después de las 05:30:00 y antes de las 18:30, la gráfica sí se altera si el dato o dato(s) perdidos se sustituye(n) por cero, por lo que debe(n) sustituirse por el valor anterior a la falla.

El comportamiento del voltaje en batería es diferente al del voltaje en el pánel fotovoltaico. Debido a que las baterías no pueden descargarse totalmente, el voltaje en las mismas nunca será cero. Si hubiera una pérdida de datos a cualquier hora del día, el valor perdido debe sustituirse por el anterior, no puede sustituirse por cero a ninguna hora.

Para probar que la generación de gráficas diarias es correcta, se crearon las gráficas para los días 28 y 30 del mes de abril de 1995 del sistema fotovoltaico 1 (EDO. MEX.). En el día 30, la generación de datos es completa y en día 28 hubo datos perdidos en los siguientes horarios: 16:30, 16:35, 16:45 y 17:05 horas.

Las cantidades de radiación, voltaje y corriente tanto en el panel fotovoltaico como en las baterías son muy similares. La validación es correcta. Estas gráficas se muestran a continuación.

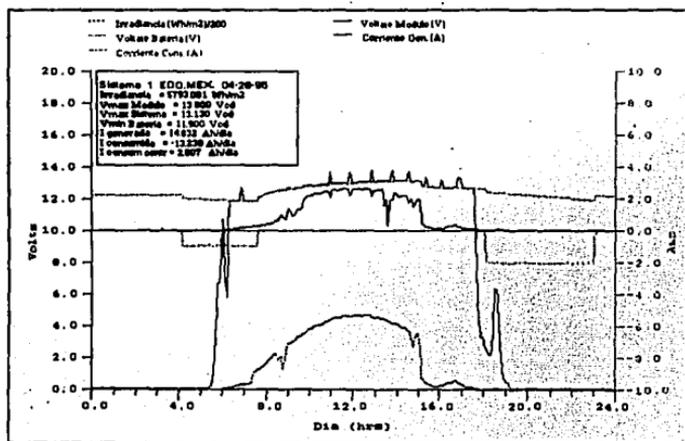


Figura 5.1 Gráfica generada sin validación de datos (con valores completos)

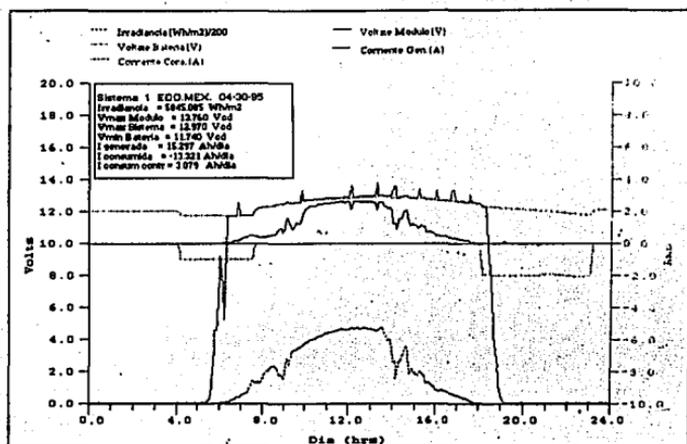


Figura 5.2 Gráfica generada validando datos perdidos por interrupciones de usuario o falla de energía eléctrica

La gráfica generada lleva un pequeño recuadro que incluye los valores que se especificaron en el inciso A) de la sección 4.2.2.3 del capítulo 4. Si hubo datos perdidos en el archivo correspondiente, la validación de éstos se realiza de la misma forma que para la gráfica.

5.2.4.2 TABLA RESUMEN

Como sabemos, la tabla resumen es la que está contenida en el recuadro de la gráfica diaria. El procedimiento de validación de datos es el mismo que se emplea en la gráfica.

Se crearon las tablas resumen diarias del primero al treinta de abril del año en curso, para los sistemas fotovoltaicos del 1 al 4 dados de alta como EDO.MEX., NAYARIT, OAXACA y CHIAPAS².

Para corroborar si la validación de datos en la tabla resumen es correcta se creó el archivo de datos para el día 28 de abril del año en curso (en este día hubo interrupciones en la adquisición de datos por parte del usuario). También se generó un programa independiente para realizar los cálculos de la tabla resumen con el mismo archivo de datos, pero validando manualmente los datos perdidos con un editor de texto (la validación se hizo utilizando los mismos criterios de programación).

Los valores calculados manualmente y con la opción TABLA RESUMEN, son muy aproximados. Esto se puede observar en la tabla y en el listado generado por el programa auxiliar que se muestran a continuación:

■ **Listado del programa auxiliar:**

Irradiancia	5793.0807	Wh/m2
Volteje máximo del Módulo	13.8000	Vcd
Volteje máximo del Sistema	13.1300	Vcd
Volteje mínimo de Batería	11.9000	Vcd
Corriente generada	14.8324	Ah/día
Corriente consumida	-13.2448	Ah/día
Corriente consumida por controlador	2.8000	Ah/día

■ **Listado del generador de tablas resumen diarias:**

.....
 TABLA RESUMEN DIARIA
 CORRESPONDIENTE AL SISTEMA 1 EDO.MEX
 CON FECHA 04-28-95

VARIABLE	CANTIDAD	UNIDAD
Irradiancia	5793.081	Wh/m2
Volteje máximo del Módulo	13.800	Vcd
Volteje máximo del Sistema	13.130	Vcd
Volteje mínimo de Batería	11.900	Vcd
Corriente generada	14.832	Ah/día
Corriente consumida	-13.238	Ah/día
Corriente consumida por controlador	2.807	Ah/día

² Este nombre se designa de acuerdo con el lugar donde serán instalados los sistemas fotovoltaicos.

Además de verificar la correcta validación de datos en la tabla resumen, se verificó que los archivos de salida cumplieran con las especificaciones de diseño: nombre del archivo de salida y extensión, así como facilidad de impresión y archivado de los mismos.

5.2.4.3 TABLA DE FALLAS

Se crearon las tablas de fallas diarias del primero al treinta de abril del año en curso, para los sistemas fotovoltaicos del 1 al 4 dados de alta como EDO.MEX., NAYARIT, OAXACA y CHIAPAS, respectivamente.

De las tablas de fallas generadas con el programa *REPORTES* se identificaron las que habían detectado datos perdidos.

Para corroborar que el contenido de estas tablas es real se crearon los archivos de datos de esos días. Se hizo una comparación visual entre el archivo de datos y las tablas de fallas generadas. En ambos archivos se compararon las horas en que había pérdida de datos. Se hizo el cálculo del porcentaje de datos perdidos en forma manual y se comparó con el registrado en las tablas de fallas. Los resultados de estas comprobaciones fueron los esperados.

Si no se registraban valores perdidos durante un día específico, el programa *REPORTES* creó correctamente el archivo de salida para identificar que no se habían producido pérdidas durante el día que especificó el usuario.

A continuación se muestra el contenido de los archivos 28AB951.ERR y 30AB951.ERR. El primero muestra los valores que se perdieron en el día 28 de abril para el sistema fotovoltaico 1 y el segunda muestra que no hubo pérdidas de datos en el mismo sistema pero para el día 30 de abril.

.....

TABLA DE FALLAS DIARIA
CORRESPONDIENTE AL SISTEMA 1 EDO.MEX
CON FECHA 04-28-85

.....

HORA	VFV	IFV	VBB	IC	RSP	IBB
16:30:00	????????	????????	????????	????????	????????	????????
16:35:00	????????	????????	????????	????????	????????	????????
16:45:00	????????	????????	????????	????????	????????	????????
17:05:00	????????	????????	????????	????????	????????	????????

.....
TABLA DE FALLAS DIARIA
NO SE ENCONTRARON FALLAS EN EL SISTEMA 1 EDO.MEX
CON FECHA 04-30-95
.....

La asignación del nombre para cada uno de los archivos de fallas creados fue también correcta.

5.2.4.4 LISTADO DE DATOS

La creación del archivo que contiene el listado de datos diarios de cada uno de los sistemas fotovoltaicos es la tarea más importante del generador de reportes, ya que de la correcta generación de esos datos depende la generación de todos los demás reportes (excepto el archivo mensual).

Se verificó que los datos obtenidos cada 5 minutos mediante los formatos creados con el programa *Report Creator* sean los mismos que el usuario obtenía en forma manual con el programa *Historical Trend Display*. Para realizar dicha verificación se crearon los archivos de datos con el programa *Historical Trend Display* para el día 28 de abril y se creó el archivo de datos por medio de esta opción del programa *REPORTES* con la misma fecha. Se compararon cada uno de los valores en forma visual y se observó que los valores obtenidos con los formatos generados con el programa *Report Creator* son los mismos que el usuario obtiene con el programa *Historical Trend Display*.

La asignación del nombre para el archivo de salida que el programador define para esta opción se hizo en forma correcta.

A continuación se muestra un fragmento del archivo creado en forma manual con el programa *Historical Trend Display* y del archivo creado mediante la opción LISTADO del menú para reportes diarios del programa *REPORTES*. Ambos listados de las 07:00:00 a las 09:00:00 horas, para comparación de datos.

Capítulo 5

Pruebas e implementación

08:40:00,	12.600,0,	12.581,0,	0.980,0,	-0.000,0,	0.815,0,	37.48,0,
24.23,0,	427.7,0,					
08:45:00,	12.600,0,	12.583,0,	0.882,0,	-0.000,0,	0.716,0,	38.81,0,
24.25,0,	267.1,0,					
08:50:00,	12.570,0,	12.555,0,	0.733,0,	-0.000,0,	0.566,0,	38.19,0,
24.27,0,	390.9,0,					
08:55:00,	12.612,0,	12.593,0,	0.999,0,	-0.000,0,	0.834,0,	38.50,0,
24.33,0,	512.1,0,					
09:00:00,	12.690,0,	12.664,0,	1.400,0,	-0.000,0,	1.237,0,	39.44,0,
24.38,0,	584.0,0,					

■ Archivo de datos obtenido mediante la opción LISTADO del generador de reportes diarios:

.....
TABLA DE DATOS DIARIA
CORRESPONDIENTE AL SISTEMA 1 EDO.MEX
CON FECHA 04-28-95

HORA	VFV	IFV	VBB	IC	RSP	IBB
07:00:00	11.90	0.20	11.90	-0.96	61.57	-0.93
07:05:00	11.90	0.22	11.90	-0.96	66.03	-0.91
07:10:00	11.90	0.23	11.90	-0.97	70.61	-0.90
07:15:00	11.90	0.25	11.90	-0.97	75.45	-0.88
07:20:00	11.90	0.27	11.90	-0.95	154.89	-0.86
07:25:00	11.90	0.28	11.90	-0.95	211.79	-0.84
07:30:00	11.90	0.30	11.90	-0.96	232.98	-0.83
07:35:00	12.10	0.31	12.10	-0.08	253.19	0.07
07:40:00	12.10	0.31	12.10	-0.08	253.19	0.07
07:45:00	12.20	0.33	12.20	-0.00	271.58	0.16
07:50:00	12.25	0.35	12.24	-0.00	280.43	0.18
07:55:00	12.29	0.37	12.28	-0.00	313.54	0.20
08:00:00	12.32	0.40	12.31	-0.00	335.14	0.23
08:05:00	12.36	0.42	12.35	-0.00	353.80	0.25
08:10:00	12.38	0.42	12.37	-0.00	372.81	0.25
08:15:00	12.40	0.46	12.39	-0.00	400.15	0.30
08:20:00	12.44	0.52	12.43	-0.00	421.37	0.35
08:25:00	12.47	0.58	12.46	-0.00	441.77	0.41
08:30:00	12.51	0.68	12.49	-0.00	374.89	0.52
08:35:00	12.55	0.82	12.54	-0.00	384.83	0.66
08:40:00	12.60	0.88	12.58	-0.00	427.70	0.82
08:45:00	12.60	0.88	12.58	-0.00	267.08	0.72
08:50:00	12.57	0.73	12.56	-0.00	390.93	0.57
08:55:00	12.61	1.00	12.59	-0.00	512.13	0.83
09:00:00	12.69	1.40	12.66	-0.00	584.01	1.24

5.2.5 FUNCIONES PARA CREAR LOS REPORTES SEMANALES Y MENSUALES

Se agruparon estas 2 opciones debido a la similitud de funcionalidad y de diseño que existe entre ellas.

5.2.5.1 TABLA RESUMEN

Para crear la tabla resumen, tanto semanal como mensual, se crearon con el generador de reportes diarios, las tablas resumen para cada día del mes de abril de los cuatro sistemas que se están probando. Como en las pruebas anteriores, sólo se listan las tablas resumen semanal y mensual para el sistema fotovoltaico 1. La tabla resumen semanal corresponde a la primera semana de abril.

.....
 TABLA RESUMEN SEMANAL
 CORRESPONDIENTE AL SISTEMA 1 EDO.MEX.
 CON FECHA DEL 04-01-95 al 04-07-95

VARIABLE	CANTIDAD	UNIDAD	FECHA[MM-DD-AA]
Irradiancia	6449.012	Wh/m2	04-01-95
Voltaje máximo del Módulo	13.760	Vcd	04-07-95
Voltaje máximo del Sistema	13.020	Vcd	04-02-95
Voltaje mínimo de Batería	11.670	Vcd	04-06-95
Corriente generada	16.770	Ah/día	04-07-95
Corriente consumida	-14.898	Ah/día	04-02-95
Corriente consumida por controlador	3.101	Ah/día	04-01-95

.....
 TABLA RESUMEN MENSUAL
 CORRESPONDIENTE AL SISTEMA 1 EDO.MEX.
 CON FECHA DEL 04-01-95 al 04-28-95

VARIABLE	CANTIDAD	UNIDAD	FECHA[MM-DD-AA]
Irradiancia	7332.719	Wh/m2	04-09-95
Voltaje máximo del Módulo	17.590	Vcd	04-10-95
Voltaje máximo del Sistema	14.060	Vcd	04-10-95
Voltaje mínimo de Batería	11.670	Vcd	04-06-95
Corriente generada	19.145	Ah/día	04-09-95
Corriente consumida	-16.533	Ah/día	04-12-95
Corriente consumida por controlador	4.090	Ah/día	04-24-95

Los resultados obtenidos en estas tablas se comprobaron con las tablas resumen diarias y semanales, respectivamente.

El programa verificó la existencia de todos los archivos necesarios para generar estos reportes y asignó los nombres de los archivos de salida en la forma esperada.

5.2.5.2 TABLA DE FALLAS

Para crear la tabla de fallas, tanto semanal como mensual, se crearon con el generador de reportes diarios, las tablas de fallas para cada día del mes de abril de los cuatro sistemas que se están probando. Como en las pruebas anteriores, sólo se listan las tablas de fallas semanal y mensual para el sistema fotovoltaico 1. La tabla de fallas semanal corresponde a la primera semana de abril.

El porcentaje de datos perdidos se calculó en forma manual y con base en las tablas de fallas diarias y semanales, respectivamente. El porcentaje de pérdidas equivale al número de datos perdidos multiplicado por 100 y dividido entre el número total de datos que se generan en intervalos de 5 minutos durante una semana o cuatro semanas (según la clase de reporte que se quiera generar). El total de datos para una semana es 2016 y para cuatro semanas es 8064. Los porcentajes calculados se compararon con los obtenidos a través del programa *REPORTES*. La diferencia fue mínima.

.....

TABLA DE FALLAS SEMANAL
CORRESPONDIENTE AL SISTEMA 1 EDO.MEX.
CON FECHA DEL 04-01-95 al 04-07-95

.....

FECHA: 04-01-95
NO SE ENCONTRARON FALLAS

FECHA: 04-02-95
21:50:00 ?????????? ?????????? ?????????? ?????????? ?????????? ??????????
21:55:00 ?????????? ?????????? ?????????? ?????????? ?????????? ??????????
22:00:00 ?????????? ?????????? ?????????? ?????????? ?????????? ??????????
22:05:00 ?????????? ?????????? ?????????? ?????????? ?????????? ??????????

FECHA: 04-03-95
NO SE ENCONTRARON FALLAS

FECHA: 04-04-95
NO SE ENCONTRARON FALLAS

FECHA: 04-05-95
 10:30:00 ?????????? ?????????? ?????????? ?????????? ??????????
 10:35:00 ?????????? ?????????? ?????????? ?????????? ??????????
 10:40:00 ?????????? ?????????? ?????????? ?????????? ??????????
 10:45:00 ?????????? ?????????? ?????????? ?????????? ??????????
 10:50:00 ?????????? ?????????? ?????????? ?????????? ??????????
 10:55:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:00:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:05:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:10:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:15:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:20:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:25:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:30:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:35:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:40:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:45:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:50:00 ?????????? ?????????? ?????????? ?????????? ??????????
 11:55:00 ?????????? ?????????? ?????????? ?????????? ??????????

FECHA: 04-06-95
NO SE ENCONTRARON FALLAS

FECHA: 04-07-95
NO SE ENCONTRARON FALLAS

PORCENTAJE SEMANAL DE DATOS PERDIDOS: 1.05 %

.....
 TABLA DE FALLAS MENSUAL
 CORRESPONDIENTE AL SISTEMA 1 EDO.MEX.
 CON FECHA DEL 04-01-95 al 04-28-95

FECHA: 04-02-95
 21:50:00 ?????????? ?????????? ?????????? ?????????? ??????????
 21:55:00 ?????????? ?????????? ?????????? ?????????? ??????????
 22:00:00 ?????????? ?????????? ?????????? ?????????? ??????????
 22:05:00 ?????????? ?????????? ?????????? ?????????? ??????????

FECHA: 04-05-95
 10:30:00 ?????????? ?????????? ?????????? ?????????? ??????????
 10:35:00 ?????????? ?????????? ?????????? ?????????? ??????????
 10:40:00 ?????????? ?????????? ?????????? ?????????? ??????????

FECHA: 04-28-95
16:30:00 ?????????? ?????????? ?????????? ?????????? ??????????
16:35:00 ?????????? ?????????? ?????????? ?????????? ??????????
16:45:00 ?????????? ?????????? ?????????? ?????????? ??????????
17:05:00 ?????????? ?????????? ?????????? ?????????? ??????????

PORCENTAJE MENSUAL DE DATOS PERDIDOS: 0.61 %

Los resultados obtenidos en estas tablas se comprobaron con las tablas de fallas diarias y semanales, respectivamente.

El programa verificó la existencia de todos los archivos necesarios para generar las tablas de fallas semanales y mensuales y asignó los nombres de los archivos de salida en la forma esperada.

5.2.6 FUNCION PARA CREAR EL ARCHIVO MENSUAL

Esta función se incluye con las opciones para generar las tablas resumen y de fallas mensuales, sin embargo la generación del archivo mensual no depende de otros archivos creados con este generador de reportes.

Para crear este archivo, el programa accesa la base de datos de FIX DMACS por medio del formato cuyos valores de las variables se especifican en la sección 4.2.2.1.

La prueba a esta función consistió en verificar que se creara un archivo por cada día del mes de abril y que se fuera uniendo hasta formar el archivo mensual por sistema fotovoltaico. Esta prueba se realizó a principios del mes de mayo con la finalidad de tener los datos completos del mes de abril.

Los archivos de salida se generaron sin interferir el funcionamiento normal del programa FIX DMACS y con los nombres correctos.

5.2.7 FUNCION DE DESPLIEGUE EN PANTALLA DE LOS REPORTES GENERADOS

El programa realizó el despliegue de los reportes creados de la siguiente manera:

- Si el reporte generado se deriva de uno de los cuatro sistemas fotovoltaicos de los cuales se están adquiriendo datos, el programa permite ver al usuario dicho reporte desplegado en pantalla y espera que éste presione cualquier tecla para activar el siguiente menú para imprimir o archivar el reporte que se desplegó.

- Si los reportes generados se derivan de los cuatro sistemas fotovoltaicos de los cuales se están adquiriendo datos, el programa genera y despliega uno por uno de los cuatro reportes en forma continua (sin esperar a que el usuario pulse tecla alguna para continuar) y activa el siguiente menú para imprimir y/o archivar los reportes desplegados.

Si el reporte seleccionado es el archivo mensual, el programa no despliega los valores del reporte ya que el archivo es muy grande.

Las pruebas de esta función se realizaron en todas las opciones de generación de reportes diarios, semanales y mensuales para el mes de abril, obteniéndose los resultados anteriores.

5.2.8 FUNCIONES DE IMPRESIÓN Y ARCHIVADO DE LOS REPORTES GENERADOS

Estas funciones se probaron antes de implementarse dentro de la interfaz de usuario del programa *REPORTES*. Para lo cual se simuló el archivado en tres unidades de disco (A, B y C) y envió a impresión de algunas gráficas, tabla resumen y tablas de fallas creadas por los programas *GRAFICA.EXE* y *DIARIOS.EXE*. Dichas pruebas se realizaron también una vez implementadas estas funciones dentro del generador de reportes de sistemas fotovoltaicos y en el ambiente *FIX DMACS*. El funcionamiento normal del programa *FIX DMACS* no se vio afectado al realizarse dichas pruebas.

5.2.9 FUNCIONES DE ALARMA

Los mensajes de alarma se activan cuando se dan determinadas situaciones de error, ya sea producidas por el programa mismo, por el usuario o por el programa *FIX DMACS*. Todas estas situaciones se simularon antes de implementar el programa *REPORTES* dentro del ambiente *FIX DMACS*. Las situaciones más importantes se listan a continuación.

- Si el programa no encuentra el archivo *ALTA.DAT* avisa al usuario lo que sucede y termina automáticamente su ejecución, ya que sin este archivo el generador de reportes para sistemas fotovoltaicos no funciona.
- Si el usuario elige la opción *ANUALES* el programa despliega el mensaje que indica que la función no se encuentra implementada.
- Si el usuario pretende generar algún tipo de reporte de un sistema que no se encuentra dado de alta, el programa envía al usuario un mensaje que así lo indique y verifica que no se active ninguna función hasta que el usuario elija un sistema que si se encuentre dado de alta.
- Si el usuario teclea valores de fecha no válidos en la ventana correspondiente, el programa despliega un mensaje que indica que la fecha no es válida, hasta que el usuario introduzca una fecha real.
- Si el usuario introduce caracteres no válidos (espacios en blanco, tabuladores) en la ventana que obtiene el identificador de los sistemas fotovoltaicos, el programa no acepta esos caracteres hasta que se introduzcan letras del alfabeto, puntos o dígitos.

Estos cuatro factores son importantes para que el generador de reportes de sistemas fotovoltaicos funcione adecuadamente.

Existen otras condiciones de alarma que son secundarias, que también se probaron, pero que por su grado de importancia no se listan.

La prueba final consistió en probar todas las funciones del programa *REPORTES* en forma integrada y dentro del ambiente del programa *FIX DMACS*. Este resultado es satisfactorio ya que fue posible realizar todas las funciones de dicho programa sin interferir el funcionamiento normal del programa *FIX DMACS*.

5.3 REQUERIMIENTOS DE SOFTWARE Y HARDWARE PARA IMPLEMENTAR EL PROGRAMA REPORTES

5.3.1 REQUERIMIENTOS DE SOFTWARE

- Sistema operativo MS DOS versión 5.0 o versión 6.2

- Programa multitareas FIX DMACS versión 3.0 con los siguientes módulos instalados:
 - *System Config Utility (SCU)*
 - *REPORT* con sus dos programas *Report Creator* y *Report Generator*
 - *HISTORICAL TRENDING* con dos de sus programas: *Historical Trend Assigment* e *Historical Trend Collection*

5.3.2 REQUERIMIENTOS DE *HARDWARE*

- Computadora personal 386 ó 486 compatible con IBM
- 2 Mb en RAM (como mínimo)
- Coprocesador matemático
- Dos unidades de disco flexible (3,5 y 5,25 pulgadas)
- Disco duro con 20 Mb libres (como mínimo)
- Un puerto paralelo
- Monitor a color EGA o VGA
- Un *ratón* de dos botones

El resultado de las pruebas de funcionalidad y estabilidad del programa *REPORTES* se considera aceptable, es por ello que dicho programa ha quedado instalado en forma permanente dentro del *SCU*, como un módulo más del programa FIX DMACS.

CAPITULO 6

Guía de usuario

6.1 QUE ES UNA GUIA DE USUARIO

El manual o guía de usuario es un instructivo para instalar, operar y mantener el sistema o paquete de programación[GER84]. Este manual deberá ser presentado en un sólo documento o como un conjunto de guías que tratan en forma independiente los aspectos básicos de instalación, operación y mantenimiento del sistema.

Dentro del manual de usuario se deberán incluir diversos aspectos tales como:

- El equipo y la programación necesarias para que el paquete de programación funcione adecuadamente.
- El procedimiento de instalación y la descripción de las pruebas que permitan constatar la correcta instalación.
- Un panorama general que permita al usuario conocer las funciones que dicho sistema pueda realizar.

- La descripción de la interfaz de usuario que incluya todos los medios existentes para la comunicación bidireccional entre el sistema de programación y el usuario (desplegados, reportes, mensajes, alarmas, etc.).
- Las recomendaciones de uso que ilustren al usuario la manera de utilizar las operaciones básicas del sistema.
- Un glosario de términos que defina el significado de palabras y claves utilizadas en la operación del sistema en términos que el usuario debe comprender.
- La guía de mantenimiento que defina la manera como el sistema fue realizado y que haga referencia a los documentos de desarrollo de programación que deben ser consultados cuando se deseen hacer correcciones, extensiones o adaptaciones.

6.2 DESCRIPCIÓN DE LOS PROGRAMAS COMPONENTES DEL MÓDULO REPORTES

Este módulo consta de diez archivos principales seis de los cuales son los programas fuente del generador de reportes; tres, los archivos ejecutables y el último, un archivo de datos que contiene los números y nombres de los sistemas fotovoltaicos que se están monitoreando.

PRINCIPA.C.- Archivo principal que contiene el código fuente para la validación de pantalla por medio de *ratón* (del inglés *mouse*), la definición de la ventana principal, las rutinas para enlazar los programas que generan a los reportes diarios y las rutinas para archivar e imprimir todos los reportes creados.

VENTANAS.C.- Archivo que contiene la definición de todas las ventanas utilizadas en el módulo de reportes.

SEMYMEN.C.- Archivo que contiene el código para generar las tablas resumen y de fallas, semanales y mensuales.

MENSUAL.C.- Archivo que contiene el código fuente para generar el archivo de datos mensual.

GRAFICA.C.- Archivo que contiene el código fuente para generar la gráfica diaria con los datos históricos que accesa el generador de reportes del programa FIX DMACS.

DIARIOS.C.- Archivo que contiene el código fuente para generar la tabla resumen, la tabla de fallas y el archivo de datos diarios del generador de reportes.

Los primeros cuatro programas se compilaron en BorlandC versión 3.0, generando el archivo ejecutable **REPORTES.EXE** que enlaza los archivos ejecutables **GRAFICA.EXE** y **DIARIOS.EXE**, resultado de la compilación en QuikC versión 2.5 de los últimos dos archivos fuente descritos anteriormente.

El archivo de datos que guarda la identificación de los sistemas fotovoltaicos tiene por nombre **ALTA.DAT**.

Los archivos fuente, así como **GRAFICA.EXE** y **DIARIOS.EXE** se encuentran en el directorio C:\DMACS\REPORTES. **REPORTES.EXE** y **ALTA.DAT** están instalados en C:\DMACS que es el directorio principal de FIX DMACS y donde se encuentran los archivos ejecutables del mismo.

En el directorio C:\DMACS\REPORTES se encuentran también **COURB.FON**, **HELVB.FON**, **MODERN.FON**, **ROMAN.FON**, **SCRIPT.FON**, Y **TMSRB.FON**, que son los archivos que definen los tipos de letra que se utilizan dentro de la gráfica que se genera en la opción **DIARIOS** del generador de reportes.

Además de estos archivos, se crearon veinte pequeños archivos que corresponden al formato que permite obtener los valores de las variables que se especificaron en la parte de requerimientos del sistema y en el Apéndice A. Dichos archivos se encuentran en el directorio C:\DMACS\REP y cuyo nombre se especificó con las siglas:

GRPAN[Número_sistema].GEN y MENS[Número_sistema].GEN

donde *Número_sistema* varía del 1 al 10, de acuerdo al sistema que se seleccione en el generador de reportes. El prefijo "GRPAN" se utilizó para los archivos que obtendrán los datos requeridos diariamente y el prefijo "MENS" para obtener los datos para la opción ARCHIVO MENSUAL.

6.3 DESCRIPCIÓN DEL FUNCIONAMIENTO DEL MÓDULO DE REPORTES

En esta parte se describe cómo funciona cada parte del generador de reportes.

6.3.1 COMO ENTRAR, COMO SELECCIONAR Y COMO SALIR DEL MÓDULO REPORTES

Para ejecutar este módulo es necesario seleccionar la opción **REPORTES**, ya sea con las teclas de movimiento del cursor o con el cursor del *ratón*. Esta selección se muestra en la figura 6.1.

Una vez seleccionada esta opción se despliega la pantalla de la figura 6.2.

```

MANAGER MENU 3 0

      FOREGROUND TASKS
PVVIEW - PIXEL OPERATOR DISPLAY
PDRAM - PIXEL DISPLAY BUILDER
PBC - DATABASE CREATE
PDB - DATABASE BUILDER
DUG - DOS WINDOW
DUI - ALARM SUMMARY DISPLAY
DUE - TUN GROUP EDITOR
DHE - KEY BOARD EDITOR
SGU - SYSTEM CONFIG UTILITY
LOG - SECURITY LOGIN
HTA - HISTORICAL ASSIGNMENT
HTD - HISTORICAL DISPLAY
RCP - EDITOR DE RECETAS
SC - PROGRAMACION M-MARIA
REP - REPORT CREATOR

      BACKGROUND TASKS
*SA4 - SAS FOR EMS 4.0
*NET - NET POLLING TASK
*SCH - SCHEDULER INTELEC
ALH - ALARM SERVICES
*HTC - HISTORICAL COLLECTION
SCHS - SCHEDULER
  
```

Figura 6.1 Selección del programa **REPORTES**

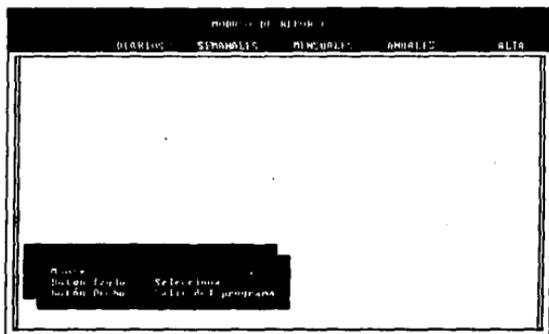


Figura 6.2 Pantalla inicial del módulo *REPORTES*

Esta pantalla inicial muestra dos indicaciones para manejar la interfaz del programa *REPORTES*:

- a) Botón izquierdo del *ratón*. Este botón permite seleccionar una de las opciones que están dentro de esta interfaz, siempre y cuando se posicione el cursor del *ratón* encima de la opción deseada y se presione este botón izquierdo.
- b) Botón derecho del *ratón*. Si se presiona este botón se desplegará una ventana que pregunta si se desea salir del programa, esta ventana se muestra a continuación:

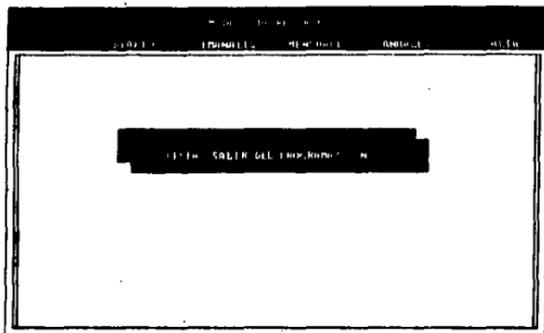


Figura 6.3 Ventana de salida para el módulo *REPORTES*

El programa mostrará esta ventana cada vez el cursor del *ratón* se encuentre activado y que se presione el botón derecho del *ratón*.

Si se encuentra activado el cursor normal de pantalla y desactivado el cursor del *ratón*, no es posible salir del programa. Este caso sólo se presenta cuando se despliegan ventanas en las cuales sea necesario introducir datos por teclado.

Para salir del programa se pulsa la tecla "s" o "S". Si el usuario no desea salir del programa se pulsa la tecla "n" o "N". En esta ventana de salida, sólo se aceptan las teclas "s" o "n" y sus respectivas mayúsculas. Si se presiona cualquier otra tecla, no se realizará ninguna acción.

6.3.2 COMO GENERAR REPORTES

Una vez que se ha seleccionado REPORTES del menú principal de FIX DMACS, se puede acceder a cualquiera de las opciones que se encuentran dentro de la franja de color azul claro en la parte superior de la interfaz del programa *REPORTES*. Dichas opciones son las siguientes:

DIARIOS.- Esta opción permite generar reportes diarios

SEMANALES.- Esta opción permite generar reportes semanales

MENSUALES.- Esta opción permite generar reportes mensuales

ANUALES.- Esta opción permite generar reportes anuales¹

ALTA.- Esta opción permite dar de alta a los sistemas fotovoltaicos de los cuales se desean generar reportes. Dar de alta significa guardar el sistema seleccionado en el archivo *ALTA.DAT* que se explicará en la parte correspondiente a esta opción.

6.3.2.1 VENTANA DE SISTEMAS

Una vez que se ha seleccionado alguna de las tres primeras opciones (DIARIOS, SEMANALES, MENSUALES) se desplegará la *ventana de SISTEMAS*, que se muestra a continuación:

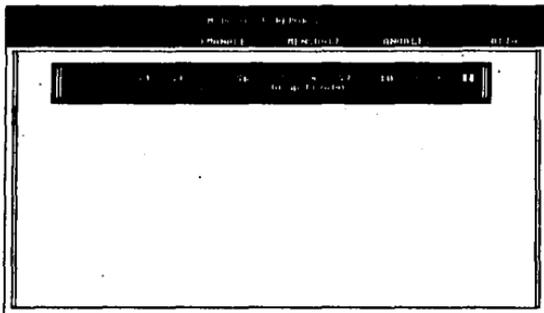


Figura 6.4. Ventana de SISTEMAS

Cada letra S identifica la palabra "sistema" y el número que le sigue corresponde al número del sistema fotovoltaico. Como se puede observar, el programa *REPORTES* está diseñado para diez sistemas fotovoltaicos que están

¹ Esta opción no se encuentra implementada dentro de la interfaz.

etiquetados como **S1**, **S2**,..., **S10**. Además de estas etiquetas aparecen a la derecha de esta ventana la palabra **TODOS** en color amarillo y dos cuadrillos de color azul.

Dentro de esta ventana se encuentran también las palabras **ACTIVADO** en color amarillo y **DESACTIVADO** en color blanco.

La palabra **ACTIVADO** indica que el identificador de sistema fotovoltaico en color amarillo es aquel que se encuentra dado de alta en el archivo *ALTA.DAT* y del cual es posible generar reportes.

Por el contrario, la palabra **DESACTIVADO** indica que el sistema que se encuentre en color blanco no está dado de alta dentro del archivo *ALTA.DAT* y, por lo tanto, no es posible generar reportes de él.

Si se selecciona con el cursor del *ratón* alguno de los sistemas en color amarillo se activará el menú que corresponde al tipo de reporte que se ha seleccionado.

Si se selecciona con el cursor del *ratón* alguno de los sistemas en color blanco se desplegará momentáneamente la ventana de la figura 6.5 que nos indica que el sistema no se encuentra dado de alta. Después de esta ventana se activa nuevamente la *ventana de SISTEMAS* que permite al usuario seleccionar otro sistema.

Si el cursor no se posiciona en ningún sistema o en la etiqueta **TODOS**, no se realizará ninguna acción y la *ventana de SISTEMAS* seguirá apareciendo hasta que se seleccione alguna opción o se seleccione uno de los dos cuadros pequeños de color azul que se encuentran a la derecha de la ventana. Estos cuadros de color azul permiten salir de esa ventana.

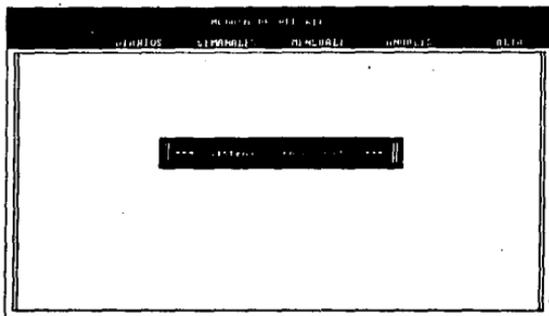


Figura 6.5 Ventana que indica que el sistema no se encuentra dado de alta

La opción **TODOS** permite generar reportes de todos los sistemas que estén dados de alta, es decir de todos los sistemas que se encuentran en color amarillo.

6.3.2.2 COMO GENERAR REPORTES DIARIOS

Para generar este tipo de reportes se posiciona el cursor del *ratón* en la opción **DIARIOS** y se selecciona presionando el botón izquierdo del *ratón*. Una vez realizada esta acción, la opción **DIARIOS** se tornará de color azul marino y aparecerá la *ventana de SISTEMAS*. Seleccionando un sistema que esté activado o la etiqueta **TODOS** se desplegará la ventana que contiene el *menú DIARIOS* para generar reportes diarios. Este menú se muestra en la figura 6.6.



Figura 6.7 Ventana de FECHA

Esta ventana permite introducir la fecha de la cual se desea el reporte. Como se puede observar, en ella se despliegan tres campos: **DIA[DD]**, **MES[MM]** y **AÑO[AA]**. Las letras que aparecen dentro de los corchetes indican que sólo es posible introducir dos dígitos por cada campo.

El campo DIA[DD] sólo acepta valores del 01 al 31, el campo MES[MM] sólo acepta valores del 01 al 12 y el campo AÑO[AA] sólo acepta valores del 95 al 99. Por cualquier valor o fecha no válidos que se introduzcan, se desplegará un mensaje que indica que no son válidos.

Si el valor que se pretende introducir se encuentra entre 1 y 9, es necesario introducir como primer valor un cero y como segundo valor el número del 1 al 9 para que sea válido.

Después de introducir los dos dígitos por cada campo, se pulsa la tecla **<ENTER>** para indicar que ya es posible pasar al siguiente campo.

Una vez llenados todos los campos de la *ventana de FECHA*, la pantalla inicial del módulo *REPORTES* se tornará de color negro y aparecerán los siguientes mensajes:

REPORT GENERATOR 3.0**REPGEN: Report completed successfully.**

Estos mensajes indican que se ha accedido correctamente a la base de datos de FIX DMACS.

Posteriormente se desplegará en pantalla la gráfica, tabla o listado con la fecha que se especificó y el programa esperará a que el usuario pulse cualquier tecla para continuar con el proceso, si es que seleccionó un solo sistema de la *ventana de SISTEMAS*.

Si la selección hecha es la etiqueta **TODOS**, el despliegue en pantalla de los reportes generados se hará en forma continua y no será necesario presionar tecla alguna para continuar con el proceso.

NOTA IMPORTANTE: Si se seleccionó un solo sistema de la *ventana de SISTEMAS*, después de introducir el valor que corresponde a ARQ(AA) no se debe pulsar ninguna otra tecla después <ENTER>. Espere a que se despliegue la gráfica, tabla o listado para presionar cualquier tecla para continuar.

Si el o los archivo(s) de salida se genera(n) correctamente, aparecerá en la interfaz de usuario la *ventana de IMPRESIÓN y ARCHIVADO* para los reportes de salida. El funcionamiento de esta ventana se explica en la sección 6.3.3.

Si la generación de todos los archivos de salida no fue correcta, entonces se desplegará momentáneamente la *ventana de la figura 6.8* y después volverá a aparecer la ventana que contiene el *menú DIARIOS*.

Si se eligió la etiqueta **TODOS** de la *ventana de SISTEMAS*, y alguno de los archivos de salida no se genera, ya sea por que no se encuentre un archivo de datos o por algún problema del sistema, entonces se desplegará la *ventana de la figura 6.9*. Se debe introducir por teclado la respuesta "s" o "n", si la respuesta es "s" entonces se desplegará la *ventana de IMPRESION y ARCHIVADO* de la sección 6.3.3. Si la respuesta es "n", entonces el archivo o archivos generado(s) se borrarán automáticamente y aparecerá el *menú DIARIOS*.

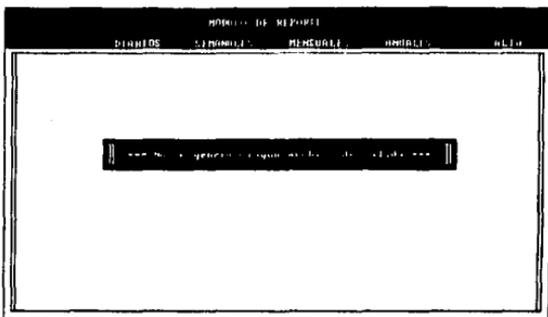


Figura 6.8 Ventana que indica que no se generó ningún archivo de salida

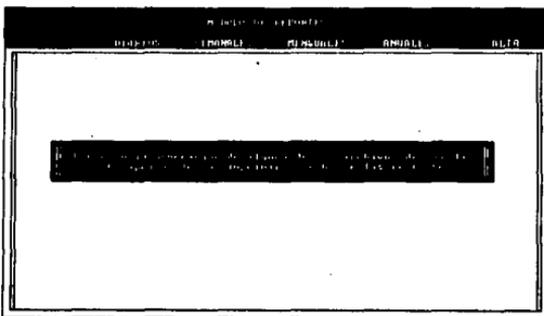


Figura 6.9 Ventana que indica la generación incompleta de los archivos de salida para la selección de la opción TODOS en la ventana de SISTEMAS

B) Ventana *OTRA FECHA* para los reportes diarios

Si se realiza una segunda selección del *menú DIARIOS* y no se ha salido de la ventana que lo contiene, aparecerá la ventana de la figura 6.10.

Esta ventana pregunta si se quiere generar el reporte que se acaba de seleccionar con la fecha anterior o con una fecha nueva.

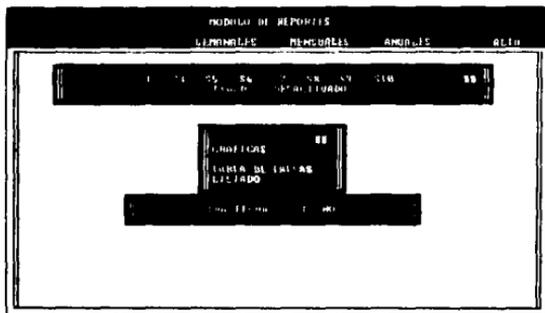


Figura 6.10 Ventana *OTRA FECHA*

Para seleccionar la respuesta posicione el cursor del *ratón* encima de **SI** o **NO** y presione el botón izquierdo.

Si la respuesta seleccionada es **SI**, entonces se desplegará la *ventana de FECHA* para reportes diarios que permitirá generar el reporte diario con otra fecha distinta a la seleccionada en el *menú DIARIOS* anteriormente.

Si la respuesta seleccionada es **NO**, entonces se procederá a generar el reporte que se eligió con la misma fecha del anterior.

NOTA: Cada vez que se salga del *menú DIARIOS* y se seleccione otro sistema o la etiqueta **TODOS**, aparecerá la *ventana de FECHA*.

6.3.2.3 COMO GENERAR REPORTES SEMANALES

Para generar este tipo de reportes se posiciona el cursor del *ratón* en la opción **SEMANALES** y se selecciona presionando el botón izquierdo del *ratón*. Una vez realizada esta acción, la opción **SEMANALES** se tornará de color azul marino y se activará la *ventana de SISTEMAS*. Seleccione un sistema que esté activado o la etiqueta **TODOS** y se desplegará la ventana que contiene el *menú SEMANALES* para generar reportes semanales. Este menú se muestra en la figura 6.11.

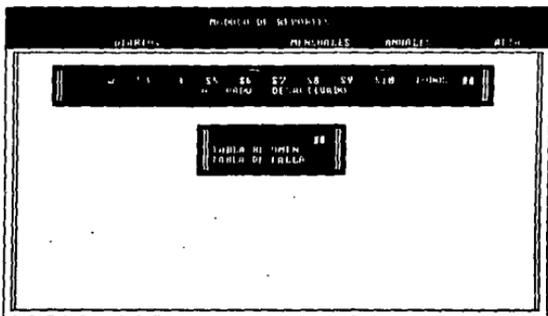


Figura 6.11 Menú **SEMANALES**

En este menú se muestran dos opciones, de las cuales sólo es posible seleccionar una a la vez. Estas opciones son

TABLA RESUMEN **TABLA DE FALLAS**

Además de estas opciones, en la parte superior derecha aparecen dos pequeños cuadros. Estos cuadros se utilizan para salir de esta ventana, si es que no se desea seleccionar alguna opción del menú.

Una vez seleccionada alguna de las opciones *por primera vez*, se activará la *ventana de FECHA*, que se explicó en el inciso A) de la sección 6.3.2.2, sólo que para reportes semanales.

Si se realiza una nueva selección sin salir de la ventana del *menú SEMANALES*, aparecerá la *ventana de OTRA FECHA*, que se explicó en el inciso B) de la sección 6.3.2.2, sólo que esta ventana sirve ahora para reportes semanales.

Para generar estos dos reportes semanales, es necesario que se hayan generado previamente, con el generador de reportes diarios, los reportes diarios TABLA RESUMEN Y TABLA DE FALLAS de una semana (siete días). El día de inicio de la semana corresponde a la fecha que se introduce en la *ventana de FECHA*. Este día de inicio puede ser cualquier día de la semana.

El generador de reportes semanal verifica que existan los siete archivos diarios a partir del día que el usuario introduce. Dichos archivos tienen las extensiones .TAB y .ERR, respectivamente y deben estar almacenados en el directorio C:\DMACS\REP².

Si todos los archivos necesarios se encuentran en ese directorio, entonces se procederá a generar y a desplegar en pantalla la tabla resumen o la tabla de fallas semanal.

Si sólo se eligió un sistema se desplegará la tabla en pantalla y en un fondo color negro. Después de desplegar dicha tabla, el generador de reportes espera que se pulse cualquier tecla para continuar y regresar a la interfaz de usuario para desplegar la *ventana de IMPRESION y ARCHIVADO* que se explica en la sección 6.3.3.

Si se eligió la etiqueta **TODOS** entonces se desplegarán las tablas de todos los sistemas fotovoltaicos dados de alta en forma continua, sin necesidad de presionar alguna tecla. Y después aparecerá la *ventana de IMPRESION y ARCHIVADO* que se explica en la sección 6.3.3.

² Estos archivos son almacenados *por default* en ese directorio de la unidad de disco duro si se elige la unidad C de la *ventana de UNIDADES*.

Si no existen todos los archivos necesarios para generar dicha tabla, entonces se desplegará un mensaje que indica el o los nombre(s) del(os) archivo(s) que no se encuentra(n) en el directorio C:\DMACS\REP.

Por ejemplo, si se introduce la fecha 26 de febrero de 1996 para la opción TABLA RESUMEN semanal del sistema 1, el generador de reportes buscará que existan los archivos 26FE961.TAB, 27FE961.TAB, 28FE961.TAB, 01MZ961.TAB, 02MZ961.TAB, 03MZ961.TAB, 04MZ961.TAB. Si todos estos archivos se encuentran, el sistema procederá a elaborar la tabla que estará contenida en el archivo 26FE961.SET. Si no se encontrara el archivo 01MZ961.TAB, se desplegará el mensaje:

*** No existe el archivo: 01MZ961.TAB ***

Presione cualquier tecla para continuar

Si la generación de todos los archivos de salida no fue correcta, entonces aparecerá momentáneamente la ventana de la figura 6.8 y después volverá a desplegarse la ventana que contiene el *menú SEMANALES*.

Si se eligió la etiqueta **TODOS** de la *ventana de SISTEMAS*, y alguno de los archivos de salida no se genera, ya sea por que no se encuentre un archivo o por algún problema del sistema, entonces se desplegará la ventana de la figura 6.9 que espera una respuesta afirmativa, por parte del usuario, para desplegar la *ventana de IMPRESION* y *ARCHIVADO* que se explica en la sección 6.3.3 y una respuesta negativa para borrar automáticamente el o los archivo(s) generado(s) y regresar a la ventana que contiene el *menú SEMANALES*.

6.3.2.4 COMO GENERAR REPORTES MENSUALES

Para generar este tipo de reportes, se posiciona el cursor del *ratón* en la opción **MENSUALES** y se selecciona presionando el botón izquierdo del *ratón*. Una vez realizada esta acción, la opción **MENSUALES** se tornará de color azul marino y aparecerá la *ventana de SISTEMAS*. Para desplegar la ventana que contiene el *menú MENSUALES* para generar reportes mensuales, seleccione un sistema que esté activado o la etiqueta **TODOS**. Este menú se muestra en la figura 6.12.

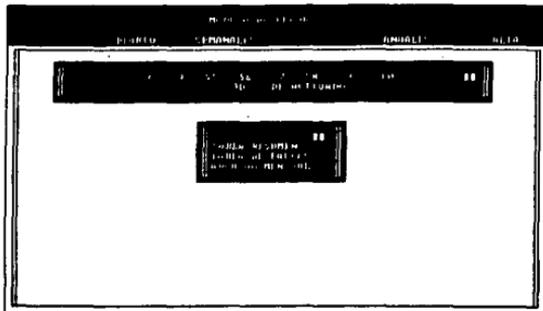


Figura 6.12 Menú MENSUALES

En este menú se muestran tres opciones, de las cuales sólo es posible seleccionar una a la vez. Estas opciones son

TABLA RESUMEN
TABLA DE FALLAS
ARCHIVO MENSUAL

Además de estas opciones, en la parte superior derecha aparecen dos pequeños cuadros. Estos cuadros se utilizan para salir de esta ventana, si es que no se desea seleccionar alguna opción del menú.

Una vez seleccionada alguna de las dos primeras opciones: **TABLA RESUMEN** y **TABLA DE FALLAS**, aparecerá la *ventana de FECHA*, que se explicó en el inciso A) de la sección 6.3.2.2, sólo que para reportes mensuales.

Si se realiza una nueva selección sin salir de la ventana del *menú MENSUALES*, aparecerá la *ventana de OTRA FECHA*, que se explicó en el inciso B) de la sección 6.3.2.2, sólo que esta ventana sirve ahora para reportes mensuales.

Para generar estos dos reportes mensuales, es necesario que se hayan generado los reportes semanales TABLA RESUMEN o TABLA DE FALLAS de cuatro semanas. El día de inicio de reporte mensual corresponde a la fecha que se introduce en la *ventana de FECHA*. Este día de inicio no es precisamente el día lunes, puede ser cualquier día de la semana.

El generador de reportes mensual verifica que existan los cuatro archivos semanales a partir del día que el usuario introduce en la *ventana de FECHA*. Dichos archivos tienen las extensiones .SET y .SEE, respectivamente y deben estar almacenados en el directorio C:\DMACS\REP³. Cabe mencionar que los archivos semanales debieron ser generados en una sesión de reportes semanales por este mismo generador de reportes.

Si los archivos necesarios se encuentran en ese directorio, entonces se procederá a generar y a desplegar en pantalla la tabla resumen o la tabla de fallas mensual, de acuerdo con la selección realizada anteriormente.

Si sólo se eligió un sistema fotovoltaico, se desplegará la tabla en pantalla y en un fondo color negro. Después de desplegar dicha tabla, el generador de reportes espera que el usuario presione cualquier tecla para continuar y desplegar la *ventana de IMPRESION y ARCHIVADO* (ver sección 6.3.3).

Si se eligió la etiqueta **TODOS** entonces se desplegarán en pantalla y en forma continua las tablas de todos los sistemas fotovoltaicos dados de alta, sin necesidad de presionar alguna tecla. Y después aparecerá la *ventana de IMPRESION y ARCHIVADO* que se explica en la sección 6.3.3.

Si no existen todos los archivos necesarios para generar dicha tabla, entonces se desplegará un mensaje que indica el o los nombre(s) del(os) archivo(s) que no se encuentra(n) en el directorio C:\DMACS\REP.

Por ejemplo, si se introduce la fecha 26 de febrero de 1996 para la opción TABLA RESUMEN mensual del sistema 1, el generador de reportes buscará que existan los archivos 26FE961.SET, 05MZ961.SET, 12MZ961.SET y 19MZ961.SET, que corresponden a los reportes semanales del 26 de febrero al 4 de marzo, del 5 al 11 de marzo, del 12 al 18 de marzo y del 19 al 25 de

³ Estos archivos son almacenados por default en ese directorio de la unidad de disco duro si se elige la unidad C de la *ventana de UNIDADES*.

marzo de 1996, respectivamente.

Si todos estos archivos se encuentran, el sistema procederá a elaborar la tabla que estará contenida en el archivo 26FE961.MET. Si el sistema no encuentra alguno de los archivos que se indicaron anteriormente, por ejemplo el archivo 05MZ961.SET, se desplegará un mensaje como el siguiente:

*** No existe el archivo: 05MZ961.SET ***

Presione cualquier tecla para continuar

Si la generación de todos los archivos de salida no fue correcta, entonces aparecerá momentáneamente la ventana de la figura 6.8 y después volverá a aparecer la ventana que contiene el *menú MENSUALES*.

Si se eligió la etiqueta **TODOS** de la *ventana de SISTEMAS*, y alguno de los archivos de salida no se genera, ya sea por que no se encuentre un archivo o por algún problema del sistema, entonces se desplegará la ventana de la figura 6.9 que espera una respuesta afirmativa, por parte del usuario, para desplegar la *ventana de IMPRESION y ARCHIVADO* (ver sección 6.3.3) y una respuesta negativa para borrar automáticamente el o los archivo(s) generado(s) y regresar a la ventana que contiene el *menú MENSUALES*.

Para generar el **ARCHIVO MENSUAL** que es la tercera opción para los reportes mensuales, selecciónela posicionando el cursor del *ratón* sobre alguna letra de la opción y presione el botón izquierdo del mismo. Al realizar dicha selección, aparecerá la ventana de la figura 6.13. Esta ventana corresponde a la *ventana de FECHA para archivos mensuales*.



Figura 6.13 Ventana de FECHA para archivos mensuales

En esta ventana sólo se requiere el mes y el año para generar el archivo mensual, ya que el generador de reportes toma como día de inicio el día 01 del mes seleccionado.

Esta ventana consta de dos campos: **MES[MM]** y **AÑO[AA]**. El campo **MES[MM]** sólo acepta valores del 01 al 12 y el campo **AÑO[AA]** sólo acepta valores de 95 al 99. Para pasar del campo **MES[MM]** al campo **AÑO[AA]** pulse la tecla **<ENTER>** después de introducir los dos dígitos correspondientes al mes seleccionado. Después teclee los dos dígitos correspondientes al campo **AÑO[AA]** y pulse una vez más la tecla **<ENTER>**.

Una vez que se han llenado los dos campos de la *ventana de FECHA para archivo mensual*, se limpia la ventana del generador de reportes y aparecerán mensajes como el siguiente:

```
GEN /RMENS1 /D[Fecha] /F
REPGEN: Report completed sucessfully.
```

Este mensaje indica que se está generando un archivo de datos por cada día del mes. El primer mensaje tiene por *Fecha* la que corresponde al primer día del mes que se especificó y se va incrementado hasta llegar al día último. Por cada día del mes va a aparecer un mensaje como este.

Conforme se va generando cada archivo diario, éste se va concatenando hasta formar el archivo mensual.

Si no se produce ningún error en la generación de los archivos diarios para esta opción, se desplegará la *ventana de IMPRESION y ARCHIVADO* que se describe en la sección 6.3.3.

El nombre del archivo generado para esta opción corresponde al día primero del mes y año introducidos en la ventana correspondiente y al sistema seleccionado con la extensión .MEN. Así, si se quiere el reporte mensual del mes de febrero de 1996 para el sistema fotovoltaico 1, el archivo de salida tendrá el nombre O1FE961.MEN.

Si se produce algún error en la generación de estos archivos diarios aparecerá un mensaje como el siguiente:

No se puede abrir el archivo MENS1.RPT
Es posible que exista un error en el archivo MENS1.GEN
que se encuentra en el directorio C:\DMACS\REP.
Presiona cualquier tecla para continuar

Esto significa que es posible que se haya borrado o modificado el formato que se ha creado en FIX DMACS y que lleva por nombre MENS1. Para mayor información sobre como crear o modificar este formato consulte el Apéndice A.

Una vez que se ha generado este mensaje de error, el generador de reportes espera a que el usuario pulse una tecla para continuar y borrar la fracción del archivo mensual que se haya concatenado hasta ese momento. Posteriormente se desplegará la ventana que contiene el *menú MENSUALES*.

6.3.2.5 COMO GENERAR REPORTES ANUALES

Esta opción no se encuentra implementada dentro del generador de reportes, pero se contempló dentro de la interfaz del mismo para una implementación futura, por lo que cada vez que se intente seleccionar se desplegará una ventana con el mensaje:

***** Función no implementada *****

6.3.3 COMO IMPRIMIR Y COMO ARCHIVAR TODO TIPO DE REPORTES

Una vez que se ha generado y desplegado cualquier tipo de reporte o cada vez que el usuario ha aceptado imprimir o archivar los archivos restantes cuando se produce un error en la generación de alguno de los archivos de salida para la selección de **TODOS** los sistemas fotovoltaicos, se despliega lo que se ha denominado **ventana de IMPRESION y ARCHIVADO**. Esta ventana se muestra en la figura 6.14 (en este caso corresponde a la ventana para imprimir y archivar reportes diarios).

Antes de continuar es preciso que el usuario conozca lo siguiente. Cada vez que se genera un reporte (gráfica, tabla o archivo de datos), éste se guarda temporalmente en el directorio C:\DMACS\REP antes de que aparezca la ventana de la figura 6.14. Este archivo se guarda definitivamente en dicho directorio sólo si se selecciona la unidad C como unidad de almacenamiento de la **ventana de UNIDADES** que se despliega una vez que se ha elegido la opción ARCHIVAR. Si el usuario selecciona la unidad A o B, el archivo es borrado automáticamente del directorio C:\DMACS\REP una vez que se da por terminada la ejecución del módulo **REPORTES** o cuando el usuario elige salir de la **ventana de IMPRESION y ARCHIVADO**.

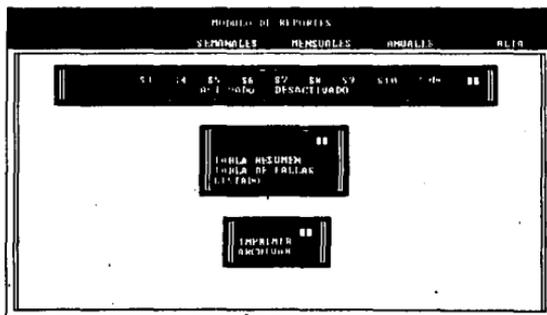


Figura 6.14 Ventana de IMPRESION y ARCHIVADO para los reportes diarios

El generador de reportes asigna un nombre predefinido por el programador a cada tipo de reporte. Dicho nombre se compone de doce letras y dígitos combinados: dos dígitos para representar el día (del 01 al 31 como máximo), dos letras para representar el mes (EN, FE, MZ, AB, MY, JN, JL, AG, SE, OC, NO, DI), dos dígitos para representar el año (95 al 99), uno o dos dígitos para representar el número de sistema fotovoltaico (1 al 10), un punto decimal para separar la extensión del nombre y tres letras como extensión. Esta extensión se asigna de acuerdo a la siguiente tabla:

Opciones Tipo de reporte	Extensión asignada al tipo de reporte		
	DIARIOS	SEMANALES	MENSUALES
GRAFICAS	.PLT		
TABLA RESUMEN	.TAB	.SET	.MET
TABLA DE FALLAS	.ERR	.SEE	.MEE
LISTADO	.DAT		
ARCHIVO MENSUAL			.MEN

En la *ventana de IMPRESION* y *ARCHIVADO*, se despliegan dos opciones:

IMPRIMIR
ARCHIVAR

Estas dos opciones se explican en las siguientes secciones.

6.3.3.1 COMO IMPRIMIR

Para imprimir el o los reporte(s) generado(s) por el módulo *REPORTES* seleccione con el *ratón* la opción **IMPRIMIR**. Antes de enviar el archivo de salida a impresora, esta opción despliega alguna de las dos ventanas que se muestran en las figuras 6.15 y 6.16.

El despliegue de estas ventanas depende del tipo de archivo a imprimir.

La ventana de la figura 6.15 se despliega cuando se va(n) a imprimir la(s) gráfica(s) generada(s) por la opción **GRAFICAS** del *menú DIARIOS*, ya que el generador de gráficas está configurado para enviar la gráfica únicamente a una impresora HP Laser Jet III o a una impresora HP Laser Jet 4 Plus, conectada al puerto paralelo **LPT1** de la computadora.

La ventana de la figura 6.16 se despliega cuando se va imprimir cualquier otro reporte diferente a una gráfica. La orden de impresión se puede enviar a cualquier impresora que se encuentre conectada en el puerto paralelo **LPT1** de la computadora donde está instalado el programa **FIX DMACS**.

Una vez que el usuario ha dado una respuesta afirmativa para continuar con la impresión o una respuesta negativa para no imprimir, el generador de reportes restaura la *ventana de IMPRESION* y *ARCHIVADO* para que envíe una nueva orden de impresión, para archivar el o los archivo(s) generado(s) o bien para salir de dicha ventana para seleccionar una nueva opción del menú que se encuentre desplegado.

Cuando el usuario da una respuesta afirmativa para imprimir el reporte generado (tabla, gráfica o archivo de datos), la pantalla principal se torna de color negro y aparece un mensaje como el siguiente:

Imprimiendo el archivo: *(Nombre del archivo)*...

Cuando la opción seleccionada de la *ventana de SISTEMAS* es **TODOS**, se desplegará este mensaje por cada archivo que se envíe a impresión.

Para salir de la *ventana de IMPRESION* y *ARCHIVADO* seleccione con el *ratón* los dos cuadros de color azul que se encuentran en la esquina superior derecha.

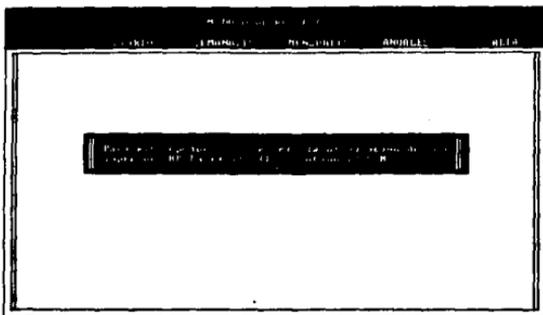


Figura 6.15 Aviso de impresión para una gráfica

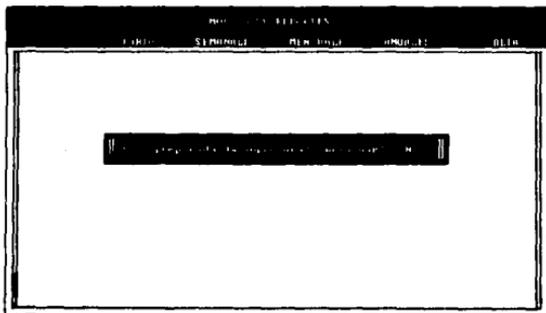


Figura 6.16 Aviso de impresión para cualquier tipo de reporte, excepto una gráfica

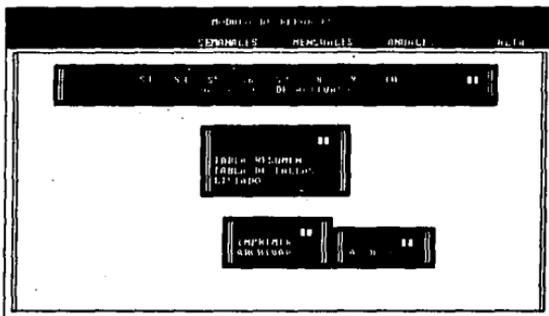


Figura 6.17 Ventana de UNIDADES

6.3.3.2 COMO ARCHIVAR

Para archivar el o los reporte(s) generado(s), seleccione con el *ratón* la opción **ARCHIVAR**. Una vez que se ha seleccionado esta opción se despliega la *ventana de UNIDADES*. Esta ventana se muestra en la figura 6.17.

Las unidades definidas para almacenar los reportes generados son A, B y C (directorio C:\DMACS\REP). Sólo es posible seleccionar una unidad a la vez. Una vez realizada la selección con el *ratón* de una de estas unidades, esta ventana se desactiva y el sistema realiza el grabado del archivo o archivos generado(s) en la unidad seleccionada. Una vez realizado el archivado aparece la *ventana de IMPRESION y ARCHIVADO* para realizar una nueva selección (IMPRIMIR o ARCHIVAR) o bien para salir de esa ventana.

Si el usuario decide archivar los reportes generados en las unidades A o B, la pantalla principal del generador de reportes se tornará de color negro y aparecerá un mensaje como el siguiente:

Copiando el archivo: [Nombre del archivo]...

Si se seleccionó la opción **TODOS** de la *ventana de SISTEMAS*, aparecerá un mensaje como el anterior por cada archivo que se guarde en la unidad de disco flexible (A o B) seleccionada.

Si la elección de usuario es la unidad C, el archivado de los reportes creados (gráficas, tablas o archivos de datos) se hará en forma automática en el directorio C:\DMACS\REP y no aparecerá ningún mensaje; el programa desactivará la *ventana de UNIDADES* y regresará a la *ventana de IMPRESION y ARCHIVADO*.

NOTA: Si el reporte que se va a archivar corresponde a la opción ARCHIVO MENSUAL, asegúrese de tener un espacio de 700 Kbytes (aprox.) en el disco que va a utilizar para tal efecto.

Una vez que se ha seleccionado la opción ARCHIVAR y se ha desplegado la *ventana de UNIDADES*, es posible salir de dicha ventana aunque el usuario no haya elegido ninguna unidad para archivar los reportes creados. Para salir de esta ventana es necesario seleccionar con el *ratón* uno de los dos cuadros de color azul que se encuentran en esquina superior derecha de la ventana.

6.3.4 COMO DAR DE ALTA UN SISTEMA DENTRO DEL GENERADOR DE REPORTES

Cuando se inicia la ejecución de este generador de reportes aparecen en la parte superior de la pantalla cinco opciones básicas. El funcionamiento de las cuatro primeras opciones ya se ha descrito en las secciones anteriores, sólo falta por describir la opción **ALTA**.

Puede decirse que esta opción es la más importante dentro del generador de reportes, ya que la generación de todos los tipos de reportes depende de los sistemas fotovoltaicos que se encuentren dados de alta mediante esta opción.

Para entrar a la *ventana de ALTA* que se muestra en la figura 6.18 posicione el cursor del *ratón* en la opción **ALTA** y presione el botón izquierdo para seleccionarla.

Como se puede ver, esta ventana es similar a la ventana de *SISTEMAS*, sólo que funciona de manera distinta.

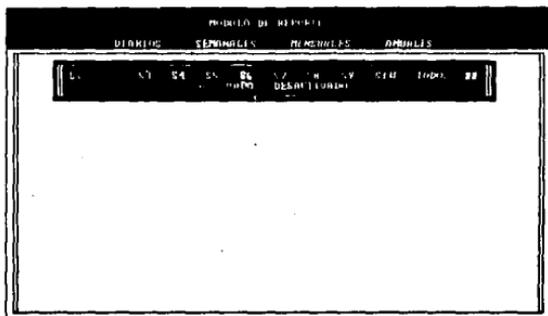


Figura 6.18 Ventana de ALTA

Los identificadores de los sistemas fotovoltaicos de color amarillo, son los sistemas que se encuentran dados de alta o activados, los sistemas en color blanco son los que se encuentran dados de baja o desactivados.

Si se selecciona con el cursor del *ratón* alguno de los sistemas activados, éste cambiará de color amarillo a blanco, lo que indica que se ha dado de baja del generador de reportes.

Si se selecciona con el cursor del *ratón* alguno de los sistemas sin activar, entonces aparecerá la ventana que se ilustra en la figura 6.19.

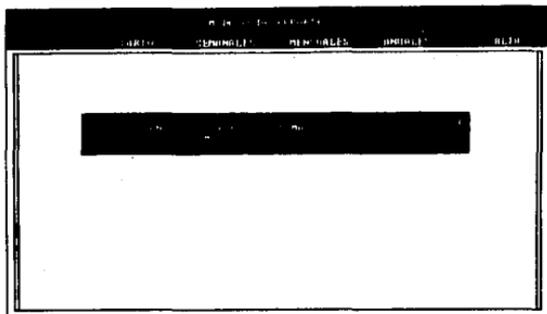


Figura 6.19 Ventana que permite introducir el identificador para cada sistema fotovoltaico

Esta ventana permite introducir el identificador que aparecerá en cada uno de los reportes que se generen del sistema fotovoltaico en cuestión. Dicho identificador puede componerse por letras, números y puntos y debe ser, de una longitud máxima de 12 letras o números, sin espacios en blanco. Si el usuario introduce un nombre de longitud 12, la ventana desaparece sin esperar que se pulse la tecla <ENTER>. Si el usuario introduce un nombre de longitud menor, entonces el sistema espera que se pulse la tecla <ENTER> para salir de la ventana en cuestión y volver a la *ventana de ALTA*.

Es posible dar de baja o de alta un sistema fotovoltaico todas las veces que el usuario lo necesite. Sólo existe una restricción ***no es posible dar de baja todos los sistemas***. Siempre que se intente dar de baja el último sistema activado, aparecerá una ventana que indica que no es posible darlo de baja.

Cada vez que se da de alta un sistema, el generador de reportes abre el archivo *ALTA.DAT* y escribe el número y el identificador del sistema fotovoltaico en dicho archivo.

Si se da de baja un sistema, el generador de reportes abre el archivo en cuestión y borra el número y el identificador del sistema fotovoltaico del mismo.

Si el archivo *ALTA.DAT*⁴, se borra por algún motivo, es posible restaurarlo como se indica en el Apéndice B, ya que sin este archivo el generador de reportes no funciona.

Si el generador de reportes no encuentra el archivo *ALTA.DAT* en el subdirectorio C:\DMACS entonces desplegará un mensaje cuando se seleccione algún tipo de reporte en la ventana principal de este programa. Este mensaje indica que no existe el archivo *ALTA.DAT* y después termina la ejecución del módulo *REPORTES*, volviendo al menú principal de FIX DMACS.

[GER84] GEREZ, VICTOR. Desarrollo y administración de programas de computadora (software). (México, CECSA, primera edición, 1984). p.75.

⁴ Este archivo forma parte de los archivos que utiliza el generador de reportes. Si por algún accidente se llegara a borrar, es posible restaurarlo como se indica en el Apéndice B.

Conclusiones y recomendaciones

En el trabajo que presento se diseñaron, programaron, compilaron y probaron las rutinas que corresponden a un generador de reportes cuyo objetivo principal es ayudar en la evaluación del comportamiento diario, semanal y mensual de los sistemas fotovoltaicos que forman parte del banco de pruebas del Instituto de Investigaciones Eléctricas.

Esta ayuda consiste en reducir el tiempo para procesar los archivos que contienen los valores que se obtienen de las mediciones de las variables necesarias para realizar dicha evaluación.

Para reducir el tiempo de procesamiento se simplificaron las siguientes tareas:

- Generación de archivos de datos en forma ordenada, es decir con valores de radiación solar, corrientes y voltajes en el panel, la batería y la carga del sistema fotovoltaico en intervalos de cada cinco minutos y de las 00:00:00 a las 23:55:00 horas. Dicha generación se realizaba en forma manual mediante el programa *Historical Trend Display* del programa multitareas FIX DMACS. La generación de estos datos se realiza dentro del generador de reportes en forma automática con sólo introducir la fecha de reporte en la interfaz de usuario correspondiente.

- Concatenación de los archivos obtenidos en forma manual. Esta tarea se eliminó, ya que el generador de reportes enlaza los programas *Report Creator* y *Report Generator* del programa FIX DMACS y genera los datos requeridos cada 5 minutos y de las 00:00 a las 23:55 Hrs. Anteriormente el usuario sólo podía generar datos de 22 horas y 30 minutos, por lo que era necesario unir los archivos resultantes y después separarlos para obtener los datos de un día completo.

■ Generación de gráficas y tablas de comportamiento de cada sistema fotovoltaico mediante un paquete de *software* comercial, diferente a FIX DMACS. Esta tarea se eliminó ya que el generador de reportes crea, imprime y archiva las gráficas y tablas necesarias que permiten observar, caracterizar y evaluar el comportamiento de los sistemas fotovoltaicos a prueba, sin necesidad de recurrir a otro *software* diferente a FIX DMACS.

La simplificación de las tareas permitirá además un ahorro de horas-hombre considerable.

La incorporación del módulo *REPORTES* al ambiente FIX DMACS se realizó en forma progresiva para preveer fallas futuras. Hubo problemas de manejo de memoria, sobre todo en la graficación de datos. Estos problemas se solucionaron con las herramientas de Quinn Curtis para manejo de memoria.

Se realizó una interfaz de usuario para el generador de reportes. Dicha interfaz de usuario es similar (en apariencia) a la interfaz del programa FIX DMACS. Esta interfaz es amigable y muestra todas sus funciones en pantalla.

La programación de todas las rutinas se realizó en lenguaje C que nos permite realizar una programación modular y transportable, además de ser rápido y eficiente.

Recomendaciones:

■ La presentación de la interfaz de usuario correspondiente al generador de reportes podría mejorarse con un poco más de tiempo, para que el despliegue de datos y gráficas se haga más flexible y elegante (despliegue a color y por seguimiento de pantallas).

■ Para que el generador de reportes quede en condiciones óptimas de funcionamiento, es necesario definir, diseñar y realizar las rutinas del módulo para reportes anuales. Con lo cual se hace necesario incrementar la capacidad de disco duro de la computadora en la que se encuentra instalado el programa FIX DMACS, para guardar los archivos de todos los meses del año para generar los reportes anuales.

■ Dada la característica de modularidad del generador de reportes, podrían adicionarse al mismo otras opciones que permitan observar y evaluar aspectos diferentes a la producción de corrientes y voltajes en función de la radiación solar captada por los sistemas fotovoltaicos a prueba. Tales aspectos pueden ser las condiciones ambientales (temperatura, humedad, velocidad del viento) en que operan los sistemas fotovoltaicos. Podrían adicionarse también al generador de reportes, opciones que permitan generar gráficas de eficiencia por sistema fotovoltaico en función de la insolación, ángulo de inclinación, temperatura modular, entre otros.

Comentario personal:

El realizar este trabajo me ha proporcionado una visión más amplia de lo importante que es la computación en nuestros días, ya que me permitió aplicar en forma real los conocimientos que adquirí, acerca de esa y otras áreas, durante mi formación profesional. Pero no sólo eso, me ha dado la oportunidad de reflexionar acerca de lo necesario y urgente que es desarrollar tecnología propia para explotar la energía que proviene del sol (y de otras fuentes) y generar energía eléctrica que complemente y/o sustituya las fuentes convencionales con que cuenta nuestro país. Sólo así tendremos electricidad para muchísimos años (ya que la energía de sol es inagotable) y un gran ahorro monetario que favorecería a nuestra nación.

Bibliografía

CEBALLOS SIERRA, FRANCISCO JAVIER. Manual para QuikC 2: Guía del programador. México: Macrobit, 1990. 520p.

CORTES GARCIA, CLAUDIA. "Análisis de las metodologías de diseño de sistemas fotovoltaicos autónomos". Veracruz, Méx.: Universidad Veracruzana, 1993. (Tesis de licenciatura).

ELECTRICAL POWER RESEARCH INSTITUTE. "Data Acquisition for Photovoltaic Power Plants", preparado por New Mexico State University, Las cruces, New Mexico: Diciembre, 1990. (EPRI GS-7082 Project 1607-6, reporte interno).

ELECTRICAL POWER RESEARCH INSTITUTE. "PG & E Photovoltaic Module Performance Assessment", preparado por Pacific Gas and Electric Company, San Ramón, California: Marzo, 1986. (EPRI AP-4464 Project 1607-2, reporte final).

FIX_DMACS versión 3.0. Intellution. Guía de usuario y manuales de instalación.

FUENTES CASTILLO, GUILLERMINA. "Desarrollo de un subsistema de reportes de control supervisorio de subestaciones eléctricas en computadora personal". México, D.F.: UNAM, ENEP Aragón, 1993. 187p., (Tesis de licenciatura).

Funciones del DOS y BIOS. Manual, México, D.F.: Addison-Wesley Iberoamericana, 1990. 171p.

GEREZ GREISER, VICTOR. Desarrollo y administración de programas de computadora (software). México: CECSA, primera edición, 1984. 299p.

HUACUZ, JORGE V. [et al]. "Electrificación rural con fuentes renovables de energía". En BOLETIN IIE, (mayo-junio, 1992), pp. 116-124.

INSTITUTO DE INVESTIGACIONES ELECTRICAS. "Programa de electrificación rural con fuentes" (1991-1994). Reporte interno. Cuernavaca, Morelos, México.

INSTITUTO DE INVESTIGACIONES ELECTRICAS. "Integración y puesta en operación de un sistema de adquisición de datos, para la evaluación de comportamiento de pequeños sistemas fotovoltaicos". Cuernavaca, Méx.: Junio, 1993. (IIE/FE/FNCE/5520/001/P).

LAFORE, ROBERT. The Waite group's Microsoft C programming for the PC. USA: Howard W. SAMS & Company, 1989. 747p.

MEDELLIN CUENCAR, SANDRA ROCIO. "Interfaz gráfica para el diseño de prototipos rápidos en un contexto de programación orientada a objetos". Coahuila, México: Universidad Autónoma de Coahuila, Facultad de sistemas, 1992. 98p., (Tesis licenciatura).

OLIVERA CASTAÑEDA AURELIO. "Generación de energía eléctrica por medio solar". México, D.F.: Universidad Nacional Autónoma de México, Facultad de Ingeniería, 1987. 141 p., (Tesis de licenciatura).

PRESSMAN, S. ROGER. Ingeniería del software: un enfoque práctico. México: McGraw Hill, tercera edición, 1993. 824p.

SCIENCE AND ENGINEERING TOOLS, revisión 7.0 para QuikC versión 2.x . Quinn-Curtis. Guía de usuario.

SCHILD, HERBERT. Programación en Turbo C. Madrid, España: McGraw Hill, segunda edición, 1990. 449p.

SCHILD, HERBERT. Turbo C: programación avanzada. España: McGraw Hill, segunda edición, 1991. 401p.

TIPPENS, PAUL E. Física: conceptos y aplicaciones. D.F. México: McGraw Hill, primera edición en español, 1981. 612p.

URBANO CASTELAN, JOSE ANTONIO. "Proyecto para la industrialización de módulos solares fotovoltaicos". México, D.F.: Instituto Politécnico Nacional, ESIME, 1985. (Tesis de licenciatura).

U. S. DEPARTMENT OF DEFENSE "Maintenance and operation of stand-alone photovoltaic system", preparado por U.S Naval facilities Engineering command, Colorado, U.S.A: diciembre, 1991.

VALENZUELA FLORES, ARTURO. "Diseño, construcción y puesta en operación de un sistema fotovoltaico de 2.45 KW intercalado a la red eléctrica". Aguascalientes, Ags.: Instituto Tecnológico de Aguascalientes, 1993. 74 p., (Tesis de licenciatura).

WEISS, EDMOND H. How to write a usable user manual. Filadelfia, E.U.: Isi press, 1985. 197p.

Apéndice A

COMO CREAR LOS FORMATOS DE DATOS PROVENIENTES DE FIX DMACS PARA EL GENERADOR DE REPORTES

Instrucciones: Para seleccionar una opción de cualquier menú que se despliegue al crear los formatos de datos posicione el cursor del *ratón* en la opción deseada y presione el botón izquierdo del mismo. Puede utilizar también las teclas para movimiento de cursor (↑ ↓ ← →) para posicionar éste en la opción deseada y luego pulsar la tecla <ENTER> para seleccionarla.

1.- Seleccione la opción **REP - REPORT CREATOR** del menú principal de FIX DMACS para activar la ventana que permitirá editar el formato de datos. Esta ventana contiene las opciones: EDIT, COPY, DELETE, DIRECTORY Y RESTORE.

2.- Seleccione la opción **EDIT** para crear el nuevo formato de datos. Aparecerá el siguiente mensaje:

Enter Report filename:

3.- Introduzca el nombre del reporte¹ y pulse la tecla <ENTER>. El sistema le preguntará lo siguiente:

Create new Report file (Y/N) ?

¹ Si el formato es para generar reportes diarios este nombre se designa de la siguiente manera: GRPAN[Número de sistema]. Si es para reportes mensuales entonces será MENS[Número de sistema].

Teclée "Y" para editar el nuevo reporte.

Se desplegará una pantalla que indica el número de columnas y de renglones de que consta la primera página del formato de datos que acaba de crear.

4.- Auxiliándose de las teclas para movimiento de cursor, posicione éste en la columna 1 y el renglón 1. Observe que se encuentra en la página 1 de 1. Para generar datos de un día completo es necesario que el reporte cuente con 5 páginas, ya que el número máximo de renglones por página es 66. Cree las 4 páginas restantes pulsando las teclas <ALT> y <A> simultáneamente. Observe cómo se va incrementando el número de páginas hasta llegar a la página 5 de 5. Pulse las teclas <CTRL> y <PAGEUP> simultáneamente, las veces que sea necesario, para regresar a la página 1. Se regresa a la página 1 para crear los enlaces del formato de datos en orden.

5.- Pulse las teclas <ALT> y <L> simultáneamente para activar la ventana que contiene el menú de valores que se pueden asociar o enlazar al formato que se está creando.

6.- Seleccionando la opción **HISTORICAL**, aparecerá el formato de enlace que te permitirá acceder los datos que necesita el generador de reportes.

7.- Llene los campos como se indica a continuación

CAMPO	VALOR
PATH	Sin valor
DATE	SYS_DATE
TIME	00:00:00
INTERVAL	00:00:05:00
SAMPLES	66
ALIGNMENT	HORIZONTAL
SKIP	SINGLE
SCIENTIFIC	NO
FORMAT	> 5.2
TAGNAMES	NO
LINE	1
COLUMN	1

	NODE	TAGNAME	FIELD	MODE
1	FOT	VFV_(No. Sisi)	F_CV	SAMPLE
2	FOT	IFV_(No. Sisi)	F_CV	SAMPLE

3	FOT	VBB_[No. Sist]	F_CV	SAMPLE
4	FOT	IC_[No. Sist]	F_CV	SAMPLE
5	FOT	RSP	F_CV	SAMPLE
6	FOT	IBB_[No. Sist]	F_CV	SAMPLE
7				
8				
9				
10				

Por cada campo que se llene asegúrese de pulsar la tecla <ENTER> al final. Si no lo hiciera así, no se respeta el valor que introdujo en el formato.

8.- Una vez que se han llenado todos los campos pulse la tecla <F10> para indicarle al sistema que el enlace está completo. Aparecerá una ventana con las opciones: SAVE CHANGES, QUIT MENU y CONTINUE.

9.- Seleccione la opción **SAVE CHANGES** para grabar en el enlace que se acaba de crear.

10.- El sistema regresará a la pantalla que despliega las columnas y renglones con que cuenta la página. Después de haber creado el enlace esta pantalla debe estar llena. Pulse las teclas <CTRL> y <PAGE DOWN> para cambiar a la siguiente página y crear el enlace que le corresponde.

11.- Cree los enlaces necesarios para las páginas restantes del formato de datos, siguiendo las instrucciones que se indican a partir del paso 5. Se debe tener cuidado con la siguiente observación: todos los campos llevan los mismos valores que se muestran en el paso 7, excepto TIME y SAMPLES. El valor para TIME varía conforme se va cambiando de página y el valor de SAMPLES varía únicamente en la última página. Asigne el valor de TIME de acuerdo a lo siguiente:

VALOR DE TIME

PAGINA 1 00:00:00 (Valor inicial)
 PAGINA 2 05:30:00
 PAGINA 3 11:00:00
 PAGINA 4 16:30:00
 PAGINA 5 22:00:00

El valor de SAMPLES en la última página será de 24.

12.- Una vez que ha creado todos los enlaces, pulse la tecla **<F10>** para salvar el formato en general. Aparecerá la ventana con las opciones que se indican en el paso 8. Seleccione **SAVE CHANGES** para grabar el formato completo y regresar al menú del generador de reportes de FIX DMACS.

13.- Si necesita crear otro formato, realice las instrucciones que se indican a partir del paso 2, si no pulse la tecla **<F10>** para regresar al menú principal de FIX DMACS.

Apéndice B

COMO CREAR EL ARCHIVO ALTA.DAT

Este archivo se crea con el editor de sistema operativo (MS DOS versión 5.0) de la siguiente forma:

- 1.- Cambie al directorio C:\DMACS.
- 2.- Teclee la siguiente orden: **EDIT ALTA.DAT** y pulse la tecla **<ENTER>**.
- 3.- Se desplegará entonces la ventana de edición de **EDIT**.
- 4.- Posicione el cursor en el renglón 1 y la columna 1.
- 5.- Escriba el siguiente dato:
1 (Identificador del sistema fotovoltaico 1)
- 6.- Pulse la tecla **<ENTER>** al final del renglón que se acaba de escribir. Asegúrese de no utilizar ningún caracter o número diferente a los que se indican.
- 7.- Pulse la tecla **<ALT>** para posicionarse en la barra de menú de **EDIT**.
- 8.- Posicione el cursor en la opción **FILE**. Pulse la tecla **<ENTER>** para desplegar el menú de esta opción.

9.- Posicione el cursor en la opción *Exit* y pulse la tecla **<ENTER>**. Una vez seleccionada esta opción se desplegará un mensaje como el siguiente:

Loaded file is not saved. Save it now ?

10.- Posicione el cursor en **<Yes>** y pulse la tecla **<ENTER>** para guardar el archivo *ALTA.DAT* y salir de *EDIT*.

NOTA: En el paso 5 se introduce el número 1 y el identificador del sistema fotovoltaico 1. Este número puede variar del 1 al 10. El identificador no debe incluir ningún espacio en blanco, sólo puntos, guiones, caracteres y números. El único espacio en blanco se debe encontrar entre el número y el identificador del sistema fotovoltaico. Lo que se pretende con ello es que el programa *REPORTES* pueda detectar la existencia de este archivo para funcionar, el contenido del mismo puede modificarse mediante la opción *ALTA* de dicho programa.

Apéndice C

RUTINAS DE PROGRAMACION DEL MODULO REPORTES

C.1 PROGRAMA PRINCIPA.C

/.....
Este programa está elaborado en lenguaje C utilizando el software BorlandC versión 3.0.

Contiene:

- El programa y la pantalla principales del generador de reportes
- La validación general de las funciones en pantalla con ratón
- Las rutinas para generar los reportes diarios
- Las rutinas para imprimir y archivar todos los reportes generados

Todas estas funciones para uno o todos los sistemas dados de alta dentro del mismo generador de reportes.

Elaboró: LETICIA HERNANDEZ RAMIREZ
Supervisó: Dr. FRANCISCO VILLASEÑOR RIVERO (Instituto de Investigaciones eléctricas).

Marzo 1995
/.....

```
#include <stdio.h>
#include <ctype.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <dos.h>
#include <string.h>
#include <process.h>
#include <mouse.h>
#include <principa.h>
```

```
#define OFF 0
#define ON 1
```

```
#define ALTA "ALTA.DAT"
#define GRAFICA "GRAFICA "
#define OTRAS_OP "DIARIOS "
```



```

    {
        if ((Noborres == 0 && activ_vl == 1) || (Noborres == 0 && unidad == 1))
            borra_todo();
        borra_todo();
        CursorOn();
    }
} while(tr);
system("cla");
}

/.....
Esta función valida las posiciones de las opciones en el generador de reportes.
...../
void valida_pantalla(int amx, int amy)
{
    switch(amx)
    {
        case 3: if (amy > 15 && amy < 23)
            tipo_reporte = 1;
            else
            if (amy > 27 && amy < 37)
                tipo_reporte = 2;
            else
            if (amy > 41 && amy < 51)
                tipo_reporte = 3;
            else
            if (amy > 55 && amy < 63)
                {
                    tipo_reporte = 4;
                    No_funcion();
                }
            else
            if (amy > 71 && amy < 76)
                alta = 1;
            else
                { tipo_reporte = 0; alta = 0; }
            if ((tipo_reporte > 0 && tipo_reporte < 5) || alta == 1)
                solo_ventas();
        break;
        case 8: if (tipo_reporte > 0 && tipo_reporte < 5)
            {
                if (amy > 8 && amy < 11)
                    Num_sist = 1;
                else
                if (amy > 13 && amy < 18)
                    Num_sist = 2;
                else
                if (amy > 18 && amy < 21)
                    Num_sist = 3;
                else
                if (amy > 23 && amy < 26)
                    Num_sist = 4;
                else
                if (amy > 28 && amy < 31)
                    Num_sist = 5;
                else
                if (amy > 33 && amy < 36)
                    Num_sist = 6;
                else
                if (amy > 38 && amy < 41)
                    Num_sist = 7;
                else
                if (amy > 43 && amy < 46)

```

```

        Num_sist = 8;
    else
        if (amy > 48 && amy < 51)
            Num_sist = 9;
        else
            if (amy > 53 && amy < 57)
                Num_sist = 10;
            else
                {
                    if (amy > 59 && amy < 65)
                        {
                            todos = 1;
                            Num_sist = 0;
                        }
                    else
                        /* Salir ventana sistemas */
                        if (amy > 66 && amy < 69)
                            {
                                todos = 0;
                                Num_sist = 0;
                                tipo_reporte = 0;
                                MouseWindow(1,632,1,192);
                            }
                        else
                            solo_ventas();
                }
    }
}
if (Num_sist > 0 && Num_sist < 11)
    busca_sistema();
if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
    { MouseCursor(OFF); reestablece(); MouseCursor(ON); }
break;
/* El case 8 permite dar de alta (dentro de este programa) a los
sistemas fotovoltaicos de los cuales está adquiriendo datos el
programa FIX DMACS */
case 8: if (alta == 1)
    {
        if (amy > 12 && amy < 15)
            Num_sist = 1;
        else
            if (amy > 17 && amy < 20)
                Num_sist = 2;
            else
                if (amy > 22 && amy < 25)
                    Num_sist = 3;
                else
                    if (amy > 27 && amy < 30)
                        Num_sist = 4;
                    else
                        if (amy > 32 && amy < 35)
                            Num_sist = 5;
                        else
                            if (amy > 37 && amy < 40)
                                Num_sist = 6;
                            else
                                if (amy > 42 && amy < 45)
                                    Num_sist = 7;
                                else
                                    if (amy > 47 && amy < 50)
                                        Num_sist = 8;
                                    else
                                        if (amy > 52 && amy < 55)
                                            Num_sist = 9;
                                        else
                                            if (amy > 57 && amy < 61)

```

```

        Num_sist = 10;
    else
        /* Salir ventana alta */
        if (amy > 62 && amy < 65)
        {
            alta = 0;
            Num_sist = 0;
            MouseWindow(1,632,1,192);
        }
        else
        {
            alta = 1;
            solo_venta1st();
            Num_sist = 0;
        }
    }
    if (Num_sist > 0 && Num_sist < 11)
        alta_sistema();
break;
case 10: if (tipo_reporte > 0 && tipo_reporte < 5)
    if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
    {
        MouseCursor(OFF); reestablece();
        MouseCursor(ON);
    }
break;
case 11: if (tipo_reporte > 0 && tipo_reporte < 5)
    if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
        /* Salir ventana diarias, semanales y mensuales */
        if (amy > 45 && amy < 48)
        {
            if (band_fech == 1 && tipo_reporte == 1)
            {
                if (todos == 1)
                    borra_todoal();
                else
                    borra_unol();
            }
            Num_sist = 0;
            todos = 0;
            band_fech = 0;
            inicializa = 0;
            solo_venta1st();
        }
        else
        {
            MouseCursor(OFF); reestablece();
            MouseCursor(ON);
        }
break;
case 12: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
        switch (tipo_reporte)
        {
            /* Gráfica diaria */
            case 1: selecciona(amx,amy,30,39); break;
            /* Tabla resumen semanal */
            case 2:
            /* Tabla resumen mensual */
            case 3: selecciona(amx,amy,30,44); break;
        }
    }
break;

```

```

case 13: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
    switch (tipo_reporte)
        {
        /* Tabla resumen diaria */
        case 1: selecciona(amx,amy,30,44); break;
        /* Tabla fallas semanal */
        case 2:
        /* Tabla fallas mensual */
        case 3: selecciona(amx,amy,30,46); break;
        }
    }
break;
case 14: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
    switch (tipo_reporte)
        {
        /* Tabla de fallas diaria */
        case 1: selecciona(amx,amy,30,46); break;
        /* No hay seleccion */
        case 2: No_opcion(); break;
        /* Archivo mensual */
        case 3: archivo_mensual(amy,30,46); break;
        }
    }
break;
case 15: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
    switch (tipo_reporte)
        {
        /* Tabla de datos diaria */
        case 1: selecciona(amx,amy,30,38); break;
        /* No hay seleccion */
        case 2:
        case 3: No_opcion(); break;
        }
    }
break;
case 16: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
    if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
        No_opcion();
    }
break;
case 18: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
    if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
        if (amy > 42 && amy < 45 && band_fech == 1 && activ_vi == 0)
            {
            MouseCursor(OFF);
            if (todos == 1 && tipo_reporte == 1)
                borra_todos();
            else
                borra_uno();
            band_fech = 0;
            lectura_fecha(); inicializa = 1;
            if (Num_sist > 0 && Num_sist < 11)
                procesa_unsist();
            else
                if (todos == 1)
                    todos_sistemas();
            }
        else
            if (amy > 46 && amy < 49 && band_fech == 1 && activ_vi == 0)
                {
                MouseCursor(OFF);
                inicializa = 0;
                }
    }
}

```

```

case 13: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
        switch (tipo_reporte)
        {
            /* Tabla resumen diaria */
            case 1: selecciona(amx,army,30,44); break;
            /* Tabla fallas semanal */
            case 2:
                /* Tabla fallas mensual */
                case 3: selecciona(amx,army,30,48); break;
        }
    }
break;
case 14: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
        switch (tipo_reporte)
        {
            /* Tabla de fallas diaria */
            case 1: selecciona(amx,army,30,46); break;
            /* No hay seleccion */
            case 2: No_opcion(); break;
            /* Archivo mensual */
            case 3: archivo_mensual(army,30,46); break;
        }
    }
break;
case 15: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
        switch (tipo_reporte)
        {
            /* Tabla de datos diaria */
            case 1: selecciona(amx,army,30,38); break;
            /* No hay seleccion */
            case 2:
            case 3: No_opcion(); break;
        }
    }
break;
case 16: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
        if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
            No_opcion();
    }
break;
case 18: if (tipo_reporte > 0 && tipo_reporte < 5)
    {
        if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
            if (army > 42 && army < 45 && band_fech == 1 && activ_vi == 0)
            {
                MouseCursor(OFF);
                if (todos == 1 && tipo_reporte == 1)
                    borra_todos();
                else
                    borra_uno();
                band_fech = 0;
                lectura_fechal(); inicializa = 1;
                if (Num_sist > 0 && Num_sist < 11)
                    procesa_unasist();
                else
                    if (todos == 1)
                        todos_sistemas();
            }
        else
            if (army > 46 && army < 49 && band_fech == 1 && activ_vi == 0)
                MouseCursor(OFF);
                inicializa = 0;
    }

```

```

        if (Num_sist > 0 && Num_sist < 11)
            processa_unsist();
        else
            if (todos == 1)
                todos_sistemas!);
            }
        else
            if (band_fech == 1 && activ_vi == 0)
                otra_fech!);
            else
                if (activ_vi == 1)
                    solo_ventimp!);
        break;
case 19: if (tipo_reporte > 0 && tipo_reporte < 5)
        if ((Num_sist > 0 && Num_sist < 11) || todos == 1)
            /* Salir ventana impresión y archivado */
            if (amy > 42 && amy < 45 && opcion > 0)
                {
                    if (Noborres == 0)
                        borra_salida!);
                    Noborres = 0;
                    activ_vi = 0;
                    No_opcion!);
                    if (todos == 1)
                        Num_sist = 0;
                }
            else
                if (activ_vi == 1 && unidad == 0)
                    solo_ventimp!);
                else
                    if (unidad == 1)
                        ventana_unidades!);
        break;
case 20: if (amy > 33 && amy < 42)
        {
            MouseCursor(OFF);
            if (todos == 1)
                imprime_todos!);
            else
                imprime_uno!);
            MouseCursor(ON);
        }
        else
            if (activ_vi == 1 && unidad == 0)
                solo_ventimp!);
            else
                /* Salir ventana unidades */
                if (amy > 57 && amy < 60 && unidad == 1)
                    {
                        unidad = 0;
                        solo_ventimp!);
                        activ_vi = 1;
                        if (todos == 1)
                            Num_sist = 0;
                    }
                else
                    if (unidad == 1)
                        ventana_unidades!);
        break;
case 21: if (amy > 33 && amy < 42)
        {
            /* Activa ventana que permita seleccionar la unidad de
            disco duro 0 disco flexible donde se guardarán los
            archivos generados */
            unidad = 1;

```

```

    ventana_unidades!);
}
else
if (amy == 50 && unidad == 1) /*Unidad A*/
{
    MouseCursor(OFF);
    if (todos == 0)
        copia_uno(1);
    else
        copia_todos(1);
    MouseCursor(ON);
}
else
if (amy == 53 && unidad == 1) /*Unidad B*/
{
    MouseCursor(OFF);
    if (todos == 0)
        copia_uno(2);
    else
        copia_todos(2);
    MouseCursor(ON);
}
else
if (amy == 56 && unidad == 1) /*Unidad C*/
{
    MouseCursor(OFF);
    if (todos == 0)
        copia_uno(3);
    else
        copia_todos(3);
    MouseCursor(ON);
}
else
if (activ_vl == 1 && unidad == 0)
    solo_ventimp!);
else
if (unidad == 1)
    ventana_unidades!);
break;
case 22: if (activ_vl == 1 && unidad == 0)
    solo_ventimp!);
else
if (unidad == 1)
    ventana_unidades!);
break;
default: tipo_reporte = 0;
        Num_sist = 0;
        opcion = 0;
        todos = 0;
        break;
}
}

```

.....
 Esta funcion permite dar de baja o alta el sistema que se selecciona,
 además de ordenar los sistemas en forma ascendente por el método de la
 burbuja.

 void alta_sistema(void)

```

{
    int temporal, cont, cont1;
    char ch, cad_paso[20], cad_arch[20], leyenda[15];
    struct sist_sis
    {

```

```

int numero;
char ley[18];
}text[10];

if ((fn = fopen(ALTA, "rt")) == NULL)
no_ALTA();
else
{
cont = 0; cont1 = 0;
while (!gets(cad_arch, 20, in) != NULL)
{
cont ++;
cad_arch[strlen(cad_arch) - 1] = '\0';
strcpy(cad_paso, cad_arch);
cad_paso[strlen(cad_paso) - (strlen(cad_paso) - 2)] = '\0';
if (atoi(cad_paso) != Num_sist)
{
text[cont].numero = atoi(cad_paso);
strcpy(text[cont].ley, cad_arch);
cont1 ++;
}
}
fclose(in);
if (cont == cont1) /*Significa que el sistema va a ser dado de alta*/
{
MouseCursor(OFF);
colores();
ventana(15, 12, 8, 68, 1, 1);
gotoxy(9, 2); cprintf("IDENTIFICADOR DEL SISTEMA: ");
gotoxy(19, 3); cprintf("Máximo 12 caracteres");
CursorOn();
i = 0;
do
{
gotoxy(36+i, 2); ch = getch();
if (ch == '\b' && i > 0)
{ i--; gotoxy(36+i, 2); cprintf(" "); gotoxy(36+i, 2); }
else
if (ch == '\b' && i == 0)
gotoxy(36+i, 2);
else
if (ch == '\r' && i > 0)
{ leyenda[i] = '\0'; i = 12; }
else
if (ch != ' ' && ch != '\r')
{ leyenda[i] = toupper(ch); i++; }
} while(i != 12); leyenda[i] = '\0';
CursorOff();
text[cont1].numero = Num_sist;
itoa(Num_sist, cad_paso, 10); strcat(cad_paso, " ");
strcat(cad_paso, leyenda); strcpy(text[cont1].ley, cad_paso);
cont1 ++;
for (j = 0; j < cont1 - 1; j++)
for (i = cont1 - 1; i > 0; i--)
if (text[i].numero < text[i-1].numero)
{
temporal = text[i].numero;
strcpy(cad_paso, text[i].ley);
text[i].numero = text[i-1].numero;
strcpy(text[i].ley, text[i-1].ley);
text[i-1].numero = temporal;
strcpy(text[i-1].ley, cad_paso);
}
MouseCursor(ON);
}
}

```

```

- /*Esta condición verifica que por lo menos un sistema esté dado de alta*/
if (cont1 > 0)
{
    in = fopen(ALTA, "wt");
    for (i = 0; i < cont1; i++)
        strcat(text[i].ley,"ln"); fputs(text[i].ley,in);
    fclose(in);
}
else /*No es posible dar de baja todos los sistemas*/
{
    MouseCursor(OFF);
    sound(250); ventana(4,15,12.8,68,11,1);
    gotoxy(9,2); cprintf("**** El archivo no puede quedar vacío ****");
    delay(800); nosound();
    puttext(1,1,80,25,pantalla);
    colorea();
    MouseCursor(ON);
}
solo_ventaist();
}
Num_sist = 0;
}
/.....
Esta función busca que el sistema seleccionado se encuentre dado de alta.
.....
void busca_sistema(void)
{
    char cad_arch[20], cad_paso[20];
    int cont = 0, cont1 = 0;

    if ((in = fopen(ALTA, "rt") == NULL)
        no_ALTA();
    else
    {
        while ((fgets(cad_arch,20,in) != NULL)
        {
            cont++;
            cad_arch[strlen(cad_arch)-1] = '\0';
            strcpy(cad_paso, cad_arch);
            cad_paso[strlen(cad_paso)-strlen(cad_paso)-2] = '\0';
            if (atoi(cad_paso) != Num_sist)
                cont1++;
            else
                strcpy(cad_leyenda, cad_arch);
        }
        fclose(in);
        if (cont == cont1)
        {
            MouseCursor(OFF);
            sound(300); ventana(4,15,22,12.57,14,1);
            gotoxy(3,2); cprintf("**** Sistema no encontrado ****");
            delay(500); nosound();
            puttext(1,1,80,25,pantalla);
            ventana_sistemas();
            MouseCursor(ON);
            Num_sist = 0;
        }
    }
}
}

```

.....
 Esta función verifica si las posiciones obtenidas mediante la función
 Mousexy corresponden a las posiciones válidas de la opción que se
 seleccionó y, si esto último sucede, genera una acción a realizar.


```
void selecciona(int amx, int amy, int lim1, int lim2)
{
    if (amy > lim1 && amy < lim2)
    {
        switch(amx)
        {
            case 12: opcion = 1; break;
            case 13: opcion = 2; break;
            case 14: opcion = 3; break;
            case 15: opcion = 4; break;
        }

        if (Num_sist > 0 && Num_sist < 11)
        {
            if (band_fecha == 0)
            {
                MouseCursor(OFF);
                lectura_fecha(); inicializa = 1;
                procesa_unsist();
            }
            else
                otra_fecha();
        }
        else
        {
            if (todos == 1 && opcion > 0 && opcion < 5)
            {
                if (band_fecha == 0)
                {
                    MouseCursor(OFF);
                    lectura_fecha(); inicializa = 1;
                    todos_sistemas();
                }
                else
                    otra_fecha();
            }
        }
        No_opcion();
    }
}

.....
Esta función permite generar una de las opciones del generador de reportes
para un sistema dado de alta, excepto la opción de ARCHIVO MENSUAL del tipo
de reportes mensual.
.....
void procesa_unsist(void)
{
    switch(tipo_reporte)
    {
        case 1: genera_diaros(); break;
        case 2:
        case 3: genera_sem_men[1]; break;
    }

    puttext(1, 1, 80, 25, pantalla);
    archivo_salida();
    system("cd \\dmcac\\vrep");
    if ((lin = fopen(Tipo_Arch, "rt")) != NULL)
        { fclose(lin); system("cd.."); ventana_impresion(); activ_vi = 1;
        }
    else { system("cd.."); No_salida(); }
    MouseCursor(ON);
}

```

```

.....
Esta función permite generar una de las opciones del generador de reportes
para todos los sistemas dados de alta, excepto la opción de ARCHIVO MENSUAL
del tipo de reportes mensual.
.....
void todos_sistemas(void)
{
FILE *out;
char cad_arch[20], cad_paso[20];

arch_gen = 0; l = 0;
if ((lin = fopen(ALTA, "rt")) == NULL)
no_ALTA();
else
{
while (fgets(cad_arch,20,lin) != NULL)
{
cad_arch[strlen(cad_arch)-1] = '\0';
l++;
strcpy(cad_paso, cad_arch);
strcpy(cad_leyenda, cad_arch);
cad_paso[strlen(cad_paso)-strlen(cad_paso-2)] = '\0';
Num_slst = atoi(cad_paso);
switch(tipo_reporte)
{
case 1: genera_diarios(); break;
case 2:
case 3: genera_sem_men(0); break;
}
archivo_salida();
system("cd \\dmacs\\rep");
if ((fout = fopen(Tipo_Arch, "rt")) != NULL)
{
fclose(fout);
strcpy(nom_arch[arch_gen].salida, Tipo_Arch); arch_gen++;
}
system("cd..");
}
fclose(lin);
Num_slst = 0;
puttext(1,1,80,25,pantalla);
if (arch_gen == l)
{ ventana_impresion(); activ_vi = 1; }
else
if (arch_gen == 0)
No_salida();
else
aviso_impresion();
MouseCursor(ON);
}
}
.....
Esta función permite generar los comandos necesarios para ejecutar los
programas realizados con las herramientas de Quinn Curtis, para generar las
cuatro opciones del generador de reportes diario.
.....
void genera_diarios(void)
{
char envia_ley[40], ejecuta[70], sistema[4], diario[50], paso[50];

system("cls");
fecha();
ltoa(Num_slst,sistema,10); strcat(sistema," ");
strcpy(diario,"GEN /Rgrpan"); strcat(diario,sistema); strcat(diario,"/D");
}

```

```

strcat(diario, fech_rep); strcat(diario, " /F");
strcpy(envia_ley, Cad_leyenda);
strcat(envia_ley, " "); strcat(envia_ley, fech_real);
if (inicializa == 1)
{
    system(diario);
    delay(2000);
    system("cls");
}
switch(opcion)
{
    case 1: archivo_salida();
            strcpy(paso, GRAFICA); strcat(paso, cad_leyenda);
            strcat(paso, " "); strcat(paso, fech_rep);
            strcat(paso, " "); strcat(paso, Tipo_Arch);
            break;
    case 2: strcpy(paso, OTRAS_OP); strcat(paso, envia_ley);
            strcat(paso, " 2");
            break;
    case 3: strcpy(paso, OTRAS_OP); strcat(paso, envia_ley);
            strcat(paso, " 3");
            break;
    case 4: strcpy(paso, OTRAS_OP); strcat(paso, envia_ley);
            strcat(paso, " 4");
            break;
}
if (todos == 1 && Num_sist > 0 && Num_sist < 11)
    strcat(paso, " 0");
else
    strcat(paso, " 1");
pasos[atrien(paso)] = '0';
strcpy(ejecuta, "cd \\dmaca\\reportes\\");
strcat(ejecuta, paso); ejecuta[atrien(ejecuta)] = '0';
system("cd \\dmaca\\rep");
system(ejecuta);
system("cd \\dmaca");
}
/*****
Esta función verifica si las posiciones obtenidas mediante la función
Mousexy corresponden a las posiciones válidas para la opción ARCHIVO
MENSUAL y, si esto último sucede, empieza a generar el archivo mensual.
*****/
void archivo_mensual(int amy, int lim1, int lim2)
{
    FILE *out;
    char cad_arch[20];

    arch_gen = 0; l = 0;
    if (amy > lim1 && amy < lim2)
    {
        opcion = 3;
        MouseCursor(OFF);
        lee_fecha_men();
        if (Num_sist > 0 && Num_sist < 11)
        {
            genera_archivo_mensual();
            puttext(1, 1, 80, 25, pantalla);
            system("cd \\dmaca\\rep");
            if ((fin = fopen(Tipo_Arch, "rt")) != NULL)
                { fclose(fin); system("cd.."); ventana_impresion(); activ_vi = 1; }
            else { system("cd.."); No_salida(); }
        }
    }
}

```

```

if (todos == 1)
{
    if ((ln = fopen(ALTA, "rt") == NULL)
        no_ALTA());
    else
    {
        while (fgets(cad_arch,20,ln) != NULL)
        {
            cad_arch[strlen(cad_arch)-strlen(cad_arch)-2] = '\0';
            Num_sist = atoi(cad_arch);
            genera_archivo_mensual();
            system("cd \\dmaca\\rep");
            if ((out = fopen(Tipo_Arch, "rt") != NULL)
                {
                    fclose(out);
                    strcpy(nom_arch[arch_gen].salida, Tipo_Arch);
                    arch_gen++;
                }
            system("cd..");
            fclose(ln);
            Num_sist = 0;
            puttext(1,1,80,25,pantalla);
            if (arch_gen == 1)
            { ventana_impresion(); activ_vi = 1; }
            else
            {
                if (arch_gen == 0)
                    No_salida();
                else
                    aviso_impresion();
            }
        }
        MouseCursor(ON);
    }
    No_opcion();
}

/*****
Esta función despliega una ventana que indica que hubo error en la
generación de alguno de los archivos de salida.
*****/
void aviso_impresion(void)
{
    ventana(1,15,8,12,72,15,1);
    gotoxy(4,2);
    cprintf("Error en la generación de alguno de los archivos de salida");
    gotoxy(4,3);
    cprintf("de esta opción, deseas imprimir o archivar los restantes?");
    obt_n_respuesta(62,3);
    if (respuesta == 'N')
    {
        for (i = 0; i < arch_gen; i++)
            remove(nom_arch[i].salida);
        puttext(1,1,80,25,pantalla);
        No_opcion();
    }
    else
    { puttext(1,1,80,25,pantalla); ventana_impresion(); activ_vi = 1; }
}

```

```

.....
/.....
Esta función imprime el archivo de salida que corresponde a la opción y al
sistema que se seleccionaron.
.....
void imprime_uno(void)
{
    char imprime[25];

    system("cd \\dmcacl\\rep");
    archivo_salida();
    strcpy(imprime, "copy "); strcat(imprime, Tipo_Arch);
    if (tipo_reporte == 1 && opcion == 1)
        { strcat(imprime, " lpt1"); ventana_laser(); obten_respuesta(47,3); }
    else
        { strcat(imprime, " prn"); ventana_impresora(); obten_respuesta(47,2); }
    if (respuesta == 'S')
        {
            system("cls");
            system(imprime);
            printf("Imprimiendo el archivo: %s...\n", Tipo_Arch);
            delay(4000);
        }
    system("cd.."); system("cls");
    puttext(1,1,80,26,pantalla); colorear(); ventana_impresion();
}
/.....
Esta función imprime todos los archivos de salida que se generaron cuando
se seleccionó la opción TODOS (independientemente de la opción
seleccionada).
.....
void imprime_todos(void)
{
    char imprime[25];

    if (tipo_reporte == 1 && opcion == 1)
        { ventana_laser(); obten_respuesta(47,3); }
    else
        { ventana_impresora(); obten_respuesta(47,2); }
    if (respuesta == 'S')
        {
            system("cls");
            system("cd \\dmcacl\\rep");
            for (i = 0; i < arch_gen; i++)
                {
                    strcpy(imprime, "copy "); strcat(imprime, nom_arch[i].salida);
                    if (tipo_reporte == 1 && opcion == 1)
                        strcat(imprime, " lpt1");
                    else strcat(imprime, " prn");
                    printf("Imprimiendo el archivo: %s...\n", nom_arch[i].salida);
                    delay (4000); system(imprime); delay(4000);
                }
            system("cd..");
        }
    system("cls");
    Num_sist = 0; puttext(1,1,80,26,pantalla); ventana_impresion();
}
/.....
Esta función copia el archivo de salida que corresponde a la opción y al
sistema que se seleccionaron a la unidad que se especificó.
.....
void copia_uno(int eleccion)
{
    char copia[25];

```

```

archivo_salida);
strcpy(copia, "copy "); strcat(copia, Tipo_Arch);
switch(eleccion)
{
    case 1: strcat(copia, " a."); break;
    case 2: strcat(copia, " b."); break;
    case 3: Nborres = 1; break;
}
if (eleccion == 1 || eleccion == 2)
{
    system("cls");
    system("cd \\dmca\\rep");
    printf("Copiando el archivo: %s...\n", Tipo_Arch);
    delay(2000); system(copia); delay (2000);
    system("cd..");
}
system("cls");
puttext(1,1,80,25, pantalla); unidad = 0; ventana_impresion();
}
/*****
Esta función copia los archivos de salida que corresponden a la opción
"todos", independientemente de la opción que se seleccionó, a la unidad
que se especificó.
*****/
void copia_todos(int eleccion)
{
    char copia[25];

    system("cls");
    system("cd \\dmca\\rep");
    for (i = 0; i < arch_gen; i++)
    {
        strcpy(copia, "copy "); strcat(copia, nom_arch[i].salida);
        switch(eleccion)
        {
            case 1: strcat(copia, " a.");
                    break;
            case 2: strcat(copia, " b.");
                    break;
            case 3: Nborres = 1; break;
        }
        if (eleccion == 1 || eleccion == 2)
        {
            system("cls");
            printf("Copiando el archivo: %s...\n", nom_arch[i].salida);
            delay(2000); system(copia); delay(2000);
        }
    }
    system("cd..");
    system("cls");
    Num_sist = 0;
    unidad = 0;
    puttext(1, 1, 80, 25, pantalla);
    ventana_impresion();
}
/*****
Esta función permite leer la fecha para las opciones de los tres tipos de
reportes, excepto para la opción ARCHIVO MENSUAL.
*****/
void lectura_fecha(void)
{
    int sigue = 1, x = 13;
    char ch, paso[4];

```

```

struct checa
{
    int day;
}mes_ano[12] = {31,28,31,30,31,30,31,31,30,31,30,31};
reestablece(); ventana(1,15,19,62,21,1);
CursorOn();
do
{
    = 0;
    gotoxy(4,2); cprintf("DIA[DD]: ");
    gotoxy(17,2); cprintf("MES[MM]: ");
    gotoxy(30,2); cprintf("AÑO[AA]: ");
    do
    {
        gotoxy(x+i, 2); ch = getch();
        if (ch == '\b' && i > 0)
            { i--; gotoxy(x+i, 2); cprintf(" "); gotoxy(x+i, 2); }
        else
            if (!isdigit(ch) && i >= 0 && ch != '\r')
                { gotoxy(x+i, 2); cprintf(" "); gotoxy(x+i, 2); }
            else
                if (ch == '\r' && i > 0)
                    { paso[i] = '\0'; i = 3; }
                else
                    if (ch != '\r')
                        { paso[i] = ch; i++; }
                    } while(i = 3); paso[i] = '\0'; i = 0;
        switch(sigue)
        {
            case 1: if (atoi(paso) > 0 && atoi(paso) < 32 && strlen(paso) == 2)
                    { sigue++; x = 28; strcpy(dia, paso); }
                    else
                        { error_fecha(x); i = 0; }
                break;
            case 2: if (atoi(paso) > 0 && atoi(paso) < 13 && strlen(paso) == 2)
                    { sigue++; x = 39; strcpy(mes, paso); }
                    else
                        { error_fecha(x); i = 0; }
                break;
            case 3: if (atoi(paso) > 93 && atoi(paso) <= 99 && strlen(paso) == 2)
                    { sigue++; strcpy(ano, paso); }
                    else
                        { error_fecha(x); i = 0; }
                break;
        }
        if (sigue == 4)
        {
            if (atoi(ano) == 96)
                { mes_ano[1].day = 29; }
            if (atoi(dia) <= mes_ano[atoi(mes) - 1].day)
                sigue++;
            else
                { textcolor(YELLOW);
                  gotoxy(12,3); cprintf("***** NO VALIDO *****");
                  sound(100); delay(100); sound(200); delay(100); nosound();
                  ventana(1,15,19,62,21,1); x = 13; sigue = 1;
                }
        }
    } while (sigue != 5);
    band_fech = 1;
} CursorOff();

```

.....
 Esta función permite leer el mes y el año para generar el archivo de datos mensual.


```

void lee_fecha_mes(void)
{
    int sigue = 2, x = 12;
    char ch, paso[4];

    reestablece(); ventana(1,15,26,19,55,21,1);
    CursorOn();
    do
    {
        i = 0;
        gotoxy(3,2); cprintf("MES(MM): "); gotoxy(17,2); cprintf("AÑO(AA): ");
        do
        {
            gotoxy(x+1, 2); ch = getch();
            if (ch == 'b' && i > 0)
                if (--gotoxy(x+1, 2); cprintf(" "); gotoxy(x+1, 2); }
            else if (!isdigit(ch) && i >= 0 && ch != 'r')
                { gotoxy(x+1, 2); cprintf(" "); gotoxy(x+1, 2); }
            else
                if (ch == 'r' && i > 0)
                    { paso[i] = '0'; i = 3; }
                else
                    if (ch != 'r')
                        { paso[i] = ch; i++; }
                } while(i = 3); paso[i] = '0'; i = 0;
            switch(sigue)
            {
                case 2: if (atoi(paso) > 0 && atoi(paso) < 13 && strlen(paso) == 2)
                        { sigue++; x = 26; strcpy(mes, paso); }
                        else
                            { error_fecha(x); i = 0; }
                        break;
                case 3: if (atoi(paso) > 94 && atoi(paso) <= 99 && strlen(paso) == 2)
                        { sigue++; strcpy(ano, paso); }
                        else
                            { error_fecha(x); i = 0; }
                        break;
            }
        } while (sigue != 4);
    }
    band_fech = 0;
    CursorOff();
}
    
```

.....
 Esta función borra el archivo de datos que genera FIX DMACS que ya no es útil.


```

void borra_uno(void)
{
    char paso[3];

    ltoa(Num_sist, paso, 10);
    strcpy(Nom_Arch, "grpan"); strcat(Nom_Arch, paso);
    strcat(Nom_Arch, ".prt");
    system("cd \\dmacal\\rep"); remove(Nom_Arch); system("cd..");
}
    
```

```

.....
/.....
Esta función borra todos los archivos de datos que genera FIX DMACS cuando
ya no son útiles.
.....
void borra_todos(void)
{
  char cad_arch[20], cad_paso[20];
  if ((ln = fopen(ALTA, "r")) == NULL)
    no_ALTA();
  else
    while (fgets(cad_arch,20,ln) != NULL)
      {
        cad_arch[strlen(cad_arch)-1] = '\0';
        strcpy(cad_paso, cad_arch);
        cad_paso[strlen(cad_paso)-strlen(cad_paso-2)] = '\0';
        Num_sist = atoi(cad_paso);
        borra_uno();
      }
    fclose(ln);
    Num_sist = 0;
}
/.....
Esta función borra los archivos de salida que del disco duro, si no fue
especificada la unidad C para almacenarlos.
.....
void borra_salida(void)
{
  if (todos == 0)
    {
      system("cd \\dmacs\\rep");
      archivo_salida();
      remove(Tipo_Arch);
      system("cd..");
    }
  else
    {
      system("cd \\dmacs\\rep");
      for (i = 0; i < arch gen; i++)
        remove(nom_arch[i].salida);
      system("cd..");
    }
}
/.....
Esta función genera el nombre del archivo de salida de acuerdo a la opción
y al sistema que se seleccionaron.
.....
void archivo_salida(void)
{
  char sist[3];

  struct Nom_mes
  {
    char Nombre[3];
  }meses[12]={"EN","FE","MR","AB","MY","JN","JL","AG","SE","OC","NO","DI"};

  strcpy(Tipo_Arch, dia);
  strcat(Tipo_Arch,meses[atoi(mes) - 1].Nombre);
  strcat(Tipo_Arch,ano);
  ltoa(Num_sist,sist,10);
  strcat(Tipo_Arch,sist);
}

```

```

switch(tipo_reporte)
{
case 1: switch(opcion)
{
case 1: strcat(Tipo_Arch, ".PLT"); break;
case 2: strcat(Tipo_Arch, ".TAB"); break;
case 3: strcat(Tipo_Arch, ".ERR"); break;
case 4: strcat(Tipo_Arch, ".DAT"); break;
}
break;
case 2: switch(opcion)
{
case 1: strcat(Tipo_Arch, ".SET"); break;
case 2: strcat(Tipo_Arch, ".SEE"); break;
}
break;
case 3: switch(opcion)
{
case 1: strcat(Tipo_Arch, ".MET"); break;
case 2: strcat(Tipo_Arch, ".MEE"); break;
case 3: strcat(Tipo_Arch, ".MEN"); break;
}
break;
}
Tipo_Arch[strlen(Tipo_Arch)] = '\0';
}
}
.....
Esta función permite generar las cadenas de caracteres que contienen la
fecha real y la fecha para generar el archivo de datos del generador de
reportes de FIX DMACS.
.....
void fechas(void)
{
strcpy(fech_rep,mes); strcpy(fech_real,dia);
strcat(fech_rep,"-"); strcat(fech_real," ");
strcat(fech_rep,dia); strcat(fech_real,mes);
strcat(fech_rep,"."); strcat(fech_real," ");
strcat(fech_rep,ano); strcat(fech_real,ano);
}
.....
Esta función únicamente obtiene la respuesta "s" o "n" para las preguntas
que se utilizan en la interfaz.
.....
void obten_respuesta(int x, int y)
{
do
{
CursorOn();
gotoxy(x,y); cprintf(" ");
gotoxy(x,y); respuesta = toupper(getche());
CursorOff();
}while (respuesta != 'N' && respuesta != 'S');
}
.....
Esta función obtiene las posiciones (en modo texto) x y y del ratón,
en valor decimal
.....
void Mousexy(void)
{
union REGS in;
div_t aux;
}

```

```

l = 0; r = 0; c = 0;
in.h.ch = 0;
in.h.dl = 3;
int86(0x33, &in, &in);
aux = div(in.h.ch * 256 + in.h.cl, 8);
mx = aux.quot;
aux = div(in.h.ch * 256 + in.h.dl, 8);
my = aux.quot;
l = in.h.bl & 1;
r = in.h.bl & 2;
c = in.h.bl & 4;
}

```

C.2 PROGRAMA VENTANAS.C

```

/.....
Este programa está elaborado en lenguaje C utilizando el software BorlandC
versión 3.0.

```

Contiene el formato para cada una de las ventanas que se muestran en el programa REPORTE.

Elaboró: LETICIA HERNANDEZ RAMIREZ
 Supervisó: Dr. FRANCISCO VILLASEÑOR RIVERO (Instituto de Investigaciones eléctricas).

Marzo 1995

```

/.....
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <ventanas.h>

#define ON 1
#define OFF 0
#define ALTA "ALTA.DAT"

/.....
Este función genera la ventana que muestran todos los sistemas de los
cuales es posible acceder datos. Los sistemas activados se muestran en
amarillo y los no activados en blanco.
.....
void ventana_sistemas(void)
{
  FILE *in;
  int i,j;
  char cad_arch[20], sistema[4], cad_paso[20], cadena[4];

  colorear();
  if (alta == 0)
    ventana(3,15,8,6,72,9,1);
  else
    ventana(3,15,12,8,68,11,1);
  if ((in = fopen(ALTA, "rt")) == NULL)
    no_ALTA();
  else
    {
      = 1; /* Inicia en sistema 1 */
      while (fgetc(cad_arch, 20, in) != NULL)
        cad_arch[strlen(cad_arch) - 1] = '\0';
    }
}

```



```

textcolor(BLUE);
gotoxy(18,2);cprintf("*****");
switch(opcion)
{
    case 1: gotoxy(3,3); cprintf("GRAFICAS"); break;
    case 2: gotoxy(3,4); cprintf("TABLA RESUMEN"); break;
    case 3: gotoxy(3,5); cprintf("TABLA DE FALLAS"); break;
    case 4: gotoxy(3,6); cprintf("LISTADO"); break;
}
MouseWindow(232,400,80,128);
}
/*****
Esta función despliega la ventana que contiene el menú de reportes
semanales.
*****/
void Ssemanales(void)
{
    ventana(3,15,30,11,51,15,1);
    gotoxy(3,3); cprintf("TABLA RESUMEN");
    gotoxy(3,4); cprintf("TABLA DE FALLAS");
    textcolor(BLUE);
    gotoxy(18,2); cprintf("*****");
    switch(opcion)
    {
        case 1: gotoxy(3,3); cprintf("TABLA RESUMEN"); break;
        case 2: gotoxy(3,4); cprintf("TABLA DE FALLAS"); break;
    }
    MouseWindow(232,400,80,112);
}
/*****
Esta función despliega la ventana que contiene el menú de reportes
manuales.
*****/
void Mensuales(void)
{
    ventana(3,15,30,11,51,16,1);
    gotoxy(3,3); cprintf("TABLA RESUMEN");
    gotoxy(3,4); cprintf("TABLA DE FALLAS");
    gotoxy(3,5); cprintf("ARCHIVO MENSUAL");
    textcolor(BLUE);
    gotoxy(18,2); cprintf("*****");
    switch(opcion)
    {
        case 1: gotoxy(3,3); cprintf("TABLA RESUMEN"); break;
        case 2: gotoxy(3,4); cprintf("TABLA DE FALLAS"); break;
        case 3: gotoxy(3,5); cprintf("ARCHIVO MENSUAL"); break;
    }
    MouseWindow(232,400,80,120);
}
/*****
Esta función despliega la ventana que contiene las opciones de impresión
y archivado para los archivos de salida del generador de reportes.
*****/
void ventana_impresion(void)
{
    reastablece();
    ventana (3,15,33,19,48,23,1);
    gotoxy(3,3); cprintf("IMPRIMIR");
    gotoxy(3,4); cprintf("ARCHIVAR");
    textcolor(BLUE); gotoxy(12,2); cprintf("*****");
    if (unidad)
    { gotoxy(3,4); cprintf("ARCHIVAR"); }
}

```

```

MouseWindow(256,376,144,176);
}
.....
/.....
Esta función despliega la ventana que indica cuáles son las unidades en que
es posible archivar los archivos de salida.
.....
void ventana_unidades(void)
{
MouseCursor(OFF);
ventana_impresion();
ventana(3,15,49,20,63,23,1);
textcolor(BLUE); gotoxy(1,2); cprintf("XXXX");
textcolor(WHITE);
gotoxy(3,3); cprintf("A"); gotoxy(6,3); cprintf("B");
gotoxy(9,3); cprintf("C");
MouseWindow(384,496,152,176);
MouseCursor(ON);
}
.....
/.....
Esta función genera la ventana que contiene el mensaje que pregunta si se
desea utilizar otra fecha, cuando ya se ha introducido una fecha
anteriormente.
.....
void otra_fecha(void)
{
MouseCursor(OFF);
reestablece();
ventana(1,15,19,18,62,20,1);
gotoxy(1,2); cprintf(" OTRA FECHA ? SI NO");
MouseWindow(144,488,144,144);
MouseCursor(ON);
}
.....
/.....
Esta función pregunta si se cuenta con una impresora HP LASER III que es
necesaria para imprimir la gráfica generada en los reportes diarios.
.....
void ventana_laser(void)
{
ventana(5,15,12,12,66,15,1);
gotoxy(4,2);
cprintf("Para esta opción es necesaria la utilización de una");
gotoxy(4,3);
cprintf("impresora HP LaserJet III. Continuar? S\N ");
}
.....
/.....
Esta función pregunta si la impresora está en línea para enviar el archivo
de salida a impresión.
.....
void ventana_impresora(void)
{
ventana(5,15,15,12,65,14,1);
gotoxy(3,2);
cprintf("Está preparada la impresora? Continuar? S\N ");
}
.....
/.....
Esta función indica que hay un error en algún dígito o valor excedido de
rango cuando se está introduciendo una fecha.
.....
void error_fecha(int x)
{

```

```

CursorOff(); textcolor(YELLOW);
if (tipo_reporte == 3 && opcion == 3) gotoxy(5,3);
else gotoxy(12,3);
cprintf("***** NO VALIDO *****");
sound(100); delay(100); sound(200); delay(100);
sound(100); delay(100); nosound();
textcolor(WHITE);
if (tipo_reporte == 3 && opcion == 3) borde(1,1,29,3);
else borde(1,1,43,3);
gotoxy(x,2); cprintf(" "); gotoxy(x,2);
CursorOn();
}

/*****
Esta función genera la ventana de salida cuando se ha presionado el botón
derecho del ratón (este botón se utiliza para salir del programa).
*****/
void ventana_salida(void)
{
    ventana(1,1,17,10,60,12,0); ventana(3,15,19,11,62,13,0);
    gotoxy(6,2); cprintf(" DESEAS SALIR DEL PROGRAMA? S/N");
    ventana(1,15,55,12,57,12,0);
    sound(250); delay(200); sound(50); delay(100); sound(250); delay(200);
    nosound();
}

/*****
Esta función genera una ventana que indica que no se generó ningún archivo
de salida.
*****/
void No_salida(void)
{
    ventana(4,15,16,12,66,14,1);
    sound(500); delay(200); nosound();
    gotoxy(4,2);
    cprintf("***** No se generó ningún archivo de salida *****");
    delay(1500);
    puttext(1,1,60,25,pantalla);
    No_opcion();
}

/*****
Esta función indica que el archivo ALTA.DAT no se encuentra en el
directorio C:\DMACS.
*****/
void no_ALTA(void)
{
    ventana(4,15,16,12,64,14,1);
    sound(500); delay(300); nosound();
    gotoxy(4,2);
    cprintf("***** No puedo abrir el archivo Alta.dat *****");
    delay(4000);
    exit(1);
}

/*****
Esta función despliega una ventana con un mensaje que indica que la función
no se ha implementado.
*****/
void No_funcion(void)
{
    coloreal();
    sound(250); ventana(4,15,22,12,57,14,1);
    gotoxy(3,2); cprintf("***** Función no implementada *****");
    delay(1000); nosound();
}

```

```

puttext(1,1,80,25,pantalla);
tipo_reporte = 0;
colorea();
}
.....
Esta función permite generar una ventana vacía con color y marco (o sin él).
.....
void ventana(int color_fondo, int color_texto, int minx,
             int miny, int maxx, int maxy, int no_borde)
{
    window(minx,miny,maxx,maxy); textbackground(color_fondo);
    textcolor(color_texto);
    clrscr();
    if (no_borde) borde (1,1,maxx-minx,maxy-miny + 1);
}
.....
Esta función genera el marco para cualquier ventana.
.....
void borde(int lnix,int lny,int flnx,int flny)
{
    int i;

    gotoxy(1,1); cprintf("  ");
    for (i=2; i<= flnx-1; i++)
        { gotoxy(i,1); cprintf("  "); }
    gotoxy(flnx,1); cprintf("  ");
    for (i=2; i<= flny-1; i++)
        { gotoxy(flnx,i); cprintf("  "); }
    gotoxy(flnx,flny); cprintf("  ");
    for (i=2; i<= flnx-1; i++)
        { gotoxy(i,flny); cprintf("  "); }
    gotoxy(lnix,flny); cprintf("  ");
    for (i=2; i<= flny-1; i++)
        { gotoxy(lnix,i); cprintf("  "); }
}
.....
Esta función cambia de color las opciones del menú principal cuando se
selecciona alguna.
.....
void colorea(void)
{
    ventana(3,15,1,4,80,4,0);
    gotoxy(17,1); cprintf ("DIARIOS"); gotoxy(29,1); cprintf ("SEMANALES");
    gotoxy(43,1); cprintf ("MENSUALES"); gotoxy(57,1); cprintf ("ANUALES");
    gotoxy(73,1); cprintf ("ALTA");
    textcolor(BLUE);
    if (alta == 1) { gotoxy(73,1); cprintf ("ALTA"); }
    else
        switch(tipo_reporte)
        {
            case 1: gotoxy(17,1); cprintf ("DIARIOS"); break;
            case 2: gotoxy(29,1); cprintf ("SEMANALES"); break;
            case 3: gotoxy(43,1); cprintf ("MENSUALES"); break;
            case 4: gotoxy(57,1); cprintf ("ANUALES"); break;
        }
}
.....
Esta función reestablece la ventana que indica los sistemas y la ventana
del tipo de reporte que se haya seleccionado.
.....
void reestablece(void)

```

```

{
    ventana_sistemas();
    switch(tipo_reporte)
    {
        case 1: Diarios();
        break;
        case 2: Semanales();
        break;
        case 3: Mensuales();
        break;
    }
}

/*.....
Esta función reestablece la ventana de opciones correspondiente a cada tipo
de reporte si el cursor del ratón no se posiciona correctamente en alguna
de las opciones.
.....*/
void No_opcion(void)
{
    opción = 0;
    MouseCursor(OFF);
    reestablece();
    MouseCursor(ON);
}

/*.....
Esta función genera únicamente la ventana de impresión, activando y
desactivando el cursor del ratón.
.....*/
void solo_ventimp(void)
{
    MouseCursor(OFF);
    ventana_impresion();
    MouseCursor(ON);
}

/*.....
Esta función genera únicamente la ventana de sistemas, activando y
desactivando el cursor del ratón.
.....*/
void solo_ventsist(void)
{
    MouseCursor(OFF);
    ventana_sistemas();
    MouseCursor(ON);
}

```

C.3 PROGRAMA SEMYMEN.C

```

/*.....
Este programa está elaborado en lenguaje C utilizando el software BorlandC
versión 3.0.

Contiene las rutinas para generar las tablas resumen semanal y mensual, así
como la tabla de fallas semanal y mensual para el programa REPORTES.

Elaboró: LETICIA HERNANDEZ RAMÍREZ
Supervisó: Dr. FRANCISCO VILLASEÑOR RIVERO (Instituto de Investigaciones
eléctricas).

Marzo 1995
.....*/

```

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <dos.h>
#include "semymen.h"

FILE *out;
char dia_final[10];

struct separa
{
float ENTRADA[7];
float SALIDA[7];
}valores;

struct guarda
{
char nombre[20];
}archivo[7], solo_datos[7], real[7], fech_sem[7];
}
.....
Esta función permite generar la tabla resumen y la tabla de fallas tanto
semanal como mensual. La tabla resumen obtiene los valores máximos
registrados semanal y mensualmente de acuerdo a la fecha de
inicio proporcionada por el usuario. La tabla de fallas obtiene el
porcentaje de datos perdidos semanal y mensualmente.
.....
void genera_sem_men(int pausa)
{
FILE *in;
int i = 0, cuenta_arch = 0, No_max_arch;

system("cls");
system("cd \\dmacs\\rep");
switch(tipo_reporte)
{
case 2: arch_diasem[7,1]; No_max_arch = 7;
if (opcion == 2) opcion = 1;
else if (opcion == 3) opcion = 2;
tipo_reporte = 2;
break;
case 3: arch_diasem[4,7]; No_max_arch = 4;
tipo_reporte = 3;
break;
}
for (i = 0; i < No_max_arch; i++)
{
if ((in = fopen(archivo[i].nombre, "rt")) == NULL)
{
sound(500);
printf("**** No existe el archivo: %s ***\n", archivo[i].nombre);
delay(50); nosound();
}
else
{
fclose(in); cuenta_arch++;
}
}
if (cuenta_arch == No_max_arch)
{
switch(tipo_reporte)
{
case 2: switch(opcion)
{
case 1: guarda_datos(No_max_arch);
archivo_salida);
}
}
}
}

```

```

        fechas();
        gen_arch_tabla();
        desplegar(pausa);
    break;
    case 2: archivo_salida();
        fechas();
        fallas_semanales(No_max_arch);
        desplegar(pausa);
        break;
    }
break;
case 3: switch(opcion)
{
    case 1: guarda_datos(No_max_arch);
        archivo_salida();
        fechas();
        gen_arch_tabla();
        desplegar(pausa);
        break;
    case 2: archivo_salida();
        fechas();
        fallas_mensuales(No_max_arch);
        desplegar(pausa);
        break;
    }
break;
}
else
{ printf("\nPresione cualquier tecla para continuar"); getch(); }
system("cd..");
}
/.....
Esta función genera los nombres de los archivos que deben existir para
que sea posible generar la tabla resumen semanal y mensual. Dichos
nombres corresponden a los archivos que contienen la tabla resumen
diaria y la tabla resumen semanal, respectivamente.
La fecha de inicio corresponde a la introducida por el usuario.
.....
void arch_diasem(int No_max_arch, int Incremento)
{
    int i = 0;
    char paso(4);

    struct guarda
    { char inicial(4); } fech[3];

    struct checa_fecha
    { int dia_max; } mensual[12] = {31,28,31,30,31,30,31,30,30,30,30,31};

    if (atoi(ano) == 98)
        mensual[1].dia_max = 29;
    strcpy(fech[0].inicial, dia);
    strcpy(fech[1].inicial, mes);
    strcpy(fech[2].inicial, ano);
    switch(tipo_reporte)
    {
        case 2: tipo_reporte = 1;
            if (opcion == 1) opcion = 2;
            else if (opcion == 2) opcion = 3;
        break;
        case 3: tipo_reporte = 2;
        break;
    }
}

```

```

do
{
if (atoi(dia) < mensual[atoi(mes) - 1].dia_max)
{
fechas(); strcpy(real[i].nombre, fech_rep);
archivo_salida(); strcpy(archivo[i].nombre, Tipo_Arch);
itoa(atoi(dia) + incremento, dia, 10);
if (atoi(dia) < 10 && dia[0] != '0')
{ strcpy(paso, dia); strcpy(dia, "0"); strcat(dia, paso); }
}
i++;
}
else
{
if (atoi(dia) == mensual[atoi(mes) - 1].dia_max)
{
fechas(); strcpy(real[i].nombre, fech_rep);
archivo_salida(); strcpy(archivo[i].nombre, Tipo_Arch);
if (atoi(mes) == 12 && atoi(dia) == mensual[atoi(mes)-1].dia_max)
{
strcpy(dia, "01"); strcpy(mes, "01");
itoa(atoi(ano) + 1, ano, 10);
}
else
{
strcpy(dia, "01");
itoa(atoi(mes) + 1, mes, 10);
if (atoi(mes) < 10 && mes[0] != '0')
{
strcpy(paso, mes); strcpy(mes, "0");
strcat(mes, paso);
}
}
}
i++;
}
else
{
itoa(atoi(dia) - mensual[atoi(mes) - 1].dia_max, dia, 10);
if (atoi(dia) < 10 && dia[0] != '0')
{ strcpy(paso, dia); strcpy(dia, "0"); strcat(dia, paso); }
if (atoi(mes) == 12)
{ strcpy(mes, "01"); itoa(atoi(ano) + 1, ano, 10); }
else
{
itoa(atoi(mes) + 1, mes, 10);
if (atoi(mes) < 10 && mes[0] != '0')
{
strcpy(paso, mes); strcpy(mes, "0");
strcat(mes, paso);
}
}
fechas(); strcpy(real[i].nombre, fech_rep);
archivo_salida(); strcpy(archivo[i].nombre, Tipo_Arch);
itoa(atoi(dia) + 7, dia, 10);
if (atoi(dia) < 10 && dia[0] != '0')
{ strcpy(paso, dia); strcpy(dia, "0"); strcat(dia, paso); }
}
i++;
}
}
if (i == No_max_arch)
strcpy(dia, fina); fech_rep);
}while(i != No_max_arch);
strcpy(dia, fech[0].inicial);
strcpy(mes, fech[1].inicial);
strcpy(ano, fech[2].inicial);
}

```

```

.....
Esta función permite generar la tabla de fallas semanales.
.....
void fallas_semanales(int No_max_arch)
{
    FILE *In;
    int i, num_lin, num_err = 0;
    char cad_arch[90];

    out = fopen(Tipo_Arch, "w");
    encabezado();
    for (i = 0; i < No_max_arch; i++)
    {
        num_lin = 0;
        In = fopen(archivo[i].nombre, "rt");
        while (!fgets(cad_arch, 90, In) != NULL)
            num_lin++;
        fclose(In);
        fprintf(out, "%s %s\n", "FECHA:", real[i].nombre);
        if (num_lin > 8)
        {
            num_lin = 0;
            In = fopen(archivo[i].nombre, "rt");
            while (!fgets(cad_arch, 90, In) != NULL)
            {
                cad_arch[strlen(cad_arch) - 1] = '\0';
                num_lin++;
                if (num_lin > 8)
                    { fprintf(out, "%s\n", cad_arch); num_err++; }
            }
            fclose(In);
        }
        else
            fprintf(out, "%s\n", "NO SE ENCONTRARON FALLAS");
    }
    fprintf(out, "\nPORCENTAJE SEMANAL DE DATOS PERDIDOS: %5.2f %ln",
        (num_err*100.0)/2098.0);
    fclose(out);
}

.....
Esta función permite generar la tabla de fallas manuales.
.....
void fallas_mensuales(int No_max_arch)
{
    FILE *In;
    int i, bandera, cont_fallas = 0;
    char cad_ant[85], cad_arch[85];

    In = fopen(archivo[No_max_arch - 1].nombre, "r");
    while (!fscanf(In, "%s", cad_arch) != EOF)
    {
        cont_fallas++;
        if (cont_fallas == 16)
            strcpy(dia_final, cad_arch);
        }
        fclose(In);
        cont_fallas = 0;
        out = fopen(Tipo_Arch, "w");
        encabezado();
        for (j = 0; j < No_max_arch; j++)
        {
            bandera = 0;
            In = fopen(archivo[j].nombre, "r");

```

```

while (fgets(cad_arch, 85, in) != NULL)
{
    if (strcmp(cad_arch, "\n") != 0)
    {
        cad_arch[strlen(cad_arch) - 1] = '\0';
        if (cad_arch[0] == 'F' && cad_arch[1] == 'E'
            && cad_arch[2] == 'C' && cad_arch[3] == 'H')
        {
            strcpy(cad_ent, cad_arch); bandera = 1;
        }
        else
        {
            if (strcmp(cad_arch, "NO SE ENCONTRARON FALLAS") != 0 &&
                strlen(cad_arch) != 48)
            {
                if (cad_arch[15] == '?' || cad_arch[26] == '?' ||
                    cad_arch[36] == '?' || cad_arch[48] == '?')
                {
                    if (bandera == 1)
                    {
                        fprintf(out, "\n%s\n", cad_ent); bandera = 0;
                        fprintf(out, "%s\n", cad_arch);
                    }
                    else
                        fprintf(out, "%s\n", cad_arch);
                    cont_fallas++;
                }
            }
        }
    }
}
fclose(in);
fprintf(out, "\nPORCENTAJE MENSUAL DE DATOS PERDIDOS: %5.2f %\n",
        (cont_fallas*100.0)/8384.0);
fclose(out);
}

```

.....
 Esta función guarda los datos (datos de los archivos).


```

void guarda_datos(int No_max_arch)
{
    FILE *in;
    int i, j, k, cont;
    float temp;
    char temporal[10], cad_arch[85];

    struct checa_posic
    { int posicion[7]; }datos = {22,29,36,43,48,53,60};

    if (tipo_reporte == 2)
    {
        datos.posicion[0] = 18; datos.posicion[1] = 24;
        datos.posicion[2] = 30; datos.posicion[3] = 36;
        datos.posicion[4] = 40; datos.posicion[5] = 44;
        datos.posicion[6] = 50;
        for (j = 0; j < No_max_arch; j++)
            strcpy(cad_datos[j].nombre, real[j].nombre);
    }
    for (k = 0; k < 7; k++)
    {
        for (j = 0; j < No_max_arch; j++)
        {
            cont = 0;
            in = fopen(archivo[j].nombre, "r");
            while (fscanf(in, "%s", cad_arch) != EOF)
            {
                cont++;
            }
        }
    }
}

```

```

if (j == No_max_arch - 1 && cont == 15 && tipo_reporte == 3)
    strcpy(dia_final, cad_arch);
if (cont == datos.posicion[k])
    valores.ENTRADA[i] = atof(cad_arch);
if (cont == datos.posicion[k]+2 && tipo_reporte == 3)
    strcpy(real[i].nombre, cad_arch);
fclose(in);
for (j = 0; j < No_max_arch - 1; j++)
    for (i = No_max_arch - 1; i > 0; i--)
        if (valores.ENTRADA[i] < valores.ENTRADA[i-1])
            {
            temp = valores.ENTRADA[i];
            valores.ENTRADA[i] = valores.ENTRADA[i-1];
            valores.ENTRADA[i-1] = temp;
            strcpy(temporal, real[i].nombre);
            strcpy(real[i].nombre, real[i-1].nombre);
            strcpy(real[i-1].nombre, temporal);
            }
/* Mínimo valor en batería y máximo en corriente consumida,
tomando en cuenta que ésta última es negativa */
if (k == 3 || k == 5)
    {
    valores.SALIDA[k] = valores.ENTRADA[0];
    strcpy(fech_sem[k].nombre, real[0].nombre);
    }
else /* Los valores máximos de los demás valores */
    {
    valores.SALIDA[k] = valores.ENTRADA[No_max_arch - 1];
    strcpy(fech_sem[k].nombre, real[No_max_arch - 1].nombre);
    }
if (tipo_reporte == 2)
    for (j = 0; j < No_max_arch; j++)
        strcpy(real[j].nombre, solo_datos[j].nombre);
}
}
/*****
Esta función genera el archivo de los datos seleccionados
*****/
void gen_arch_tabla(void)
{
    out = fopen(Tipo_Arch, "w");
    encabezado();
    fprintf(out, "%-10s%-18s\n", "UNIDAD", "FECHA[MM-DD-AA]");
    fprintf(out, "%-37s %8.3f %9s %s\n", "Irradiancia", valores.SALIDA[0], "Wh/m2",
    fech_sem[0].nombre);
    fprintf(out, "%-37s %8.3f %9s %s\n", "Voltaje máximo del Módulo", valores.SALIDA[1], "Vcd",
    fech_sem[1].nombre);
    fprintf(out, "%-37s %8.3f %9s %s\n", "Voltaje máximo del Sistema", valores.SALIDA[2], "Vcd",
    fech_sem[2].nombre);
    fprintf(out, "%-37s %8.3f %9s %s\n", "Voltaje mínimo de Batería", valores.SALIDA[3], "Vcd",
    fech_sem[3].nombre);
    fprintf(out, "%-37s %8.3f %9s %s\n", "Corriente generada", valores.SALIDA[4], "Ah/día",
    fech_sem[4].nombre);
    fprintf(out, "%-37s %8.3f %9s %s\n", "Corriente consumida", valores.SALIDA[5], "Ah/día",
    fech_sem[5].nombre);
    fprintf(out, "%-37s %8.3f %9s %s\n", "Corriente consumida por controlador", valores.SALIDA[6],
    "Ah/día", fech_sem[6].nombre);
    fclose(out);
}

```

```

if (j == No_max_arch - 1 && cont == 15 && tipo_reporte == 3)
  strcpy(dia_final, cad_arch);
if (cont == datos.posicion(k))
  valores.ENTRADA[j] = atof(cad_arch);
if (cont == datos.posicion(k)+2 && tipo_reporte == 3)
  strcpy(real[j].nombre, cad_arch);
}
fclose(in);
for (j = 0; j < No_max_arch - 1; j++)
for (i = No_max_arch - 1; i > 0; i--)
if (valores.ENTRADA[i] < valores.ENTRADA[i-1])
{
temp = valores.ENTRADA[i];
valores.ENTRADA[i] = valores.ENTRADA[i-1];
valores.ENTRADA[i-1] = temp;
strcpy(temporal, real[i].nombre);
strcpy(real[i].nombre, real[i-1].nombre);
strcpy(real[i-1].nombre, temporal);
}
/* Mínimo valor en batería y máximo en corriente consumida,
tomando en cuenta que ésta última es negativa */
if (k == 3 || k == 5)
{
valores.SALIDA[k] = valores.ENTRADA[0];
strcpy(fech_sem[k].nombre, real[0].nombre);
}
else /* Los valores máximos de los demás valores */
{
valores.SALIDA[k] = valores.ENTRADA[No_max_arch - 1];
strcpy(fech_sem[k].nombre, real[No_max_arch - 1].nombre);
}
if (tipo_reporte == 2)
for (j = 0; j < No_max_arch; j++)
strcpy(real[j].nombre, solo_datos[j].nombre);
}
}
/*****
Esta función genera el archivo de los datos seleccionados
*****/
void gen_arch_tabla(void)
{
out = fopen(Tipo_Arch, "w");
encabezado();
fprintf(out, "%-10s%-18s\n", "UNIDAD", "FECHA[MM-DD-AA]");
fprintf(out, "%-37s %8.3f %-9s %s\n", "Irradiancia", valores.SALIDA[0], "Wh/m2",
fech_sem[0].nombre);
fprintf(out, "%-37s %8.3f %-9s %s\n", "Voltaje máximo del Módulo", valores.SALIDA[1], "Vcd",
fech_sem[1].nombre);
fprintf(out, "%-37s %8.3f %-9s %s\n", "Voltaje máximo del Sistema", valores.SALIDA[2], "Vcd",
fech_sem[2].nombre);
fprintf(out, "%-37s %8.3f %-9s %s\n", "Voltaje mínimo de Batería", valores.SALIDA[3], "Vcd",
fech_sem[3].nombre);
fprintf(out, "%-37s %8.3f %-9s %s\n", "Corriente generada", valores.SALIDA[4], "Ah/día",
fech_sem[4].nombre);
fprintf(out, "%-37s %8.3f %-9s %s\n", "Corriente consumida", valores.SALIDA[5], "Ah/día",
fech_sem[5].nombre);
fprintf(out, "%-37s %8.3f %-9s %s\n", "Corriente consumida por controlador", valores.SALIDA[6],
"Ah/día", fech_sem[6].nombre);
fclose(out);
}

```

```

/*****
Esta función genera el encabezado del archivo a generar
*****/
void encabezado(void)
{
    struct paso
    {
        char reporte[25];
    } selec[2] = {"TABLA RESUMEN SEMANAL", "TABLA RESUMEN MENSUAL"};

    if (opcion == 2)
    {
        strcpy(selec[0].reporte, "TABLA DE FALLAS SEMANAL");
        strcpy(selec[1].reporte, "TABLA DE FALLAS MENSUAL");
        fprintf(out, "%s\n", selec[0].reporte);
        fprintf(out, "%s\n", selec[1].reporte);
        fprintf(out, "%48s\n", selec[tipo reporte-2].reporte);
        fprintf(out, "%44s %s\n", "CORRESPONDIENTE AL SISTEMA", cad leyenda);
        fprintf(out, "%33s %s al %s\n", "CON FECHA DEL", fech rep, dia fin);
        fprintf(out, "%s\n", "*****");
        fprintf(out, "%s\n", "*****");
        if (opcion == 1)
            { fprintf(out, "%-38s", "VARIABLE"); fprintf(out, "%-11s", "CANTIDAD"); }
    }
}

/*****
Esta función despliega los datos generados, si se genera el archivo.
*****/
void desplegar(int pausa)
{
    FILE *in;
    char mas[25];

    system("cls");
    if ((in = fopen (Tipo_Arch, "rt")) == NULL)
    {
        sound(250); gotoxy(1,1);
        printf("**** No se generó el archivo: %s ****", Tipo_Arch);
        delay(1000); nosound(); gotoxy(1,2);
        printf("Presione cualquier tecla para continuar");
        getch();
    }
    else
    {
        fclose(in);
        if (pausa)
        {
            strcpy(mas, "more <"); strcat(mas, Tipo_Arch);
            system(mas); getch();
        }
        else
        {
            strcpy(mas, "type "); strcat(mas, Tipo_Arch);
            system(mas);
            printf("\n\nEspere un momento...");
            delay(2000);
        }
    }
}

```

C.4 PROGRAMA MENSUAL.C

.....
 Este programa está elaborado en lenguaje C utilizando el software BorlandC
 versión 3.0.

Contiene las rutinas para generar el archivo mensual del programa REPORTES.

Elaboró: LETICIA HERNANDEZ RAMIREZ
 Supervisó: Dr. FRANCISCO VILLASEÑOR RIVERO (Instituto de Investigaciones
 eléctricas).

Marzo 1995


```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <dos.h>
#include "mensual.h"

FILE *output;

.....
Esta función permite generar el archivo mensual de los datos especificados
en el análisis de requerimientos. Los datos son generados tomando como fecha
de inicio al día primero del mes que especifica el usuario.
.....
void genera_archivo_mensual(void)
{
    char dia_mes[3], mes_mes[3], ano_mes[3];
    strcpy(dia_mes, "01"); strcpy(mes_mes, mes); strcpy(ano_mes, ano);
    system("cls");
    strcpy(dia, "01");
    archivo_salida(); system("cd \\dmacs\\rep");
    mensual_FDMACS();
    system("cd \\dmacs");
    strcpy(dia, dia_mes); strcpy(mes, mes_mes); strcpy(ano, ano_mes);
}

.....
Esta función genera el comando necesario para obtener los datos mensuales,
mediante el generador de reportes del programa FIX DMACS. Dichos nombres
corresponden al nombre del archivo que los genera, sólo que con la extensión
".rpt" en lugar de ".gen". P.ej, en lugar de mens1.gen se genera el
archivo mens1.rpt con la fecha que se especificó.
.....
void mensual_FDMACS(void)
{
    FILE *in;
    int bandera_error = 0;
    char sistema[3], paso[4], gen_mes[30], cad_arch[90], arch_FDMACS[13];

    struct fecha
    { int dia_max; } mensual[12] = {31,28,31,30,31,30,31,30,30,30,30,31};

    if (atoi(ano) == 98)
        mensual[1].dia_max = 29;
    output = fopen(Tipo_Arch, "wt");
    itoaNum(sis, sistema, 10);
    strcpy(arch_FDMACS, "MENS"); strcat(arch_FDMACS, sistema);
    strcat(arch_FDMACS, ".RPT");
}
    
```

```

do
{
  fechas(); strcpy(gen_men,"GEN /RMENS"); strcat(gen_men, sistema);
  strcat(gen_men, " /D"); strcat(gen_men, fech_repl); strcat(gen_men, " /F");
  system("cd \\dmacs");
  printf("%s\n",gen_men);
  system(gen_men);
  system("cd \\dmacs\\rep");
  if ((lin = fopen(arch_FDMACS, "rt")) == NULL)
  {
    system("cls");
    sound(300);
    printf("No se puede abrir el archivo: %s\n", arch_FDMACS);
    printf("Es posible que exista un error en el archivo MENS%.GEN\n", sistema);
    printf("que se encuentra en el directorio C:\\DMACS\\REP.\n");
    delay(50); nosound();
    printf("\nPresione cualquier tecla para continuar");
    getch();
    fclose(output);
    remove(arch_FDMACS);
    remove(Tipo_Arch);
    bandera_error = 1;
    toa(mensual[atoi(mes)-1].dia_max + 1, dia, 10);
  }
  else
  {
    if (!bandera_error)
      while (fgets(cad_arch, 90, in) != NULL)
      {
        cad_arch[strlen(cad_arch) - 1] = '\0';
        strcat(cad_arch, "\n");
        fputs(cad_arch, output);
      }
    fclose(in);
    remove(arch_FDMACS);
    toa(atoi(dia) + 1, dia, 10);
    if (atoi(dia) < 10 && dia[0] != '0')
      { strcpy(paso, dia); strcpy(dia, "0"); strcat(dia, paso); }
  }
} while (atoi(dia) <= mensual[atoi(mes)-1].dia_max);
if (!bandera_error) fclose(output);
}

```

C.5 PROGRAMA GRAFICA.C

Este programa está elaborado en lenguaje C utilizando el software QuickC v2.6 y las herramientas de Quinn Curtis para graficación.

Se grafican los siguientes datos:

- Voltaje producido por el módulo (VfV)
- Corriente producida por el módulo (IfV)
- Voltaje en la batería (VfB)
- Corriente de carga (IC)
- Radiación solar de plano (RSP)
- Corriente en la batería (IBB)

En la gráfica también se incluye una tabla resumen que contiene los siguientes datos:

- Número e identificador del sistema
- Fecha de la gráfica

- Irradiancia, voltaje máximo del módulo, voltaje máximo del sistema, voltaje mínimo en batería, corriente generada, corriente consumida por la carga y por el controlador.

Elaboró: LETICIA HERNANDEZ RAMIREZ
Supervisó: Dr. FRANCISCO VILLASEÑOR RIVERO (Instituto de Investigaciones eléctricas).

Marzo 1996.

```

#include <math.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <conio.h>
#include <stdlib.h>
#include <process.h>
#include "asynch.h"
#include "worddir.h"
#include "segraph.h"
#include "realtype.h"

void acceso(void);
void grafica (void);

int pausa;
char Num_sist[3], Arch_graf[13], Arch_FDMACS[12], fecha[10], Titulo[45];
char leyenda[16];

/*****
PROGRAMA PRINCIPAL
*****/
main(int argc, char *argv[])
{
    if (argc == 6)
        exit(0);
    strcpy(Num_sist,argv[1]);
    strcpy(leyenda,argv[2]);
    strcpy(fecha,argv[3]);
    strcpy(Arch_graf,argv[4]);
    pausa = atoi(argv[5]);
    acceso();
    grafica();
}

/*****
Genera el nombre del archivo de datos que se ha generado previamente en el
software de FIX DMACS.
*****/
void acceso(void)
{
    strcpy(Arch_FDMACS,"grpan");
    strcat(Arch_FDMACS,Num_sist);
    strcat(Arch_FDMACS,".rpt");
}

/*****
GENERA LA GRAFICA DIARIA DE CADA SISTEMA
Genera la gráfica correspondiente al sistema que se seleccionó mediante
las herramientas gráficas de Quinn-Curtis
*****/
void grafica(void)
{
    FILE *In;

```

```

int i, j = 0, cont = 0, *color, *linea;
char paso[15], T1X[10], T1Y1[10], T1Y2[10];
char cad_val[12];

realtype Irrad, Igen, Icon, Icontr, Ibat, min, max;
realtype *Tiempo, *VFV, *IFV, *VBB, *IC, *RSP;
string80 *cad_leys;

Irrad = 0.00; Igen = 0.00; Icon = 0.00; Icontr = 0.00; Ibat = 0.00;

/***** Reservando memoria para los datos a graficar *****/
Tiempo = GetMem(288); if (BadMem(Tiempo)) exit(0);
VFV = GetMem(288); if (BadMem(VFV)) exit(0);
IFV = GetMem(288); if (BadMem(IFV)) exit(0);
VBB = GetMem(288); if (BadMem(VBB)) exit(0);
IC = GetMem(288); if (BadMem(IC)) exit(0);
RSP = GetMem(288); if (BadMem(RSP)) exit(0);
color = GetIntMem(5); if (BadMem(color)) exit(0);
linea = GetIntMem(5); if (BadMem(linea)) exit(0);
cad_leys = (string80 *) calloc(5, sizeof(string80));
if (BadMem(cad_leys)) exit(0);

color[0] = 11; color[1] = 14; color[4] = 12;
color[2] = 10; color[3] = 13;

linea[0] = 2; linea[1] = 0; linea[4] = 3;
linea[2] = 1; linea[3] = 0;

strcpy(cad_leys[0], "Irradiancia (Wh/m2/200)");
strcpy(cad_leys[1], "Voltaje Modulo (V)");
strcpy(cad_leys[2], "Voltaje Bateria (V)");
strcpy(cad_leys[3], "Corriente Gen. (A)");
strcpy(cad_leys[4], "Corriente Cons. (A)");

/***** Accesando el archivo que genera FIX DMACS *****/
if ((lin = fopen("Arch_FDMACS", "r")) == NULL)
{
    system("cls");
    printf("Hay un error en la apertura del archivo fuente de datos\n");
    printf("Presione cualquier tecla para terminar la ejecución");
    getch(); exit(0);
}

/***** Leyendo datos del archivo *****/
while ((facn = fgetc(lin, "%s", cad_val)) != EOF)
{
    switch(j)
    {
        case 0: j++; break;
        case 1: if (strcmp(cad_val, "????????") == 0)
            VFV[cont] = VFV[cont-1];
            else
                VFV[cont] = (realtype) atof(cad_val);
            j++;
        break;
        case 2: if (strcmp(cad_val, "????????") == 0 && (cont > 65 &&
            cont < 223)) { atof(cad_val) = 0.00 && (cont > 65 &&
            cont < 223); }
            { IFV[cont] = IFV[cont-1]; Igen = Igen + IFV[cont-1]; }
            else
            {
                IFV[cont] = (realtype) atof(cad_val);
                Igen = Igen + (realtype) atof(cad_val);
            }
    }
}

```

```

}++;
break;
case 3: if (strcmp(cad_val,"????????") == 0 && cont > 0)
    VBB[cont] = VBB[cont-1];
    else
        if (atof(cad_val) == 0.00 || atof(cad_val) < 0.00)
            VBB[cont] = VBB[cont-1];
        else
            VBB[cont] = (realtype)atof(cad_val);
    }++;
break;
case 4: if (strcmp(cad_val,"????????") == 0 && cont > 0)
    IC[cont] = IC[cont-1]; lcon = lcon + IC[cont-1];
    else
        if ((atof(cad_val) == 0.00 && (cont > 48 && cont < 92))
            || (atof(cad_val) == 0.00 && (cont > 216 && cont < 278)))
            IC[cont] = IC[cont-1]; lcon = lcon + IC[cont-1];
        else
            if ((atof(cad_val) != 0.00 && (cont > 0 && cont < 49))
                || (atof(cad_val) != 0.00 && (cont > 277)))
                IC[cont] = 0.00; lcon = lcon + 0.00;
            else
                IC[cont] = (realtype)atof(cad_val);
                lcon = lcon + (realtype)atof(cad_val);
        }
    }++;
break;
case 5: if (strcmp(cad_val,"????????") == 0 &&
    (cont > 65 && cont < 223) || (atof(cad_val) == 0.00 &&
    (cont > 65 && cont < 223)))
    RSP[cont] = RSP[cont-1];
    lrrad = lrrad + (RSP[cont-1]*200.00);
    else
        RSP[cont] = (realtype)(atof(cad_val)/200);
        lrrad = lrrad + (realtype)(atof(cad_val));
    }++;
break;
case 8: if (strcmp(cad_val,"????????") == 0 || stof(cad_val) == 0.00)
    lcontr = lcontr + IFV[cont] + IC[cont] - lbat;
    else
        lbat = (realtype)atof(cad_val);
        lcontr = lcontr + IFV[cont] + IC[cont] - lbat;
    cont++; } = 0;
break;
}
fclose(in);
/***** Generando el vector para dibujar el eje X *****/
for (i = 0; i < cont; i++)
    Tiempo[i] = (realtype)(0.083333*i);

```

/***** Abriendo archivo para guardar la grafica plotada *****/

```
PlotterFileOn(Arch_graf,1);
SetupPlotter(0,2,0,1,0,j,0,9,0,9,0);
```

/***** Iniclando gráfica *****/
 InitSEGGraphics(-3,defaultdirect,2);

```
SetPercentWindow(0,0,0,12,1,0,1,0,1);
SetCurrentWindow(1);
SetWin2PlotRatio(1,0,12,0,05,0,12,0,12);
SetAxesType(0,0);
ScalePlotArea(0,0,0,0,24,0,20,0);
SetXYIntercepts(0,0,0,0);
SelectColor(15);
DrawYAxis(2,0,0);
LabelYAxis(1,0);
strcpy(TitY1,"Volts");
TitleYAxis(TitY1,0);
DrawXAxis(1,0,0);
LabelXAxis(4,0);
strcpy(TitX,"Dia (hrs)");
TitleXAxis(TitX,0);
```

```
LinePlotData(Tiempo,VFV, cont, 14, 0);
LinePlotData(Tiempo,VBB, cont, 10, 1);
LinePlotData(Tiempo,RSP, cont, 11, 2);
```

```
ScalePlotArea(0,0,-10,0,24,0,10,0);
SetXYIntercepts(24,0,0,0);
SelectColor(7);
DrawYAxis(2,0,1);
LabelYAxis(1,1);
DrawXAxis(24,0,0);
strcpy(TitY2,"Amp");
TitleYAxis(TitY2,1);
```

```
LinePlotData(Tiempo,IC, cont, 12, 3);
LinePlotData(Tiempo,IFV, cont, 13, 0);
```

```
SetPercentWindow(0,126,0,16,0,42,0,38,2);
SetWin2PlotRatio(2,0,1,0,1,0,05,0,05);
SetCurrentWindow(2);
BorderCurrentWindow(15);
settextstyleXX(1,0,5,5);
strcpy(Titulo,"Sistema ");
strcat(Titulo, Num_sist);
strcat(Titulo, " "); strcat(Titulo,leyenda); strcat(Titulo, " ");
strcat(Titulo, fecha);
LabelGraphWindow(15,850,Titulo,0,0);
```

```
settextstyleXX(2,0,4,4);
sprintf(paso,"%*.*f",2,3,1/rad/12);
strcpy(Titulo,"Irradiancia = ");
strcat(Titulo,paso);
strcat(Titulo," Wh/m2");
LabelGraphWindow(15,750,Titulo,0,0);
```

```
FindMinMax(VFV,cont,&min,&max);
sprintf(paso,"%*.*f",2,3,max);
strcpy(Titulo,"Vmax Modulo = ");
strcat(Titulo,paso);
strcat(Titulo," Vcd");
LabelGraphWindow(15,850,Titulo,0,0);
```

```

FindMinMax(VBB,cont,&min,&max);
sprintf(paso,"%*.*f",2,3,max);
strcpy(Titulo,"Vmax Sistema = ");
strcpy(Titulo,paso);
strcat(Titulo," Vcd");
LabelGraphWindow(15,550,Titulo,0,0);

sprintf(paso,"%*.*f",2,3,min);
strcpy(Titulo,"Vmin Bateria = ");
strcpy(Titulo,paso);
strcat(Titulo," Vcd");
LabelGraphWindow(15,450,Titulo,0,0);

sprintf(paso,"%*.*f",2,3,gen/12);
strcpy(Titulo,"I generada = ");
strcpy(Titulo,paso);
strcat(Titulo," Ah/dia");
LabelGraphWindow(15,350,Titulo,0,0);

sprintf(paso,"%*.*f",2,3,icon/12);
strcpy(Titulo,"I consumida = ");
strcpy(Titulo,paso);
strcat(Titulo," Ah/dia");
LabelGraphWindow(15,250,Titulo,0,0);

sprintf(paso,"%*.*f",2,3,contr/12);
strcpy(Titulo,"I consum contr = ");
strcpy(Titulo,paso);
strcat(Titulo," Ah/dia");
LabelGraphWindow(15,150,Titulo,0,0);

SetPercentWindow(0,1,0,0,1,0,0,1,3);
SetCurrentWindow(3);
StringLegends(cad_leya, color, linea, 5, 0);

EjectPage();
if (pausa == 1)
    getch();
CloseSEGraphics();
ClosePlotter();
PlotterFileOff();
free(color); free(linea); free(cad_leya);
free(Tiempo); free(VFV); free(IFV); free(VBB); free(C); free(RSP);
}

```

C.6 PROGRAMA DIARIOS.C

Este programa está elaborado en lenguaje C utilizando el software QuikC v2.5 y las herramientas de Quinn Curtis para espacio de memoria.

Realiza tres de las funciones del generador de reportes:

- Tabla resumen diaria,
 - Tabla de fallas diaria y
 - Tabla de datos diaria,
- donde cada una tiene un encabezado que contiene los siguientes datos:
- Tipo de tabla
 - Número e identificador del sistema
 - Fecha del reporte
 - Datos calculados, fallos en la adquisición de datos y datos obtenidos cada 5 minutos durante el día que especifica la fecha, respectivamente.

Elaboró: LETICIA HERNANDEZ RAMIREZ

Supervisado: Dr. FRANCISCO VILLASEÑOR RIVERO (Instituto de Investigaciones eléctricas).

Marzo 1995

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <process.h>
#include "asyn2.h"
#include "worldr.h"
#include "segraph.h"
#include "reatype.h"

```

```

void acceso(void);
void archivo_salida(void);
void tabla_resumen(void);
void tabla_fallas(void);
void tabla_datos(void);

```

```

void encabezado(void);
void desplegar(void);
void no_fallas(void);

```

```

int pausa, opcion;
char Num_siat[3], Tipo_Arch[15], Arch_FDMACS[12], fecha[10];
char leyenda[15], dia[3], mes[3], ano[3];
FILE *in, *out;

```

PROGRAMA PRINCIPAL

```

main(int argc, char *argv[])
{
if (argc != 8)
exit(1);
strcpy(Num_siat, argv[1]);
strcpy(leyenda, argv[2]);
strcpy(dia, argv[3]);
strcpy(mes, argv[4]);
strcpy(ano, argv[5]);
opcion = atoi(argv[6]);
strcpy(fecha, mes); strcat(fecha, "-");
strcat(fecha, dia); strcat(fecha, "-");
strcat(fecha, ano);
acceso();
archivo_salida();
pausa = atoi(argv[7]);
switch(opcion)
{
case 2: tabla_resumen(); break;
case 3: tabla_fallas(); break;
case 4: tabla_datos(); break;
}
}

```

Genera el nombre del archivo de datos que se ha generado previamente en el software de FIX DMACS.

```

void accesar(void)
{
strcpy(Arch_FDMACS, "gran");
strcat(Arch_FDMACS, Num_sist);
strcat(Arch_FDMACS, ".rpt");
}

.....
          GENERA LA TABLA RESUMEN DIARIA DE CADA SISTEMA
Genera la tabla resumen correspondiente al sistema fotovoltaico que se
seleccionó, mediante las herramientas Quinn-Curtis.
.....
void tabla_resumen(void)
{
int cont = 0, j = 0;
realtype Irrad, Igen, Icon, Icontr, Ibat, min, max;
realtype *VFV, *IFV, *VBB, *IC, *RSP;
char cad_val[12];

Irrad = 0.00; Igen = 0.00; Icon = 0.00; Icontr = 0.00; Ibat = 0.00;

/***** Reservando memoria para los datos a calcular *****/

VFV = GetMem(288); if (BadMem(VFV)) exit(0);
IFV = GetMem(288); if (BadMem(IFV)) exit(0);
VBB = GetMem(288); if (BadMem(VBB)) exit(0);
IC = GetMem(288); if (BadMem(IC)) exit(0);
RSP = GetMem(288); if (BadMem(RSP)) exit(0);

/***** Accesando el archivo que genera FIX DMACS *****/

if ((lin = fopen (Arch_FDMACS, "r")) == NULL)
{
system("cls");
printf("Hay un error en la apertura del archivo fuente de datos\n");
printf("Presione cualquier tecla para terminar la ejecución");
getch(); exit(2);
}

/***** Leyendo datos del archivo *****/

while (fscanf(lin, "%s", cad_val) != EOF)
{
switch(j)
{
case 0: j++; break;
case 1: if (strcmp(cad_val, "????????") == 0)
VFV[cont] = VFV[cont-1];
else
VFV[cont] = (realtype)atof(cad_val);
j++;
break;
case 2: if (strcmp(cad_val, "????????") == 0 && (cont > 65 &&
cont < 223) || (atof(cad_val) == 0.00 && (cont > 85 &&
cont < 223)))
{ IFV[cont] = IFV[cont-1]; Igen = Igen + IFV[cont-1]; }
else
{ IFV[cont] = (realtype)atof(cad_val);
Igen = Igen + (realtype)atof(cad_val);
}
j++;
break;
}
}
}

```

```

case 3: if (strcmp(cad_val,"????????") == 0 && cont > 0)
    VBB[cont] = VBB[cont-1];
    else
    {
        if (atof(cad_val) == 0.00 || atof(cad_val) < 0.00)
            VBB[cont] = VBB[cont-1];
        else
            VBB[cont] = (realtype)atof(cad_val);
    }
    j++;
break;
case 4: if (strcmp(cad_val,"????????") == 0 && cont > 0)
    { IC[cont] = IC[cont-1]; Icon = Icon + IC[cont-1]; }
    else
    {
        if ((atof(cad_val) == 0.00 && (cont > 48 && cont < 92))
            || (atof(cad_val) == 0.00 && (cont > 216 && cont < 278)))
            { IC[cont] = IC[cont-1]; Icon = Icon + IC[cont-1]; }
        else
        {
            if ((atof(cad_val) != 0.00 && (cont > 0 && cont < 49))
                || (atof(cad_val) != 0.00 && (cont > 91 && cont < 217))
                || (atof(cad_val) != 0.00 && cont > 277))
                { IC[cont] = 0.00; Icon = Icon + 0.00; }
            else
            {
                IC[cont] = (realtype)atof(cad_val);
                Icon = Icon + (realtype)atof(cad_val);
            }
        }
    }
    j++;
break;
case 5: if (strcmp(cad_val,"????????") == 0 &&
    (cont > 65 && cont < 223) || (atof(cad_val) == 0.00 &&
    (cont > 65 && cont < 223)))
    { RSP[cont] = RSP[cont-1];
    Irrad = Irrad + (RSP[cont-1]*200.00);
    }
    else
    {
        RSP[cont] = (realtype)atof(cad_val)/200;
        Irrad = Irrad + (realtype)atof(cad_val);
    }
    j++;
break;
case 6: if (strcmp(cad_val,"????????") == 0 || atof(cad_val) == 0.00)
    { Icontr = Icontr + IFV[cont] + IC[cont] - Ibat;
    }
    else
    {
        Ibat = (realtype)atof(cad_val);
        Icontr = Icontr + IFV[cont] + IC[cont] - Ibat;
    }
    cont++; j = 0;
break;
}
fclose(in);
encabezado();
printf(out,"%-38s%-11s%-10s\n","VARIABLE","CANTIDAD","UNIDAD");
printf(out,"%-37s %8.3f %-9s\n","Irradiancia",Irrad/12,"Wh/m2");
FindMinMax(IFV,cont,&min,&max);
printf(out,"%-37s %8.3f %-9s\n","Voltaje máximo del Módulo",max,"Vcd");

```

```

FindMinMax(VBB,cont,&min,&max);
fprintf(out, "%-37s %8.3f %-9s\n", "Voltaje máximo del Sistema", max, "Vcd");
fprintf(out, "%-37s %8.3f %-9s\n", "Voltaje mínimo de Batería", min, "Vcd");
fprintf(out, "%-37s %8.3f %-9s\n", "Corriente generada", Igen/12, "Ah/día");
fprintf(out, "%-37s %8.3f %-9s\n", "Corriente consumida", Icon/12, "Ah/día");
fprintf(out, "%-37s %8.3f %-9s\n", "Corriente consumida por controlador", Icontr/12, "Ah/día");
fclose(out);
free(VFV);free(IFV);free(VBB);free(IC);free(RSP);
desplegar();
}

```

```

.....
GENERA LA TABLA DE FALLAS DE CADA SISTEMA
Busca las fallas que se encuentran en el archivo de datos que generó el
programa FIX DMACS.
.....

```

```

void tabla_fallas(void)
{
    int cont1 = 0;
    char error [120];
    if ((lin = fopen (Arch_FDMACS, "rt")) == NULL)
    {
        system("cls");
        printf("Hay un error en la apertura del archivo fuente de datos\n");
        printf("Presione cualquier tecla para terminar la ejecución");
        getch(); exit(2);
    }
    while (fgetc(error,120,in) != NULL )
    {
        error[strlen(error)-1] = '\0';
        if (error[15] == '?' || error[26] == '?' || error[36] == '?' || error[48] == '?' ||
            error[59] == '?' || error[70] == '?' || error[15] == '!' || error[26] == '!' ||
            error[36] == '!' || error[48] == '!' || error[59] == '!' || error[70] == '!' )
        {
            if (cont1 == 0)
                encabezado();
            strcat(error, "\n"); fputs(error, out);
            cont1 ++;
        }
    }
    fclose(in);
    if (cont1 > 0)
        { fclose(out); desplegar(); }
    else
        no_fallas();
}

```

```

.....
GENERA LA TABLA DE DATOS DIARIA DE CADA SISTEMA FOTOVOLTAICO
Genera la tabla de datos correspondiente al sistema fotovoltaico que se
seleccionó.
.....

```

```

void tabla_datos(void)
{
    char cad_arch[120];
    if ((lin = fopen (Arch_FDMACS, "rt")) == NULL)
    {
        system("cls");
        printf("Hay un error en la apertura del archivo fuente de datos\n");
        printf("Presione cualquier tecla para terminar la ejecución");
        getch(); exit(2);
    }
    encabezado();
    while (fgetc(cad_arch,120,in) != NULL )

```

```

    fputs(cad_arch,out);
    fclose(in); fclose(out);
    desplegar();
}

/.....
Esta función coloca un encabezado a cada archivo que se genera. Este
encabezado describe las características generales del sistema que se
selecciona, así como la fecha a la que corresponde el reporte.
/.....
void encabezado(void)
{
    struct paso
    {
        char reporte[25];
    } selec[3] = {"TABLA RESUMEN DIARIA", "TABLA DE FALLAS DIARIA",
        "TABLA DE DATOS DIARIA"};

    out = fopen(Tipo_Arch, "w");
    fprintf(out, "%s", ".....");
    fprintf(out, "%s\n", ".....");
    fprintf(out, "%47s\n", selec[opcion-2].reporte);
    fprintf(out, "%44s %s %s\n", "CORRESPONDIENTE AL SISTEMA", Num_sist, leyenda);
    fprintf(out, "%38s %s\n", "CON FECHA", fecha);
    fprintf(out, "%s", ".....");
    fprintf(out, "%s\n", ".....");
    if (opcion == 3) { opcion == 4
        fprintf(out, "%-10s%9s%11s%11s%11s%11s%11s\n",
            "HORA", "VFV", "IFV", "VBB", "IC", "RSP", "IBB");
    }
}

/.....
Esta función sólo se utiliza en caso de que no se encuentren errores en el
sistema que se seleccionó.
/.....
void no_fallas(void)
{
    out = fopen(Tipo_Arch, "w");
    fprintf(out, "%s", ".....");
    fprintf(out, "%s\n", ".....");
    fprintf(out, "%49s\n", "TABLA DE FALLAS DIARIA");
    fprintf(out, "%38s", "NO SE ENCONTRARON FALLAS");
    fprintf(out, "%s %s %s\n", "EN EL SISTEMA", Num_sist, leyenda);
    fprintf(out, "%38s %s\n", "CON FECHA", fecha);
    fprintf(out, "%s", ".....");
    fprintf(out, "%s\n", ".....");
    fclose(out);
    desplegar();
}

/.....
Esta función despliega los datos obtenidos de cada tabla, mediante los
comandos "more.com" y "type.com" de sistema operativo, dependiendo del
valor de "pausa".
/.....
void desplegar(void)
{
    char mas[25];

    if ((lin == fopen (Tipo_Arch, "rt")) == NULL)
    {
        printf("\nNo existe o no se generó el archivo: %s\n\n", Tipo_Arch);
        printf("Presione cualquier tecla para continuar"); getch();
    }
    else

```


Apéndice D

GLOSARIO

Base de datos.- Es un conjunto de datos almacenados en forma integrada y compartida; integrada porque puede estar compuesta de varios archivos independientes; y compartida porque varios usuarios diferentes pueden acceder a la misma fracción de la base de datos, incluso al mismo tiempo, y utilizarla con fines distintos.

Biomasa.- Toda la materia orgánica que existe en la naturaleza (árboles, algas marinas, animales, estiércol, etc.) y que sea susceptible de transformarse en energía.

Búffer.- Memoria temporal de una computadora.

Corto circuito.- Se dice que se produce un corto circuito cuando se unen las terminales positiva y negativa de un circuito. Si el circuito es de corriente alterna, la corriente eléctrica que circula entre ellas tiende a ser infinita.

Dirección de memoria.- Es el número que tiene asociado una localidad de memoria. Dicho número se utiliza para efectos de almacenamiento y recuperación de la información que está contenida en esa localidad de memoria.

Electrodo.- Elemento terminal en un circuito eléctrico.

Electrón.- Carga de electricidad negativa que forma parte del átomo.

Energía eólica.- La energía proveniente del viento.

Insolación.- Radiación solar que llega a una superficie en un periodo de tiempo. Generalmente se expresa en kilowatts-hora por metro cuadrado.

Kilowatt.- Se refiere a mil watts. Un watt es la potencia que tiene una máquina que efectúa un trabajo de un joule en un segundo.

Memoria expandida.- Es una área de memoria accesible a programas que pueden acceder memoria por debajo de los 640 K. La memoria expandida se divide en segmentos de 16 K llamados páginas. Cuando un programa requiere información de memoria expandida el manejador mapea o copia la página apropiada en el área llamada una página enmarcada y la transfiere al área de memoria superior. La memoria expandida requiere manejadores especiales como EMM386.EXE.

Programa multitareas.- Es aquel que permite a una computadora ejecutar varias tareas a la vez, en forma independiente.

PLC.- Sus siglas significan Controlador Lógico Programable. La National Electric Manufacturing Association (NEMA) lo define como un aparato electrónico de operación digital que utiliza una memoria programable para almacenar internamente las instrucciones que realicen funciones específicas de lógica secuencial, temporización, conteo y aritmética para controlar, a través de módulos de entrada/salida analógicos y digitales, diversos tipos de máquinas o procesos.

Semiconductividad.- Propiedad que tienen los cuerpos de actuar en determinadas condiciones ya sea como conductores de corriente o como aisladores. Dichas condiciones dependen de la pureza, la temperatura, el grado de iluminación, etc. a las que se encuentre expuesto en material.

Transductor.- Es aquel dispositivo que permite convertir un tipo de energía en otra (v.b. calorífica en eléctrica).

UCP.- Unidad central de proceso de una computadora. Está compuesta por la unidad de control que dirige todas las acciones del sistema y la unidad de aritmética y lógica (UAL) que es la encargada de realizar todas las operaciones de aritmética y lógica que involucren esas acciones.

Yarda.- Unidad de medida inglesa equivalente a 0.9144 metros.