1ej.

FACULTAD DE INGENIERIA

INGENIERIA DE PROGRAMACION PARA EL PROGRAMA DE ESTIMACION DE CARGA HORARIA PARA EL CENTRO DE INFORMACION Y CONTROL EN TIEMPO REAL DEL SISTEMA ELECTRICO NACIONAL

Tesis para optar a la Licenciatura en Ingeniería en Computación presenta :

INOUE KUBO, MOTOKI

México , D.F. 1981 **U.N.A.M.**

TESIS CON FALLA DE ORIGEN

Resumen.

El propósito de este trabajo es, introducir las nuevas técnicas de "Ingeniería de Programación" (Software Engineering) a fin de mejorar los métodos existentes de desarrollo de programación.

1) Introduce un nuevo enfoque de la metodología para el desarrollo de programación, empleando las técnicas de Diseño Estructurado y Programación Estructurada.

2) Presenta una aplicación de este nuevo enfoque al desarrollo de un proyecto de programación específico.

CONTENIDO PAGINA

CONTENIDO PAGINA

CONTENIDO PAGINA

CONTENIDO PAGINA

CONTENIDO PAGINA

CONTENIDO PAGINA

CAPITULO I. INTRODUCCION

I.1 Antecedentes

Puede afirmarse que en la década de los 60's empezó a desarrollarse la actividad de programación. Inicialmente, después de una fase muy corta de análisis, se procedía de inmediato a escribir código; se pensaba que sólo si se contaba con código se había avanzado en el proyecto.

Puede afirmar que la única metodología específica empleada en estos primeros sistemas de cómputo eran los diagramas de flujo, una forma esquemática de registrar desde su fuente la lógica de la operación de un programa pasando por los diferentes procesos de operación hasta llegar al producto final. Estos diagramas de flujo dan una imagen gráfica de las operaciones que hay que ejecutar y al nivel de programación representan detalladamente los pasos lógicos que tiene que realizar la computadora con objeto de llegar al resultado deseado. Sin embargo, estos diagramas no ayudan a darle a los programas la estructura y la flexibilidad que se requiere en la actualidad.

Pronto, sin embargo, la carencia de una metodología más extensa y adecuada para el desarrollo de programas, sobre todo al aumentar éstos de complejidad, llevó a los siguientes problemas:

- Sobre costos: Con frecuencia cuando el programa estaba apenas terminado en un 50% ya se había ejercido casi el 100% del presupuesto.

- Retraso en la entrega de los programas: Al principio, los retrasos eran de algunos meses, pero al aumentar su complejidad, estas demoras crecieron fuera de control.

- Dificultad en el mantenimiento y cambio a otro sistema: Debido a la carencia de un procedimeinto para estructurar los programas resultaba muy difícil mantenerlos; la falta de una documentación adecuada hacía que este proceso fuera muy caro. Ante la necesidad de cambiar programas de un sistema a otro, los programadores, ante programas mal estructurados, frecuentemente caían en la tentación de rediseñar y reprogramarlos totalmente, No era por lo tanto de extrañarse que los costos de mantenimiento del programa llegasen a ser del orden del 90% del costo total de ciclo de vida útil de estos.

- Baja confiabilidad : Por las mismas razones anteriores y la carencia de un plan adecuado de pruebas de estos programas, resultaban poco confiables en su operación.

- Inconformidad del Usuario : Era muy frecuente, que al terminar un programa, el usuario argumentara que realmente el programa terminado, no realizaba las funciones que él pensaba que debería de hacer. Esto se debía fundamentalmente a la falta de una definición precisa de cuales deberían ser los requerimientos del mismo. Se procedía a codificar sin contar con esta información en forma exacta.

- De Integración : Los primeros sistemas generalmente se diseñaban e implementaban no tomando en cuenta la interdependencia con otros. Este procedimiento desde luego minimizaba los costos de diseño pero complicaba futuros cambios a los programas cuando el usuario identificaba nuevas funciones que podían ser computarizadas y que deseaba incluir en el sistema. Cuando la empresa deseaba integrarlos a un sistema general de manejo de información, su estructura lo impedía.

I.2 Complejidad de Proyectos.

Lo anterior no es verdad para todos los proyectos de programación, pero con proyectos de gran escala, es evi-- dente la presencia de los problemas antes mencionados.

Es conveniente realizar una observación acerca de la magnitud del proyecto y su complejidad :

MAGNITUD DEL PROYECTO	COMPLEJIDAD
a) Hasta 1000 líneas de código	Trivial
b) Entre 1000 y 10,000 líneas	Simple
c) Entre 10,000 y 100,000 líneas	Difícil
d) Entre 100,000 y 1,000,000 líneas	Complejo
e) Entre 1,000,000 y 10,000,000 líneas	Casi imposible
f) Más de 10,000,000 líneas	Completamente absurdo

- Proyectos Triviales :Estos no necesitan ningún tipo de administración formal. Un programa de computadora que involucra unos cuantos cientos de líneas de código, generalmente lo puede realizar una sola persona en un período de pocos meses. Casi siempre el único factor a-considerar, es el límite de tiempo, "¿Cuándo tiene que estar?". Aún cuando las técnicas de análisis estructurado, diseño estructurado y programación estructurada-pueden ser de gran ayuda, el analista-programador puede, por fuerza bruta o por sentido común, realizar el-trabajo a su manera.

- Proyectos Simples : Se pueden decir muchas cosas similares para proyectos "simples", que involucran programas de computadora o sistemas hasta de 10,000 líneas - de código. Este tipo de proyectos es desarrollado generalmente, con 3 ó 4 analista-programadores en 6 ó 12 meses. Es suficientemente grande y de una duración --

tal que, se necesita cierto tipo de administración formal del Proyecto. Porque precisamente estos Proyectos están dentro del campo de comprensión del responsable, éste puede organizar, administrar y controlar. Aunque las técnicas que se mencionan en este trabajo pueden ser muy útiles, algunos directores pueden argumentar que es una " artillería pesada innecesaria ". Las técnicas estructuradas pueden orientar a la construcción de un producto mantenible, pero no necesariamente acelerar el desarrollo del Proyecto.

- Proyectos Difíciles : En proyectos que involucran la escritura de 10,000 a 100,000 líneas de código, las cosas van más allá de la habilidad de un director, como para que éste lo pueda administrar fácilmente. Este tipo de proyectos pueden incluir de 6 a 12 programadores y abarcar 2 ó 3 años para su desarrollo. Se hace obvia la necesidad de una organización formal del Proyecto, así como de un método de análisis y diseño adecuados. Una "casa de Software", puede garantizar la "terminación" de un Proyecto de esta magnitud; existe sin embargo, el peligro de terminarlo meses más tarde con el consecuente exceso en el presupuesto.
  Las técnicas de análisis, diseño y programación estructurada empiezan a cobrar una gran importancia, en proyectos de esta magnitud. Con el uso correcto de éstas técnicas existe la oportunidad de terminar el proyecto a tiempo y dentro de los límites de presupuesto.
  Tal vez en esta categoría de proyectos es donde las técnicas y los procesos de desarrollo de programación tienen mayor importancia. Un proyecto que involucra hasta 100,000 líneas de código es suficientemente complejo como para que, ni el analista ni el usuario tengan una idea clara de lo que debe realizar el sistema.

Aún cuando exista un claro entendimiento, el usuario es-partidario de cambiar de ideas sobre algunos requerimientos. Además, la duración del proyecto es de 2 a 3 años,-existiendo la posibilidad de que el medio ambiente (como la tecnología, condiciones de comercio, política o hasta el mismo usuario) cambien durante el desarrollo del proyecto.

- Proyectos Complejos : Con los proyectos "Complejos"(entre 100,000 y 1,000,000 líneas de código), aún el más experimentado director de proyectos tendrá que admitir que --- siente un cierto nerviosismo. Para un proyecto de esta - magnitud, estamos hablando de aproximadamente de 50 a -- 100 personas involucradas en la programación, y con una-duración de aproximadamente 3 a 5 años. Esto significa - que :

  - Cualquier miembro del equipo puede abandonar antes de la terminación del proyecto.

  - Existirá por lo menos dos niveles jerárquicos de administración del proyecto.

  También podemos estar casi seguros de que el producto no estará dirigido a un solo usuario, sino a una comunidad-de usuarios, cada uno de los cuales, observará un punto-de vista particular de los requerimientos del sistema.

  Con métodos convencionales de desarrollo de programa--ción, se corre el riesgo de :

  - Que el proyecto nunca se concluya.
  - Terminar el proyecto mucho después del tiempo estipulado y sobre-presupuestado a tal grado que puede --- arruinar la carrera de un director.
  - Que el usuario rechace el producto final, por ser inadecuado a sus necesidades.

- Todo lo anterior,

Como los métodos que se describen en el presente trabajo, no presentan una técnica infalible, no se puede garantizar el éxito del proyecto si, éste está bajo control de un jefe o de un grupo de programadores incompetentes. -- Esto es escencial si se quiere terminar el proyecto a -- tiempo y bajo presupuesto.

- Proyectos Imposibles : La siguiente categoría de proyectos, son de tal magnitud(entre 1,000,000 y 10,000,000 de líneas de código) que, casi ningún director de proyectos ha afrontado. Las técnicas convencionales de desarrollo de programación, garantizan un fracaso completo del proyecto. Aún con los métodos que se describe en este trabajo, es casi imposible de terminar con éxito el proyecto, por los problemas de política de empresas y de manejo de personal.

I.3 Evolución de la Ingeniería de Programación

Estudios de las causas de los errores que se observaban en los programas llevaron a los resultados que se ilustran en la figura I.3.(1). Alrededor de 55% de los errores son causados durante las fases de análisis del problema; sin embargo, estos errores se detectan en la fase de construcción y como se muestra en la figura I.3.(2), a medida de que el período comprendido entre el orígen del error y su detección aumenta, el costo de su corrección crece fuertemente. Por esto actualmente en el desarrollo de programas, a la fase de codificación o construcción preceden fases extensas de análisis y diseño, tal como se discute en el siguiente capítulo.

Entre las primeras técnicas desarrolladas para resolver los problemas encontrados en los desarrollos iniciales de Software puede citarse el álgebra de información {1,2} . Empleando este análisis se pueden especificar los datos relevantes y después determinar las relaciones y reglas asociadas a la manipulación de estos. A través de estas relaciones se obtiene un nuevo juego de variables incluyendo el resultado. Con esta metodología se pensaba o se esperaba optimizar el proceso de desarrollo de Software. Nunca tuvo un gran impacto sobre trabajos específicos. Sin embargo mostraba que los investigadores empezaban a preocuparse por un método de diseño más formal que la programación.

Al principio de la década de los 60's surgió lo que se llamó Plan para el Estudio de la Organización {1} . Definía tres elementos: datos generales, estructurales y operacionales. Los primeros contenían básicamente información histórica. La información estructural presentaba un panorama conciso de la empresa incluyendo detalles sobre

FIGURA I.3.(1) ORIGEN DEL ERROR Y LUGAR DE SU DETECCION

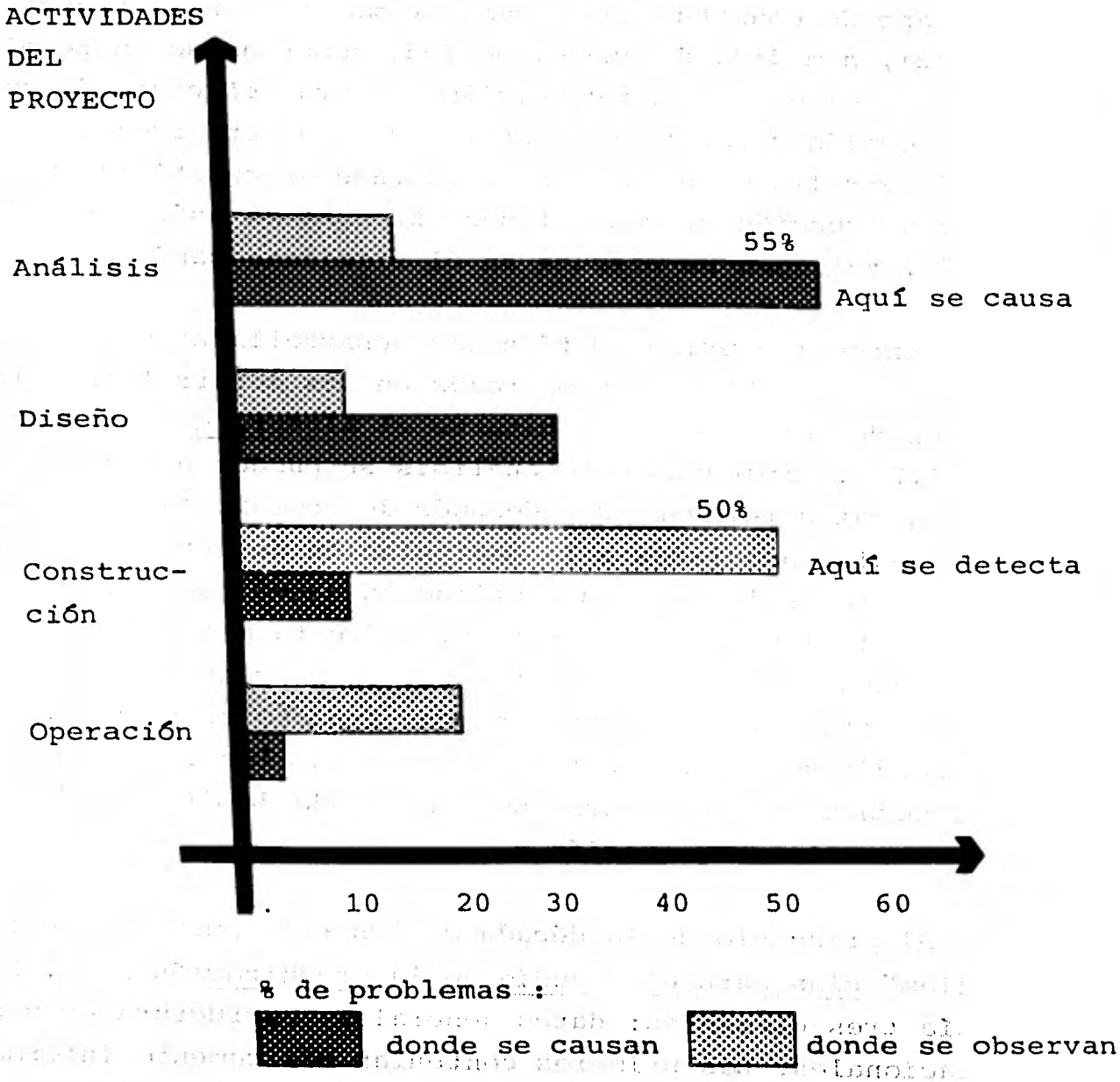
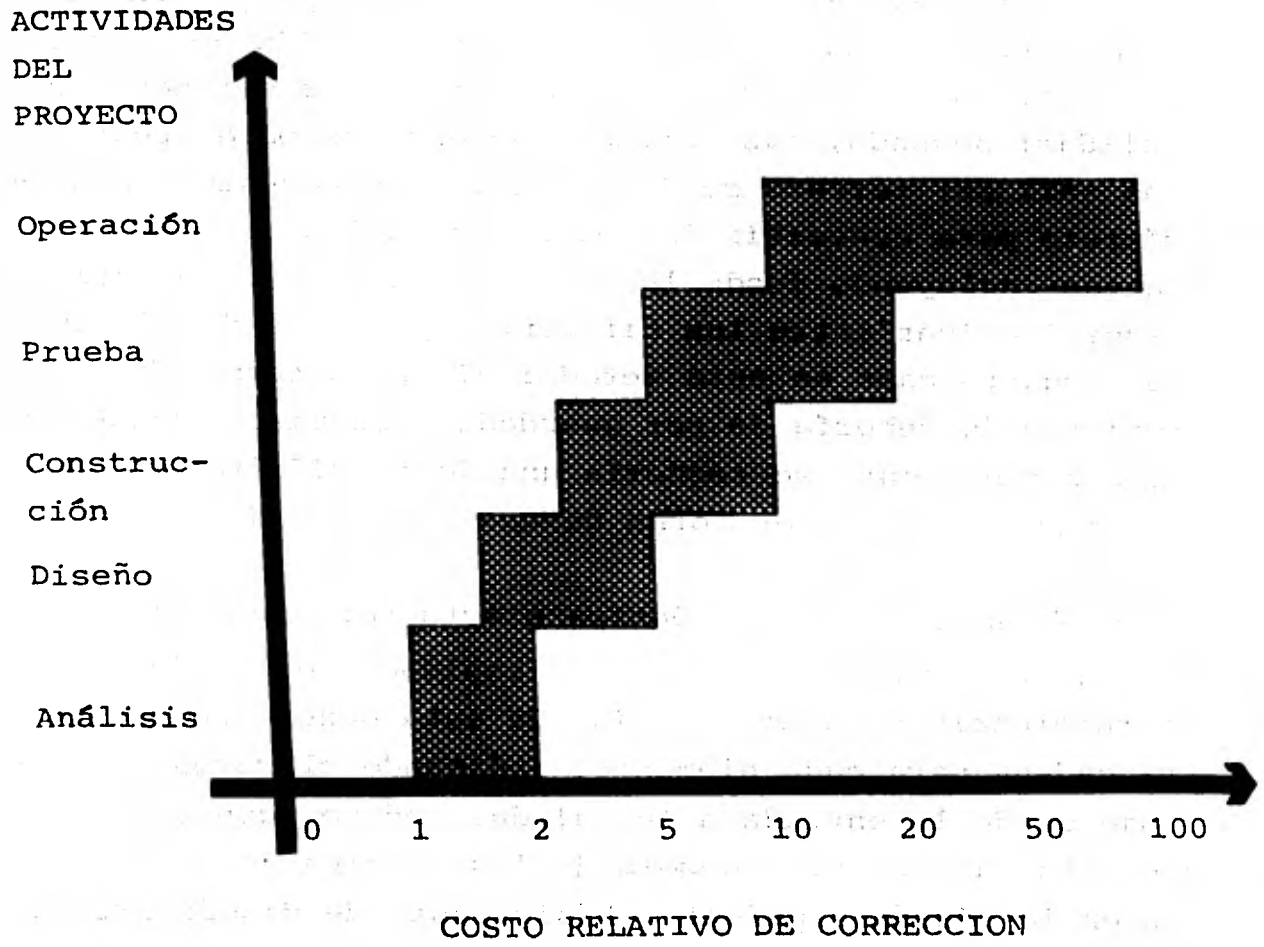


FIGURA I.3.(2) COSTO DE CORRECCION DE ERRORES



sus productos materiales, mercados, situación financiera y personal, entre otros. Finalmente los elementos operacionales proveían la información que describía la operación del sistema, como diagramas de flujo y distribución total de los recursos dentro de las actividades operativas. Esta técnica muestra claramente una preocupación -- por el flujo de información dentro de un sistema, característica, como se verá más adelante, de las técnicas modernas de programación.

También aparecieron lo que se llaman sistemas exactamente desarrollados {3} . En la construcción de éstos, - se distinguen 5 pasos formales: primero se definen las - salidas; segundo, las entradas necesarias para crear estas salidas; tercero se identifican las necesidades de - cómputo para convertir las entradas en las salidas necesarias incluyendo desde luego restricciones y relaciones significativas entre los diferentes procesos de cómputo; como cuarto paso en esta metodología se determinaba qué información debería de ser guardada para pasos subsecuentes, y finalmente se requería una definición formal de - relaciones lógicas en forma de una tabla de decisión.

Además se seguía una técnica exacta para referir los - datos, cada vez que un dato era empleado se le ligaba a - su referencia o fuente previa. De esta manera se creaba - una cadena para cada elemento, indicando claramente el - flujo desde la entrada a la salida. Podemos considerar - que esta fue una predecesora de las técnicas que surgieron en la década siguiente, básicamente de diseño arriba hacia abajo.

En paralelo se empezaron a desarrollar técnicas asistidas por computadora para el diseño. Sin embargo, se descubrió que si bien fueron importantes en la evolución de

la metodología del diseño de sistemas, resultaron inapropiadas para depender totalmente de ellas en el diseño de un programa.

I.4 Técnicas de Diseño Contemporáneas

Todas las técnicas de diseño contemporáneas tienen una característica común: tratan de descomponer un sistema complejo en partes que sean más manejables. En particular se pone gran énfasis en las fases iniciales del diseño. Ello produce una estructura de costos durante las fases de desarrollo como la mostrada en la figura I.4.(1).

Las personas identificadas con la revolución en la programación que ahora se conoce como programación estructurada, fueron Bohm y Jacobini {1} que mostraron que era posible escribir programas usando solamente tres instrucciones lógicas: 'DO WHILE', 'IF THEN ELSE', y 'SEQUENCE'. Posteriormente se han introducido otras instrucciones lógicas adicionales. Dijkstra {5} mostró un poco después, que era conveniente eliminar el 'GO TO' de la programación. Empleando las tres instrucciones lógicas indicadas arriba es posible estructurar programas que se puedan leer de arriba hacia abajo sin necesidad de regresar a instrucciones precedentes.

Ya en la década de los 70's Harlan Mills {6} indicó los módulos de un programa deberían tener una sola entrada y una sola salida. Además se estableció la necesidad de seguir normas estrictas de programación, para que los programas quedaran liberados de características y peculiaridades personales de su realizador.

Como resultado de la evolución en las técnicas de desarrollo de Software es posible afirmar que hoy es factible desarrollar programas con las siguientes características:

FIGURA I.4.(1) ESTRUCTURA DE COSTOS DE FASES DE DESARROLLO

F A S E	ANALISIS	DISEÑO	CONSTRUCCION	PRUEBA	OPERACION

45% de los Recursos empleados antes de proceder a la codificación

- El programa debe de trabajar. Esto es lo más importante; la velocidad, los requerimientos de máquina, el número de líneas de código y otras variables cuantificables no tienen significado si el programa no trabaja. Debe ser correcto antes de ser rápido, debe ser factible antes de ser rápido, debe ser claro antes de ser rápido.
- Bajos costos de desarrollo.
- Bajos costos de prueba.
- Costos mínimos de mantenimiento. Organizaciones que emplean masivamente procesamiento de información gastan entre 50 y 90% de su presupuesto anual en el mantenimiento de sistemas existentes.
- Fácilmente modificable. Dentro de los costos señalados arriba, se in cluye los dos de modificación, ya que, dentro del dinámico ambiente de una empresa moderna, los requerimientos de los programas cambian continuamente. Hasta el módulo más pequeño debe estar diseñado tomando en cuenta las eventuales necesidades de revisión y cambio.
- De diseño poco complicado. Siempre debe diseñarse un programa con la idea de que alguna otra persona lo va a mantener.
- Eficiente. Debe recordarse que en general cerca de 50% del tiempo de ejecución de un programa se gasta solamente en el 3% de las instrucciones. Para obtener mayor eficiencia, el programa debe tener una estructura lógica, con énfasis en simplicidad y confiabilidad.

Para obtener programas con las características anteriores se siguen metodologías que se conocen con el nombre genérico de técnicas de diseño estructurado {1,7,8,17}.

Básicamente estas incorporan :

- Diseño de Arriba hacia Abajo. {9,10} Este permite descomponer una función compleja en una jerarquía de funciones más sencillas; cada nivel jerárquico tiene que ser lógicamente completo; el nivel superior contiene la lógica de máximo nivel y cada nivel siguiente debe estar subordinado al nivel jerárquico inmediatamente superior. Mediante esta técnica el problema se descompone en un número cada vez mayor de segmentos lógicos.

  Asociado a este diseño de arriba hacia abajo debe de ir una documentación también de arriba hacia abajo. Esta debe realizarse antes de codificar y sirve como un documento para revisar y aprobar el diseño antes de iniciar la codificación. Entre las técnicas que estan surgiendo para documentar, se pueden señalar los diagramas de jerarquía y de entrada-proceso-salida (HIPO) y la pseudocodificación {11}.

- Implementación de Arriba hacia Abajo. {1} La codificación, verificación e implementación deben seguir este orden jerárquico. Se empieza con el más alto nivel; niveles jerárquicos inferiores que se necesiten para probar una interfase se codifican como módulos ciegos que no realizan ningún cálculo útil.
  Lo importante es integrar y verificar programas a cada nivel en combinación con sus antecesores, antes de proceder a codificar el siguiente nivel.

- Empleo de Programación Estructurada. {12,13} Esta implica o presupone el uso de solamente cinco instrucciones lógicas que son:

  * DO WHILE
  * IF THEN ELSE
  * SEQUENCE
  * CASE
  * DO UNTIL

Aunque es posible desarrollar un programa empleando solamente las tres primeras, el uso de las dos últimas, facilita este esfuerzo sin aumentar la complejidad del mismo.

- Revisiones Estructuradas. {1} Estas son revisiones técnicas internas de diseño, implementación y documentación de un programa o sistema realizadas por el grupo de diseño. Su objeto es proveer retroalimentación positiva al programador y diseñador. Estas revisiones son calendarizadas por los programadores y diseñadores. Asisten a ellas todos los integrantes del grupo, con lo que se evita que la programación adquiera características personales.

- Pseudo Codificador. {1,14} Este emplea un lenguaje de muy alto nivel para describir la estructura de control y la organización general del programa de computadora. Está diseñador para que pueda ser leído y comprendido fácilmente por otras personas.

- Administración de la Programación. {15} Para facilitarla se recomienda dividir o separar claramente los esfuerzos de análisis de sistemas de los esfuerzos de desarrollo de la programación.

I.5 Tendencias Futuras. [14]

En el futuro se hará cada vez más énfasis en contar con una definición adecuada del problema, un diseño estructurado, un análisis completo de los datos y la lógica del programa antes de empezar a codificar. La documentación es cada vez más importante para satisfacer las necesidades futuras de mantenimiento y modificación del sistema. Hay que hacer notar que con la metodología del desarrollo de programas que se ha resumido, la actividad de programación se requiere hasta las fases finales del proyecto.

En el diseño del sistema el énfasis aumentará en las fases iniciales y conceptuales de desarrollo del programa. Es posible que en el futuro existan procesadores de alto nivel capaces de producir un programa completo e integrado. Se informará a este procesador cuales son sus necesidades y el procesador analizaría el problema y daría resultados en forma automática.

En las secciones subsecuentes se describen las fases del proceso de programación y su control de calidad. Finaliza este trabajo con un ejemplo de aplicación de estas metodologías a un proyecto específico.

CAPITULO II PROCESO DE PROGRAMACION.

II.1 Introducción.{16,18}

Para la obtención de productos con las cualidades mencionadas en el capítulo anterior, es necesario seguir ciertos pasos para su elaboración. El conjunto de pasos es denominado "Proceso de Programación" y consta de las siguientes etapas :

- Formulación y Análisis del Problema.
  Esta etapa consiste en definir el problema y proporcionar los requerimientos básicos y restricciones operativas de la programación.

- Análisis de Alternativas.
  En esta fase, un conjunto de soluciones factibles se investiga.

- Selección de Alternativas.
  Esta es una fase del proceso de programación en la cual cada una de las posibles soluciones investigadas anteriormente es, evaluada y comparada a fin de elegir la más adecuada.

- Requerimientos del Programa de Computadora (RPC).
  Esta etapa del proceso, define formalmente los requerimientos funcionales de la programación a desarrollar.

- Requerimientos de Interfase (DIE).
  Esta es una actividad crítica en el proceso de programación. En ella se detallan tanto las restricciones operativas de la programación, así como las estructuras de archivos que permi-

ten la interacción de nuestra programación con la del resto del sistema.

- Diseño del Programa de computadora (DPC).
  Aquí se muestra, utilizando las más novedosas técnicas de diseño y programación estructurada (como son los diagramas de estructura {7,10}, etc....), el orden que permite satisfacer los RPC.

- Diseño detallado del Programa de Computadora (DDPC).
  Esta actividad permite formalizar las variables de entrada y salida a cada módulo del diagrama de estructuras. Asimismo, mediante el uso de un LDP (lenguaje para diseñar programas {26, Apéndice B.2}) da margen al eventual código que se escribirá en la siguiente etapa del proceso de programación.

- Código.
  En este período se obtiene el programa fuente producto del proceso de programación.

- Plan y Procedimientos de pruebas (PPP).
  Esta es una actividad que permite concebir las pruebas que certificarán la calidad de la programación, así como el procedimiento computacional necesario para satisfacerlas.

En la figura II.1.(1) puede observarse esquemáticamente esta metodología.

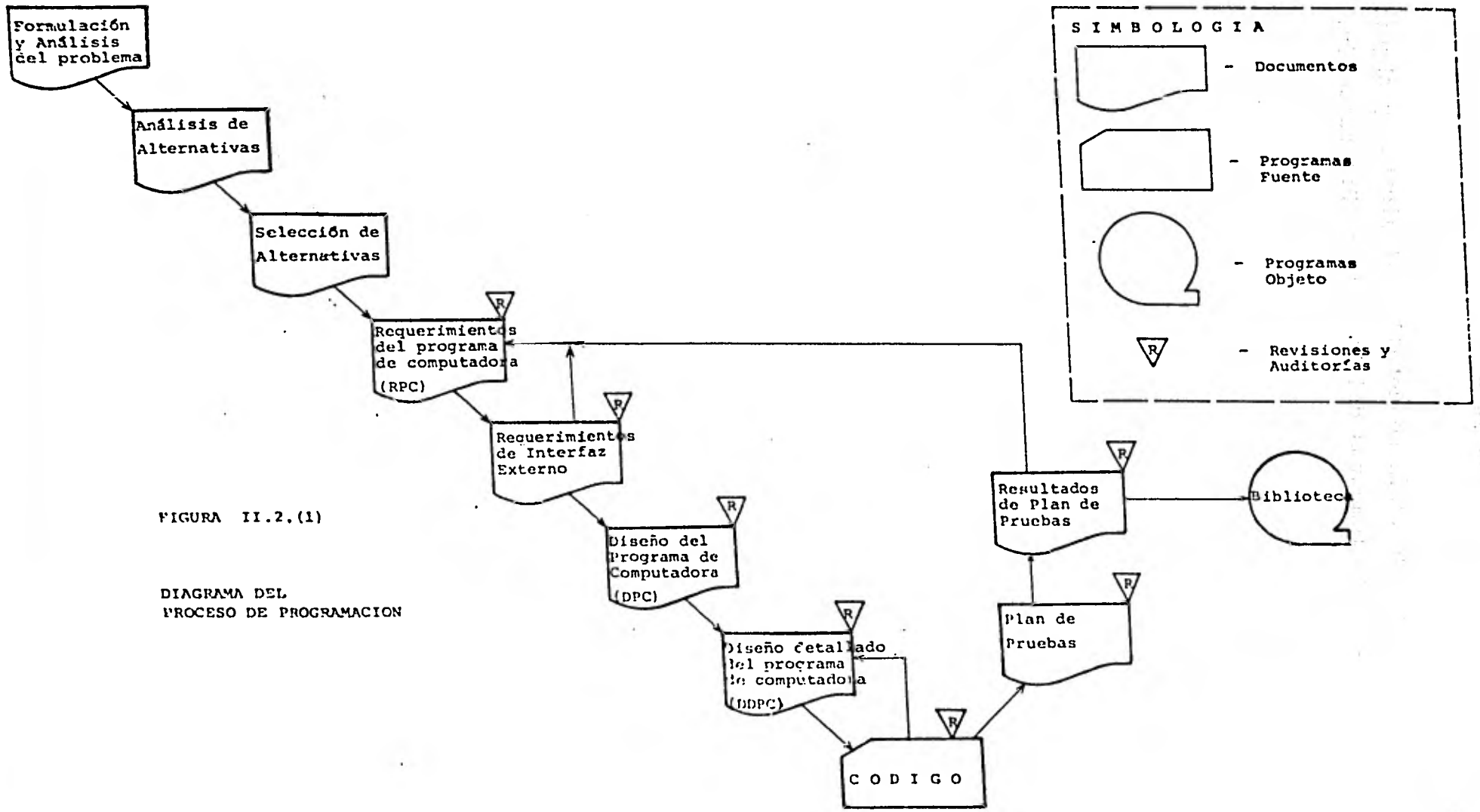


FIGURA II.2.(1)

DIAGRAMA DEL PROCESO DE PROGRAMACION

A continuación se detallan el alcance y objetivo de los elementos del Proceso de Programación.

II.2 Formulación y Análisis del Problema.

Esta etapa del proceso de programación define el problema en base a los requerimientos del usuario. Para el desarrollo de esta etapa, se deben realizar las siguientes funciones :

- Definir las necesidades del usuario.
- Determinar los requerimientos básicos del programa de computadora.
- En base a lo anterior, definir a grandes rasgos las características del equipo de cómputo. En caso de estar definido el sistema de cómputo, determinar las modificaciones y adiciones necesarias.
- Proponer un programa de trabajo para el desarrollo del programa de computadora. Este programa de trabajo debe contemplar los siguientes puntos :

  i) Obtención de información para la etapa de análisis de alternativas.

  ii) Definición de los requerimientos básicos y restricciones operacionales del programa, como son el interfaz-hombre-máquina, interacción con el sistema existente, etc.

El documento que se elabore, debe sentar las bases del proceso de programación.

II.3 Análisis de Alternativas.

En esta etapa se proporcionan las herramientas requeridas para la selección de las técnicas que vayan a ser empleadas para resolver el problema planteado.

Se proponen los siguientes pasos para el análisis de alternativas :

- Recopilación y análisis de datos.
- Propuesta de alternativas factibles.
- Simulación de alternativas.
- Obtención de resultados.

A continuación se describen cada uno de estos pasos.

II.3.1 Recopilación y Análisis de Datos.

En esta actividad se determinan :

- Las fuentes de información, indicando en donde pueden encontrarse los datos necesarios para el análisis de alternativas.

- La naturaleza de los datos, describiendo como se obtienen los datos en las fuentes de información.

- La confiabilidad de los datos, determinando la calidad de la información.

II.3.2 Propuesta de Alternativas Factibles.

En base a los resultados del análisis de datos,- los requerimientos básicos y operativos del programa y las restricciones del mismo, se investiga la existencia de soluciones a problemas similares.

Generalmente para esta etapa, se utilizan un --- gran número de manuales técnicos, publicaciones, textos, etc.

II.3.3 Simulación de Alternativas.

Con los algoritmos seleccionados como soluciones factibles, se desarrollan prototipos de prueba.- El objetivo de estos programas es realizar la simulación de las condiciones de operación del programa. La información arrojada servirá para realizar mejoras y modificaciones a las alternati-- vas propuestas.

II.3.4 Obtención de Resultados.

Los resultados de la simulación obtenidos ante-- riormente, se ordenan y clasifican para ser utilizados como herramienta para la siguiente etapa del proceso de programación.

II.4 Selección de Alternativas.

Esta fase del proceso de programación, permite decidir cuál de las técnicas propuestas para resolver el problema planteado es la más adecuada. Diversos criterios deberán ser analizados exhaustivamente, a fin de elegir el algoritmo que finalmente desee entregarse al usuario.

A continuación se mencionan algunos puntos que merecen especial consideración :

- Requerimientos de entrada
  Es indispensable que para una evaluación comparativa de alternativas se utilice la misma información de entrada.

- Calidad de la respuesta
  Es necesario definir índices, que permitan comparar las soluciones de los prototipos analizados con otras técnicas de solución publicadas en la literatura comercial.

- Tiempos de ejecución
  Es importante conocer los tiempos de respuesta de la programación propuesta, ya que un comportamiento pobre de este índice puede eliminar por completo téncicas con calidades de respuestas satisfactorias.

II.5 Requerimientos del Programa de Computadora (RPC)

II.5.1 Introducción.

El objetivo de esta fase del proceso es delinear y describir las características funcionales de cada programa a un nivel de detalle suficiente que permita el diseño, así como asignar los criterios y métodos para verificar que el producto de la programación satisface los requerimientos. El resultado de esta fase es un documento de requerimientos del programa de computadora, para cada uno de los programas que vayan a ser escritos.

Una vez que este documento sea aprobado, será puesto bajo el control de la biblioteca de soporte de programas y de los procedimientos que integran el Control de Calidad de programación, a fin de permitir que nuevos requerimientos, cambios a requerimientos existentes, etc., den margen a actualizaciones a las especificaciones, que pueden requerir una repetición completa o parcial de algunas de las etapas anteriores en el proceso de programación.

II.5.2 Descripción de la estructura del documento RPC.

1. Introducción y Propósito.

Indica a grandes rasgos cuál es el objetivo del programa y funciones a cumplir. Asimismo señala finalidad del documento RPC y su importancia.

2. Referencias.

2.1 Especificación del Sistema

Indica el medio ambiente en el que operará el programa y las especificaciones preliminares del programa.

2.2 Especificación funcional de Interfase del programa

Muestra las especificaciones de Interacción del programa con otros programas del sistema.

2.3 Estudios y reportes

Indica los estudios y reportes publicados que tengan relación con el diseño del programa.

2.4 Normas y Convenciones

Indica los manuales ó estándares que se emplearon como base para elaborar el documento RPC.

2.5 Glosario, abreviaciones y símbolos

Lista todos los mnemónicos, abreviaciones y símbolos que se emplean en el documento.

3. Definición del Programa.

Describe para el programa :

- Carcterísticas básicas.
- Una breve descripción de propósito u objetivo.
- Flujo de información.
- Diagrama de interacción del programa.

3.1 Definición de subprogramas.

Describe para el programa :

- Diagrama de estructuras que muestra los principales subprogramas.
- Diagrama de interacción entre subprogramas.

3.1.X Definición del subprograma X.

En esta sección se describe todos los requerimientos necesarios para definir un subprograma X.

Cabe señalar que habrá una definición para cada uno de los subprogramas que se hayan definido en la sección anterior.

3.1.X.1 Interfase con subprogramas.

Indica las entradas, salidas, mensajes y relaciones con otros subprogramas.

3.1.X.2 Desglose del subprograma X en módulos.

Se proporcionan :

- diagrama de estructuras describiendo los principales módulos que componen el subprograma (4 ó 5 niveles).
- diagrama de bloques indicando la secuencia y transferencia de control entre módulos.

3.1.X.3 Diagrama de Interacción funcional de módulos, base de datos y archivos.

3.2 Diagrama de interacción funcional de subprogramas.

Indica la interacción entre diferentes subprogramas señalando sus interfases.

3.3 Resumen de subprogramas y módulos.

Describe brevemente el funcionamiento de subprogramas y módulos del programa.

4. Requerimientos de Subprogramas.

4.X Requerimientos del subprograma X.

4.X.1 Requerimientos de interfaz Hombre-Máquina.

Indica el número de desplegados y su - contenido.

4.X.2 Requerimientos de elaboración de reportes.

Describe brevemente el número y contenido de reportes.

4.X.3 Requerimientos de los módulos del subprograma X.

4.X.3.Y Requerimientos del módulo Y.

4.X.3.Y.1 Requerimientos de Interfase.

Presenta los argumentos de entrada y salida, transferencia de control y mensajes.

4.X.3.Y.2 Desglose del módulo en funciones.

Introduce las funciones que -- componen el módulo por medio - de diagrama de estructura; además se indica la transferencia de control y secuencia de ejecución con diagrama de bloques ó con lenguaje para diseñar -- programas (LDP Apéndice B.2).

4.X.3.Y.3 Definición detallada de las -- funciones del módulo Y.

Es importante mencionar que la siguiente sección se realizará para todas las funciones que - componen el subprograma en --- cuestión.

4.X.3.Y.3.W Definición de la función W.

a) Entradas.

b) Proceso.

c) Salidas.

4.X.3.Y.4 Requerimientos de almacenamiento de información para el módulo Y.

Debe indicarse todos los arreglos matriciales y variables que se utilizan en el módulo Y, especificando su dimensión, tipo (real ó entero), precisión (simple ó doble), etc.

4.X.3.Y.5 Restricciones de utilización del módulo Y.

Indica los modos de uso, precisión de cálculo, frecuencia y tiempo de ejecución para el módulo Y.

4.X.3.Y.6 Restricciones de operación del módulo Y.

Define :

- Equipo de interfase.
- Programación de interfase, como son las rutinas de acceso a base de datos.
- La interacción con el resto del programa, especificando la activación del módulo por otros subprogramas y, activación de otros subprogramas por el módulo.

- La interfase hombre-máquina, especificando el número y contenido de desplegados, reportes, alarmas y/o mensajes.

4.X.3.Y.7 Restricciones de desarrollo del módulo Y.
Especifica las características del equipo de cómputo, sistema operativo, lenguajes y programas disponibles para el desarrollo del módulo.

4.X.3.Y.8 Requerimientos de mantenimiento y modificación.
Indica las modificaciones o expanciones necesarias para futuros requerimientos del módulo.

5. Certificación de Calidad.

5.X Requerimientos de certificación de calidad par el subprograma X.
En esta parte se asientan las bases para la realización de un plan de prueba para la programación a desarrollar.

5.X.1 Enfoque y métodos.
Indica la forma y los métodos con que se realizará la certificación de calidad para el subprograma.

5.X.2 Requerimientos de certificación de calidad para el módulo Y.

5.X.2.Y Requerimientos de certificación de calidad para el módulo Y.

5.X.2.Y.1 Enfoque.
Asienta la forma de realizar la certificación de calidad para el módulo Y.

5.X.2.Y.2 Métodos.
Indica qué métodos se utilizarán para llevar a cabo la certificación de calidad del módulo Y.

5.X.2.Y.3 Criterios de aceptación.
Indica las condiciones con las que se acepta la certificación del módulo Y.

5.X.2.Y.4 Lista de especificaciones a verificar.
Lista todas las especificaciones del módulo y las condiciones que deben cumplirse.

5.X.2.Y.4.W Especificación W.
a) Métodos de prueba.
b) Precisión.
c) Criterios de aceptación

5.X.2.Y.5 Matriz de certificación de calidad para el módulo Y.
Lista una a una las especificaciones solicitadas, y el cumplimiento de ellas.

6. Apéndices.

6.1 Desarrollo matemático de algoritmos, demostraciones, teoremas, etc.

6.2 Desarrollo y/o justificación de modelos.

6.3 Resumen de ecuaciones.

6.4 Definiciones de términos.

6.5 Referencias.
Indica las revistas técnicas, textos y publicaciones varias que se utilizaron para el desarrollo del documento RPC.

II.6 Requerimientos de Interfase.

Esta especifiación de interfase es un documento -- que identifica las interacciones funcionales y físicas, externas al programa de computadora. Una - interfase es definida como una región común a dos- o más elementos, sistemas, proyectos o programas,- caracterizados por propiedades físicas, funciona-- les y tipo de procedimientos. Es necesario defi-- nir con claridad toda la información residente en- las bases de datos del sistema.

Adicionalmente a la descripción de los datos, la - especificación deberá identificar el sistema que - genera los elementos de información y el sistema - que recibe los elementos de información. Dos documentos deberán ser escritos :

- Interfase Externa.

  Describe la interacción entre el programa y el- resto del sistema, ejemplo : Bases de datos, - archivos, etc.

- Interfase Interna entre Subprogramas.

  Describe la interacción entre los programas que se desarrollarán. (Intercambio de información).

II.7 Diseño del Programa de Computadora (DPC).

II.7.1 Introducción.

El objetivo de la fase de diseño, es traducir los requerimientos del programa de computadora definidos en el documento RPC a un sistema de programación bien definido y lógicamente organizado, para cada una de las tareas, por ejemplo, el diseño de un subsistema detallándolo en módulos de programación.

El diseño preliminar de la base de datos, es producido en paralelo con la estructura de cada subsistema. A medida que los módulos del programa se van definiendo, el plan para su prueba es elaborado. La fase de diseño termina con la elaboración del documento de diseño de programa de computadora (DPC), que cuando es revisado y aprobado por el cliente, constituye la base para el desarrollo de cada uno de los módulos a programar.

Los productos que se obtienen de esta fase son :

- Documento de diseño del programa de computadora para cada una de las tareas a realizar.

- Plan de pruebas para cada subsistema.

II.7.2 Descripción de la estructura del documento DPC.

1. Introducción y propósito.

Indica los objetivos del programa de computadora, así como su modo de operación. Menciona el objetivo del DPC y su importancia.

2. Referencias.

2.1 Especificación del sistema.

Indica el medio ambiente en que operará el programa.

2.2 Especificación funcional de Interfase del programa.

Indica la interacción del programa con el resto del sistema.

2.3 Estudios y reportes.

Indica qué documentos previos tiene como antecedente del presente documento, documentos técnicos de la computadora, documentos de programas del proveedor del sistema, etc., que tengan relación con el diseño del programa.

2.4 Normas y convenciones.

Muestra qué manuales ó estándares se emplearon como base para elaborar el documento.

2.5 Glosario, abreviaciones y símbolos.

Lista todos los mnemónicos, abreviaciones y símbolos que se emplean en el documento.

3. Definición del programa.

Esta sección especifica la estructura del programa, asignación de los requerimientos entre los subprogramas y módulos.

3.1 Definición de Subprogramas.

Muestra mediante un diagrama, la jerarquización entre subprogramas y módulos (con un máximo de 5 niveles de jerarquía). Así mismo la descripción funcional de programas y módulos, interfases, estructura de datos, métodos de entrada salida, modos de operación y uso de recursos.

3.1.X Definición del Subprograma X.

3.1.X.1 Interfase con subprogramas.

Especifica las interfases, de entrada, salida, mensajes y alarmas.

3.1.X.2 Desglose del subprograma X en módulos.

Para el subprograma X proporciona :

- Diagrama de estructu

ra que muestre las principales módulos del subprograma.

- Descripción breve de los módulos.
- Diagrama de bloques indicando la secuencia y transferencia de control entre los módulos.

3.1.X.3 Diagrama de interacción funcional de módulos, bases de datos y archivos.

3.2 Diagrama de interacción funcional de subprogramas.

3.3 Resumen de subprogramas y módulos.

3.4 Resumen de desplegados de salida y resultados del programa.

4. Diseño de los programas.

4.X Diseño del subprograma X.

En esta sección se debe incluir todo lo necesario para el diseño del subprograma, pudiendo el subíndice "X" los valores correspondientes a todos los subprogramas que componen el programa a desarrollar.

4.X.Y Diseño del módulo Y.

4.X.Y.1 Introducción.

4.X.Y.1.1 Identificación.
Define qué módulo cubre esta-parte del DPC.

4.X.Y.1.2 Objetivo.
Indica el objetivo particular del módulo que se está dise--ñando.

4.X.Y.1.3 Resumen del funcionamiento.

4.X.Y.2 Referencias.

4.X.Y.2.1 Documentación.
Indica qué documentos son los que se utilizan para el dise-ño del módulo (estudios, re--portes, documentación de apo-yo, etc.).

4.X.Y.2.2 Glosario, abreviaturas y sím-bolos.
Lista todos los mnemónicos, - abreviaturas y símbolos utili-zados para el diseño del módu-lo Y.

4.X.Y.3 Especificaciones para el diseño - del módulo Y.

4.X.Y.3.1 Desglose del módulo Y en fun-- ciones.

4.X.Y.3,1.1 Diagrama de estructuras -- del módulo Y.
Detalla las funciones que- integran la totalidad del- módulo Y del subprograma X (utilizando aproximadamen- te 5 niveles de detalle).

4.X.Y.3.2 Diagrama jerarquizado de entra da-proceso-salida (HIPO) para- el módulo Y.

4.X.Y.3.2.1 Entradas.
Especifica la estructura - de datos y/o variables de- entrada para el módulo Y , así como la interfase con- diferentes programas de -- computadora.

4.X.Y.3.2.2 Proceso.
Detalla el funcionamiento- del módulo con el lenguaje para diseñar programas.

4.X.Y.3.2.3 Salidas.
Especifica la estructura - de datos y/o variables de- salida para el módulo Y,-- así como la interacción -- con otros programas median te los datos de salida.

4.X.Y.3.2.4 Variable Internas.
Especifica las estructuras de datos para las varia-- bles internas del módulo Y

4.X.Y.3.3 Diseño de las funciones.
En esta sección se debe deta-- llar el diseño para cada una - de las funciones que componen- un módulo del programa. Existi rá tantas secciones como fun-- ciones haya.

4.X.Y.3.3.W Diseño de la función W.

4.X.Y.3.3.W.1 Diagrama jerarquizado- de entrada-proceso-sa- lida (HIPO) de la fun- ción W.

4.X.Y.3.3.W.1.1 Entradas.
Especifica la es-- tructura de datos - para las variables- de entrada.

4.X.Y.3.3.W.1.2 Proceso.
Describe el proceso con el lenguaje pa- ra diseñar progra-- mas (LDP).

4.X.Y.3,3.W.1.3 Salidas.
Especifica la es-- tructura de datos -

para las variables-
de salida.

4.X.Y.3.3.W.1.4 Variables internas.
Especifica la es--
tructura de datos -
para las variables-
internas.

4.X.Y.3.4 Operación del módulo Y.

4.X.Y.3.4.1 Interfase de entrada.
Indica la procedencia de -
los argumentos de entrada-
para el módulo Y.

4.X.Y.3.4.2 Interfase de salida.
Indica el destino de los -
argumentos de salida del -
módulo Y.

4.X.Y.3.4.3 Recursos del Sistema.
Detalla los requerimientos
de almacenamiento en memo-
ria y disco necesarios pa-
ra el módulo y la informa-
ción que utiliza.

4.X.Y.3.5 Restricciones para el módulo Y
Indican las limitaciones y res
tricciones de operación del mó
dulo.

4.X.Y.3.6 Matriz de correspondencia del-
RPC.
Presenta una matriz el la que-

se muestra la correspondencia- entre el documento de requeri- mientos del programa de compu- tadora y el documento diseño - del programa de computadora -- del módulo Y.

4.X.Y.4 Certifiacación de calidad para el- módulo Y.

El contenido es muy parecido al -- del RPC (capítulo II.5.2, subíndi- ce 5.X.2.Y), pero con un enfoque - orientado hacia la certificación - de calidad del código.

II.8 Diseño detallado del Programa de computadora (DDPC).

II.8.1 Introducción.

En este documento se debe describir cada módulo a un nivel de detalle tal, que permita realizar su codificación en FORTRAN estructurado.

Para la descripción de los módulos se deben emplear todas aquellas herramientas necesarias que permitan dejar completamente claro y definido el contenido del módulo. Entre las herramientas que se pueden emplear para este propósito se encuentran :

- Diagramas de estructura detallada para el módulo. (Apéndice A).

  Este diagrama debe incluir hasta el último nivel de detalle, incluyendo una representación gráfica de las entradas y salidas.

- Diagramas de bloques. (Apéndice A).

  Estos diagramas muestran la secuencia y transferencia de control entre las tareas del módulo.

- Diagramas de flujo. (Apéndice A).

  Estos sirven para describir gráficamente el código del módulo.

- Pseudo código. (Apéndice B.2).

  Se debe elaborar un documento para cada subprograma o módulo principal del diagrama de estructura general del sistema, es decir, para los módulos principales descritos en DPC.

II.8.2 Descripción de la estructura del documento DDPC.

1. Introducción.

1.1 Identificación.
Identifica a qué subprograma de computadora corresponde el DDPC, su clave y número de volumen.

1.2 Propósito.
Indica a grandes rasgos cuál es el objetivo del subprograma y sus funciones a cumplir.

1.3 Ubicación.
Muestra el alcance del DDPC del programa de computadora y números de volúmenes que abarca, incluyendo la identificación, clave y descripción de cada volumen.

2. Referencias.

2.1 Especificaciones.
Indica tanto el DPC en el que aparece el diseño global del programa y las especificaciones preliminares del subprograma, como los DDPC de otros subprogramas que tengan alguna interacción con el subprograma que se esté detallando.

2.2 Normas.
Lista tanto las normas de diseño empleadas para eleborar el documento como las normas de programación que deberán ser empleadas para la codificación de los módulos.

2.3 **Documentación de apoyo.**

Aquí se incluye la referencia de otros documentos, manuales o textos que puedan -- ser de utilidad tanto para entender el diseño de módulos como para implementar el- código de los mismos.

3. Descripciones de Diseño del Subprograma.

3.1 Estructura del subprograma y secuencia de implementación.

Proporciona el diagrama de estructura del subprograma, flujo de información entre - los módulos. Además, indica la forma en - la que se implementará el programa.

3.1.1 Descripción general del propósito del subprograma.

3.1.2 Matriz de requerimientos vs. módulos.

Esta matriz muestra qué módulos del - subprograma satisfacen los requeri--- mientos y especificaciones que tiene- el DPC.

3.1.3 Descripción operativa del subprograma

Indica cómo inicia sus operaciones; - cuál es su proceso normal de opera--- ción; cómo recibe la información; a - dónde envía la información; comandos- que procesa y mensajes que recibe o - genera.

3.1.4 Diagrama de Estructura detallado del- subprograma.

Muestra todos los módulos requeridos para llevar a cabo las funciones del subprograma; se emplea un diagrama jerarquizado.

3.1.5 Diagrama de bloques del subprograma.
Muestra la secuencia de las actividades a realizar y cómo se realiza el control de las actividades y tareas en el subprograma.

3.1.6 Pseudo-código de proceso detallado del subprograma.

3.2 Diseño Detallado de los Módulos.

3.2.X Diseño detallado del Módulo X.

3.2.X.1 Objeto.
Dá una descripción de la función a realizar por el módulo. Indica cómo se accesa, bajo qué condiciones genera mensajes y qué acciones se toman cuando se detecta algún error.

3.2.X.2 Detalle del proceso.
Indica mediante un diagrama de bloques, de flujo o pseudo-código el proceso o actividades a realizar en el módulo.

3.2.X.3 Entradas, Salidas e Interfases.
Muestra cuáles son los parámetros de entrada, salida e interfases del módulo.

3.3 Interfases del Subprograma.

Indica cuáles son las principales interfases que tiene el subprograma con otros subprogramas ó módulos y con el medio ambiente (usuario, equipo y programas del sistema).

3.3.1 Forma de accesar el subprograma.

Indica de qué manera se puede llamar al subprograma para su ejecución y quién puede hacerlo.

3.3.2 Rutinas externas empleadas.

Proporciona el nombre y descripción de módulos externos o de biblioteca que sean usados por el subprograma, cuyo diseño detallado no se incluye en este documento.

3.3.3 Archivos y Bases de Datos.

Indica qué archivos y regiones de la base de datos utiliza el subprograma.

3.3.4 Condiciones de Entrada y Salida.

Menciona qué situaciones deben estar presentes para poder accesar al subprograma o para terminar su ejecución (concurrencia o sincronización con otros programas).

3.3.5 Mensajes al operador.

Presenta una tabla que muestra los mensajes que envíen al operador así como el módulo en el que se generan.

3.3.6 Mensajes entre tareas.

Presenta una tabla en la que se muestran los mensajes que puede recibir el subprograma de otros subprogramas, el orígen del mensaje y la acción a tomar. Asimismo, se presenta otra tabla que muestra los mensajes que genera el subprograma, a dónde los envía y qué módulo los genera.

3.3.7 Diccionario de Datos del Subprograma.

Lista todos los identificadores empleados en el subprograma con su correspondiente descripción.

3.3.7.1 Constantes

3.3.7.2 Variables.

3.3.7.3 Archivos.

3.3.8 Medio ambiente del Subprograma.

Indica en qué sitios se encuentra descrita la ubicación del subprograma dentro del sistema, los despliegues que produce y los comandos que procesa.

3.3.9 Limitaciones.

Indica las restricciones de operación.

Una vez definido este documento, se puede empezar a generar código del subprograma diseñado. En forma paralela a esta generación del código, puede realizarse la elaboración del Plan de Pruebas y procedimiento de pruebas.

II.9 Codificación. {14,19,20}

Dado que la programación a desarrollar será utilizada por personas ajenas a quienes escribieron originalmente el código, es necesario mantener un estilo uniforme de programación basado en ciertas normas, de tal forma que el código sea fácil de seguir, revisar, modificar y mantener por aquellas personas directamente interesadas en su aplicación y en su operación.

Es necesario verificar que cada uno de los programas escritos sea realizado bajo las normas de programación mencionadas anteriormente. Para ello es necesario contar con un "Auditor o Checador de Código" (Apéndice B.3), cuyo objeto es controlar la calidad de los programas desarrollados.

El programa auditor, verifica cada una de las normas de programación detectando diferencias con respecto al estándar. El auditor de código no tiene como objetivo optimizar, ni limitar la creatividad del programador; sólo desea, mediante la unificación del estilo obtener programas que presenten una estructura coherente y de mayor claridad. Las normas de programación que permiten alcanzar tales objetivos, pueden ser agrupadas en dos bloques principales :

- Normas de documentación en los encabezados.
- Normas de código.

Las primeras, normas de documentación en los encabezados, proporcionan a los programas una mayor -

claridad; mientras que las normas de código ayudan al programador en la elaboración de programas con- la esctructura coherente.

A continuación se mencionan algunos de los rubros- señalados en las normas de documentación en los en cabezados :

- Título
- Objetivo o propósitos del módulo
- Referencias
- Notación de diseño de programas
- Entradas
- Salidas
- Restricciones en el uso del módulo
- Módulos utilizados
- Identificadores locales
- Programador
- Fecha del último cambio
- Nombre de la persona que hace el cambio

Algunas de las normas de código que deben ser ob-- servadas, son las siguientes :

- Longitud y nombre de las variables
- Longitud y nombre de las subrutinas o módulos
- Tamaño del módulo
- Localización de los formatos
- Secuencia de las etiquetas
- Restricciones
- Tamaño de los arreglos
- Comentarios adicionales
- Separación entre el encabezado y el código
- Empleo de estructuras lógicas y de control apro- piados

II.10 Plan de Pruebas. {14,22}

Es muy importante saber antes de escribir código, como se va a probar un programa. De otra manera no hay forma de garantizar que la estructura de la programación permita verificar en forma incremental y sencilla el producto de la programación.

Un plan de pruebas deberá ser escrito para cada uno de los requerimientos del programa de computadora. El objeto del plan de pruebas es definir una lista de características del programa a ser probadas. Estas se han descrito en el documento de requerimientos de computadora (RPC) y deberán ser satisfechos por un módulo de programación. Adicionalmente el plan de pruebas describe los criterios de aceptación para verificar que cada una de las características mencionadas ha sido satisfecha así como un método de verificación.

La información deberá ser desplegada en forma de tablas que permitan verificar objetivamente las características que deben de ser satisfechas por cada uno de los requerimientos. Los procedimientos deberán contener la siguiente información :

- Identificación de la prueba
- Características a ser demostradas
- Archivos de entrada
- Descripción del procedimiento de prueba
- Matriz de requerimientos de prueba

II.11 Control de Calidad de Programación.

II.11.1 Objetivo. {14,23,24,25}

El objetivo asignado a esta función, es administrar los productos del proceso de programación-enunciados en las secciones anteriores y establecer la biblioteca de soporte de programas -- (BSP). Estas labores estan asociadas a la responsabilidad de desarrollar normas para la documentación, así como convenciones y estándares - para la programación.

Otra función que acompaña al control de calidad de programación es la revisión de los documentos y el código elaborados a fin de determinar-su congruencia con las normas establecidas.

II.11.2 Biblioteca de Soporte de Programas.

La biblioteca de soporte de programas es un sistema organizacional diseñado para controlar y - mantener toda la programación y documentación - relacionada con el proyecto.

Las funciones de esta biblioteca son desarrolladas utilizando, tanto procedimientos manuales,- como algunos programas de computadora,(ej.: auditor de código estático).

La biblioteca de soporte de programas almacenará y mantendrá todos los elementos de programación. Será función del grupo de Control de Calidad, mantener las últimas versiones relaciona

das, tanto con el código escrito durante el proceso de programación, como con la documentación entregada al cliente.

Los componentes de la biblioteca son los elementos de programación, documentación y procedimientos que la gobiernan. Estos procedimientos tienen como objeto controlar las últimas versiones de documentación y código. No se autoriza a los elementos ajenos al grupo de Control de Calidad introducir modificaciones a los productos terminados. Para dicho efecto es necesario establecer un procedimiento, a fin de que toda la detección de errores, cambios en el código, cambios en la documentación, etc., sean documentados. Es así como es posible rastrear todas las modificaciones introducidas a la documentación y código entregados.

II.11.3 Normas. {19,20,26,27}

En la actividad correspondiente al Control de Calidad se debe generar, mantener y recopilar todas las normas, convenciones y estándares relacionados cn el proyecto.

Ejemplo de estos estándares son los siguientes documentos :

- Normas de diseño
- Normas de Análisis
- Normas de codificación
- Normas para nombrar variables, archivos y módulos

- Normas para una notación de diseño de programas
- Normas para planes y procedimientos de prueba
- Normas para construcción de pláticas estructuradas
- Normas para elaborar el manual de usuario
- Auditor de código para programación en FORTRAN 77

II.11.4 Auditorías. {14,28}

II.11.4.1 Introducción.

Como se señaló en el capítulo I, el número de errores que se cometen en la etapa de Análisis es elevado y su detección en fases posteriores del proyecto es muy costoso. Resulta muy conveniente durante la vida del proyecto y en el desarrollo del mismo, calendarizar una serie de auditorías, a fin de señalar errores en las etapas jóvenes del proyecto.

Antes de la escritura de código se deben llevar a cabo una serie de auditorías, a fin de constatar, que todos los requerimientos especificados han sido incluídos. Asimismo, es conveniente analizar si las personas responsables de estos documentos han cumplido con las normas de análisis y diseño especificados.

Durante la escritura del código es necesario calendarizar auditorías, a fin de analizar que el código se elabore bajo las normas establecidas. Es conveniente que para realizar -

estas funciones se desarrolle o utilice un -- checador estático de código.

II.11.4.2 Etapas de Auditorías.

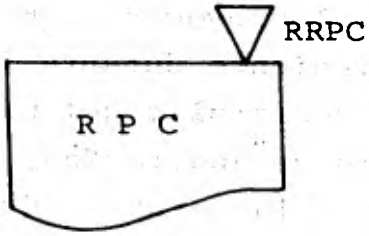
Las etapas que se deben de calendarizar se -- describen a continuación.

- Etapa 1

  Revisión del documento RPC (RRPC)

  Objetivo : asegurar que los requerimientos del programa de computadora, -- han sido especificados correcta y completamente.

  Prerequisitos : (1) RPC

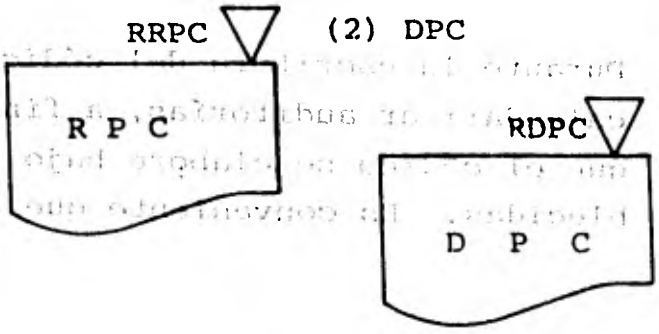


- Etapa 2

  Revisión del documento DPC (RDPC)

  Objetivo : evaluar que las especificacio-- nes técnicas del DPC son adecua dos.

  Prerequisitos : (1) RRPC
  (2) DPC

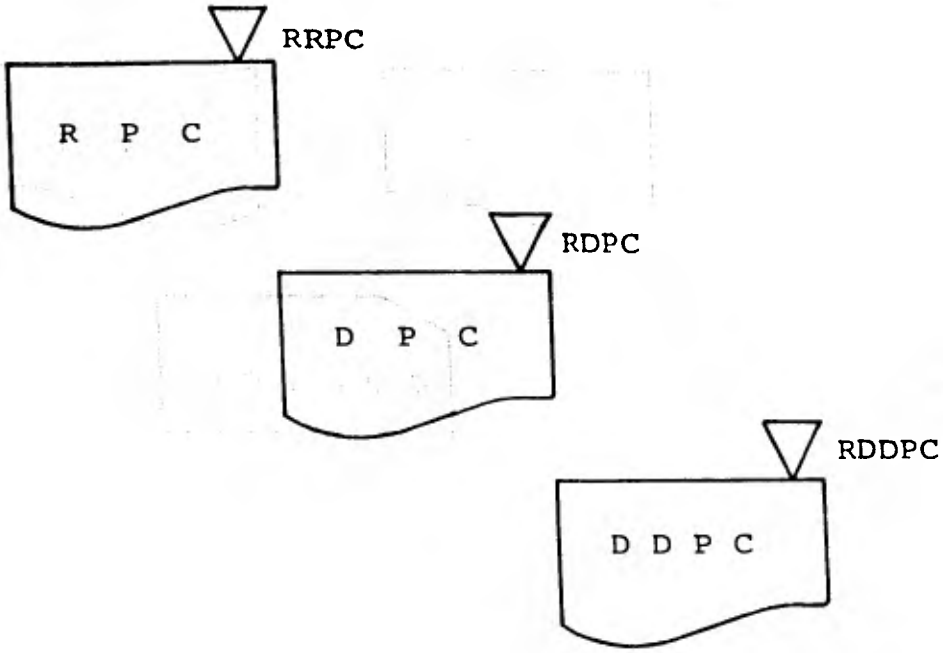


- Etapa 3

Revisión del documento DDPC (RDDPC)

Objetivo : determinar si el documento DDPC representa lo establecido en el DPC. Además, verificar si satisface los requerimientos del RPC.

Prerequisitos: (1) RRPC
(2) RDPC
(3) DDPC

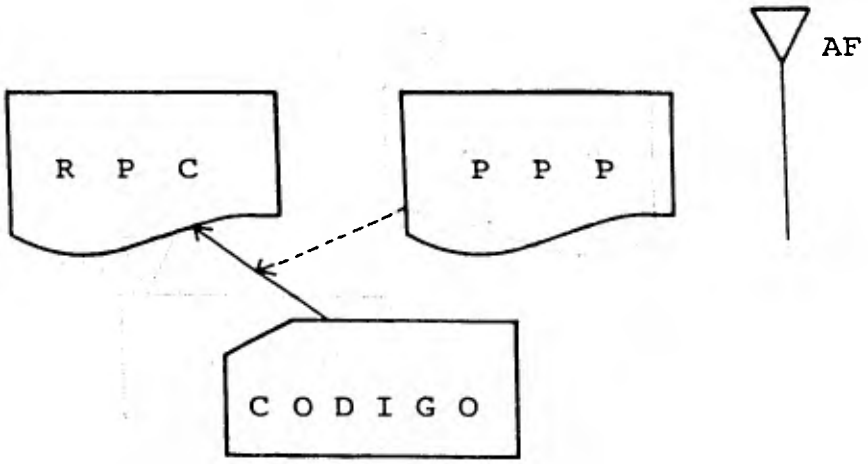


- Etapa 4

Auditoría funcional (AF)

Objetivo : verificar que se cumplan todos los requerimientos especificados en el RPC. En esencia, es una revisión de los resultados del plan de pruebas.

Prerequisitos : (1) RRPC
(2) RDPC
(3) RDDPC
(4) PPP

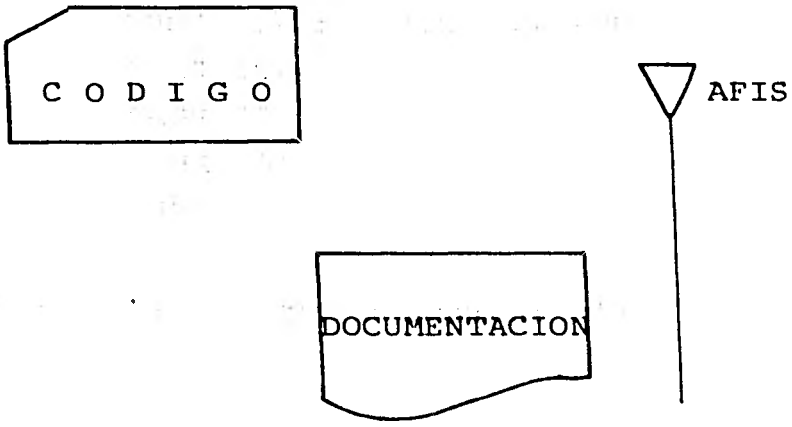


- Etapa 5

Auditoría física (AFIS)

Objetivo : verificar que el código y su documentación son consistentes y están listos para su entrega.

Prerequisitos : (1) RRPC
(2) RDPC
(3) RDDPC
(4) AF
(5) Código y documentación del mismo



- Etapa 6

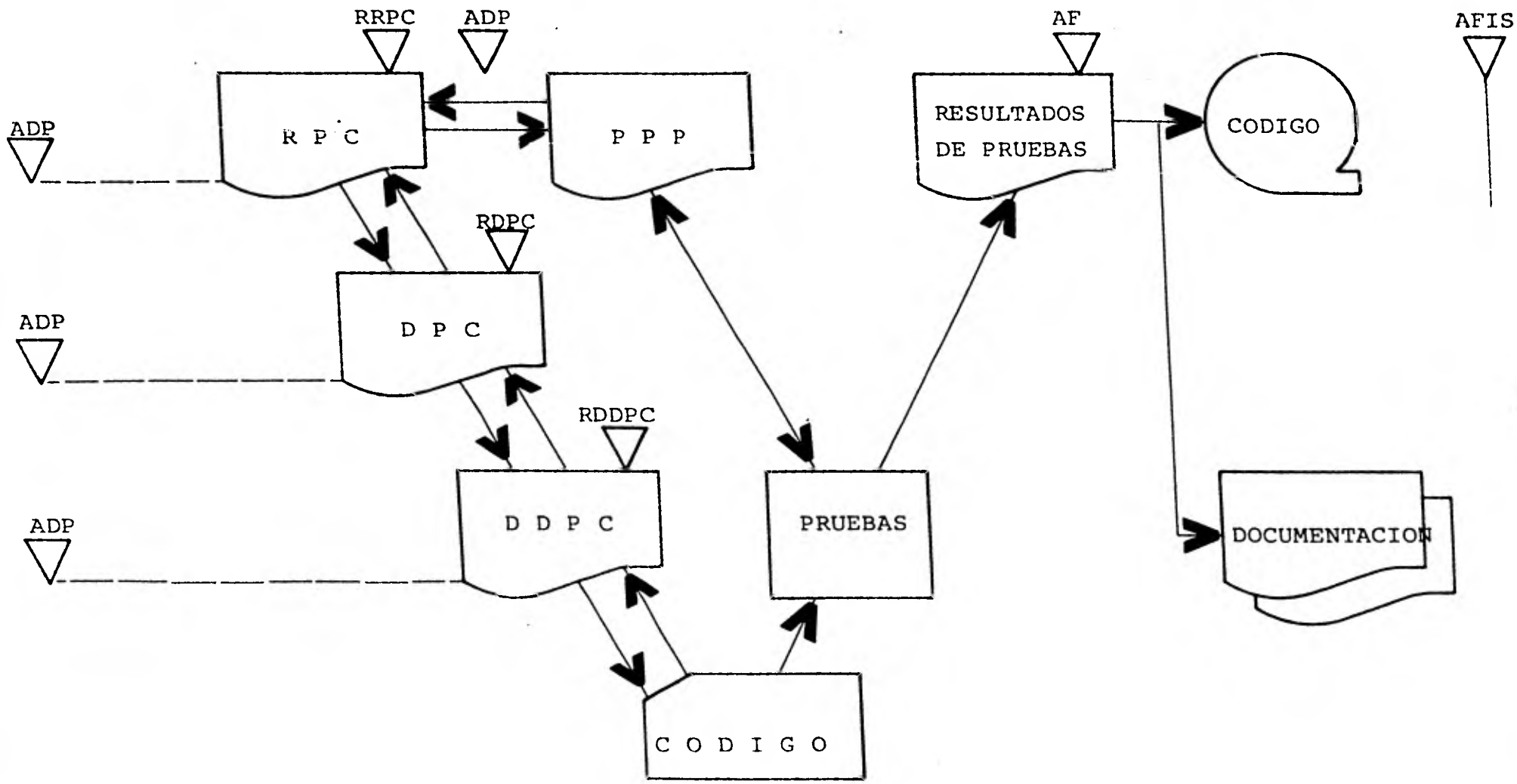
Auditoría del Proceso (ADP)

Objetivo : verificar la consistencia del-- diseño incluyendo :

- código vs. documentación del diseño
- especificación del interfase
- implementación del diseño vs. requerimientos funcionales
- requerimientos funcionales - vs. descripción de pruebas

Prerequisitos : (1) RRPC
(2) RDPC
(3) RDDPC
(4) AF
(5) AFIS

El diagrama se presenta a continuación.


RRPC
ADP
AF
AFIS
ADP
R P C
P P P
RESULTADOS
DE PRUEBAS
CODIGO
RDPC
ADP
D P C
RDDPC
ADP
D D P C
PRUEBAS
DOCUMENTACION
C O D I G O

CAPITULO III. INGENIERIA DE PROGRAMACION PARA EL PROGRAMA DE ESTIMACION DE CARGA HORARIA (PECHO).

III.1 Introducción.

El proceso de programación delineado en el capítulo anterior, es ensayado en el Instituto de Investigaciones Eléctricas como parte del desarrollo de Programas de Aplicación Avanzada en tiempo real para el "Sistema de Información y Control en Tiempo-Real del Sistema Eléctrico Nacional".

El desarrollo de programas de Aplicación Avanzada-en tiempo real involucrará aproximadamente 35 investigadores durante 3 años para la construcción - de aproximadamente 50,000 líneas de código.

Lo anterior nos permite clasificar el proyecto en-la categoría de "difíciles", según la descripción-del capítulo introductorio.

Las etapas del Proceso de Programación descritas - en el capítulo II, se aplican al desarrollo del - PECHO, que ha requerido hasta el momento,de un esfuerzo de 8 años/hombre. El Instituto de Investigaciones Eléctricas ha laborado en las siguientes - etapas del Proceso de Programación de este algoritmo :

- Formulación y análisis del problema.
- Análisis de alternativas.
- Selección de alternativas.
- Documento de requerimientos del programa de computadora.

PROGRAMA DE ESTIMACION DE

CARGA HORARIA

FASE : FORMULACION Y ANALISIS DEL PROBLEMA

III.2 Formulación y Análisis del problema de Estimación de carga eléctrica para cada una de las 6 áreas - operativas de la CFE.

III.2.1 Características de la Red Eléctrica Nacional.

La República Mexicana tiene una superficie cercana a los 2000 000 Km^2 y la capacidad instalada es de aproximadamente 16 GW. La demanda crece a ritmos superiores al 7% anual. De esta capacidad instalada, 41% es hidroeléctrica, 45% - termoeléctrica y el resto lo cubren unidades de ciclo combinado, turbo gas y geotérmicas.

Para transmitir esta generación a los centros - de consumo, se cuenta con una red de transmi--- sión de 400, 230 y 115 KV, que tiene una extensión de 20 000 Km. Este sistema, exceptuando - las penínsulas de Yucatán y Baja California, se encuentra interconectado, formando 6 áreas geográficas, eléctricamente bien definidas (ver -- figura III.2.1.1).

La estructura del sistema interconectado es tal, que puede ser operado como dos o más subsiste-- mas eléctricamente separados por períodos de -- tiempo relativamente largos ('islas').

Esta red cuenta con 450 nodos, 750 líneas de -- transmisión y 128 plantas generadoras, de éstas plantas pueden despacharse hasta 144 unidades - térmicas y 32 plantas hidráulicas considerando- las restricciones sobre los flujos de potencia- activos en 28 líneas inter-áreas y 20 enlaces -

entre áreas de control.

Figura III.2.1.1 AREAS DE CONTROL

■ Centro Nacional de Control de Energía

○ Centros de Control Regionales

III.2.2 Estructura jerárquica del sistema de control.

A fin de lograr la supervisión del sistema interconectado nacional, CFE ha planeado un sistema de adquisición de datos y control en tiempo-real y ha adquirido el equipo necesario.

El sistema consiste fundamentalmente de dos niveles :

1. El Centro Nacional de Control de Energía.
2. Los Centros de Control Regionales.

III.2.2.1 Centro Nacional de Control de Energía.

Este centro es el de más alta jerarquía. Sus principales funciones son :

- Coordinación de la transmisión en bloque de energía.
- Asignación de los recursos de generación.
- Cálculos de despacho económico.
- Programación de los intercambios de energía.
- Evaluación de la seguridad del sistema.

Basa estas funciones en información que recibe de los Centros de Control de Area a través de un sistema de adquisición de datos basado en unidades terminales remotas (UTR) y canales de comunicación que trabajan a 2400 bauds.

La porción del sistema correspondiente al Centro Nacional de Control de Energía (CNC) está formada por el sistema Harris 9400 (Fig. --- III.2.2.1.1.) Este sistema está integrado por

un sistema dual de cómputo y equipos periféricos asociados. El equipo de cómputo está formado por máquinas del tipo Slash/8 con memoria compartida. El proveedor. Harris, proporcionará para este centro los siguientes programas :

- Configurador de la red. Determina y mantiene la topología en tiempo real del modelo del sistema eléctrico.
- Estimador de estado de sistemas. Identifica errores de medición y estima cual es el estado del sistema.
- Evaluación de costos de producción.
- Monitoreo de la reserva. Determina la reserva de generación, tanto en base a las áreas como a todo el sistema y permite minimizar los excesos de reserva no planeados.
- Análisis de seguridad. Empleando en tiempo real el modelo de sistema de potencia y una serie de contingencias pre-programadas alerta al despachador sobre potenciales problemas de sobrecarga.
- Flujo de carga del despachador. Permite estudiar al despachador los efectos sobre el sistema de acciones de control proyectadas empleando un modelo real de la red.
- Generador de Reportes.

Adicionalmente se requieren cuatro programas:

- Pronóstico de Carga Nacional y de Area a -

corto plazo.

- Coordinación hidroeléctrica restringida a corto plazo.
- Despacho económico restringido.
- Programación de intercambios.

CFE ha tomado la decisión de desarrolar estos programas en México. Como razones principales para esta decisión pueden citarse las siguientes :

Conveniencia de desarrollar una capacidad de programación en tiempo real en México.

Necesidad de incorporar a los programas, características peculiares de la red nacional de tipo no comercial.

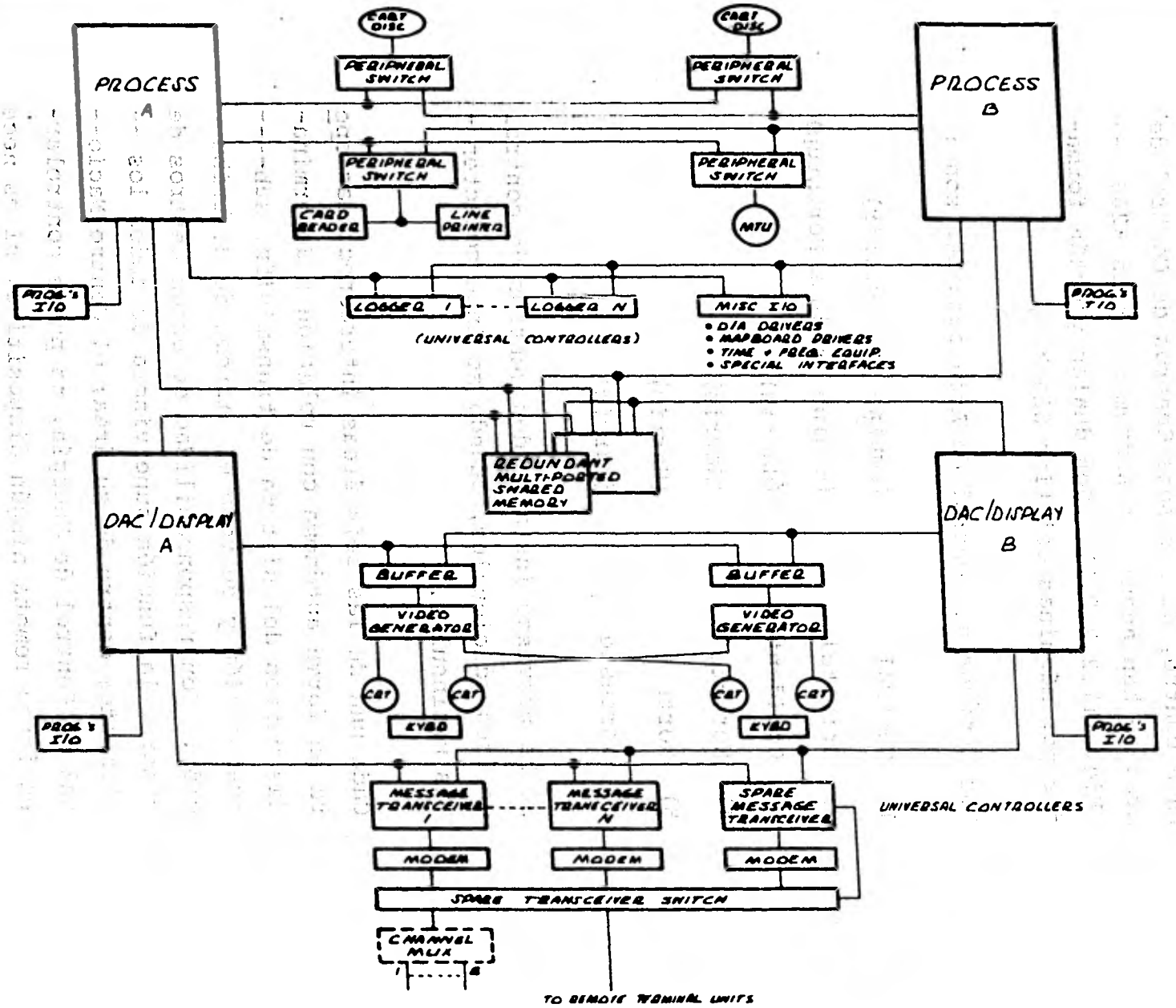


FIG. III 2.2.1.1 DIAGRAMA DE BLOQUES DEL SISTEMA HARRIS M 9400

III.2.2,2 Centros de Control de Area.

Para cuatro, de los seis Centros de Control de Area se han adquirido sistemas Harris 9200 -- (Fig. III.2,2,2.1). Son duales y están formados por máquinas del tipo Slash/8.

Los Centros de Control de Area actuales son :

1)	Central	(Equipo por definir)
2)	Oriental	(Equipo Leeds & Northrup)
3)	Occidental	(Harris)
4)	Noreste	(Harris)
5)	Norte	(Harris)
6)	Noroeste	(Harris)

Se ha previsto instalar dos centros de control de área adicionales, cuando se interconecten las penínsulas.

Cada una de las seis áreas interconectadas opera en forma autónoma con relación a determinadas partes del sistema de transmisión, sub---- transmisión y de todo el sistema de distribución. Son responsabilidad de estos centros de control la función supervisora de todos los -- elementos dentro de su área; el Centro Nacional de Control de Energía, no puede controlar en forma remota ningún dispositivo, si es necesario enviará directrices a las diferentes áreas para implementar las funciones de control adecuadas.

Para la realización de sus funciones, los centros de control regionales tendrán las siguientes capabilidades :

- Control automático de generación.
- Monitoreo del funcionamiento de los equipos.
- Chequeo de la telemedición en los enlaces.
- Generación de reportes sobre el sistema.
- Rechazo de carga y restauración.
- Colección de datos de disturbio y análisis.

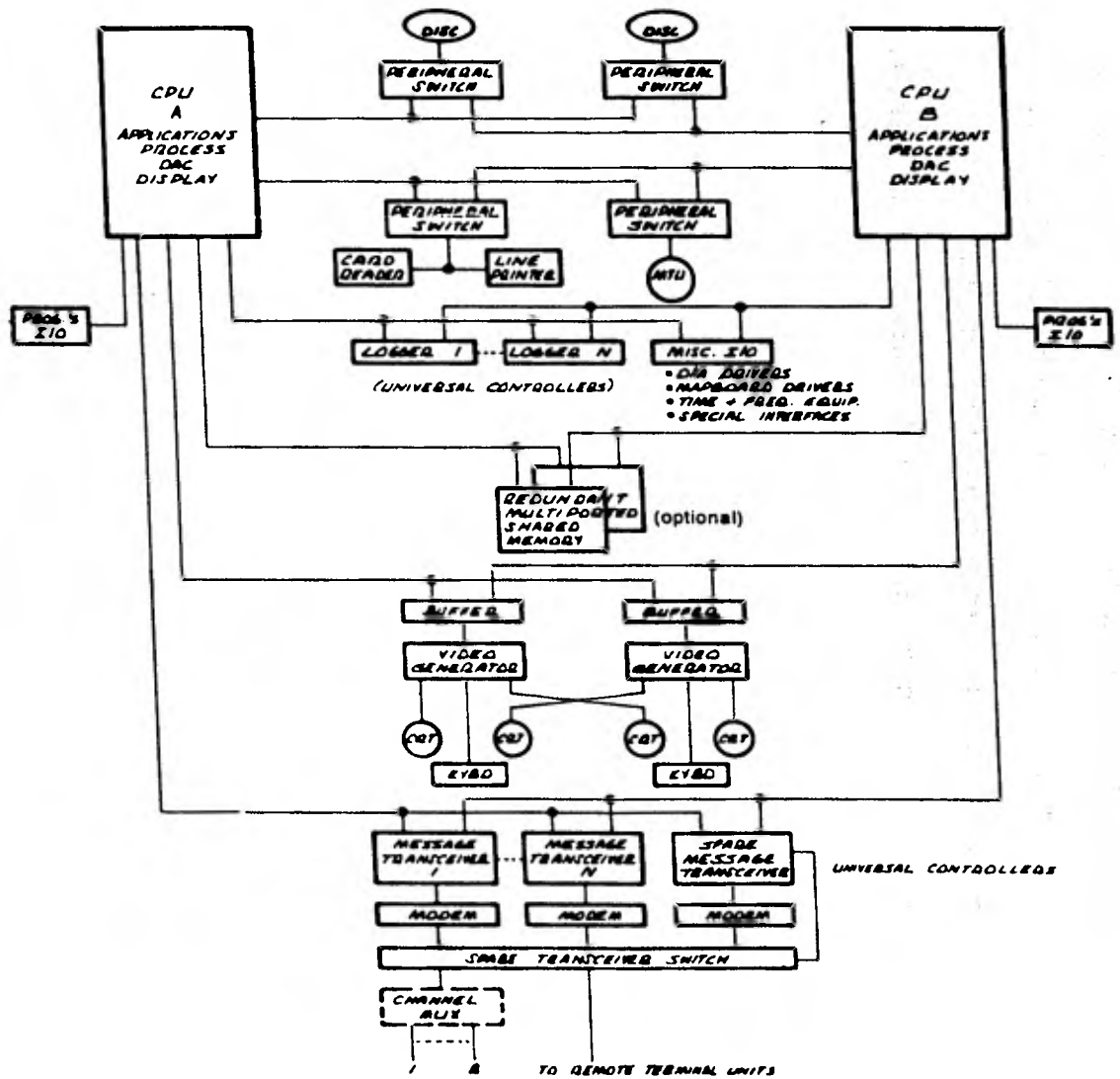


FIG. III.2.2.2.1 DIAGRAMA DE BLOQUES DEL SISTEMA HARRIS M 9200

III.2.3 Objetivo del PECHO.

El objetivo del PECHO es predecir la carga total horaria en MWH para el sistema nacional interconectado y para cada una de las seis áreas operativas de la Comisión Federal de Electricidad, considerando regiones climatológicas dentro de cada área y para períodos de tiempo hasta de siete días de adelanto. Asimismo, proporcionará el pronóstico semanal de energía cada seis horas.

III.2.4 Origen del problema, generalidades.

Los planes para una expansión y un control adecuado de un Sistema Eléctrico de Potencia comienzan con un pronóstico de los requerimientos futuros de Energía Eléctrica.

III.2.5 Programa del Trabajo Propuesto.

III.2.5.1 Enfoque del trabajo.

Ya se han investigado esquemas de pronóstico que han sido instrumentados por Centros de Control Automático de Energía Eléctrica; así mismo, se han analizado otras técnicas que en la literatura existente parecen ser las más poderosas.

Es necesario estudiar alternativas de modelos de carga dependientes de variables meteorológicas, para integrarlos al algoritmo que predecirá finalmente las cargas eléctricas en forma horaria.

III.2.5.2 Obtención de Información.

Como complemento se deberán aportar todos los datos necesarios para la modelación del sistema en forma oportuna y se tomarán las medidas necesarias para que sean obtenidos aquellos datos del sistema que, siendo necesarios no estén actualmente en disponibilidad.

Se deberá así mismo contribuir con toda su experiencia y conocimiento de las particularidades del sistema para facilitar el trabajo a realizar. Se contará con un grupo de ingenieros a quienes se podrá recurrir para obtener los datos y la información necesaria y colaborar al desarrollo de los programas.

III.2.5.2.1 Información eléctrica.

Es indispensable contar con las demandas horarias de cada una de las seis áreas de CFE de los últimos dos años (1978-1979). Se recopilará esta información en los correspondientes Centros de Control de Area.

Debe tomarse en cuenta que el pronóstico de carga eléctrica nunca podrá mejorar la calidad de la información empleada.

III.2.5.2.2 Información meteorológica.

En base a los estudios realizados con la información meteorológica del área NW y del Area Central de CFE, a las experiencias de-

diversos centros automáticos de control de energía en el mundo y a la revisión de la literatura existente, se ha concluido que los parámetros climáticos que pueden afectar al comportamiento del consumo de energía eléctrica son :

- Temperatura promedio.
- Temperatura máxima.
- Temperatura mínima.
- Humedad relativa.
- Luminosidad.
- Precipitación pluvial integrada cada tres horas.

Se consultará con diferentes dependencias (tanto nacionales como extranjeras) que manejan parámetros meteorológicos, con objeto de :

- Conocer qué organismos pueden suministrar esta información.
- Recopilar la información disponible que sea necesaria para el desarrollo de modelos de carga eléctrica dependientes de variables climáticas.
  La información deberá incluir lecturas horarias o trihorarias de 1978 y 1979 para los centros de consumo de energía eléctrica más importantes en cada área.

III.2.5.2.3 Pronóstico Meteorológico.

El pronóstico meteorológico a utilizar en la predicción de demanda deberá incluir los parámetros defindios en los modelos climáticos para las diferentes áres de CFE. Debe consultarse con los organismos que pueden ofrecer servicios de pronóstico meteorológico (tanto nacionales como extranjeros), a fin de entregar una propuesta que valore las distintas alternativas y sugerir los procedimientos de captura de la información bajo condiones normales de operación del programa.

Entre las dependencias que podrían proveer esta información se encuentran :

- Servicio Meteorológico Nacional.
- SENEAM.
- Global Weather Dynamicas, Inc.
- A.H. Glenn and Associates.

Cabe aclarar que, una desviación en el pronóstico meteorológico disminuirá la calidad del pronóstico de la demanda de energía eléctrica.

III.2.5.3 Requerimientos básicos y restricciones operativas del PECHO.

III.2.5.3.1 Requerimientos Básicos.

El programa proporcionará un resumen indi--

cando el valor del pronóstico semanal de energía cada seis horas, y la estimación horaria de los primeros 2 días en MWH.

El programa que estimará las cargas horarias y la energía integrada cada seis horas (en MWH) se ejecutará diariamente.

Cuando el pronóstico se desvíe sustancialmente de la carga real se podrá ejecutar otro algoritmo que actualizará las demandas horarias con la información más reciente.

El programa de pronóstico deberá estar estructurado en base a un modelo meteorológico (siempre y cuando exista en el área la dependencia con respecto a parámetros climáticos) y tendrá una característica adaptiva que permita la actualización automática de sus parámetros.

El operador (en el correspondiente centro de control de área) tendrá la facultad de modificar los pronósticos así como de cancelar una actualización si, en su juicio, el antiguo modelo refleja mejor las características del consumo de energía eléctrica.

En caso de modificarse los valores del pronóstico, se llevará una estadística para evaluar la intervención manual del operador. (No se permitirá al operador modificar los pronósticos para horas anteriores).

III.2.5.3.2. Restricciones Operativas.

- El algoritmo deberá verificar automáticamente los datos de entrada.
- La información meteorológica será alimentada por las terminales de video.
- Los resultados del pronóstico deberán estar disponibles para el operador, para otros programas de aplicación en la base de datos del Centro Nacional de Control y para ser transmitidos al correspondiente Centro de Control de Area.
- Se incluirá un calendario que permita detectar los días que deban ser considerados como días anormales para efectos del pronóstico.
- Se estimarán las cargas horarias de todos los días festivos del año, con exactitudes comparables a las de los días anormales, siempre y cuando exista información suficiente en la historia para caracterizar los diferentes tipos de días anormales.
- El tiempo de adelanto del pronóstico podrá ser definido arbitrariamente por el operador.
- Cada pronóstico de carga horaria deberá ir acompañado de su desviación estándar.
- Dependiendo de la metodología empleada, el programa de pronóstico deberá incluir un filtrado automático de información con objeto de reconocer los días que tie

nen características especiales o anormales en los patrones de consumo de energía eléctrica, para no emplearlos en el proceso de aprendizaje del algoritmo de pronóstico.

- El archivo de pronóstico será actualizado una vez que el operador esté satisfecho con los resultados.
- Para cada una de las seis áreas operativas de CFE existirá información en la base de datos del Centro Nacional de Control de las cargas industriales que vayan a librarse en determinados períodos de tiempo en el futuro.
- Con objeto de inicializar los modelos del programa de pronóstico, será necesario contar con la información eléctrica y meteorológica correspondiente a los últimos 24 meses. Esta información estará disponible en cinta además de la información que se encuentra almacenada en el banco de datos del Centro Nacional.
- Los datos de entrada que manejará el programa son :

  a) Parámetros del modelo (dependiendo de la metodología).

  b) Valor horario de las cargas almacenadas en la base de datos del Centro Nacional de Control.

  c) Pronóstico Meteorológico.

  d) Tipo de día por predecir.

  e) Fecha.

III.2.5.3.2 Modelos Meteorológicos.

Se desarrollarán modelos meteorológicos con las siguientes características :

- Los modelos para un área incluirán, dentro de lo posible, información meteorológica de varios lugares en el área.
- Los modelos deberán estar estructurados en forma tal, que utilicen al máximo las características adaptivas del programa de pronóstico de carga.
- Se minimizará el número de observatorios meteorológicos a utilizar en el pronóstico de carga de un área dada.

III.2.5.3.3 Programación para modelar carga eléctrica en función de parámetros meteorológicos.

Se proporcionará un paquete de programación que permita actualizar fuera de línea, los modelos meteorológicos que afectan la carga eléctrica.

El paquete incluirá , entre otras funciones:

- Prueba Chi cuadrada.
- Estadística de los datos.
- Auto correlación.
- Cross correlación.

III.2.5.3.4 Interfase Hombre-Máquina.

Los resultados de los cálculos involucrados en el pronóstico de carga serán editados en forma tal, que puedan ser utilizados por la función de interfase hombre-máquina para -- presentar en las terminales de video u -- otras formas de impresión, la siguiente información :

- Pronóstico de carga horario (tabular y - gráfico).
- Pronóstico de energía diaria (cada 24 horas).
- Carga mínima.
- Carga máxima.
- Pronóstico de carga horaria para períodos específicos de tiempo en el futuro.
- Pronóstico de la carga base para períodos específicos de tiempo en el futuro- (la carga base será definida en base al algoritmo seleccionado).
- Cargas horarias para períodos de tiempo específicos en la historia.
- Residuos horarios de carga para períodos de tiempo específicos en la historia.
- Estimación de la desviación estándar -- horaria de la carga.
- La componente meteorológica del modelo- de carga (siempre y cuando exista la - dependencia meteorológica en el área -

analizada).

- Información sobre mediciones anormales - que incluye : carga medida, carga pronosticada y la fecha.
- Pronóstico de variables meteorológicas.

La información misma de las tablas de valores presentados en als pantallas de video - será hecha por personal de la CFE. Se asegurará que todos los datos necesarios estarán disponibles en la base de datos del --- Centro Nacional de Control a través del uso de los procedimientos de generación de imágenes en pantalla del sistema Harris.

PROGRAMA DE ESTIMACION DE

CARGA HORARIA

FASE : ANALISIS DE ALTERNATIVAS

III.3 Análisis de Alternativas para el PECHO.

III.3.1 Introducción.

El objetivo de esta etapa, es obtener las alternativas para el algoritmo a emplear en el proceso de estimación de carga.

Para analizar las alternativas, es necesario empezar desde la recopilación de datos, pasando por el análisis estadísticos de los mismos y la realización de programas de prueba, hasta obtener resultados para su evaluación comparativa.

En la sección se muestra la recopilación y análisis estadístico de los datos correspondiente al área Noroeste de CFE.

En III.3.9 se describen brevemente los algoritmos propuestos para el PECHO.

III.3.2 Descripción del Area Noroeste.

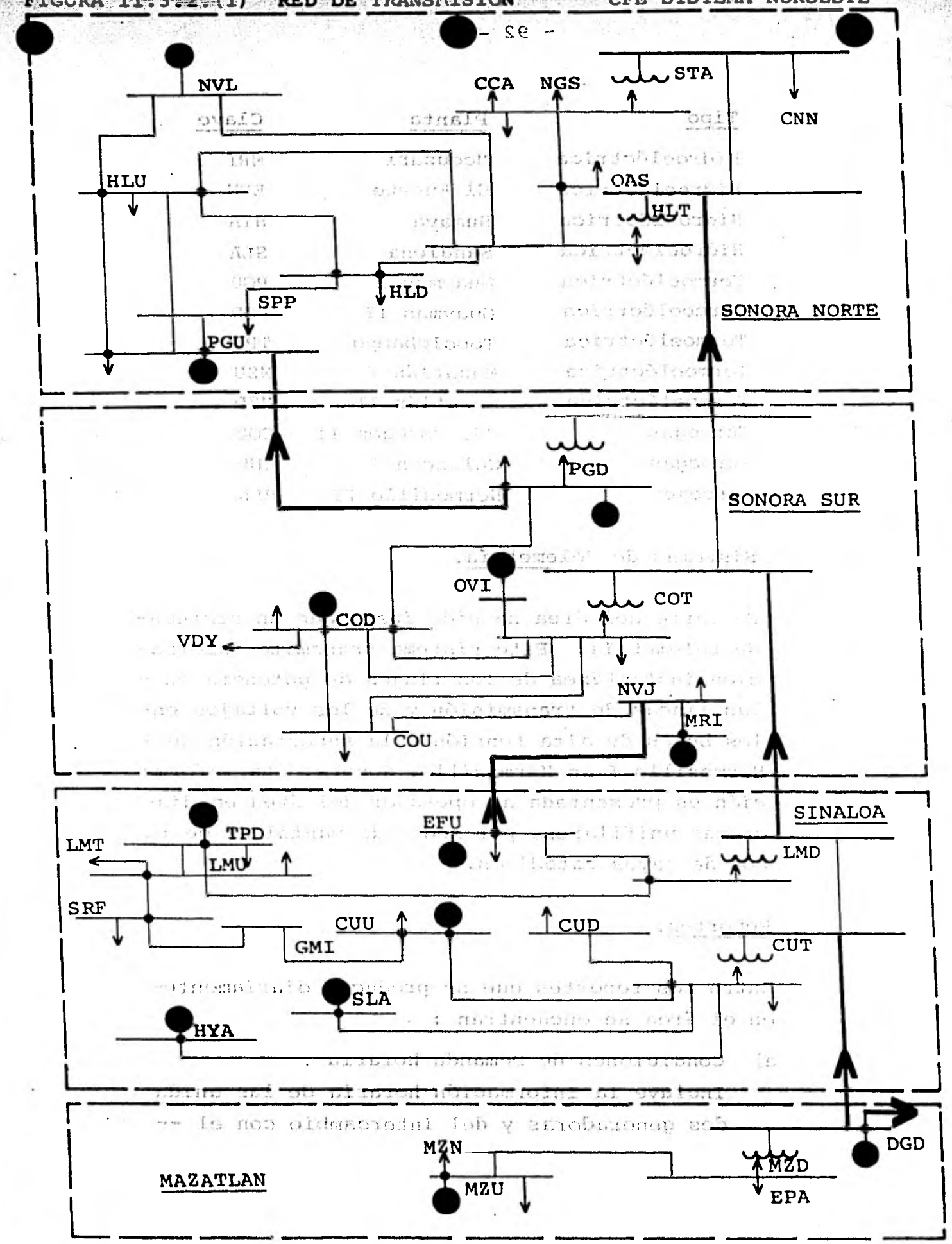
El Area Noroeste comprende los Estados de Sonora y Sinaloa. El diagrama unifilar de la Figura III.3.2.(1) muestra la red de transmisión del área y su intercambio con el Area Norte.

Plantas.

La producción de energía eléctrica es aportada por las siguientes plantas :

Tipo	Planta	Clave
Hidroeléctrica	Novillo	NVL
Hidroeléctrica	Oviachic	OVI

FIGURA II.3.2.(1) RED DE TRANSMISION CFE SISTEMA NOROESTE



Tipo	Planta	Clave
Hidroeléctrica	Mocuzari	MRI
Hidroeléctrica	El Fuerte	EFU
Hidroeléctrica	Humaya	HYA
Hidroeléctrica	Sanalona	SLA
Termoeléctrica	Guaymas I	PGU
Termoeléctrica	Guaymas II	PGD
Termoeléctrica	Topolobampo	TPO
Termoeléctrica	Mazatlán I	MZU
Termoeléctrica	Mazatlán II	MZD
Turbogas	Cd. Obregón II	COD
Turbogas	Culiacán I	CUU
Turbojet	Hermosillo II	HLD

Sistemas de Telemetría.

En parte del área ha sido instalado un sistema de telemetría. Este sistema transmite información instantánea de los flujos de potencia en las líneas de transmisión y de los voltajes en los buses de alta tensión a la subestación de Hermosillo I en Hermosillo, Sonora. La información es presentada al operador del área en diagramas unifilares, por medio de pantallas de tubos de rayos catódicos.

Reportes.

Entre los reportes que se producen diariamente en el área se encuentran :

a) Condiciones de Demanda Horaria :
Incluye la información horaria de las unidades generadoras y del intercambio con el --

Area Norte.

b) Reporte de Novedades :
Historia diaria de los problemas que se --- presentan en el área, así como detalles de- las influencias climatológicas observadas.

c) Reporte de Capacidades de las Unidades Gene radoras :
Registra la capacidad de las diferentes uni dades generadoras disponibles en el área.

d) Reporte de Unidades Libradas :
Contiene la fecha de salida, la fecha de -- entrada y la razón de la libranza de las di ferentes unidades generadoras.

Subáreas.

Para efectos de distribución y por razones his- tóricas, el Area Noroeste ha sido subdividida - en cuatro subáreas :

a) Sonora Norte
b) Sonora Sur
c) Sinaloa
d) Mazatlán

La localización geográfica de las subáreas se - muestra en el mapa de la Figura III.3.2.(2).

De pláticas con el personal encargado de la --- Operación del Sistema en el Area Noroeste, se - afirmó que la demanda en Sonora es afectada se- riamente por la irrigación. Como ejemplo se ci tó que un cambio de cultivo ha afectado los pa-

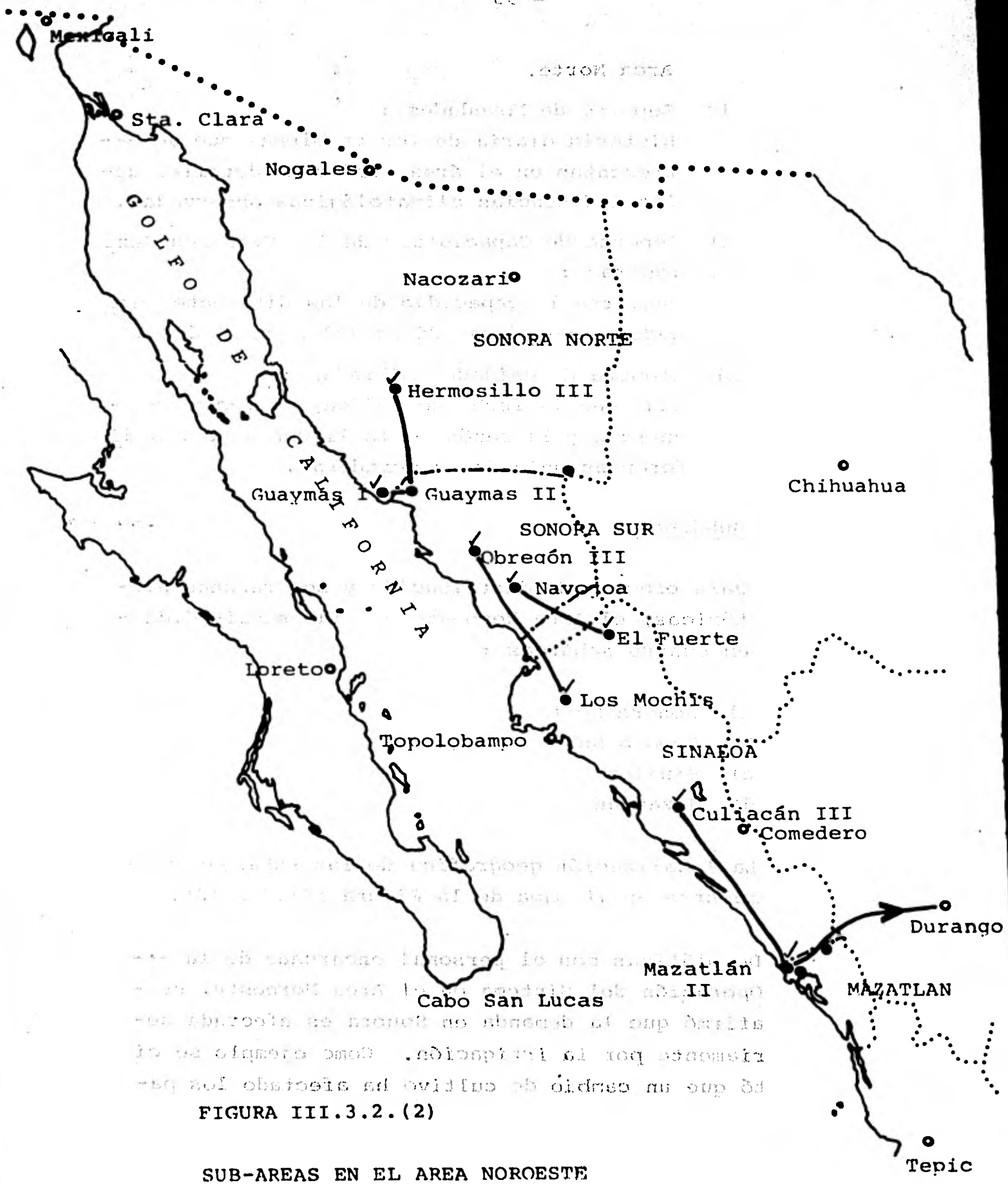


FIGURA III.3.2.(2)

SUB-AREAS EN EL AREA NOROESTE

trones en la demanda al variar la temporada de irrigación. Así mismo, se nos informó que, en Sonora, la carga agrícola es más importante que la carga poblacional, especialmente durante los ciclos de riego; en Sinaloa la carga poblacional es muy importante y finalmente que, en Mazatlán, la temperatura tiene mucha influencia sobre los patrones de demanda.

Intercambios.

El área Noroeste tiene un solo enlace con el resto del Sistema eléctrico del país, por medio de la línea Mazatlán II-Durango II, que la une al Area Norte.

Coordinación de Recursos Hidráulicos.

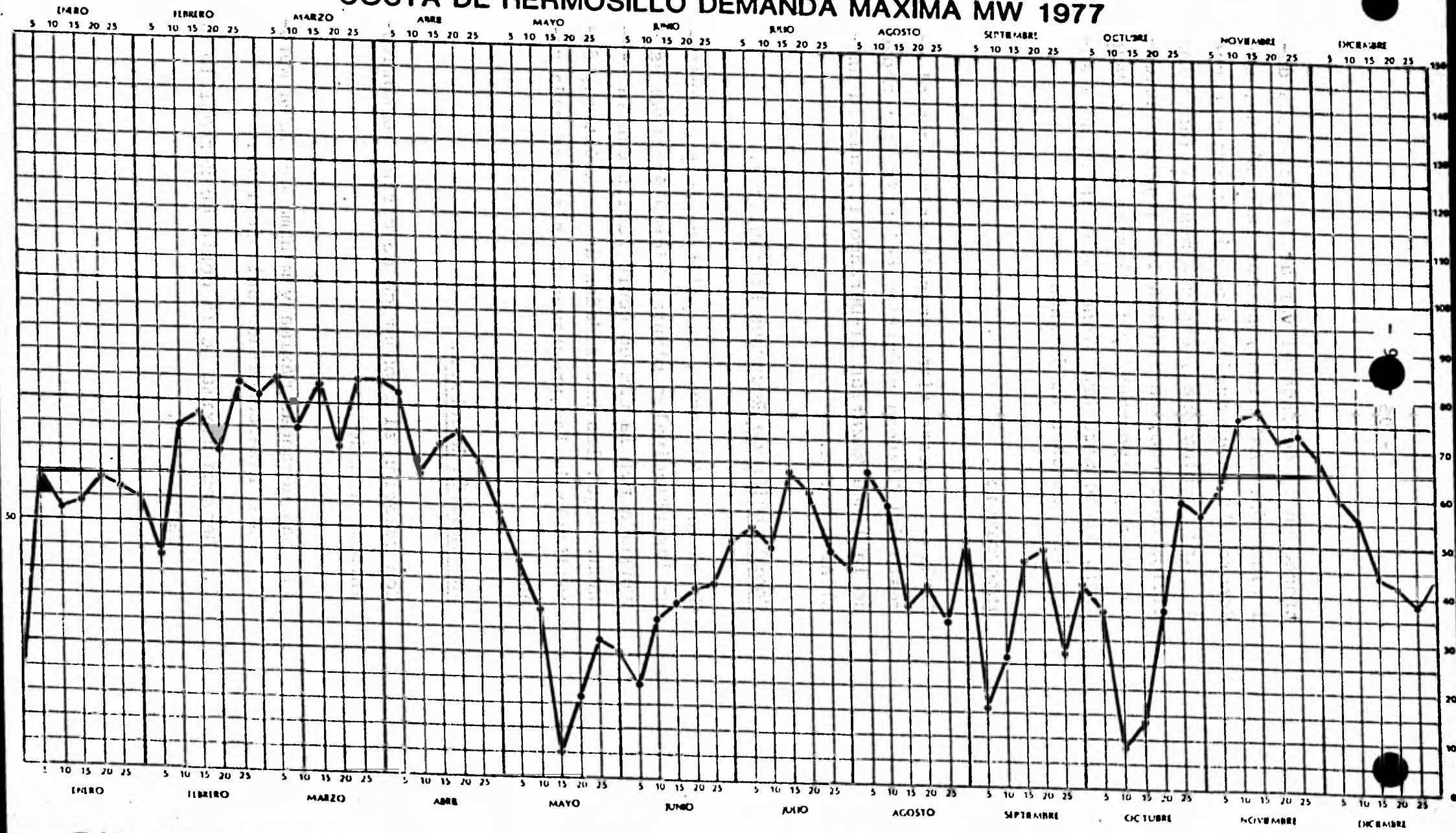
En el Area Noroeste, la política de producción en las principales plantas hidroeléctricas es manejada por la CFE en coordinación con la SARH, según las necesidades de esta última, dependiendo de los programas de riego en cada zona.

Información Recabada.

Con relación al pronóstico de carga, se obtuvo información de :

1. Las demandas máxima y mínima del Area Noroeste interconectado del año 1977.
2. Demanda máxima de la costa de Hermosillo durante 1977.

COSTA DE HERMOSILLO DEMANDA MAXIMA MW 1977



3. Información horaria de los enlaces entre las subáreas y del intercambio con el Area Norte del sistema noroeste desde Abril/77 a Abril/78. Asimismo, se obtuvieron los valores de las constantes de los watthorímetros empleados (Ver Tabla III.3.2.(3)).

4. Información horaria de las unidades generadoras en todo el sistema de Abril/77 a Abril/-78. Se obtuvieron los valores de las constantes de los watthorímetros empleados (Ver Tabla III.3.2.(4)).

LINEA MEDIDA	CONSTANTE DEL MEDIDOR
Santa Ana - Cananea	10^4
Santa Ana - Caborca	6×10^4
Guaymas II - Guaymas I	10^3
Hermosillo III - Guaymas II	10^4
El Fuerte - Navojoa(recibir)	2×10^4
El Fuerte - Navojoa(mandar)	2×10^3
Mochis II - Cd. Obregón III	16×10^4
Mazatlán II - Culiacán III	10^4
Mazatlán II - Durango II	10^4

TABLA III.3.2.(3) CONSTANTES DE LOS MEDIDORES EN LOS ENLANCES DEL AREA NOROESTE.

GENERADORES	CONSTANTE DE MEDIDORES
Novillo	10^4
Oviachic	10^3
Mocuzari	10^3
El Fuerte	10^4
Humaya	10^4
Sanalona	10^3
Guaymas I	10^4
Guaymas II	10^4
Topolobampo	10^3
Mazatlán I	10^3
Mazatlán II	10^4
Cd. Obregón II	10^3
Culiacán I	10^3
Hermosillo II	10^3

TABLA III.3.2.(4) CONSTANTES DE LOS MEDIDORES EN LOS GENERADORES.

(RWM)

III.3.3 Información Eléctrica.

Los reportes utilizados para registrar la generación horaria(en MWH)detallan la energía eléctrica entregada por cada unidad generadora en las plantas. Asimismo, se reporta el valor de la energía eléctrica de intercambio que fluye por el enlace con el Area Norte(Mazatlán II --- Durango II).

Este último valor procede de la Subestación Durango II. Para simplificar la explicación definamos :

G(i,j)	Energía generada por la unidad i durante la hora j (MWH).
I(k,j)	Valor del intercambio k durante la hora j (MWH).
NG	Número total de unidades generadoras en el área.
NI	Número total de intercambios con --- otras áreas (para el Area Noreste -- NI = 1).
LG(i,j)	Lectura del watthorímetro de la unidad generadora i a la hora j (MWH).
LI(k,j)	Lectura del watthorímetro del intercambio k a la hora j (MWH).
ETG(i)	Energía total generada por la unidad generadora i durante el día (MWH).
ETI(k)	Energía total de intercambio que fluye por el enlace k durante el día -- (MWH).

Los valores G(i,j) e I(k,j), registrados en los reportes denominados "Condiciones de Demanda -- Horaria", se obtienen en la forma :

$$G(i,j) = A_i \{ LG(i,j) - LG(i,j-1) \} \quad (1)$$

$$I(k,j) = B_k \{ LI(k,j) - LI(k,j-1) \} \quad (2)$$

A_I Y B_K SON LOS VALORES DE LAS CONSTANTES DE -- LOS WATTHORÍMETROS DE LA UNIDAD GENERADORA I Y- DEL INTERCAMBIO K, RESPECTIVAMENTE.

En el Area Noroeste se sigue la convención de - considerar al intercambio como positivo cuando- se entrega energía al Area Norte (que es lo que generalmente ocurre). En caso de recibirse --- energía, el intercambio se toma negativo.

Con la información horaria recibida determinan- a las 6, 14, 22 y 24 horas las cantidades C1 y- C2 definidas en la siguiente forma :

$$C_1 (HI,HF) = \sum_{i=1}^{NG} \sum_{j=HI}^{HF} G(i,j) - \sum_{k=1}^{NI} \sum_{j=HI}^{HF} I(k,j) \quad (3)$$

$$C_2 (HI,HF) = \sum_{j=HI}^{HF} \sum_{i=I}^{NG} G(i,j) - \sum_{j=HI}^{HF} \sum_{k=1}^{NI} I(k,j) \quad (4)$$

En donde :

C_1 Corresponde al cálculo del consumo del área por columnas.

C_2 Corresponde al cálculo del consumo del área por renglones.

HI Es la hora a partir de la cual se calculan- los valores de C_1 y C_2.

HF Es la hora hasta la cual se calculan los valores de C_1 y C_2.

En el Area Noroeste, HI y HF toman los valores- 6, 14, 22 ó 24.

Las expresiones C_1 y C_2 proporcionana la entrega neta del Area Noroeste durante el período de tiempo HI a HF.

En caso de coincidir los valores C_1 y C_2 se "Entrega el Turno" al siguiente lecturista. Si hubo error en alguna de las sumatorias 3 ó 4, serán diferentes (salvo por el poco probable caso de errores compensados), por lo que entonces se procede a verificar la suma de los valores G(i, j) e I(k,j) desde la hora HI hasta la hora HF,- hasta encontrar y corregir el error.

A las 24 horas se procede a "Cerrar" la información del día en la siguiente forma :

1. Cierre de Unidades Generadoras :

Se transmiten a Hermosillo los valores :

A_i LG(i,24) donde i=1,...NG (5)

B_k LI(k,24) donde k=1,...NI (6)

que corresponden a las lecturas de los wattthorímetros de las unidades generadoras y- la energía que fluye por los enlaces a la - hora 24.

Con el conocimiento de estos valores para - el día inmediato anterior se obtienen las - energías totales producidas por las unida--

des generadoras y, las enviadas a través de los enlaces. Se calculan entonces :

$$ETG(i) = A_i\{LG(i,24)_{Hoy} - LG(i,24)_{Ayer}\} \quad (7)$$

$$ETI(k) = B_k\{LI(k,24)_{Hoy} - LI(k,24)_{Ayer}\} \quad (8)$$

A continuación se investiga si se cumplen las igualdades :

$$\sum_{j=1}^{24} G(i,j) = ETG(i) \quad (9)$$

$$\sum_{j=1}^{24} I(k,j) = ETI(k) \quad (10)$$

En donde $i = 1...NG$ y
$k = 1...NI$

En caso afirmativo, se acepta como correcta la información de las plantas y de las energías en los enlaces. En caso contrario se procede a verificar las lecturas de cada hora hasta determinar el error.

2. <u>Cierre del Area</u>.

Se calcula la expresión :

$$CT_1 = \sum_{i=1}^{NG} ETG(i) - \sum_{k=1}^{NI} ETI(k) \quad (11)$$

que proporciona la energía neta total del área durante el día.

OBSERVACIONES

Cabe mencionar aquí los siguientes detalles :

1. Tanto en las unidades generadoras como en los enlaces, se cuenta con medidores de corriente y potencia instantánea (real y reactiva), además de los watthorímetros mencionados.

2. Las lecturas de los flujos en los enlaces LT(k,j), y de generación, LG(i,j), se toman en instrumentos que generalmente se calibran una vez al año.

3. Los lecturistas comienzan sus mediciones faltando cinco minutos para la hora y terminan aproximadamente a la hora.

4. En casos de disturbios, las mediciones se toman hasta que la falla ha sido corregida.

5. Se nos comentó que la información en los turnos nocturnos tiende a ser poco confiable.

6. Toda la información horaria está disponible únicamente en documentos fuente (reportes), pero no ha sido captada en ningún medio que facilite su proceso.

7. Se nos informó que el Area Noroeste no tiene problemas para satisfacer su demanda, esto se debe a que, en caso necesario, se dispone de energía procedente de Texas, de la Central Power & Light a través de los enlaces entre las Areas Noroeste, Norte y Noreste. Por tanto, no es necesario corregir los datos del reporte de Condiciones de De-

manda Horaria como se hace en otras áreas-- cuando se opera a baja frecuencia o cuando- es necesario cortar carga.

3. Cierre de Subestaciones.

Personal del Centro Nacional de Control de Energía comentó que en ninguna subestación de la Comisión Federal de Electricidad se- hace cierre de información.

III.3.4 Procedimiento de Pronóstico Empleado en el Area Noroeste.

Método.

Los pronósticos de carga se preparan actualmente a base de gráficas. Estas son utilizadas para observar en forma horaria la tendencia de la carga. La predicción es resultado de una apreciación visual de la tendencia, a criterio del analista.

El personal del área ha observado que los días Martes, Miércoles, Jueves y Viernes tienen un comportamiento similar, y que la curva de demanda es relativamente independiente de la temperatura los días Sábado, Domingo y Lunes.

Se observó que el principal objetivo del pronóstico es la estimación de la carga del sistema a la hora pico.

Factores Climatológicos y Agrícolas.

Aunque los factores climatológicos y de irrigación son muy importantes en esta área, no se toman en cuenta para el pronóstico a 24 horas, por diversas razones. Entre otras podemos mencionar :

a) La falta de información meteorológica oportuna.
b) Si la información existiera, la falta de metodología para incorporarla en un esquema de predicción de carga a 24 horas.

Curva de Demanda.

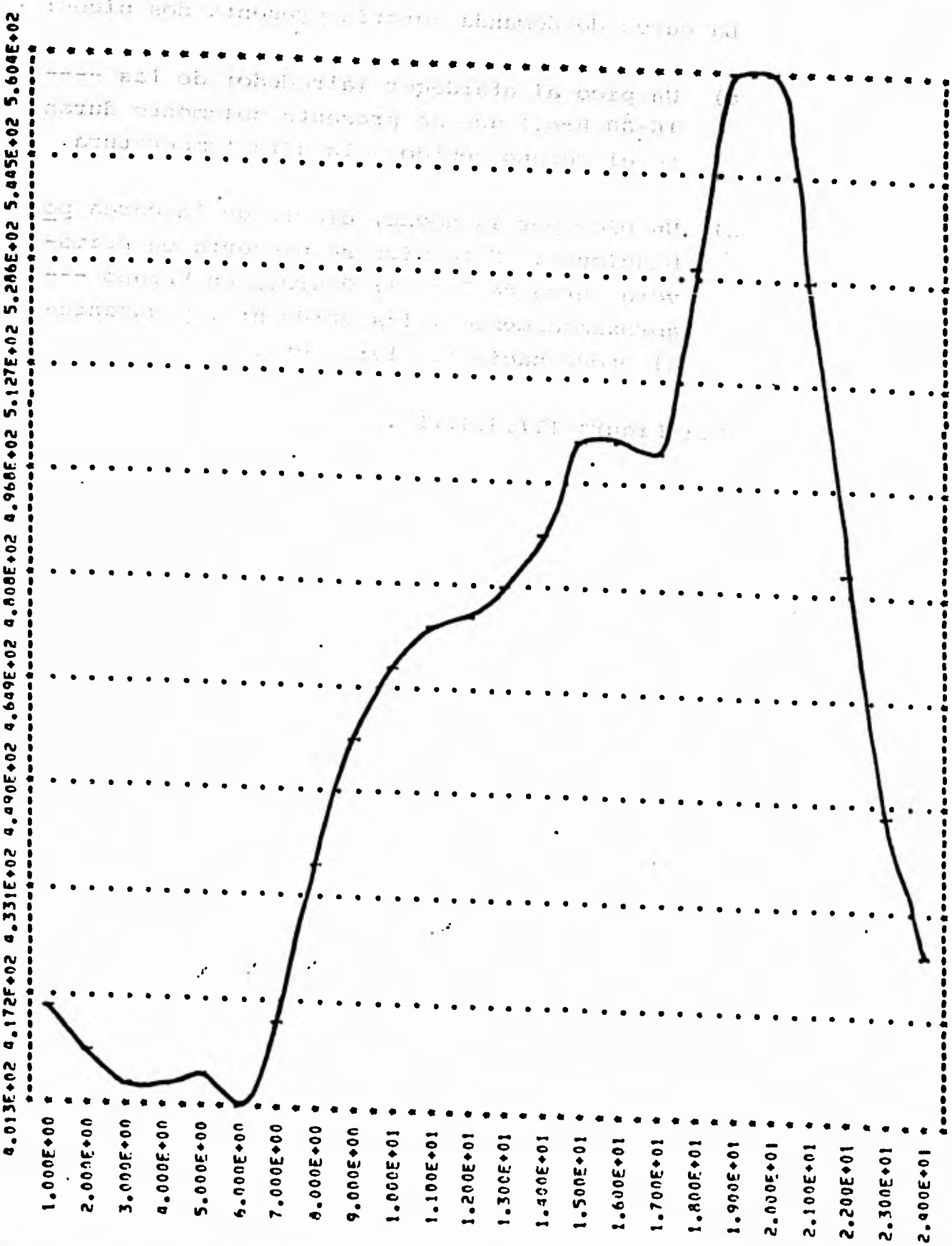
La curva de demanda horaria presenta dos picos:

a) Un pico al atardecer (alrededor de las 16:00 hrs.) que se presenta solamente durante el Verano debido a la alta temperatura.

b) Un pico por la noche, efecto de la carga poblacional. Este pico se presenta en Primavera cerca de las 21:00 hrs., en Verano aproximadamente a las 20:00 hrs. y durante el Otoño hacia las 19:00 hrs.

(Ver Figura III.3.4.(1).

FIGURA III.3.4.(1) CURVA DE DEMANDA



III.3.5 Recomendaciones.

Registros de Energía.

La información de energía horaria es inexacta - debido a diversos factores. Entre ellos los -- más importantes son :

III.3.5.1 Tipo de Medidores.

Los medidores actualmente utilizados no tie-- nen la precisión necesaria para registrar la- energía en forma horaria. Para tener una idea de la magnitud de los errores que es posible- cometer, se muestra el siguiente caso :

Línea :	Mochis II-Cd. Obregón III
Localización del medidor :	Mochis II
Fecha :	Marzo 1o./78
Constante del medidor :	160,000
Lecturas del Watthoríme-- tro :	A las 8 hrs.:1488.7 A las 9 hrs.:1488.8

Con estas lecturas, la energía fluyendo por - el enlace entre las 8 y las 9 hrs.sería de :

$$\begin{aligned} \text{energía} &= 160{,}000 \times (1488.8 - 1488.7) \\ &= 160{,}000 \times (0.1) \\ &= 16 \text{ MWH} \end{aligned} \qquad 1$$

Supóngase que se tuviera un dígito más de re- solución, y que las lecturas fueran :

a las 8 hrs.: 1488.70
a las 9 hrs.: 1488.89

En este caso, la energía sería :

$$\text{energía} = 160{,}000\ (1488.89 - 1488.70)$$
$$= 160{,}000\ (0.19) \qquad 2$$
$$= 30.4\ \text{MWH}$$

Comparando 1 con 2 se tiene un error de 14.4 MWH (47.36%).

Si bien el ejemplo analizado es un caso extremo, sirve para mostrar la poca utilidad de estos medidores para registrar la energía horaria (en promedio el error sería sólo de 23.68%).(ver anexo).

Estos errores, para fines de pronóstico de carga hora por hora, introducen desviaciones importantes en la forma de la curva de demanda y, en particular, en la hora del pico.

III.3.5.2 Hora de Lectura.

Las lecturas no se toman siempre a la hora exacta. Como ya se mencionó, en casos de disturbio, las lecturas se toman hasta que la falla ha sido corregida.

Para solucionar estas dificultades se recomienda :

III.3.5.3 Recopilación de Información.

Dado que se está instalando un sistema de telemetría en el Area Noroeste, sería conveniente obtener la información de energía horaria a partir de las mediciones instantáneas de potencia que recibirá el Centro de Control de Area. Es conveniente realizar un esfuerzo conjunto para analizar cuantitativamente las posibilidades de programación en la computadora de Hermosillo, Son.

Como la solución sugerida anteriormente sólo podrá implantarse a relativamente largo plazo, es aconsejable analizar, a la brevedad posible, algunas formas para mejorar la resolución de los medidores existentes.

Se sugiere que las lecturas de los watthorímetros se hagan a la hora exacta, ya que a partir de éstos se calcula la demanda. A continuación pueden leerse el resto de las variables : valores instantáneos de corriente, voltaje, potencia real y reactiva, temperatura del aceite en los interruptores, etc., que no es tan importante registrar a la hora exacta.

III.3.5.4 Reportes.

Actualmente se cuenta con el reporte de "Condiciones de Demanda Horaria". Para fines de pronóstico de carga, no es necesario contar con la energía de cada unidad o planta generadora. En cambio, si sería muy deseable contar con archivos de demanda horaria por subárea, una vez que estas se definan como resul

tado del análisis de la información recopilada. Por el momento es conveniente contar con la demanda de las subáreas existentes para -- proceder a la validación de esa regionaliza-- ción.

Así mismo sería muy adecuado que la verificación de la información se realizara cuando menos al cerrarse cada turno, obteniendo con -- ello datos confiables durante el día sin esperar al "Cierre de las 24 hrs.". Esto permitiría que los programas de pronóstico de carga- pudieran ser ejecutados con una mayor frecuencia, con información verificada.

III.3.5.5 Método de Pronóstico.

El área cuenta con una terminal de computadora conectada al sistema de tiempo compartido- de CFE. Por esta terminal el área transmite- y recibe información diaria del Centro Nacional de Control de Energía. Dicha terminal no se utiliza actualmente para el análisis de datos históricos de demanda ni para calcular el pronóstico diario.

El método actual de pronóstico (apreciación - visual de la tendencia) puede ser mejorado si se proporciona al área paquetes estadísticos- para el rápido análisis de la información --- existente. Entre otros, sería hoy de gran -- utilidad disponer de programas para definir - el agrupamiento de los diferentes días de la- semana (prueba de chicuadrada) y hacer análisis de regresión. Se recomienda se proporcionen al área los paquetes estadísticos de la -

PDP-11/70 de CFE y se instruya al personal en la utilización y la aplicación de las diferentes subrutinas.

III.3.6 Información Meteorológica.

Con objeto de recopilar información meteorológica relevante a los programas de pronóstico de carga horario se visitaron las siguientes Instituciones :

1) Radio Navegación Aérea Mexicana,S.A.
2) Servicio Meteorológico Nacional.
3) Instituto de Geofísica de la UNAM.
4) Departamento de Comercio de USA.
5) Dirección de Estudios del Territorio Nacional.
6) Departamento de Climatología de CFE.

A continuación se describe la información existente en cada Institución :

1) Radio Navegación Aérea Mexicana,S.A.

Se cuenta con información horaria digitalizada de los siguientes aeropuertos :

Acapulco
Guadalajara
México
Monterrey
Mazatlán
Puerto Vallarta
Veracruz
Mérida

Las variables meteorológicas reportadas cada hora son :

Visibilidad

Techo (altura de las nubes más bajas)
Precipitaciones líquida y helada
Obstrucciones a la visión
Presión
Temperatura
Punto de rocío
Velocidad y dirección del viento
Temperaturas de bulbo seco y bulbo húmedo
Humedad relativa
Nubes y fenómenos de obscurecimiento

Existen otros aeropuertos que reportan información horaria, pero que no ha sido digitalizada. Se trata de los aeropuertos de :

Zihuatanejo
Oaxaca
Tampico
Tapachula
Tuxtla Gutiérrez
Cd. del Carmen
Chetumal
Hermosillo
Nautla
Tulancingo
Tuxpan

La información de los aeropuertos mencionados data de 1970 a la fecha. El mapa de la figura III.3.6.(1) muestra la localización geográfica de las estaciones meteorológicas mencionadas. Se anexa, así mismo, una copia del formato en que se recopila la información.

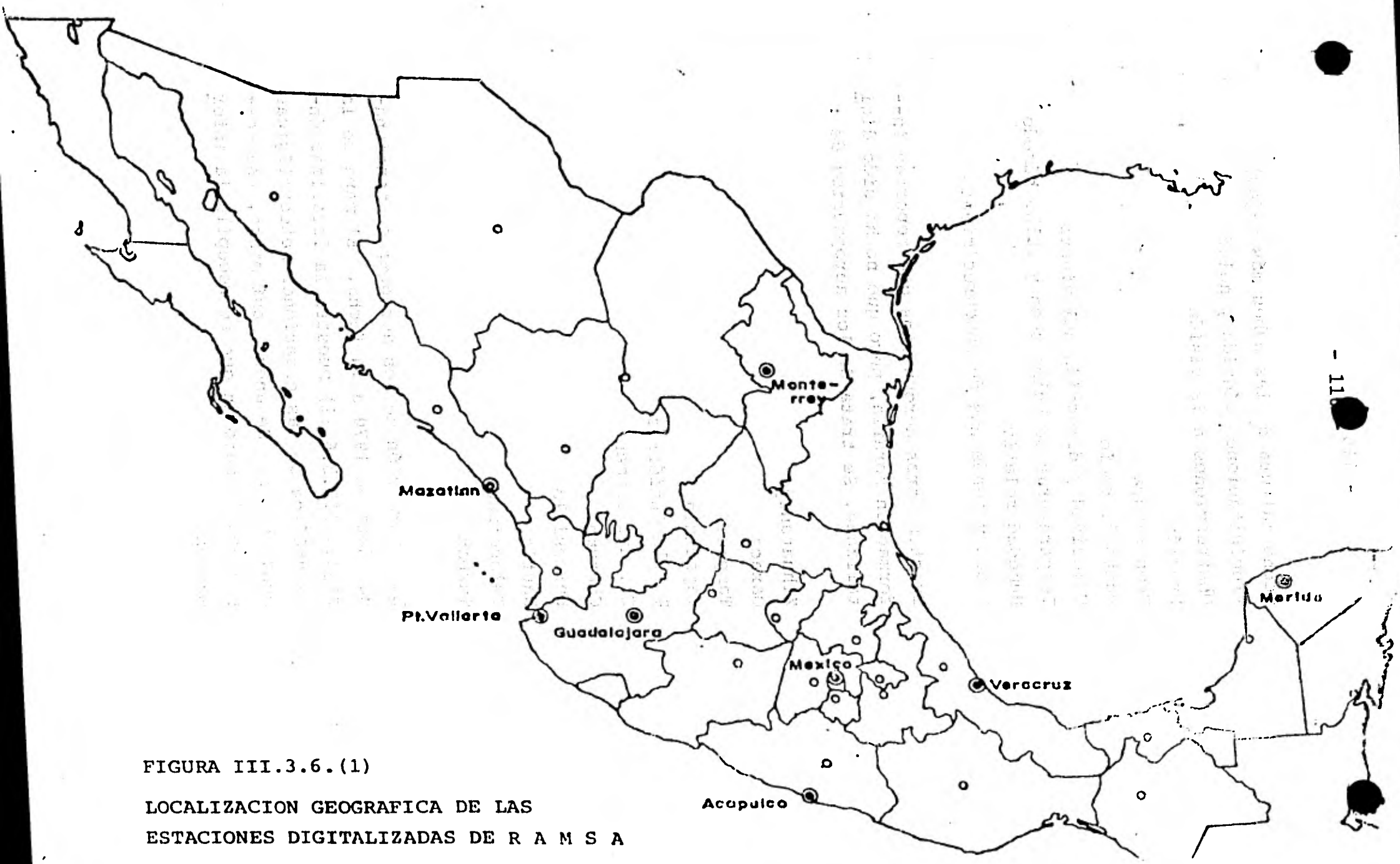


FIGURA III.3.6.(1)

LOCALIZACION GEOGRAFICA DE LAS
ESTACIONES DIGITALIZADAS DE R A M S A

codificación de las observaciones del tiempo para su vaciado en tarjetas

FECHA

NUMERO DE LA ESTACION.	FECHA			HORA	TECHO (CIENTOS PIES)	VISIBILIDAD (MILLAS)	TIEMPO			OBSTRUCCIONES A LA VISION	PRESION AL NMM (MBS)	TEMP. PUNTO DE ROCIO	VIENTO		PRESION ESTACION (MBS)	TEMP. BULBO SECO (°C)	TEMP. BULBO HUMEDO (°C)	HUMEDAD REL. %	NUBES Y FENOMENOS DE OBSCURECIMIENTO																			
																					CAPA MAS BAJA				SEGUNDA CAPA				TERCERA CAPA				CUARTA CAPA					
	AÑO	MES	DIA				TMTA-TORNADO	PRECIPITACION LIQUIDA	PRECIPITACION HELADA				DIR	INT. NUBOS					OPACIDAD TOTAL	CANTIDAD TOTAL	OPACIDAD	CANTIDAD	TIPO	ALTURA CIENTOS PIES	OPACIDAD	CANTIDAD	TIPO	ALTURA CIENTOS PIES	OPACIDAD	CANTIDAD	TIPO	ALTURA CIENTOS PIES	OPACIDAD	CANTIDAD	TIPO	ALTURA CIENTOS PIES	CANTIDAD PCPN

0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75

2) Servicio Meteorológico Nacional.

Posee información trihoraria y horaria de un gran número de variables como puede identificarse en el formato en que recopila la información. (ver figura III.3.6.(2)).

En el Area Noroeste se cuenta con las siguientes estaciones :

Empalme
Choix
Cd. Obregón
Mazatlán
Culiacán
Hermosillo

Esta información no está completa, ya que en algunas estaciones faltan meses completos de información y en otras no se cuenta con todas las lecturas trihorarias. La figura III.3.6.(3), muestra la localización geográfica de estas estaciones.

3) Instituto de Geofísica de la UNAM.

Cuenta con una o dos lecturas al día de temperatura (0 y 12 horas tiempo meridiano de Greenwich) desde 1975 a septiembre de 1977.- Esta información proviene de los libros de Registro del Servicio Meteorológico Nacional (SMN) y corresponde a las estaciones que el SMN tiene en el país.

REGISTRO DE OBSERVACIONES SIMULTANEAS (HORA DEL MERIDIANO 90°.)

ESTACION _____ FECHA _____

DATOS DE LAS OBSERVACIONES SINOPTICAS.

		03:00		06:00		09:00		12:00		15:00		18:00		21:00		24:00	
		OBS.	CIF.	OBS.	CIF.	OBS.	CIF.	OBS.	CIF.	OBS.	CIF.	OBS.	CIF.	OBS.	CIF.	OBS.	CIF.
	SIMBOLOS																
INDICATIVO DE LA ESTACION	iii																
CANTIDAD TOTAL DE NUBES	N																
DIRECCION DEL VIENTO	dd																
VELOCIDAD EN NUDOS	ff																
VISIBILIDAD HORIZONTAL	VV																
ESTADO DEL TIEMPO	ww																
CONDICION DEL TIEMPO EN PERIODO ANTERIOR	W																
PRESION AL NIVEL DEL MAR	PPP																
TEMPERATURA SECA (°F)	TT																
CANTIDAD DE NUBES BAJAS	Nh																
CLASE DE NUBES St. Sc. cu. ch.	CL																
ALTURA DE NUBES MAS BAJAS	h																
CLASE DE NUBES Ac. As. & Ns	CM																
CLASE DE NUBES Ci. Cs & Cc	CH																
PUNTO DE ROCIO	TdTd																
TOTAL DEL CAMBIO DE PRESION	P24P24																
	6																
PRESION DE LA ESTACION	PoPoPoPo																
	7																
LLUVIA TOTAL	RRR																
HORA DE LA PRECIPITACION	Rt																
	8																
TOTAL DE NUBES SIGNIFICATIVAS	Ns																
CLASE DE NUBES SIGNIFICATIVAS	C																
ALTURA DE NUBES SIGNIFICATIVAS	HsHs																
	9																
FENOMENOS ESPECIALES	ScSc																
FENOMENOS ESPECIALES EN DETALLE	spsp																
	9																
	19																
TOTAL DEL CAMBIO DE PRESION STANDARD	P_s P_s																
	1																
ESTADO DEL SUELO TROPICAL	C_t																
DIRECCION DEL MOV DE LAS NUBES BAJAS	D_L																
DIRECCION DEL MOV DE LAS NUBES MEDIAS	D_M																
DIRECCION DEL MOV DE LAS NUBES ALTAS	D_H																
	2																
TOTAL DE PRECIPITACION EN 24 HORAS	RRRRRRRRRR																
	3																
[illegible]	1a1a																
ALTURA DE LAS NUBES	HwHw																
	5																
TEMPERATURA MAXIMA	JxTx																
TEMPERATURA MINIMA	JnTn																
	5																
DIRECCION DEL VIENTO	[illegible]																
FUERZA DEL VIENTO	FxFm																
NOMBRE DEL OBSERVADOR																	

FIGURA III.3.6.(2)

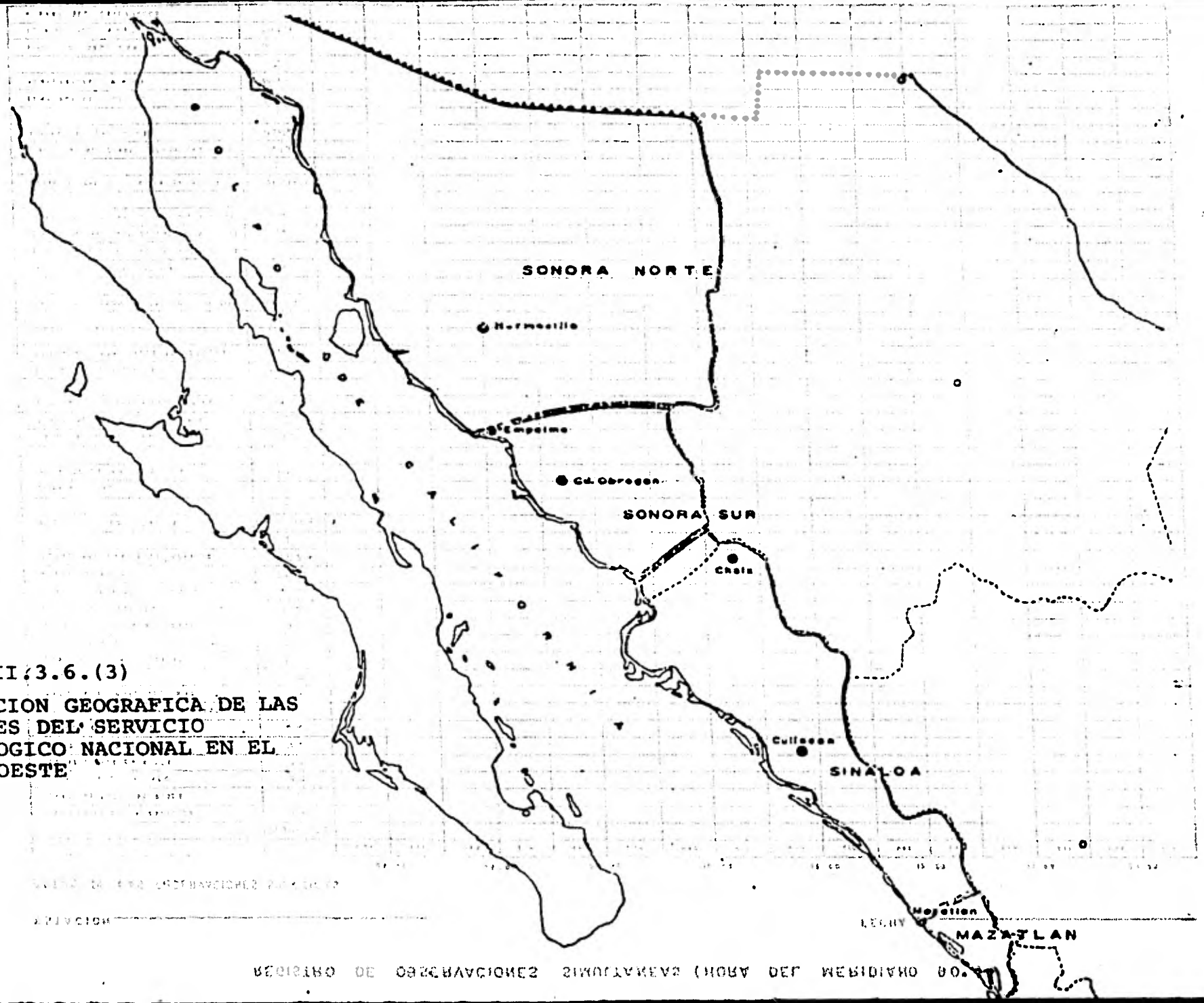


FIGURA III.3.6.(3)

LOCALIZACION GEOGRAFICA DE LAS ESTACIONES DEL SERVICIO METEOROLOGICO NACIONAL EN EL AREA NOROESTE

4) Departamento de Comercio de los EUA.

Posee en forma digitalizada la información del servicio meteorológico Nacional hasta el año de 1975.

La información corresponde a 28 estaciones del SMN. En el área Noroeste se cuenta con las estaciones de Mazatlán y Hermosillo, exclusivamente.

5) Dirección de Estudios del Territorio Nacional.

Se tiene información referente a los promedios mensuales de temperatura y precipitación pluvial de todas las Regiones del país. La información se encuentra almacenada tanto en las cartas de climas que vende Detenal como en cinta magnética. Esta segunda forma de almacenamiento de información es manejada por la Dirección General de Sistemas y Procesos Electrónicos de la Secretaría de Programación y Presupuesto.

6) Departamento de Climatología de CFE.

Posee la información recopilada por 256 "Estaciones Meteorológicas" con que cuenta CFE y complementada por las estaciones del servicio Meteorológico Nacional.

Parte de la información ha sido digitalizada y se encuentra en el sistema de información integrada para estudios de potencia (SIIPEP)

de la CFE. La información digitalizada incluye :

Temperatura máxima y mínima mensual
Lluvias mensuales
Número de días con lluvia en el mes
Evaporaciones total y media mensuales
Temperatura media mensual

El resto de la información se encuentra en documentos fuente. Incluye temperaturas máxima y mínima diaria y temperaturas a las 0 y 12 horas tiempo meridiano de Greenwich para cada día.

III.3.7 Análisis Estadístico de la Información Recopilada.

Generación del Banco de Datos Eléctrico.

Con la información horaria recabada (mayo/77 a abril/78) de las 30 unidades generadoras y los 6 enlaces existentes en el área Noroeste de la CFE se procedió a la formación de un banco de datos Eléctrico, cuya estructura se muestra en la figura III.3.7.(1).

Generación del Banco de Datos Meteorológico.

Se captó la información trihoraria (mayo/77 a abril/78) de 20 variables meteorológicas del servicio Meteorológico Nacional (SMN). Las variables seleccionadas fueron :

1) Fracción de la bóveda Celeste cubierta por nubes (CLOUDS).
2) Dirección del viento (WINDDIRC).
3) Velocidad del viento (WINDVELO).
4) Visibilidad Horizontal en superficie (HORVISIB).
5) Estado del tiempo a la hora de la observación (GENLSTAT).
6) Temperatura del aire a la hora de la observación (TEMPTURE).
7) Fracción de la bóveda celeste cubierta por las nubes más bajas (LOWCLOUD).
8) Clase de las nubes bajas (CLOUDTYP).
9) Altura sobre el suelo de la base de las nubes más bajas (HEIGHTLW).
10) Temperatura del punto del rocío (DEWPOINT).

PROMEDIO DE DEMANDA HORARIA — LUNES

GENERADORAS

HORA	NVL 1	NVL 2	NVL 3	PGU 1	PGU 2	PGU 3	PGU 4	PGD 1	PGD 2	COM 1	COM 2	OVI 1	OVI 2	MRI 1	EFU 1	EFU 2	EFU 3	TPO 1	CUU 1	HYA 1	HYA 2	SLA 1	SLA 2	MZU 1	MZU 2	MZU 3	PVU 1	MZD 1	MZD 2	[illegible]
1	9	0	4	5	5	19	25	52	54	0	0	3	3	4	6	8	9	21	0	9	14	3	2	6	5	0	0	48	92	[illegible]
2	9	0	3	5	4	19	25	51	53	0	0	3	4	4	6	8	8	20	0	9	15	3	2	6	5	0	0	47	89	[illegible]
3	9	0	3	5	4	19	25	50	52	0	0	3	3	4	6	8	9	20	0	9	15	3	2	6	5	0	0	46	88	[illegible]
4	8	0	3	5	4	19	24	50	52	0	0	3	4	4	6	8	8	20	0	9	15	3	2	6	5	0	0	46	87	[illegible]
5	8	0	2	5	4	19	24	49	52	0	0	3	4	4	6	8	8	20	0	9	15	3	2	6	5	0	0	46	87	[illegible]
6	9	0	2	5	4	19	24	49	53	0	0	3	4	4	6	8	9	20	0	9	15	3	2	6	5	0	0	46	87	[illegible]
7	9	0	3	5	5	19	24	49	53	0	0	3	4	4	6	8	9	21	0	9	15	3	2	6	5	0	0	44	94	[illegible]
8	10	1	6	5	5	19	24	52	55	0	0	3	4	4	6	8	9	22	0	8	15	3	2	6	5	0	0	46	99	[illegible]
9	12	4	12	5	5	19	25	55	59	0	0	3	3	4	6	9	11	24	0	8	15	3	2	6	5	0	0	49	103	[illegible]
10	15	7	18	5	5	19	26	57	60	0	0	3	4	4	7	9	10	24	0	8	15	3	2	7	6	0	0	50	104	[illegible]
11	16	9	21	6	5	19	26	59	62	0	0	3	4	4	7	9	11	25	0	9	15	3	2	7	6	0	0	52	104	[illegible]
12	17	11	23	6	5	20	26	59	63	0	0	3	4	4	7	9	11	26	0	9	15	3	2	7	6	0	0	51	105	[illegible]
13	17	11	24	6	5	20	27	59	63	0	0	3	3	4	7	10	11	26	0	9	15	3	3	7	6	0	0	52	107	[illegible]
14	17	10	22	6	5	20	27	59	63	0	0	3	3	4	8	10	11	26	0	9	15	3	2	7	6	0	0	52	108	[illegible]
15	18	10	23	6	5	20	26	58	62	0	0	3	4	4	8	10	11	26	0	9	15	3	2	7	6	0	0	53	109	[illegible]
16	19	12	24	6	5	20	26	60	64	1	0	3	4	4	8	10	12	26	0	8	15	3	3	7	6	0	0	53	111	[illegible]
17	21	15	26	6	5	19	27	60	65	1	0	3	4	4	9	11	12	27	0	8	15	3	2	7	6	0	0	54	114	2
18	22	15	25	6	5	20	27	59	64	1	0	4	4	4	9	11	12	27	1	9	16	3	2	7	6	0	0	53	114	2
19	25	24	29	6	6	20	27	59	65	2	1	3	4	4	11	13	14	28	1	9	15	3	3	8	6	2	0	53	114	4
20	27	30	33	6	6	21	27	61	66	3	1	3	4	4	12	13	15	29	2	9	15	3	2	8	7	4	0	54	115	8
21	26	29	33	6	6	21	27	62	66	2	1	3	4	4	11	13	15	29	2	9	15	3	3	8	7	4	0	54	115	7
22	23	20	27	6	6	20	26	61	66	1	0	3	4	4	11	12	15	28	0	8	15	3	3	8	7	1	0	54	115	[illegible]
23	15	13	18	5	5	20	25	59	64	0	0	3	4	4	9	11	12	25	0	8	15	3	3	7	6	0	0	52	112	[illegible]
24	17	6	11	5	5	20	25	57	63	0	0	3	4	4	8	10	11	23	0	8	15	3	3	6	5	0	0	50	106	[illegible]

HORA	ENERGIA PGD-PGU	ENERGIA PGD-HLT	ENERGIA EFU-NVJ	ENERGIA UMD-COT	ENERGIA M7D-CUI	ENERGIA M7D-DGD	INTERCAMB DGD-M7D	CONSUMOS	CONSUMOS	CONSUMOS	CONSUMOS	CONSUMOS TOTAL
1	11	60	10	24	38	80	76	142	82	77	37	340
2	11	60	10	25	38	80	73	139	80	76	36	334
3	10	58	9	25	38	82	74	137	80	77	34	329
4	10	58	8	23	33	81	77	136	77	75	35	325
5	11	59	8	24	36	77	74	137	77	77	35	327
6	11	60	8	23	35	76	74	137	75	79	35	328
7	12	58	8	24	33	80	78	137	77	76	38	330
8	16	69	8	29	39	90	82	158	72	79	35	347
9	16	62	8	30	40	94	87	166	06	84	38	374
10	16	61	9	39	40	93	87	177	101	74	40	392
11	17	60	10	28	42	94	87	184	95	87	40	408
12	15	57	12	28	38	95	90	185	103	83	43	414
13	15	60	12	21	39	99	93	188	94	92	41	416
14	17	62	12	21	41	97	90	190	90	95	42	419
15	17	60	11	24	44	88	86	169	93	95	44	422
16	19	63	12	26	51	90	65	199	95	101	41	438
17	15	57	13	31	45	98	93	195	113	92	44	445
18	14	55	13	21	39	106	101	194	105	99	42	[illegible]
19	8	49	13	20	30	106	102	201	117	97	51	[illegible]
20	2	42	14	14	31	107	102	206	130	107	55	498
21	4	43	16	10	34	107	102	206	125	111	53	496
22	9	51	16	5	39	105	97	197	103	115	49	466
23	13	54	14	11	41	91	88	179	95	104	49	428
24	17	62	12	18	39	95	88	174	83	95	42	394

11) Cambio de la presión en el nivel de la estación durante las 24 horas anteriores (PRESS CHG).

12) Presión de la estación (PRESSTAT).

13) Precipitación durante el periodo anterior - (TOTALPRC).

14) Hora del principio o fin de la precipitación (HOURPREC).

15) Fenómenos especiales (SPECEVNT).

16) Cambio de la presión en el nivel de la estación durante las 3 últimas horas (PRESSCH3).

17) Temperatura MAXIMA (MAXTEMPR).

18) Temperatura MINIMA (MINTEMPR).

19) Dirección del viento MAXIMO (PRESDIRC).

20) Fuerza del viento máximo (WINDFORC).

Como se mencionó anteriormente, la información del SMN es incompleta; por ello fue necesario diseñar un banco de Datos Meteorológicos que incluyera los códigos de identificación de la información faltante. El banco de datos comprende las 20 variables meteorológicas mencionadas, para cada una de las siguientes estaciones :

i) Hermosillo
ii) Empalme
iii) Choix
iv) Cd. Obregón
v) Culiacán
vi) Mazatlán

Análisis de los Datos.

Para facilitar la depuración y el análisis de la

información eléctrica y meteorológica se procedió a la Instalación del lenguaje estadístico - BMDP-77, en la computadora UNIVAC 1110 de la Comisión Federal de Electricidad. Este lenguaje permite combinar con gran flexibilidad una serie de análisis. (ver formato de la figura III. 3.7.(2).

Inicialmente se realizaron algunas pruebas que permiten la depuración de los bancos de datos - creados. Especial consideración se dió a los - siguientes problemas :

1) Los datos contienen errores de perforación que será necesario corregir.
2) Hay valores faltantes que deben identificarse antes de efectuar los análisis estadísticos. (Figura III.3.7.(2).A).
3) Los datos climatológicos tienen una codificación específica que obliga a realizar --- transformaciones con la información.
4) No es posible especificar el análisis más - apropiado hasta examinar los datos en diversas formas.
5) Los datos deben ser estratificados y agrupados en forma lógica (Figura III.3.7.(2).B).
6) Los datos contienen valores extremos que será necesario eliminar de los análisis estadísticos (Figura III.3.7.(2).C).
7) Es necesario identificar en forma global la calidad y validez de los datos.

Para dar respuesta a algunos de los problemas - anteriores se procedió a la realización de las siguientes pruebas estadísticas :

```
OBLEM      'PRUEBA CON METEOROLOGIA EN ZONAS 1 2 3 4 CON COVARIABLES'.
PUT        UNIT IS 15.
           VARIABLES ARE 26.
MAT IS '(2F7.1,4F3.0,2F3.0,4F4.0,3F3.0,2F4.0,2F5.0,F3.0,4F4.0,F3.0,F4.0)'.
AR       NAMES ARE CONSH,GENH,HORA,DIA,MES,ZONA,
         CLOUDS,WINDDIRC,WINDVELO,HORVISIB,
         GENLSTAT,TEMPTURE,LOWCLOUD,CLOUDTYP,HEIGHTLW,
         DEWPOINT,PRESSCHG,PRESSTAT,TOTALPRC,HOURPREC,
         SPECEVNT,PRESSCH3,MAXTEMPR,MINTEMPR,PRESDIRC,
         WINDFORC.
(A)      MISSING = (7)99,(8)99,(9)999,(10)999,(11)999,
                   (12)999,(13)99,(14)99,(15)99,(16)999,
                   (17)999,(18)9999,(19)9999,(20)99,(21)999,
                   (22)999,(23)999,(24)999,(25)99,(26)999.
(C)      MAXIMUM= (1)3,(2)1500,(7)9,(8)36,(9)99,(10)99,(11)99,
                  (12)50,(13)9,(14)9,(15)9,(16)99,(17)99,
                  (18)9999,(19)999,(20)9,(21)99,(22)99,(23)60,
                  (24)36,(25)36,(26)70.
         MINIMUM=11*0,(12)-50,(13)0,(14)0,(15)0,(16)-50,(17)-50,(18)0,
                 (19)0,(20)0,(21)0,(22)-50,(23)0,(24)0,(25)0,(26)0.
         USE = 1 TO 18.
ESIGN    GROUPING ARE HORA,DIA,ZONA.                    (B)
         INCLUDE = 1,2,3,12,23.
         DEPENDENT ARE CONSH.
         COVARIATES ARE
         CLOUDS,WINDDIRC,WINDVELO,HORVISIB,
         GENLSTAT,TEMPTURE,LOWCLOUD,CLOUDTYP,HEIGHTLW,
         DEWPOINT,PRESSCHG,PRESSTAT.
ROUP        CODES(3) ARE 2,5,8,11,14,17,20,23.
            NAMES(3) ARE HORA02,HORA05,HORA08,HORA11,HORA14,HORA17,
                         HORA20,HORA23.
            CODES(4) ARE 1,2,6,7.
          NAMES(4) ARE LUNES,MMJV,SABADO,DOMINGO.
            CODES(5) ARE 1 TO 12.
            NAMES(5) ARE ENE,FEB,MAR,ABR,MAY,JUN,
                         JUL,AGO,SEP,OCT,NOV,DEC.
            CODES(6) ARE 1 TO 4.
            NAMES(6) ARE SONNORTE,SONSUR,SINALOA,MAZATLAN.
ND
ROBLEM       'PRUEBA CON METEOROLOGIA EN ZONAS 1 2 3 4 SIN COVARIABLES'.
NPUT         UNIT IS 15.
             VARIABLES ARE 6.
              FORMAT IS '(2F7.1,4F3.0)'.
AR       NAMES ARE CONSH,GENH,HORA,DIA,MES,ZONA.
         MAXIMUM=(1)3,(2)1500.
ESIGN    GROUPING ARE HORA,DIA,ZONA.
         INCLUDE = 1,2,3,12,23.
           DEPENDENT ARE CONSH.
ROUP       CODES(3) ARE 2,5,8,11,14,17,20,23.
             NAMES(3) ARE HORA02,HORA05,HORA08,HORA11,HORA14,HORA17,
                          HORA20,HORA23.
             CODES(4) ARE 1,2,6,7.
             NAMES(4) ARE LUNES,MMJV,SABADO,DOMINGO.
             CODES(5) ARE 1 TO 12.
             NAMES(5) ARE ENE,FEB,MAR,ABR,MAY,JUN,
                          JUL,AGO,SEP,OCT,NOV,DEC.
             CODES(6) ARE 1 TO 4.
             NAMES(6) ARE SONNORTE,SONSUR,SINALOA,MAZATLAN.
END
```

FIGURA III.3.7.(2)

1) Descripción detallada de datos incluyendo frecuencias.

2) Análisis de factores.

3) Comparación de grupos con pruebas T.

4) Gráficas en 2 variables. Correlación.

5) Análisis de varianza y covarianza.

A continuación se describe el propósito de cada una de las pruebas anteriores :

1) Descripción detallada de Datos incluyendo Frecuencias.

Los objetivos de esta prueba son : contar y listar los distintos valores de las variables eliminando información incompleta o fuera de los límites permisibles; estimar la media y la varianza; proporcionar la forma de la distribución de las variables (densidad de probabilidad) y su porcentaje acumulado (función de distribución).

2) Análisis de Factores.

Este algoritmo fue utilizado para identificar tanto el posible agrupamiento existente entre los parámetros meteorológicos como la posible analogía en la operación de las diversas unidades generadoras. Puede también ser empleado para identificar componentes principales dentro del grupo de parámetros meteorológicos.

3) Comparación de Grupos con pruebas T y Pruebas Chicuadrado.

Las pruebas T fueron utilizadas para comparar los consumos y las generaciones hora---

rias por zona para cada una de las diferentes horas del día.

Las comparaciones por día de la semana para las diferentes zonas de consumo (generación) fueron realizadas utilizando la prueba Chi-cuadrado. Esta prueba fue elaborada en dos diferentes formas con objeto de :

a) Determinar los días de la semana que -- presentan patrones de consumo (genera-- ción) similares (suponiendo iguales las energías diarias de los días compara--- dos) y

b) Definir qué tanto se parecen en forma - las curvas de carga horaria en los días de la semana (suponiendo diferentes las energías diarias de los días por comparar y proporcionarles las curvas de carga).

4) Gráficas en 2 variables. Correlación.

Este algoritmo es empleado para definir los modelos matemáticos del consumo (generación) en función de cada una de las variables meteorológicas que vayan a ser requeridas en-un esquema de pronóstico de carga. Los resultados presentan, por ejemplo, el pico -- diario de una zona en función de MAXTEMPR - (temperatura máxima) de esa región en forma gráfica. Así mismo, es posible computar e - imprimir la regresión lineal relacionando - una variable con la otra.

5) Análisis de Varianza y Covarianza.

Este programa fue utilizado para identificar las variables climáticas que impactan el consumo (generación) de energía eléctrica en cada una de las cuatro zonas del área Noroeste.

Los resultados permiten cuantificar tanto el grado de dependencia entre el consumo (generación) y las diferentes variables meteorológicas, como el efecto que los factores de -- agrupamiento (hora, día, mes, zona) tienen - sobre el consumo (generación).

Un ejemplo de los modelos analizados se presenta a continuación :

$$Y_{ijkl} = U+D_i+M_j+Z_k+DZ_{ik}+MZ_{jk}+\beta_i X^{(1)}_{ijk}+\beta_2 X^{(2)}_{ijk}+\varepsilon_{ijkl} \qquad (1)$$

en donde :

$Y_{i,j,k,l}$ - representa la estimación l del consumo (generación) - diario del día de la semana I, del mes J, en la zona K.

U - es la media general del -- consumo diario.

D_i - representa el efecto del - día I.

M_j - representa el efecto del - mes J.

Z_k - representa el efecto de la zona K.

DZ_{ik} - representa el efecto de la interacción del día I en - la zona K.

MZ_{jk} - representa el efecto de la interacción del mes J en la zona K.

β_i - son los coeficientes de participación de los distintos parámetros $X_{ijk}^{(i)}$.

$X_{ijk}^{(i)}$ - representan los distintos parámetros meteorológicos correspondientes al día I, del mes J en la zona K.

ε_{ijkl} - es una variable aleatoria denominada residual, con densidad de probabilidad Normal de media cero y varianza σ^2, correspondiente a la estimación L del día I, mes J y zona K.

III.3.8 Conclusiones y Recomendaciones.

A continuación se reportan los resultados, producto del análisis estadístico, obtenidos del análisis inicial emprendido con los bancos de datos eléctrico y meteorológico correspondientes al área Noroeste de la Comisión Federal de Electricidad.

La disponibilidad de computadoras de tiempo compartido (UNIVAC 1110) y de un lenguaje estadístico (BMPD-77) para el procesamiento de la información recopilada señalan muy buenas posibilidades de éxito para nuestro proyecto de Pronóstico de Carga.

Para formar un criterio sobre la validez de la información captada, se plantearon hipótesis "obvias" que describen efectos conocidos (aunque no necesariamente han sido cuantificados).

Por ejemplo, se investigó si la temperatura máxima del día está relacionada con el consumo diario de energía en determinada zona; si los consumos a diferentes horas del día tienen patrones de comportamiento similares, etc.

Esto se hizo con el fin de determinar el grado de confiabilidad de nuestra información y ver si se justificará un análisis de mayor detalle.

III.3.8.1 Conclusiones y Recomendaciones Generales.

Datos de carga.

Los resultados de análisis estadísticos preliminares muestran que los datos existentes son

razonablemente buenos y confiables solamente para modelar la demanda de energía eléctrica a niveles diario y trihorario.

Datos de Climatología.

Las seis estaciones meteorológicas del Area Noroeste ofrecen información trihoraria para cada una de las cuatro zonas. La información es confiable y algunas de las variables son necesarias para modelar el consumo (temperatura, nubosidad, punto de rocío). Por falta de recursos computacionales, en el tiempo disponible no ha sido posible llegar a conclusiones precisas respecto a las variables que en los análisis preliminares no han mostrado niveles de significancia importantes en la modelación del consumo de energía eléctrica.

La falta de información horaria no permite cuantificar la importancia de contar con ella en la modelación. Creemos que de existir la información climática horaria, los modelos mostrarían una mejora substancial. (Esta suposición no es posible demostrarla en el Area Noroeste. Podría analizarse, por ejemplo, en el Area Central donde se cuenta con datos meteorológicos horarios).

III.3.8.2 Conclusiones Particulares.

A. Análisis de Factores (INFORMACION METEOROLOGICA).

El análisis de factores realizado con las 20 variables meteorológicas de las 6 esta

ciones del SMN que se encuentran en el -- Area Noroeste indica que no es posible agrupar los parámetros meteorológicos de las estaciones en factores meteorológicos, (combinaciones lineales de parámetros climáticos). Por tanto, será necesario incluir en cualquier análisis posterior los conjuntos de variables completos.

NOTA: Dado que las estaciones del SMN no reportan temperatura de bulbo húmedo, no es posible calcular la humedad relativa. Sería recomendable analizar la posibilidad de medir dicha temperatura en las estaciones del SMN.

B. Análisis de grupos con pruebas CHI-CUADRADO y pruebas T. (INFORMACION ELECTRICA).

La primera de estas pruebas permite integrar los diferentes días de la semana en grupos con características similares de consumo de energía eléctrica en cada una de las zonas del Area Noroeste.

En la siguiente tabla se resumen los resultados obtenidos de la prueba Chi-cuadrado :

ZONA	POSIBLE AGRUPAMIENTO DE DIAS DE LA SEMANA EN BASE A UN CONSUMO DE ENERGIA SIMILAR
Sonora Norte	Martes, Miércoles, Jueves, Viernes, Sábado
Sonora Sur	Martes, Miércoles, Jueves, Viernes, Sábado
Sinaloa	Martes, Miércoles, Jueves, Viernes
Mazatlán	Lunes, Martes, Miércoles, Jueves, Viernes, Sábado
TOTAL DEL AREA	Martes, Miércoles, Jueves, Viernes

Z O N A	POSIBLE AGRUPAMIENTO DE DIAS DE LA SEMANA EN BASE A UNA PROPORCIONALIDAD EN -- LAS CURVAS DE CONSUMO HORARIO
Sonora Norte	Lunes, Martes, Miércoles, Jueves, Viernes
Sonora Sur	Lunes, Martes, Miércoles, Jueves, Viernes Sábado
Sinaloa	Domingo, Lunes, Martes, Miércoles, Jueves Viernes, Sábado
Mazatlán	Domingo, Lunes, Martes, Miércoles, Jueves Viernes, Sábado
TOTAL DEL AREA	Martes, Miércoles, Jueves, Viernes, Sábado

El análisis de grupos con prueba T (Student) permite agrupar las diferentes horas del día que presentan patrones de comportamiento similares.

Los resultados obtenidos no permiten agrupar con la confiabilidad necesaria las diferentes horas del día en ninguna de las 4 zonas del área Noroeste.

C. Análisis de Varianza y Covarianza.

Se propusieron diferentes modelos de consumo de energía eléctrica para las varias zonas del Noroeste. Entre ellos podemos mencionar los siguientes :

1. Modelo del Consumo Diario por zona con sus propias variables Meteorológicas.
2. Modelo del Consumo Diario por zonas sin variables Meteorológicas.
3. Modelos por zona del Consumo Diario con variables meteorológicas de una de las zonas.
4. Modelo del Consumo Diario de cada zona con variables meteorológicas relevantes de todas las zonas.

Los resultados obtenidos a la fecha permiten concluir, en forma general, que la inclusión de parámetros meteorológicos en un esquema de pronóstico de carga para el Area Noroeste de CFE mejorará el modelo de predicción. Así mismo, el análisis de los resultados del mo-

delo 1., permite identificar los parámetros meteorológicos relevantes para la predicción de carga eléctrica diaria en el Area Noroeste. Estos parámetros serían los siguientes :

- temperatura de bulbo seco
- temperatura del punto de rocío
- visibilidad horizontal
- presión atmosférica
- temperatura máxima
- temperatura mínima

De la misma forma, el análisis de los resultados del conjunto de modelos 3., permite afirmar que las variables :

- temperatura de bulbo seco
- temperatura máxima
- temperatura mínima

de una de las zonas afectan el consumo diario de las demás zonas.

Esto concuerda con lo observado por el personal de operación del Area.

Deberá tomarse muy en cuenta que los resultados obtenidos inicialmente sugieren análisis estadísticos más elaborados, con los cuales pueden complementarse las inferencias a las que se llega incialmente. Por ejemplo, el conjunto de modelos 4., permite definir cuáles de aquellas variables que afectan el consumo diario en las diferentes

zonas (temperaturas de bulbo seco, temperaturas máximas y temperaturas mínimas), son las más importantes en cada una de las regiones del Area Noroeste.

Los resultados de estos modelos de consumo diario pueden observarse en la siguiente tabla :

VARIABLES QUE IMPACTAN EL CONSUMO DIARIO

SONORA NORTE

Temperatura de Sonora Norte
Temperatura máxima de Sonora Norte
Temperatura de Mazatlán
Temperatura máxima de Sinaloa

SONORA SUR

Visibilidad horizontal de Sonora Sur
Máxima temperatura de Sonora Sur
Mínima temperatura de Sonora Sur
Temperatura de Maztlán

SINALOA

Visibilidad horizontal de Sinaloa
Temperatura de Sinaloa
Temperatura de Sonora Norte
Máxima temperatura de Sonora Norte
Temperatura de Mazatlán

MAZATLAN

Punto de rocío de Mazatlán
Presión atmosférica de Mazatlán
Máxima temperatura de Sonora Sur
Máxima temperatura de Sinaloa

III.3.9. Algoritmos Propuestos para el PECHO.

Se ponen a consideración los siguientes modelos de carga eléctrica :

1. Componentes con agrupamiento de días (COM-G)
2. Armónicas y suavizamiento exponencial (GES-D)
3. Compensación de errores (COMP)
4. Armónicas y filtros de Kalman modelo diario (KBF-D)

A continuación se describe brevemente cada uno de los modelos mencionados.

III.3.9.1 Componentes con agrupamiento de días.

El esquema pronóstico propuesto por Gupta y Yamada {38,39,42} modela la carga eléctrica en la siguiente forma :

$$Z(I,J) = T(I,J) + S(I,J) \qquad (III.3.9.1\text{-}1)$$

Donde :

Z(I,J) - carga horaria medida en el día "I" a la hora "J". (MWH/H)

T(I,J) - componente de tendencia para el día "I" a la hora "J".

S(I,J) - componente cíclica para el día "I" a la hora "J".

III.3.9.1.1 Componente de Tendencia.

Esta componente explica el efecto estacional y crecimiento anual de la carga horaria, y - se modela de la siguiente forma :

III.3.9.1.1.1 Inicialización.

Se obtiene como un promedio de "M" días.

$$T(I,J)=\frac{1}{M}\sum_{I=1}^{M} Z(I,J) \qquad (III.3.9.1-2)$$

Donde :

M - longitud de la ventana deslizante

III.3.9.1.1.2 Actualización.

$$T(I,J)=T(I-1,J)+\frac{1}{M}(Z(I,J)-Z(I-M,J)) \qquad (III.3.9.1-3)$$

Donde :

T(I-1,J) - componente de tendencia para el día "I-1", a la hora "J".

Z(I-M,J) - carga horaria medida para el -- día "I-M", a la hora "J"(MWH/H).

Z(I,J) - carga horaria medida para el -- día "I", a la hora "J"(MWH/H).

M - longitud de la ventana deslizante.

III.3.9.1.1.3 Extrapolación.

Para fines de pronóstico, se considera que -

la componente de tendencia no cambia de un-día al otro.

$$\hat{T}(I+1,J) = T(I,J) \qquad (III.3.9.1\text{-}4)$$

Donde :

$\hat{T}(I+1,J)$ - componente de tendencia extrapolada para el día "I+1", a la hora "J".

III.3.9.1.2 Componente Cíclica.

Esta componente refleja el comportamiento de los diferentes días de la semana, y se modela de la siguiente forma :

III.3.9.1.2.1 Inicialización.

$$S(I,J) = Z(I,J) - T(I,J) \qquad (III.3.9.1\text{-}5)$$

Donde :

$S(I,J)$ - componente cíclica para el día "I" a la hora "J".

$Z(I,J)$ - carga horaria medida para el día "I" a la hora "J" (MWH/H).

$T(I,J)$ - componente de tendencia para el día "I" a la hora "J".

III.3.9.1.2.2 Actualización.

$$S(I,J)=(1-\alpha)(S(I-7,J))+\alpha(Z(I,J)-T(I,J)) \qquad (III.3.9.1\text{-}6)$$

Donde :

α - constante de suvizamiento

III.3.9.1.2.3 Extrapolación.

$$\hat{S}(I+1,J) = S(I,J) \qquad \text{(III.3.9.1-7)}$$

Donde :

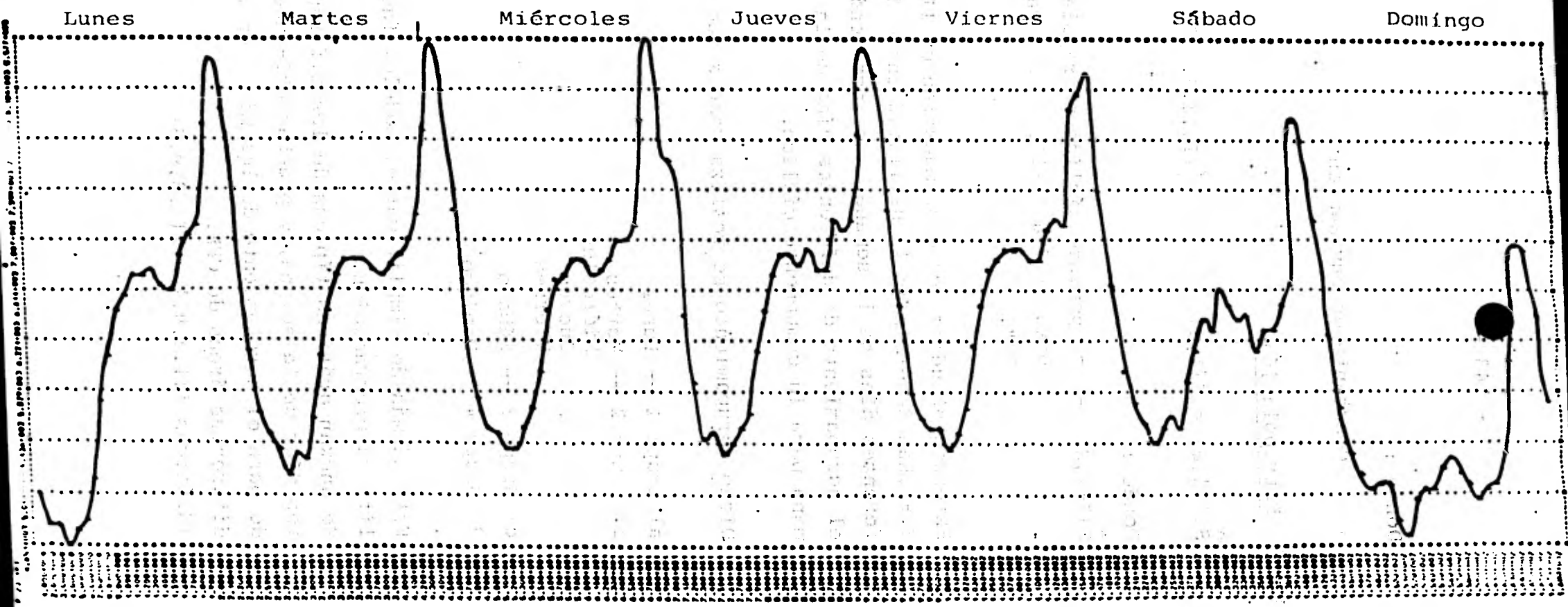
$\hat{S}(I+1,J)$ - componente cíclica extrapolada para el día "I+1", a la hora - "J".

Extensos estudios (como se muestran en III. 3.8),han determinado la similitud entre diferentes días de la semana. Se recomienda el agrupamiento de diferentes días de la semana para la componente cíclica.

Dicho agrupamiento se realiza como sigue :

a) Grupo 1 -- Lunes
b) Grupo 2 -- Martes
Miércoles
Jueves
Viernes
c) Grupo 3 -- Sábado
d) Grupo 4 -- Domingo

Esto además de simplificar el modelo, facilita su programación.
Se anexan gráficas típicas de los diferentes días de la semana, y pruebas Chi-cuadrado para el agrupamiento de los mismo en las diferentes áreas de CFE.
(Gráficas III.3.9.1 a III.3.9.5)



GRAFICA III.3.9.1

Gráfica Típica de una semana.

Area Central.

PRUEBA CHI-CUADRADA PARA LOS DIAS PARA EL SISTEMA NACIONAL

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
LUNES	.00000	347.10630	474.30910	621.31010	526.00737	804.92608	6481.88513
MARTES	381.16270	.00000	14.19499	54.99461	23.20380	576.02702	7595.32452
MIERCOLES	525.46452	14.35410	.00000	13.89667	3.80238	703.81213	8220.33203
JUEVES	693.99516	56.18374	14.04793	.00000	11.30721	857.05033	8857.38733
VIERNES	587.30600	23.61938	3.82449	11.19336	.00000	678.08059	8192.06519
SABADO	838.13512	532.76044	644.69669	779.58280	622.48194	.00000	4001.39645
DOMINGO	5042.41864	5791.47186	6222.52045	6662.78827	6214.37946	3272.93417	.00000

END PROGRAM EXECUTION

PRUEBA CHI-CUADRADA PARA LOS DIAS PARA EL SISTEMA CENTRAL

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
LUNES	.00000	347.10052	422.60440	455.48231	436.66393	1000.71637	5741.30206
MARTES	412.86682	.00000	5.07991	9.35501	9.97335	986.08807	7060.16449
MIERCOLES	510.10838	5.15750	.00000	1.32123	3.9[illegible]	1050.31172	7330.21924
JUEVES	553.55015	9.56879	1.33160	.00000	4.13793	1071.08223	7416.72443
VIERNES	532.81500	10.16422	3.94935	4.09327	.00000	959.21745	7100.71893
SABADO	1005.02457	835.32775	884.91138	901.51894	814.90567	.00000	2522.69669
DOMINGO	3860.86633	4574.32487	4729.21991	4782.20447	4612.32355	1905.79152	.00000

END PROGRAM EXECUTION

GRAFICA III.3.9.2

RESULTADOS DE LA PRUEBA CHI-CUADRADA. (1)

PRUEBA CHI-CUADRADA PARA LOS DIAS PARA EL SISTEMA NORTE

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
LUNES	.00000	9.12296	17.20476	24.68719	24.21585	24.62248	186.06089
MARTES	9.58748	.00000	2.62906	6.15924	5.20271	5.16513	191.35019
MIERCOLES	18.26271	2.67039	.00000	.82232	.75400	5.54862	234.93846
JUEVES	26.41965	6.30758	.83025	.00000	.29287	7.21327	258.39890
VIERNES	26.04965	5.32890	.76415	.29160	.00000	5.26097	243.42609
SABADO	26.48832	5.25125	5.41846	6.95873	5.09359	.00000	187.45823
DOMINGO	161.75875	163.57160	198.11055	216.51448	205.12764	160.81064	.00000

END PROGRAM EXECUTION

PRUEBA CHI-CUADRADA PARA LOS DIAS PARA EL SISTEMA OCCIDENTAL

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
LUNES	.00000	55.91382	71.54102	80.28709	88.07094	90.71405	799.68535
MARTES	61.08748	.00000	2.25281	2.97184	5.81285	24.75144	951.55421
MIERCOLES	78.48053	2.27702	.00000	.94795	1.99545	35.02097	1043.41295
JUEVES	88.69651	3.01345	.95460	.00000	1.05714	32.74495	1045.29521
VIERNES	97.24528	5.91186	2.03118	1.06660	.00000	41.78635	1097.52248
SABADO	100.05254	23.79261	33.16117	30.94807	39.27615	.00000	729.41808
DOMINGO	644.28223	746.96763	811.16571	613.95408	851.26736	589.55269	.00000

END PROGRAM EXECUTION

GRAFICA III.3.9.3

RESULTADOS DE LA PRUEBA CHI-CUADRADA (2)

PRUEBA CHI-CUADRADA PARA LOS DIAS PARA EL SISTEMA ORIENTAL

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
LUNES	.00000	20.49225	31.47540	31.96277	34.69232	43.13415	375.99048
MARTES	21.48932	.00000	3.00149	2.69779	3.47691	15.96878	405.63346
MIERCOLES	33.16893	3.04171	.00000	.97967	1.02243	23.01454	464.45753
JUEVES	33.82909	2.73458	.97805	.00000	.73199	18.31800	447.44291
VIERNES	36.76183	3.52237	1.02187	.73119	.00000	19.32232	452.53003
SABADO	45.28856	15.58961	22.16615	17.69374	18.62589	.00000	295.95518
DOMINGO	331.21512	352.90490	401.18778	387.84558	391.98087	263.69443	.00000

END PROGRAM EXECUTION

PRUEBA CHI-CUADRADA PARA LOS DIAS PARA EL SISTEMA NORESTE

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
LUNES	.00000	9.93949	17.65610	20.96886	23.38638	44.97395	545.32742
MARTES	10.38647	.00000	1.44197	2.86143	3.76931	27.31684	581.50040
MIERCOLES	18.72495	1.46235	.00000	.56937	1.04802	28.05900	607.22066
JUEVES	22.25618	2.90761	.57332	.00000	.33474	34.81617	641.95565
VIERNES	24.87929	3.83959	1.05794	.33505	.00000	36.30849	655.68660
SABADO	45.33177	26.00103	27.17433	32.61360	33.93027	.00000	398.78009
DOMINGO	438.48598	465.70613	486.09565	511.51988	520.93316	331.47429	.00000

END PROGRAM EXECUTION

GRAFICA III.3.9.4

RESULTADOS DE LA PRUEBA CHI-CUADRADA (3)

PRUEBA CHI-CUADRADA PARA LOS DIAS PARA EL SISTEMA NOROESTE

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
LUNES	.00000	15.27745	25.47815	35.60264	28.18462	27.28048	212.69693
MARTES	16.36506	.00000	2.07497	5.94799	2.49821	11.02452	240.81830
MIERCOLES	27.51429	2.10741	.00000	1.12639	.26103	17.93240	284.74985
JUEVES	38.81965	6.10030	1.13970	.00000	1.13364	24.98107	315.57526
VIERNES	30.68532	2.54715	.26167	1.11994	.00000	16.01838	279.19753
SABADO	28.86284	10.51113	16.89498	23.40107	15.14234	.00000	161.21106
DOMINGO	180.02294	200.19969	233.92630	257.67107	230.03010	138.79841	.00000

END PROGRAM EXECUTION

GRAFICA III.3.9.5

RESULTADOS DE LA PRUEBA CHI-CUADRADA (4)

III.3.9.2 Armónicas y Suavizamiento Exponencial Modelo Diario.

El esquema de Pronóstico propuesto por Christiaanse, modela la carga en la siguiente forma : {41}

$$Z(K) = \sum_{k=1}^{N} C_k \phi_k + R(K) \qquad \text{(III.3.9.2-1)}$$

Donde :

C_k - coeficientes generalizados de Fourier.

ϕ_k - funciones ortogonales (Sen W_k, Cos W_k).

N - número de funciones a considerar en la expansión.

R(K) - Componente residual.

K - hora del día.

Este modelo está compuesto por la superposición de tres componentes :

a) Componente base.

b) Componente horaria.

c) Componente residual.

La ecuación anterior puede expresarse de la siguiente forma :

$$Z(K) = C + \sum_{I=1}^{M} (A_I \operatorname{Sen} W_I K + B_I \operatorname{Cos} W_I K) \qquad \text{(III.3.9.2-2)}$$

Donde :

Z(K) - carga horaria medida a la hora K- (MWH/H).

C - componente de directa de la serie de Fourier (componente base).

W_I - múltiplos de la velocidad angular que se obtienen de análisis espectral.

C, B_I, A_I - coeficientes de la serie de Fourier.

M - número de armónicas a considerar.

Para la obtención y actualización de los coeficientes de la serie de Fourier, se propone el método de suavizamiento exponencial.

III.3.9.2.1 <u>Inicialización</u>.

Los valores de los coeficientes de Fourier, se obtienen empleando la siguiente ecuación:

$$\text{MIN} \sum_{J=0}^{N} \beta^{N-J} \{Z(J)-A^T F(J)\}^2 \qquad \text{(III.3.9.2-3)}$$

Donde :

N - número de puntos muestrales disponibles para la inicialización

β - constante de suavizamiento exponencial ($0 < \beta < 1$)

Z - carga horaria medida a la hora considerada

F - funciones ortogonales (senos y cosenos)

A - matriz de coeficientes a determinar

I - hora considerada

La expresión III.3.9.2-3 se aplica a cada día de la semana.

III.3.9.2.2 Actualización.

Para la actualización de los coeficientes de Fourier se utiliza la siguiente ecuación:

$$A^TN = A^TVL+H\ \{Z(J)-XB(J)-XH(J)\} \qquad \text{(III.3.9.2-4)}$$

Donde :

A^TN - valor actualizado de los coeficientes

A^TV - valor anterior de los coeficientes

$= (C\ A_1\ B_1\ .\ .\ .\ A_M\ B_M)$

L - matriz de transición

$$L = \begin{bmatrix} 1 & & & & \\ \cos W1 & \operatorname{sen} W1 & & 0 & 0 \\ -\operatorname{sen} W1 & \cos W1. & & . & . \\ 0 & 0 & . & 0 & 0 \\ & & & . & \\ . & . & & \cos WM & \operatorname{sen} WM \\ 0 & 0 & & -\operatorname{sen} WM & \cos WM \end{bmatrix}$$

H - vector de suavizamiento exponencial

Z(J) - valor real de la carga a la hora "J" - (MWH/H)

XB(J) - componente base a la hora "J"

XH(J) - componente horaria a la hora "J"

El vector de suavizamiento exponencial es de la forma :

$$H = \{ \sum_{J=0}^{N} \beta^{N-J} F(-J)F^T(-J)\}^{-1} F(0) \qquad \text{(III.3.9.2-5)}$$

Donde :

N - número de puntos muestrales

β - constante de suavizamiento exponencial ($0 < \beta < 1$)

F - matriz de funciones ortogonales

J - hora del día considerado

La ecuación (III.3.9.2-5), se aplica a cada día de la semana.

III.3.9.2.3 Extrapolación.

Una vez actualizados los coeficientes de la serie, se utilizan para determinar la carga estimada.

Estudios realizados permiten recomendar el uso de modelos diarios.

Resultado de esto, es el definición de la velocidad angular $W = 2\pi/24$ y el número de armónicas a utilizar $M = 11$.

III.3.9.3 Armónicas y filtro de Kalman modelo diario.

El esquema de pronóstico propuesto por Sharma y Mahalanabis {37,40}, modela la carga eléctrica en la siguiente forma :

$$Z(K) = Bo(K) + \sum_{I=1}^{M} \{B_I(K)\,Cos W_I K + A_I(K)\,Sen W_I K\} + V(K) \qquad \text{(III.3.9.3-1)}$$

Donde :

$Z(K)$ - carga horaria medida (MWH/H)

M - número máximo de armónicos a considerar

B_I y A_I - coeficientes de Fourier

W_I - velocidad angular fundamental

V(K) - secuencia de ruido aleatorio Gaussiano de media cero y varianza R(K)

K - indica la hora del día

Con objeto de actualizar los coeficientes de la serie de Fourier $A_I(K)$ y $B_I(K)$, propuestos en III.3.9.3-1 se sugiere el siguiente modelo de estado :

$$X(K+1) = \phi^T X(K) + W(K) \qquad \text{(III.3.9.3-2)}$$

Donde :

X(K+1) - extrapolación del vector de estado

X(K) - vector de estado

ϕ - matriz de transición

$$\phi = \begin{bmatrix} 1 & & & & \\ \cos W1 & \operatorname{sen} W1 & 0 & 0 & \\ -\operatorname{sen} W1 & \cos W1 & & & \\ 0 & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & & \cos WM & \operatorname{sen} WM \\ & & & -\operatorname{sen} WM & \cos WM \end{bmatrix}$$

W(K) - ruido aleatorio Gaussiano de media cero y varianza Q(K)

Las expresiones III.3.9.3-1 y III.3.9.3-2 pueden ser utilizadas por un filtro de Kalman Bucy para actualizar en forma óptima el vector de estados X(K) al momento en que la observación Z(K) esté disponible.

Las ecuaciones que caracterizan este filtro - se bosquejan en las tablas III.3.9.3.1 a --- III.3.9.3.4.

Estudios realizados desde septiembre de 1976, permiten recomendar nuevamente el empleo de - modelos diarios lo cual define los valores -- "M" y "W" de la expresión III.3.9.3-1, esto implica :

$M = 11$

$W = 2\pi/24$

M O D E L O H A R M - K B F

$$Y(K) = b_0(K) + \sum_{i=1}^{M} b_i(K) \cos \frac{2\pi i}{168} K + a_i(K) \operatorname{sen} \frac{2\pi i}{168} K$$

$$Y(K) = f(K)\, a(K) + V(K) \qquad (1)$$

Donde :

$$a(K) = \begin{bmatrix} b_0(K) \\ a_1(K) \\ b_1(K) \\ a_2(K) \\ b_2(K) \\ \vdots \end{bmatrix}$$

$$V(K) = N(0, R(K))$$

$$f(K) = \begin{bmatrix} 1 \\ \operatorname{sen} f_1 K \\ \cos f_1 K \\ \operatorname{sen} f_2 K \\ \cos f_2 K \\ \vdots \end{bmatrix}$$

$$fi = \frac{2\pi i}{168}$$

$$a(K+1) = a(K) + W(K) \qquad (2)$$

Donde : $W(K) = N(0, Q(K))$

TABLA III.3.9.3.1

MODELO DE ESTADO

$$Y(K) = f'(0)\ a(K) + V(K)$$

$$a(K+1) = L'a(K) + W(K)$$

Donde :

$$L' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos f_1 & \operatorname{sen} f_1 & 0 & 0 \\ 0 & -\operatorname{sen} f_1 & \cos f_1 & 0 & 0 \\ \vdots & & \vdots & & \\ 0 & 0 & 0 & \cos fm & \operatorname{sen} fm \\ 0 & 0 & 0 & -\operatorname{sen} fm & \cos fm \end{bmatrix}$$

TABLA III.3.9.3.2

M O D E L O D E P R O N O S T I C O

Estimación: $\hat{a}(K/K-1) = L' \hat{a}(K-1/K-1)$

$$P(K/K-1) = L' P(K-1/K-1)L + Q(K-1)$$

Ganacia de Kalman : $K(K) = P(K/K-1) f(0) \{f'(0)P(K/K-1)f(0)+R(K)\}^{-1}$

Actualización :

$$a(K/K) = a(K/K-1)+K(K) \{Y(K)-f'(0)\hat{a}(K/K-1)\}$$

$$P(K/K) = \{I-K(K)f'0\} P(K/K-1)$$

Donde : $a(K)$ = vector de estado

$P(K)$ = covarianza del error

Y

$$R(K) = \frac{1}{K} \{(K-1)\hat{R}(K-1) + r^2(K) - f'(0)P(K/K-1) f(0)\}$$

$$W(K) = \frac{1}{K}\{(K-1)\hat{W}(K-1)+r^2(K)K(K)K'(K)+P(K/K)-L'P(K-1/K-1)L\}$$

TABLA III.3.9.3.3

$$\lim_{K \to \infty} R(K) = \text{ctte}$$

$$\longrightarrow K(K) \to Ks$$

$$\lim_{K \to \infty} W(K) = \text{ctte}$$

$$Y(K + j/K) = f'(0)\ L'\hat{As}(K/K)$$

$$As(K/K) = As(K/K-1) + Ks\ \{Y(K) - f'(0)\ \hat{a}(K/K-1)\}$$

$$As(K/K-1) = L'\hat{As}(K-1/K-1)$$

TABLA III.3.9.3.4

III.3.9.4 Compensación de Errores

Este esquema de pronóstico propuesto por el Ing. José Luis Martínez M. (CLYFC), utiliza el siguiente modelo de pronóstico :

$$\hat{Z}(I+1,J)=Z(I,J)+\hat{K}(I+1,J)\{Z(I,J)-\hat{Z}(I,J)\} \qquad \text{(III.3.9.4-1)}$$

Donde :

$\hat{Z}(I+1,J)$ - pronóstico de carga para el día "I+1", a la hora "J"

$Z(I,J)$ - carga horaria medida del día "I", a la hora "J" (MWH/H)

$\hat{K}(I+1,J)$ - ganancia extrapolada para compensar el error para el día "I+1", a la hora "J"

$\hat{Z}(I,J)$ - pronóstico de carga para el día "I" a la hora "J"

Como puede observarse existe un modelo para cada hora de la semana.
A partir de la expresión anterior, pueden calcularse los valores K(I,J) en la forma :

$$K(I,J)=\frac{\hat{Z}(I,J)-Z(I-1,J)}{\hat{Z}(I-1,J)-Z(I-1,J)} \qquad \text{(III.3.9.4-2)}$$

Donde :

Z - cargas medidas (MWH/H)

$\hat{Z}$ - pronósticos de carga

I - indica el día

J - indica la hora

Esta ganancia se necesita extrapolar para el día a pronosticar; esto se realiza de la si--

guiente manera:

$$\hat{K}(I+1,J) = K(I,J) \qquad \text{(III.3.9,4-3)}$$

III.3.10 Información.

Antes de juzgar a la bondad de un algoritmo de pronóstico de carga, es conveniente analizar - la información que se emplea.

Aspectos importantes a considerar en la información actualmente disponible son :

- hora de medición
- precisión de medidores
- errores de paralaje

Detalles de estos problemas pueden ser consultados en las secciones anteriores de este capítulo.

Dicha información se utilizará para establecer una comparación entre los algoritmos que pueden ser instrumentados para el esquema en tiempo real.

A N E X O 1.

ERROR EN LECTURAS DE WATTHORIMETROS

Se supone un medidor analógico para el dígito menos significativo (N_o).

Sean (*)

$\tilde{V}_i = N_4\ N_3\ N_2\ N_1\ N_0$ (valor leído del medidor a la hora i)

$V_i = N_4\ N_3\ N_2\ N_1\ N_0 + \varepsilon_i$ (valor real del medidor a la hora i)

$V_i = \tilde{V} + \varepsilon_i$

Modelo :

$_i$ es una variable normal (gaussiana) de media

$m_\varepsilon = 0$

desviación estándar

$\sigma_\varepsilon = 0.5$

y variancia $\sigma_\varepsilon^2 = 0.25$

Esto implica

	1σ	2σ	3σ	4σ
	$\alpha=0.5$	$\alpha=1$	$\alpha=1.5$	$\alpha=2$
$P\{\lvert\varepsilon_i\rvert<\alpha\}$	68%	95%	99%	100%
$P\{\lvert error\rvert>\alpha\}$ ε_i	32%	5%	1%	0%

*Nota : Sin perder generalidad, se supondrá que NO está en el lugar de las <u>unidades</u> y no, por ejemplo, -

en el de las decenas ($\tilde{V}_i = N_4N_3N_2N_1N_0 0$) ni décimas ($\tilde{V}_i = N_4N_3N_2N_1N_0$). La constante del medidor deberá escalarse adecuadamente para tomar en cuenta esta condición.

Si k es la constante del watthorímetro (escalada como se indicó en la nota anterior), la energía en MWH entre la hora i y la hora j se calcula como :

$\tilde{E} = k(V_i - V_j)$ (valor registrado de la energía - MWH)

Puesto que la energía <u>real</u> es :

$E = k(V_i - V_j)$ (valor real de la energía MWH)

el error de la medición es :

$$e_m = k(\varepsilon_i - \varepsilon_j)$$

que es una variable aleatoria gaussiana de media cero.

Suponiendo que los errores a la hora i y a la hora j son independientes, la variancia del error de medición es :

$$\sigma^2_{e_m} = 2\,k^2\,(0.25)$$

$$= 0.5\,k^2$$

y la desviación estándar es :

$$\sigma_{e_m} = 0.7071\,k.$$

En la siguiente tabla se resumen los resultados obtenidos con la información del Area Noroeste.- Los errores han sido calculados analizando con respecto a las demandas promedio en cada región considerada.

P { |ERROR| < α }				
REGIONES	DEMANDA PROMEDIO	1σ α = 0.5	2σ α = 1	3σ α = 1.5
Sonora Norte σem = 2.12	176.02	1.20%	2.41%	3.61%
Sonora Sur σem = 11.57	92.85	12.46%	24.93%	37.39%
Sinaloa σem = 13.58	90.31	15.03%	30.07%	45.11%
Mazatlán σem = 10.14	41.92	24.18%	48.37%	72.61%
Area Noroeste σem = 3.97	399.73	0.983%	1.96%	2.94%

A N E X O 2.

MUESTPAS DE RESULTADOS DEL ANALISIS ESTADISTICO.

SORTED ROTATED FACTOR LOADINGS (PATTERN)

		FACTOR 1	FACTOR 2	FACTOR 3	FACTOR 4	FACTOR 5	FACTOR 6	FACTOR 7
CLOUDS	7	.815	.000	.000	.000	.000	.000	.000
LOWCLOUD	13	.873	.000	.000	.000	.000	.000	.000
CLOUDTYP	14	.595	.000	.511	.274	.000	.000	.000
WINDVELO	9	.000	.763	.000	.000	.000	.000	.000
WINDFORC	26	.000	.759	.000	.000	.000	.000	.000
WINDDIRC	8	.000	.725	.000	.000	.000	.000	-.279
DEWPOINT	16	.000	.000	.810	.000	.000	.000	.000
TEMPTURE	12	.000	.384	.686	.000	.000	-.278	.000
HEIGHTLW	15	-.492	.000	-.640	.000	.000	.000	.000
HOURPREC	20	.000	.000	.000	.823	.000	.000	.000
TOTALPRC	19	.000	.000	.000	.712	.000	.000	.000
GENLSTAT	11	.231	.000	.000	.568	.000	.000	.000
PRESSTAT	18	.000	.000	.000	.000	.804	.000	.000
HORVISIB	10	.000	.000	.000	.000	.714	-.252	.000
PRESSCH3	22	.000	.000	.000	.000	.000	.734	.000
PRESSCHG	17	.000	.000	.000	.000	.000	.713	.000
SPECEVNT	21	.000	.000	.000	.000	.000	.000	.733
PRESDIRC	25	.000	.335	.000	.000	.000	.000	-.531
	VP	2.237	2.048	1.965	1.817	1.349	1.326	1.022

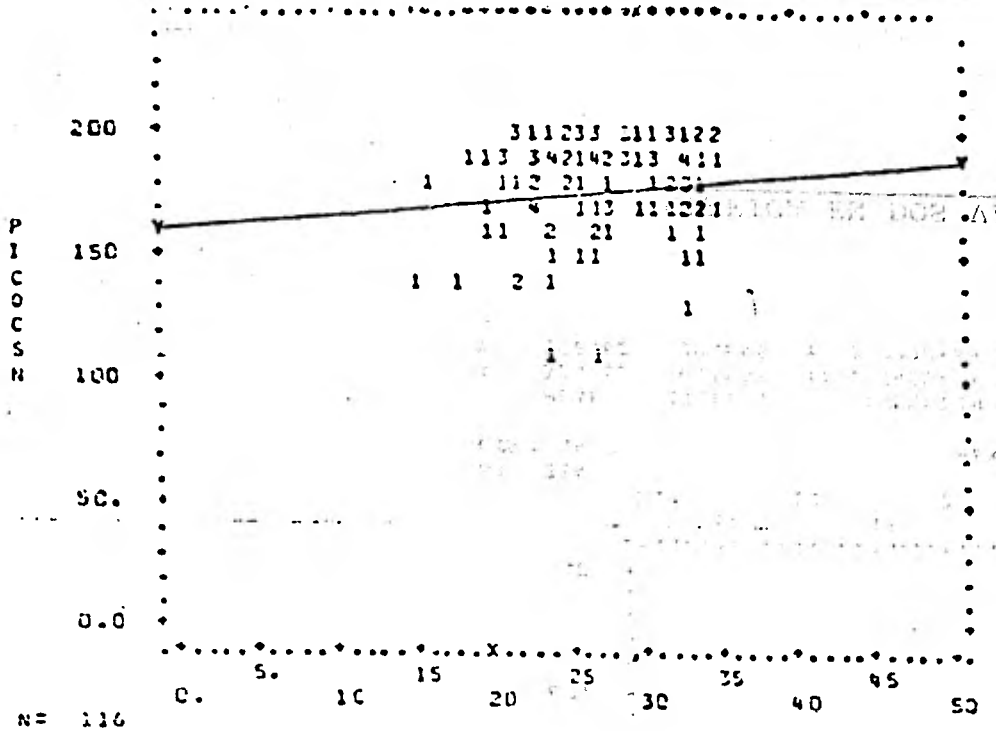
THE ABOVE FACTOR LOADING MATRIX HAS BEEN REARRANGED SO THAT THE COLUMNS APPEAR IN DECREASING ORDER OF VARIANCE EXPLAINED BY FACTORS. THE ROWS HAVE BEEN REARRANGED SO THAT FOR EACH SUCCESSIVE FACTOR, LOADINGS GREATER THAN .5[illegible] APPEAR FIRST. LOADINGS LESS THAN .2500 HAVE BEEN REPLACED BY ZERO.)

ABSOLUTE VALUES OF CORRELATIONS IN SORTED AND SHADED FORM

```
 7 CLOUDS     ■
13 LOWCLOUD   ■■
14 CLOUDTYP   ■X■
 9 WINDVELO   . .■
26 WINDFORC   ..-X■
 8 WINDDIRC    .X+■
16 DEWPOINT   --+   ■
12 TEMPTURE   . .--XX■
15 HEIGHTLW   ++■...+-■
20 HOURPREC   --- . . .■
19 TOTALPRC   --- . . .X■
11 GENLSTAT   +-+ - . -+-■
18 PRESSTAT          .   ■
10 HORVISIB          - .  --■
22 PRESSCH3    . . +.     -■
17 PRESSCHG              .■
21 SPECEVNT               ■
25 PRESDIRC    .+ .        ■
```

THE ABSOLUTE VALUES OF THE MATRIX ENTRIES HAVE BEEN PRINTED ABOVE IN SHADED FORM ACCORDING TO THE FOLLOWING SCHEME

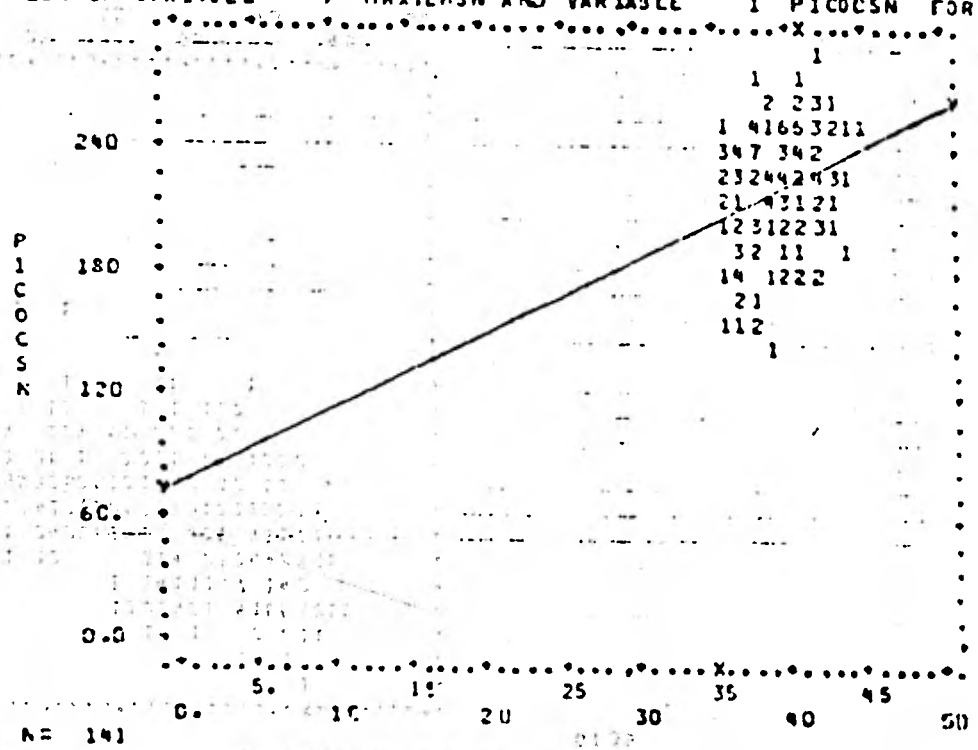
	LESS THAN OR EQUAL TO		.099
.	.099	TO AND INCLUDING	.198
-	.198	TO AND INCLUDING	.297
+	.297	TO AND INCLUDING	.396
X	.396	TO AND INCLUDING	.495
N	.495	TO AND INCLUDING	.594
■	.594	TO AND INCLUDING	.693
■		GREATER THAN	.693



N= 116
COR= .153

	MEAN	ST.DEV.	REGRESSION LINE	RES.MS.
X	27.319	4.7229	X= .03762*Y+ 20.587	21.974
Y	178.91	19.221	Y= .63303*X+ 161.89	363.96

PLOT OF VARIABLE 2 MAXTEMSN AND VARIABLE 1 PICOCSN FOR GROUP GT3

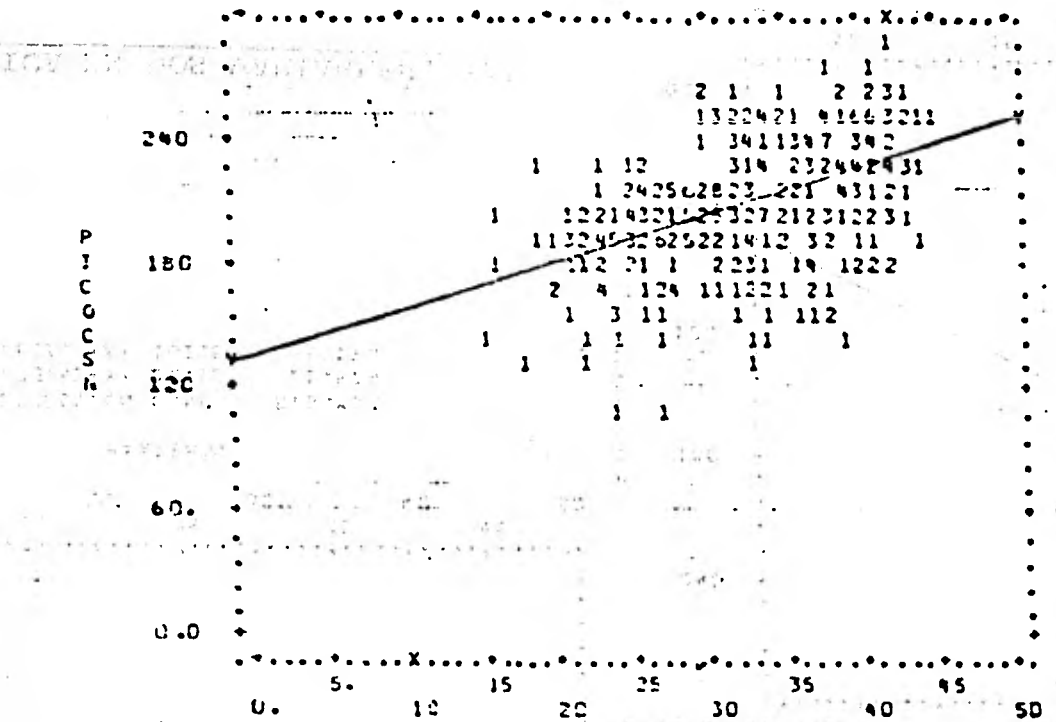


N= 141
COR= .279

	MEAN	ST.DEV.	REGRESSION LINE	RES.MS.
X	39.574	2.2014	X= .02121*Y+ 34.89	4.5015
Y	223.02	28.941	Y= 3.6660*X+ 77.94	777.99

GRAFICA EN DOS VARIABLES. (1)

PLOT OF VARIABLE 9 MAXTEMSN AND VARIABLE 1 PICOCSN FOR GROUP LT35
GT35



N= 364
COR= .487

MAXTEMSN

	MEAN	ST.DEV.	REGRESSION LINE	RES.MS.
X	32.852	6.5598	X= .10400*Y+ 11.154	32.931
Y	208.62	30.695	Y= 2.2770*X+ 133.81	721.05

GRAFICA EN DOS VARIABLES (2)

PROGRAMA DE ESTIMACION DE

CARGA HORARIA

FASE : SELECCION DE ALTERNATIVAS

III.4 Selección de Alternativas del PECHO.

III.4.1 Introducción.

Los resultados de los algoritmos de estimación de carga analizados en la sección anterior, son ordenados para realizar la elección del algoritmo que será objeto del diseño de programas.

Para realizar la evaluación comparativa de los algoritmos propuestos es necesario definir :

a) Tipo de Información utilizada, indicando el período que abarca la información, así como el tratamiento que se le ha dado a la misma (filtrado, suavizamiento, etc....).

b) Indices de evaluación, que son los criterios con que se evaluarán los resultados obtenidos con los prototipos. Estos índices se ordenarán de una forma tal que, facilitará la comparación de los resultados.

c) Recursos de cómputo utilizados, que puede considerarse como otro índice de evaluación, ya que un algoritmo que arroje los mejores resultados, puede no ser la más eficiente en el uso de los recursos de cómputo.

A continuación se realiza el desarrollo de la etapa de Selección de Alternativas para el programa de Estimación de Carga.

III.4.2 Evaluación.

III.4.2.1 Filtrado de Información.

Es necesario contar con un filtro de información eléctrica con objeto de analizar los días que tienen características anormales en su comportamiento y eliminarlos, tanto del proceso de aprendizaje de los predictores como de la evaluación de los pronósticos.

Estudios realizados en colaboración con el personal de la división Centro de CFE, permiten sugerir un filtro de energía para satisfacer estos fines.

La figura III.4.2.1.1 muestra las ecuaciones del filtro y el esquema de evaluación propuesto.

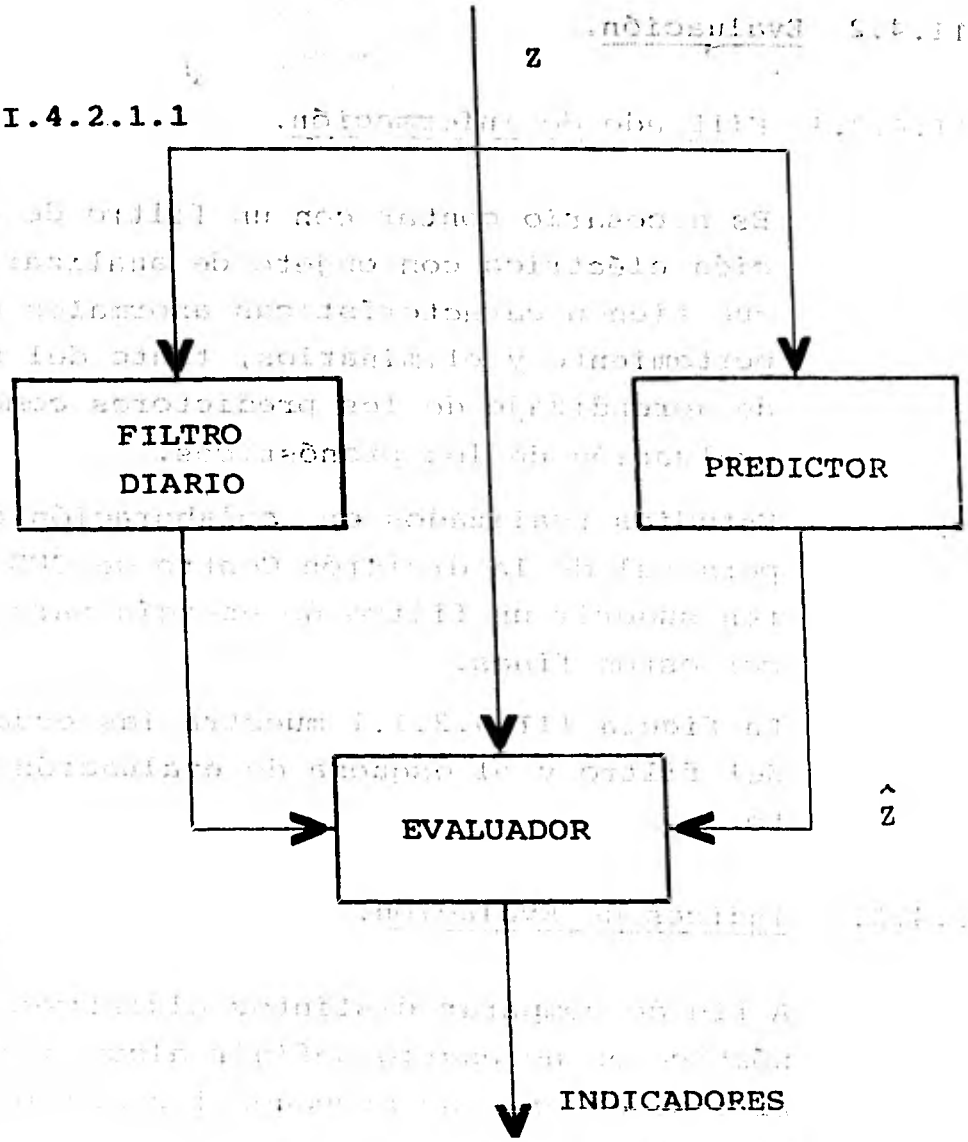
III.4.2.2 Indices de Evalución.

A fin de comparar distintos algoritmos de pronóstico es necesario definir algunos criterios de error. Se proponen los siguientes criterios :

- residuos normalizados (NAR)
- valor medio cuadrático (RMS)
- residuos de energía diaria normalizados (DEAE)
- residuos a la hora pico normalizado (PAE)

Las expresiones que definen estos errores se presentan en la figura III.4.2.2.1.

FIGURA III.4.2.1.1



EL DIA ES NORMAL SI :

$E_{DI} > \alpha E_{PI}$

DONDE :

E_{DI} - ENERGIA DEL DIA I

E_{PI} - ENERGIA PROMEDIO DIARIO HASTA EL DIA I

α - LIMITE DE TOLERANCIA (APROXIMADAMENTE 97%)

FIGURA III.4.2.2.1 **E V A L U A C I O N**

$$NAR = \frac{1}{N} \sum_{J=1}^{N} \left| \frac{\hat{Z}(J) - Z(J)}{Z(J)} \right|$$

$$RMS = \left(\frac{1}{N} \sum_{J=1}^{N} \left(\hat{Z}(J) - Z(J) \right)^2 \right)^{1/2}$$

$$DEAE = \frac{1}{N} \sum_{I=1}^{N} \left| \frac{\hat{E}(I) - E(I)}{E(I)} \right|$$

$$PAE = \frac{1}{N} \sum_{I=1}^{N} \left| \frac{\hat{P}(I) - P(I)}{P(I)} \right|$$

Donde :

$\hat{Z}(J)$ - estimación de carga a la hora J

$Z(J)$ - medición de carga a la hora J

$\hat{E}(I)$ - energía estimada para el día I

$E(I)$ - energía real para el día I

$\hat{P}(I)$ - valor de pico estimado para el día I

$P(I)$ - valor pico real para el día I

III.4.2.3 Información Empleada.

Los datos empleados para el proceso de inicia lización abarcan el período comprendido entre el 2 de enero y el 2 de julio de 1978, para - cada una de las seris áreas operativas de CFE.

El pronóstico fue hecho para 84 días contados a partir del 3 de julio de 1978.

III.4.2.4 Resultados de la Evaluación Comparativa.

Los índices de error mencionados en el párrafo III.4.2.2 se presentan para cada una de -- las 6 áreas de CFE y para el Sistema Nacional Interconectado en las siguientes tablas :

Tabla III.4.2.4.1 -- AREA CENTRAL

Tabla III.4.2.4.2 -- AREA ORIENTAL

Tabla III.4.2.4.3 -- AREA OCCIDENTAL

Tabla III.4.2.4.4 -- AREA NOROESTE

Tabla III.4.2.4.5 -- AREA NORTE

Tabla III.4.2.4.6 -- AREA NORESTE

Tabla III.4.2.4.7 -- SISTEMA NACIONAL INTERCONECTADO

TABLA DE EVALUACION III.4.2.4.1

A R E A C E N T R A L

METODO / INDICES	COM - G	GES - D	KBF - D	COMP
ERROR ABSOLUTO (%)	3.427	3.910	4.085	3.438
DESVIACION ESTANDAR (MWH)	82.586	97.384	99.395	83.715
ERROR EN ENERGIA (%)	1.910	2.308	1.621	1.151
ERROR EN PICO (%)	2.555	4.239	3.409	2.709
DIAS RECHAZADOS (DIAS)	3	3	3	3

TABLA DE EVALUACION III.4.2.4.2

A R E A O R I E N T A L

INDICES \ METODO	COM - G	GES - D	KBF - D	COMP
ERROR ABSOLUTO (%)	3.961	4.295	4.523	4.075
DESVIACION ESTANDAR (MWH)	64.742	71.482	72.840	66.014
ERROR EN ENERGIA (%)	1.299	1.887	1.642	1.554
ERROR EN PICO (%)	3.689	6.030	4.163	4.013
DIAS RECHAZADOS (DIAS)	4	4	4	4

TABLA DE EVALUACION III.4.2.4.3

A R E A O C C I D E N T A L

METODO / INDICES	COM - G	GES - D	KBF - D	COMP
ERROR ABSOLUTO (%)	2.985	4.023	3.806	6.255
DESVIACION ESTANDAR (MWH)	40.439	54.999	50.453	81.203
ERROR EN ENERGIA (%)	1.613	3.045	2.447	5.288
ERROR EN PICO (%)	2.068	3.065	2.728	3.40721
DIAS RECHAZADOS (DIAS)	61	61	61	61

TABLA DE EVALUCION III.4.2.4.4

A R E A N O R O E S T E

METODO / INDICES	COM - G	GES - D	KBF - D	COMP
ERROR ABSOLUTO (%)	2.725	16.292	5.027	10.813
DESVIACION ESTANDAR (MWH)	19.333	96.123	34.428	71.313
ERROR EN ENERGIA (%)	1.889	16.241	4.267	10.449
ERROR EN PICO (%)	1.955	14.165	3.396	6.236
DIAS RECHAZADOS (DIAS)	41	41	41	41

TABLA DE EVALUACION III.4.2.4.5

A R E A N O R T E

METODO / INDICES	COM - G	GES - D	KBF - D	COMP
ERROR ABSOLUTO (%)	2.835	7.514	4.479	3.223
DESVIACION ESTANDAR (MWH)	19.735	45.424	29.404	21.774
ERROR EN ENERGIA (%)	2.338	7.463	4.093	2.403
ERROR EN PICO (%)	2.435	6.528	3.391	1.717
DIAS RECHAZADOS	57	57	57	57

TABLA DE EVALUACION III.4.2.4.6

A R E A N O R E S T E

INDICES \ METODO	COM - G	GES - D	KBF - D	COMP
ERROR ABSOLUTO (%)	2.513	6.014	3.244	3.128
DESVIACION ESTANDAR (MWH)	24.674	55.480	32.096	30.871
ERROR EN ENERGIA (%)	1.342	5.726	1.851	1.950
ERROR EN PICO (%)	1.742	6.474	2.668	2.519
DIAS RECHAZADOS (DIAS)	37	37	37	37

TABLA DE EVALUACION III.4.2.4.7

A R E A N A C I O N A L

METODO / INDICES	COM - G	GES - D	KBF - D	COMP
ERROR ABSOLUTO (%)	1.492	3.561	1.971	2.675
DESVIACION ESTANDAR (MWH)	114.672	249.930	152.965	210.595
ERROR EN ENERGIA (%)	0.752	2.507	1.094	1.637
ERROR EN PICO (%)	1.22	3.402	2.212	2.957
DIAS RECHAZADOS (DIAS)	18	18	18	18

III.4.3 Conclusiones.

III.4.3.1 Método de Componentes.

El algoritmo denominado "componentes con agrupamiento de días" ha mostrado ser un esquema que proporciona consitentemente buenos resultados tal y como puede observarse en las tablas III.4.2.4.1 a III.4.2.4.7.

Los requerimientos computacionales de este modelo son :

Inicializador	--Memoria Central	- 2.5k palabras
	Tiempo CPU	- 4 seg.
Pronóstico	--Memoria Central	- 3.2k palabras
	Tiempo CPU	- 40m seg.

Cabe mencionar que este esquema es utilizado en diversos centro de control de energía tales como, Wisconsin Electric Power, Pacific Gas and Electric y Boneville Power Administration.

III.4.3.2 Armónicas y Suavizamiento Exponencial.

El método "armónicas y suavizamiento exponencial modelo diario", no tiene la flexibilidad necesaria para ajustarse adecuadamente a las series de tiempo de nuestro país (ver tablas III.4.2.4.1 a III.4.2.4.7).

Los requerimientos computacionales son :

Inicializador--Memoria Central - 23k palabras
Tiempo CPU - 27 min.

Pronóstico --Memoria Central - 16k palabras
Tiempo CPU - 59m seg.

III.4.3.3 Compensación de Errores.

El algoritmo "compensación de errores", ha -- probado ser muy rápido y con bajos requeri--- mientos de memoria. Existe implementado en - CLYFC a manera de respaldo de otros esquemas- de pronóstico.

La calidad de sus pronósticos es deficiente - en algunas áreas como el Noroeste, el Occiden te y el Sistema Nacional Interconectado.

Requerimientos computacionales :

Incializador--Memoria Central - 2k palabras
Tiempo CPU - 3 seg.

Pronóstico --Memoria Central - 2k palabras
Tiempo CPU - 10m seg.

III.4.3.4 Armónicos y Filtro de Kalman.

El modelo "armónicas y filtro de Kalman mode- lo diario" requiere de un gran espacio en me- moria central durante la ejecución de su pro- grama inicializador que se extiende por perío dos muy largos de tiempo,que se exceden los - 45 minutos de procesamiento.

El módulo de pronóstico tiene requerimientos-

computacionales inferiores :

Pronóstico--Memoria Central - 16k palabras
Tiempo CPU - 59m seg.

Esta técnica ofrece mejores resultados que -- los algoritmos de suavizamiento exponencial y compensación de errores.

III.4.3.5 Comentarios Finales.

Se propone sin reserva el algoritmo denominado "componentes con agrupamiento de días", en conjunto con el modelo meteorológico de "descomposición" para realizar los pronósticos en el sistema de control en tiempo real.

** NOTA ** a) Todos los requerimientos computacionales, fueron determinados en base a la computadora UNIVAC-1110 de CFE.

b) Los tiempos determinados en el pronóstico incluyen :

b.1 Actualización
b.2 Pronóstico
b.3 Evaluación

PROGRAMA DE ESTIMACION DE CARGA HORARIA

FASE : REQUERIMIENTOS DEL PROGRAMA DE COMPUTADORA

III.5 Requerimientos del Programa de Computadora (RPC) para el PECHO.

III.5.1 Introducción.

Una vez que se ha seleccionado el algoritmo para el PECHO, se deben de establecer los "requerimientos" referentes a :

a) Diseño
b) Desarrollo
c) Prueba

para el programa que se implementará en el Sistema de Control de Energía en tiempo real.

Estos requerimientos se plasman en el documento RPC, cuya esctructura se definió en la sección II.6 del capítulo anterior.

En esta sección se muestra una parte del documento RPC, correspondiente al PECHO.

No se incluye la totalidad del documento por ser demasiado voluminoso.

Para facilitar la lectura de la siguiente sección, se definen algunos términos que se incluyen en ella.

- Glosario de Definiciones.

1) Sistema - es el conjunto de los programas que residirán en el sistema de información y control en tiempo real.

2) Subsistema - se denomina de esta manera a - cada uno de los programas de - aplicación avanzada.
Ejemplos de subsistemas : despacho económico restringido, estimación de carga horaria, - coordinación hidro-térmica y - cálculo automático de intercambios.

3) Pronóstico de carga - es la denominación -- que recibe el programa de Estimación de - carga horaria.

4) Tareas - son los programas que componen un- subsistema.

5) Funciones - son los programas o rutinas -- que componen una tarea.

6) Monitor - programa que controla las ejecu-- ciones de las diferentes tareas.

7) CNC (NCC) - Centro Nacional de Control

8) CCA (ACC) - Centro de Control de Area

9) MMI - Interfase Hombre-Máquina

10) DCR - rutinas de recolección de datos.

11) RGS - subsistema de generación de reportes.

12) SCADA - sistema automático de adquisición - de datos.

PROGRAMA DE ESTIMACION DE

CARGA HORARIA

MUESTRA DEL DOCUMENTO RPC

3.1.1 Definición del Subsistema Pronóstico de Carga

El objetivo del subsistema Pronóstico de Carga es predecir la carga total horaria y hexahoraria en MWH para:

a) El sistema Nacional Interconectado.
b) Las 6 áreas operativas de CFE.
c) Las regiones que componen cada una de las áreas operativas.

Para períodos de tiempo hasta de 8 días de adelanto.

Los requerimientos básicos se listan a continuación.

- Requerimientos básicos del programa Pronóstico de Carga.

1) Proporcionar valores del pronóstico semanal de energía horaria y hexahoraria.
2) El pronóstico nacional se obtendrá mediante la sumatoria de los pronósticos de cada una de las seis áreas operativas de CFE.
3) El programa se ejecutará por lo menos una vez al día.
4) Cuando el pronóstico se desvíe sustancialmente de la carga real, se podrá ejecutar otro algoritmo que actualizará la demanda horaria con la información más reciente.
5) El programa deberá estar estructurado en base a un modelo meteorológico y tendrá característica adaptiva que permita la actualización automática de sus parámetros.

6) El operador de cada centro de control de las áreas , tendrá la facultad de modificar los pronósticos, así como de cancelar una actualización si, en su juicio, el antiguo modelo refleja mejor las características del consumo de energía eléctrica.

7) Se llevará una estadística para evaluar la intervención manual del operador.

8) El algoritmo deberá verificar automáticamente los datos de entrada.

9) La información meteorológica será alimentada por las terminales de video.

10) Los resultados del pronóstico deberán estar disponibles para el operador y para otros programas de aplicación en la base de datos del CNC.

11) Se incluirá un calendario que permita detectar los días que deberá ser consideradas como días anormales para efectos del pronóstico.

12) Se estimarán las cargas horarias de todos los días festivos del año con exactitudes comparables a las de los días normales, siempre y cuando exista información suficiente en la historia para caracterizar los diferentes tipos de días anormales.

13) El tiempo de adelanto podrá ser definido arbitrariamente por el operador.

14) Cada pronóstico de carga horaria deberá ir acompañado de su desviación estándar.

15) Deberá incluir un filtrado automático de información con objeto de reconocer los días que tienen características especiales o anormales, para

no emplearlos en el proceso de aprendizaje del algoritmo de pronóstico.

16) El archivo de pronóstico será actualizado una vez que el operador esté satisfecho con los resultados.

17) Para cada una de las seis áreas operativas de CFE existirá información en el centro de control de Area de las cargas industriales que vayan a librarse en determinados períodos de tiempo en el futuro de cada región.

18) Para los fines de inicialización de los modelos de pronóstico, es necesario contar con la información eléctrica y meteorológica correspondiente a los últimos 24 meses.

19) Los modelos para una región o para un área incluirán dentro de lo posible, información meteorológica de varios observatorios meteorológicos.

20) Se proporcionará un paquete de programas que permitan actualizar fuera de línea los modelos meteorológicos que afectan la carga eléctrica.

21) Se definen:

a) 5 regiones de pronóstico por área.

b) 6 áreas.

c) 8 días de pronóstico hacia adelante.

d) 0 días de pronóstico hacia atrás.

22) En las áreas el respaldo a disco se realizará diariamente y a cinta cada 7 días.

23) Se almacenarán en disco las cargas correspondientes a los últimos 28 días. Se construirán-

archivos de usuario pra ello.

24) Se manejarán 5 parámetros meteorológicos diaros y sus respectivos pronósticos por cada estación meteorológica. Asimismo, se llevará una estadística por cada pronóstico meteorológico. se manejarán 15 estaciones meteorológicas por región - como máximo.

25) Se desplegarán N1 generadores y N2 enlaces como máximo por región al operador del área en forma horaria, se desplegarán al operador N3 cargas - horarias correspondiente a eventos especiales y a cargas industriales. Este validará la información del día anterior y ejecutará en demanda el programa de pronóstico de carga.(Los valores de N1,N2 y N3 se determinarán al definir el modelo de la red nacional)

26) Requerimientos de interfase hombre-máquina.
Los resultadosde los cálculos involucrados en - el pronóstico de carga serán editados en formatal, que puedan ser utilizados por la función - de interfase hombre-máquina para presentar en - las terminales de video u otras formas de impresión, la siguiente información:

- Pronóstico de carga horario
- Pronóstico de carga hexahoraria
- Pronóstico de energía diaria
- Carga máxima
- Carga mínima
- Pronósticos de carga horaria para períodos - específicos de tiempo en el futuro
- Pronóstico de la carga base para períodos -- específicos de tiempo en el futuro

- Cargas horarias para períodos de tiempo específicos en la historia
- Residuos horarios de carga para períodos de tiempo específicos en la historia
- Estimación de la desviación estándar horaria de la carga
- La componente meteorológica del modelo de carga
- Pronóstico de variables meteorológicas

3.1.1.1 Interfase con Subsistemas

El subsistema Pronóstico de carga no tiene interacción directa con otros subsistemas, sólo proporciona sus resultados a disposición de otros programas o sumsistemas.

3.1.1.2 Desglose del subsistema Pronóstico de carga en tareas

En la figura F3.1.1.2(1) se muestra el diagrama de estructuras del subsistema Pronóstico de Carga.

A continuacón se realiza una breve descripción de las tareas que compone el subsistema.

- Breve descripción de las tareas

1) Monitor de Pronóstico

Controla la secuencia de ejecución de las diferentes tareas que componen el subsistema Pronóstico de Carga.

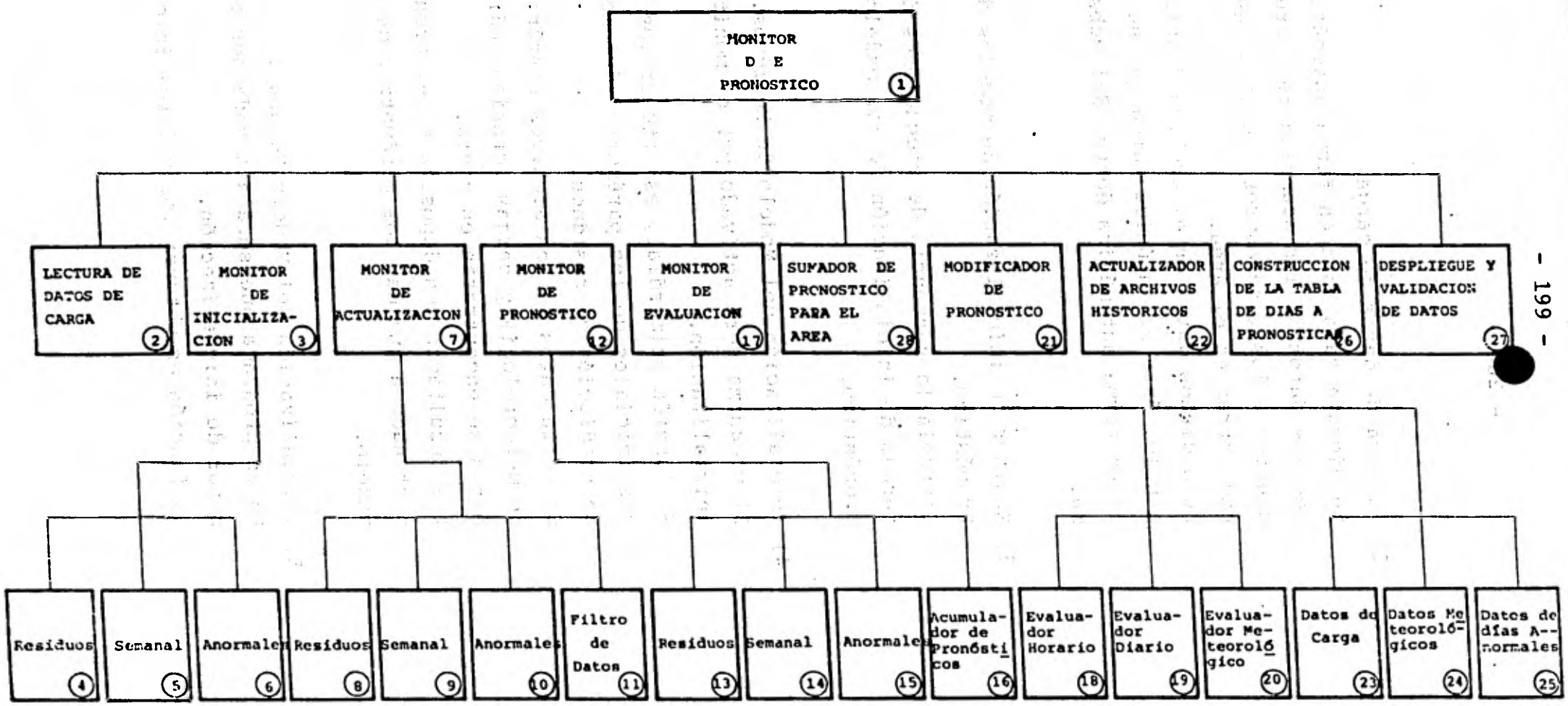


DIAGRAMA DE ESTRUCTURAS DEL SUBSISTEMA PRONOSTICO DE CARGA

2) Lectura de datos de carga

Se ejecuta cada hora en forma automática.

Lee los datos de los acumuladore correspondientes a los generadores y enlaces que se encuentran en el área.

Deja dichos datos en el archivo destinado a ello, éste es la única tarea que se ejecuta en forma automática dentro del subsistema.

3) Monitor de inicialización de modelos de Pronóstico.

Controla la secuencia de ejecuciones de las tareas de inicialización de los modelos de pronóstico.

4) Inicialización del Modelo de Residuos.

El progarma será activado por el monitor de inicialización.

Su función primordial, es obtener los valores iniciales de los parámetros del modelo de predicción de residuos.

Solicita datos de los archivos históricos de meteorología y cargas horarias medidas, éstos se encontrarán en cinta.

Los resultados los deposita en el archivo destinado a guardar los parámetros de este modelo.

5) Inicialización del Modelo Semanal.

Es un programa que será activada por en monitor de inicialización.

Su función es obtener los valores iniciales

de los parámetros del modelo de predic-- ción horaria con adelanto de 8 días.
Solicita datos del archivo histórico de- cargas horarias medidas.
Los resultados son depositados en el ar- chivo destinado a los parámetros del mo- delo semanal.

6) Inicialización del Modelo de días Anormales.
Es un programa que será activado por el- monitor de inicialización.
Su función es,obtener los valores inicia les de los parámetros del modelo de pre- dicción de días anormales.
Solicita datos del archivo histórico de- cargas reales de días anormales.
Los resultados son depositados en el ar- chivo destinado para dichos parámetros.

7) Monitor de actualización de los modelos de pronóstico.
Es activado por el monitor de pronóstico y su función es, controlar la secuencia- de ejecución de las tareas de actualiza- ción de modelos de predicción.

8) Actualización del Modelo de Residuos.
Se activa por medio del monitor de actua lización.
Su función es, actualizar los parámetros del modelo de predicción de residuos.
Utiliza datos de carga horaria provenien te del filtro de datos y los datos meteo rológicos, correspondiente al día ante-- rior.

9) Actualización del Modelo Semanal.

Es activado por el monitor de actualización.

Este programa actualiza los valores de los parámetros del modelo de predicción semanal.

Utiliza los datos de carga medidos del día anterior proveniente del filtro de datos.

10) Actualización del modelo de días Anormales.

Es activado por el monitor de actualización, bajo la condición de que el filtro de datos haya detectado anormalidad en los datos del día anterior.

Su función, es actualizar los parámetros del modelo de predicción de días anormales con los datos que le envía el filtro de datos.

11) Filtro de Datos.

Este programa será activado por el monitor de pronóstico.

Su función, es checar si los datos medidos de carga horaria del día anterior corresponden a un día anormal.

En caso de que los datos resulten ser anormales, suaviza dichos datos para proporcionarlos a los programas de actualización de los modelos de residuos y el semanal.

Los datos sin suavizar se envían al programa de actualización del modelo de pre

dicción de días anormales.

12) Secuenciador de Pronósticos.

Para realizar la estimación de carga horaria, es necesario obtener los resultados de los diferentes modelos en una secuencia determinada. Este programa controla dicha secuencia de obtención de resultados.

Se activa mediante el monitor de pronóstico.

13) Pronóstico de Residuos.

Es activado por el secuenciador de pronósticos.

Realiza el pronóstico de la componente de la demanda que es sensible a factores meteorológicos.

Sus entradas son:

a) Pronóstico de las variables meteorológicos.

b) Parámetros del modelo de predicción de residuos.

14) Pronóstico Semanal.

Es activado por el secuenciador de pronósticos.

Realiza la predicción de la demanda horaria para 8 días en el futuro, en base a los parámetros de este modelo y la fecha de los días a predecir.

15) Pronóstico de días Anormales.

Es activado por el secuenciador de pronósticos, cada vez que se detecte que el día a predecir es anormal.

Utiliza los parámetros correspondiente a este modelo y el pronóstico normal.

16) Acumulador de Pronósticos.
Es activado por el secuenciador de pronósticos.
Su función es, integrar los resultados de los diferentes pronósticos (residuos, semanal y anormales), junto con los datos de eventos especiales y cargas industriales.
El resultado se coloca en el archivo destinado al pronóstico final automático.

17) Monitor de Evaluación.
Es un programa que será activado por el monitor de pronóstico, y controla la secuencia de ejecución de las diferentes tareas de evaluación.

18) Evaluación Horaria.
Es activado por el monitor de evaluación.
Calcula los índices de error de los pronósticos horarios para evaluar la bondad de las estimaciones.
Los resultados se colocan en el archivo destinado a los índices de error horarios.

19) Evaluación Diaria
Es activado por el monitor de evaluación.
Determina los índices de error de los pronósticos a nivel diario, es decir, tomando en cuenta las 24 horas de un día.
Estos cálculos lo realiza para los pronósticos automáticos y modificados por el operador.

Los resultados son colocados en el archivo destinado a los índices de error diarios.

20) Evaluación Meteorológica.

Es activado por el monitor de evaluación. Calcula los índices de error para determinar la bondad de los pronósticos meteorológicos. Dicha evaluación se realiza para cada uno de los parámetros utilizados para realizar el pronóstico de residuos.

Los resultados se guardan en el archivo construído para tal fin.

21) Modificador de Pronósticos.

Es activado por el monitor de pronósticos. Da al operador de área la facilidad de modificar los resultados del pronóstico automático, según su criterio. Las modificaciones realizadas al pronóstico se guardan en un archivo diferente del que contiene los resultados del pronóstico automático.

22) Monitor de Actualización de los Archivos Históricos.

Es necesario que en forma priódica se realice una actividad que es el de guardar la información de carga y meteorología reciente. Este programa es activado por el monitor de pronóstico y controla la ejecución de dicha actividad.

23) Actualización del Archivo Histórico de datos de Carga Horaria Medida.

Es activado por el monitor de actualización de archivos históricos.

Actualiza la información de datos de carga existente en cinta con la información de los últimos 7 días, de esta manera siempre se tendrá un período de datos de carga siempre actualizado.

24) Actualizador del Archivo Histórico de datos Meteorológicos.

Es activado por el monitor de actualización de archivos históricos.

Realiza la misma función que la tarea anterior, pero trabaja con datos meteorológicos.

25) Actualizador del Archivo Histórico de datos de Días Anormales.

Es activado por el monitor de actualización de archivos históricos cada vez que exista datos de días festivos (anormales).

Efectúa la misma función que las dos tareas anteriores, pero con datos de días anormales

26) Construcción de la Tabla de días a Pronosticar.

Es activado por el monitor de pronóstico.

Construye una tabla de fechas para 8 días en el futuro, e indica si esos días deben ser pronosticados como normales o anormales.

27) Despliegues y Validación de Datos por el Operador.

Es activado por el monitor de pronóstico.

Realiza despliegues de la información que solicite el operador, asimismo obtiene las modificaciones que realice el operador sobre

la información desplegada.

28) Sumador de Pronósticos (Area)

Es activado por el monitor de pronóstico.

Realiza la suma de pronósticos de las regiones para así obtener el pronóstico correspondiente al área.

Los programas antes descritos residirán en los centro de control de Area. Sin embargo, nos falta la tarea que obtiene el pronóstico de carga para el Sistema Nacional. Dicho programa residirá en el centro de Control Nacional y se describe a continuación.

29) Sumador de Pronósticos (Nacional)

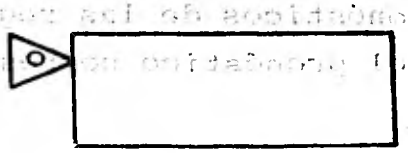
Es activado bajo demanda del operador.

Realiza la suma de los pronóstico de las áreas para así obtener el pronóstico de carga para el sistema Nacional Interconectado.

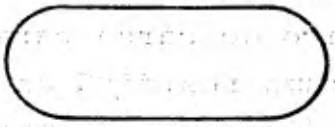
En los siguientes diagramas se muestra la secuencia de ejecución del Subsistema Pronóstico de Carga.

Asimismo los despliegues, mensajes, alarmas y actividades asociadas a cada uno de los pasos de ejecución. Estos diagramas facilitarán la comprensión del funcionamiento de Pronóstico de Carga.

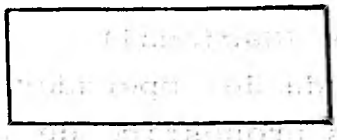
Resumen de simbología utilizada en los diagramas de desplieques



- Desplieques en el que existe una respuesta ó datos de entrada por parte del operador



- Indica alguna actividad, en su mayoría indican activaciones de tareas.



- Desplieques y mensajes informativos, sin intervención -- del operador.

- Conectores

DIAGRAMA 1

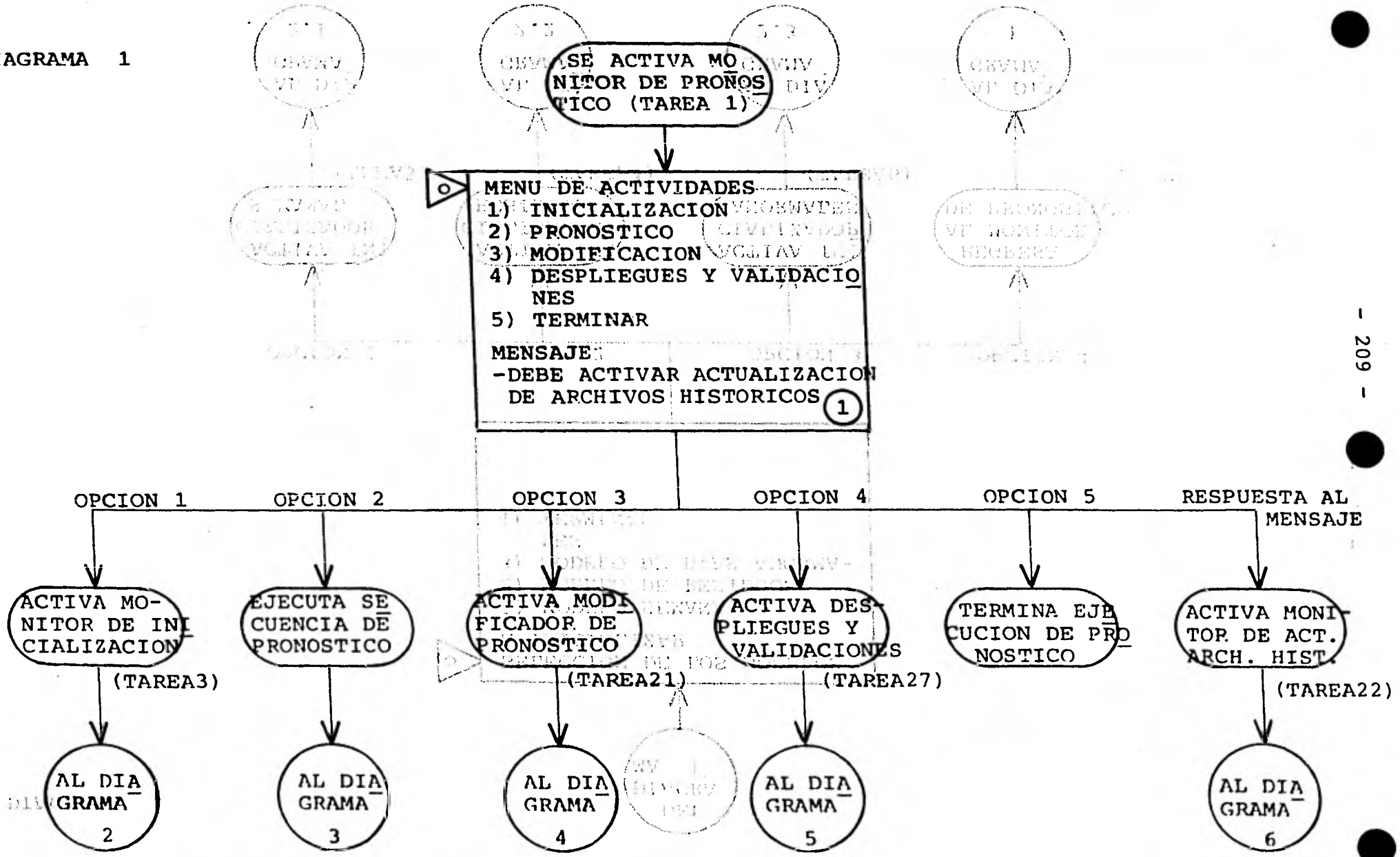
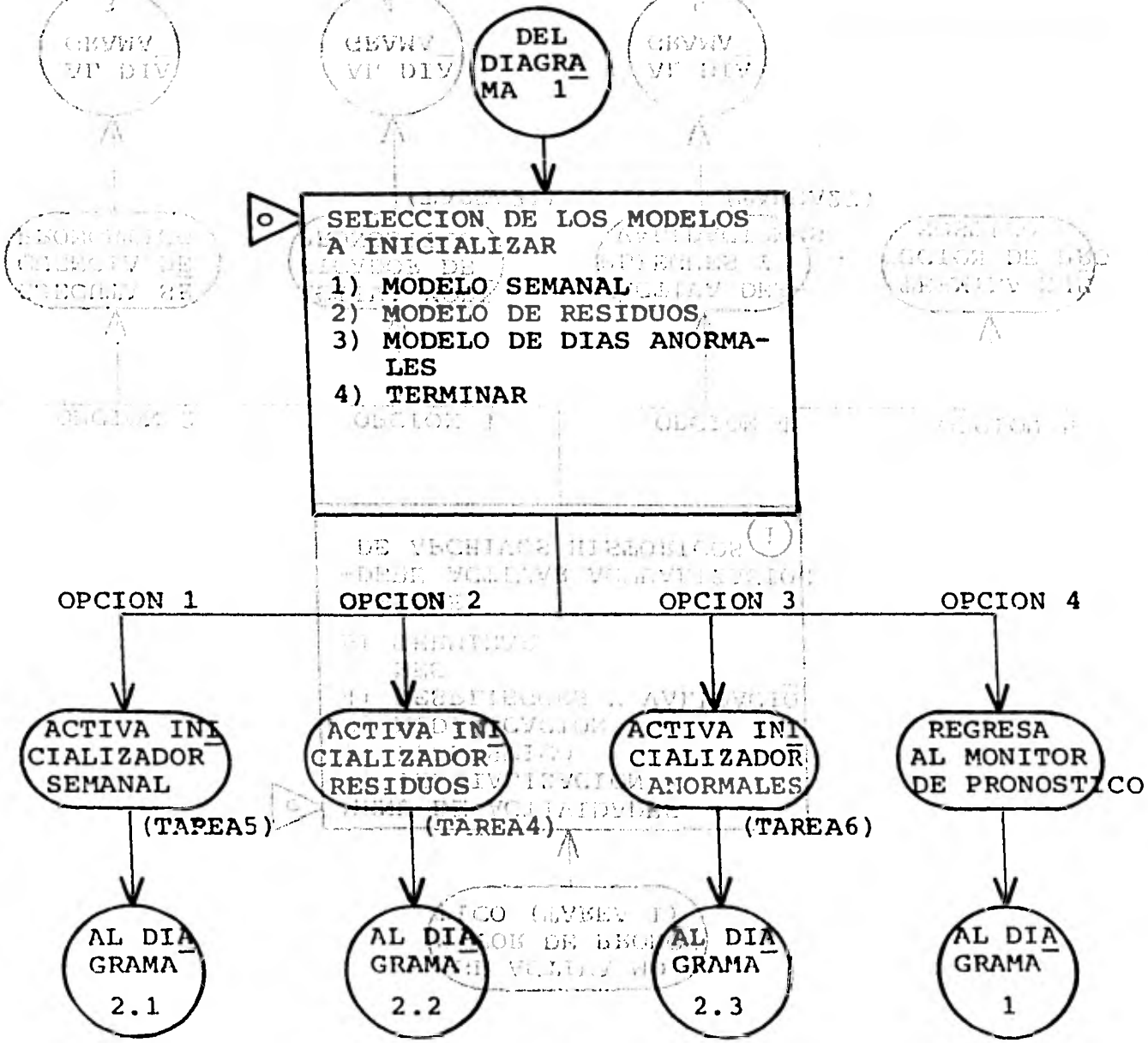

SE ACTIVA MONITOR DE PRONOSTICO (TAREA 1)
MENU DE ACTIVIDADES
1) INICIALIZACION
2) PRONOSTICO
3) MODIFICACION
4) DESPLIEGUES Y VALIDACIONES
5) TERMINAR
MENSAJE
-DEBE ACTIVAR ACTUALIZACION DE ARCHIVOS HISTORICOS
1
OPCION 1
OPCION 2
OPCION 3
OPCION 4
OPCION 5
RESPUESTA AL MENSAJE
ACTIVA MONITOR DE INICIALIZACION
(TAREA3)
EJECUTA SECUENCIA DE PRONOSTICO
ACTIVA MODIFICADOR DE PRONOSTICO
(TAREA21)
ACTIVA DESPLIEGUES Y VALIDACIONES
(TAREA27)
TERMINA EJECUCION DE PRONOSTICO
ACTIVA MONITOR DE ACT. ARCH. HIST.
(TAREA22)
AL DIAGRAMA 2
AL DIAGRAMA 3
AL DIAGRAMA 4
AL DIAGRAMA 5
AL DIAGRAMA 6

DIAGRAMA 2


DEL DIAGRAMA 1
SELECCION DE LOS MODELOS A INICIALIZAR
1) MODELO SEMANAL
2) MODELO DE RESIDUOS
3) MODELO DE DIAS ANORMALES
4) TERMINAR
OPCION 1
OPCION 2
OPCION 3
OPCION 4
ACTIVA INICIALIZADOR SEMANAL
ACTIVA INICIALIZADOR RESIDUOS
ACTIVA INICIALIZADOR ANORMALES
REGRESA AL MONITOR DE PRONOSTICO
(TAREA5)
(TAREA4)
(TAREA6)
AL DIAGRAMA 2.1
AL DIAGRAMA 2.2
AL DIAGRAMA 2.3
AL DIAGRAMA 1

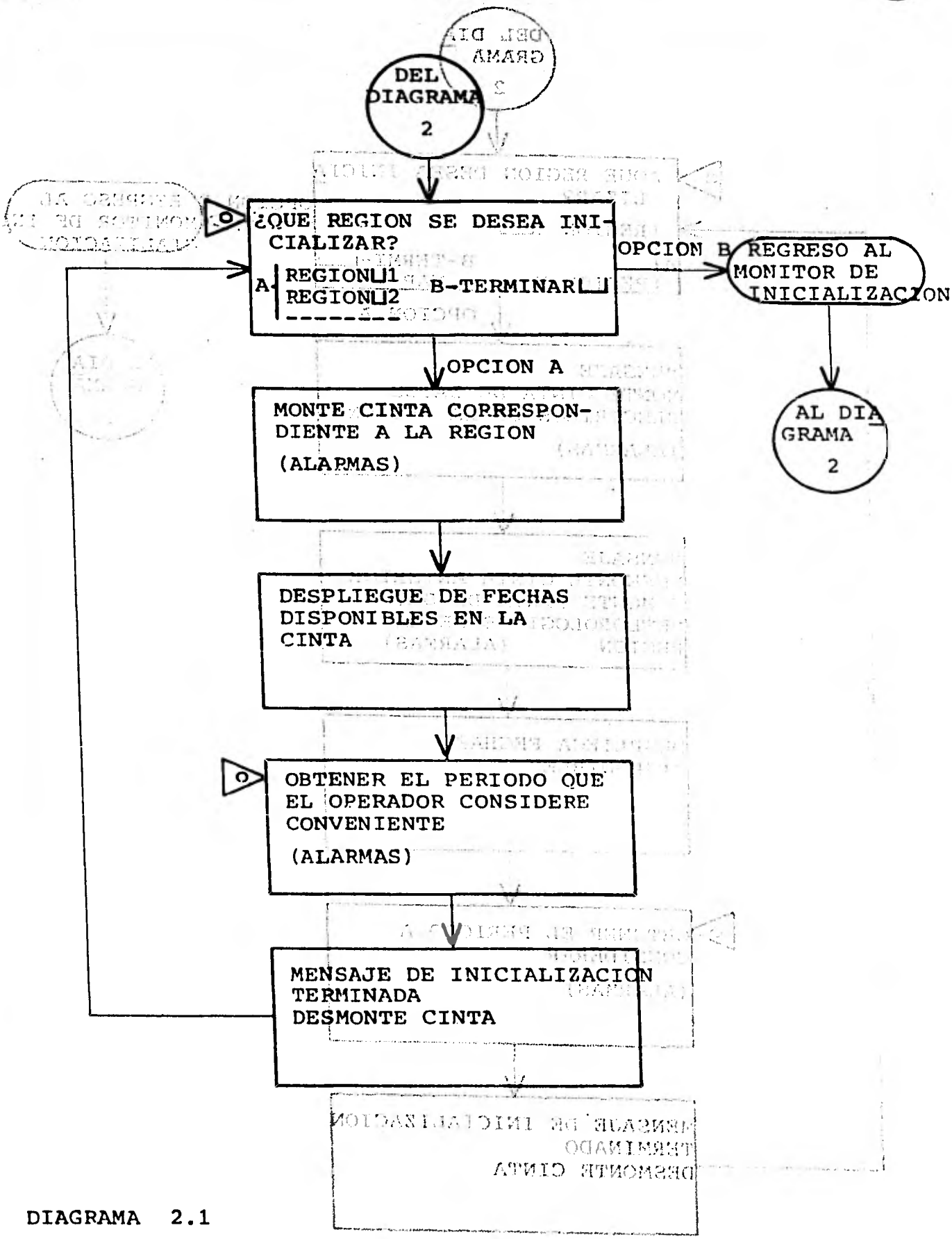

DEL DIAGRAMA 2
¿QUE REGION SE DESEA INICIALIZAR?
A- REGION 1
REGION 2
B-TERMINAR
OPCION B
REGRESO AL MONITOR DE INICIALIZACION
AL DIAGRAMA 2
OPCION A
MONTE CINTA CORRESPONDIENTE A LA REGION
(ALARMAS)
DESPLIEGUE DE FECHAS DISPONIBLES EN LA CINTA
OBTENER EL PERIODO QUE EL OPERADOR CONSIDERE CONVENIENTE
(ALARMAS)
MENSAJE DE INICIALIZACION TERMINADA
DESMONTE CINTA

DIAGRAMA 2.1

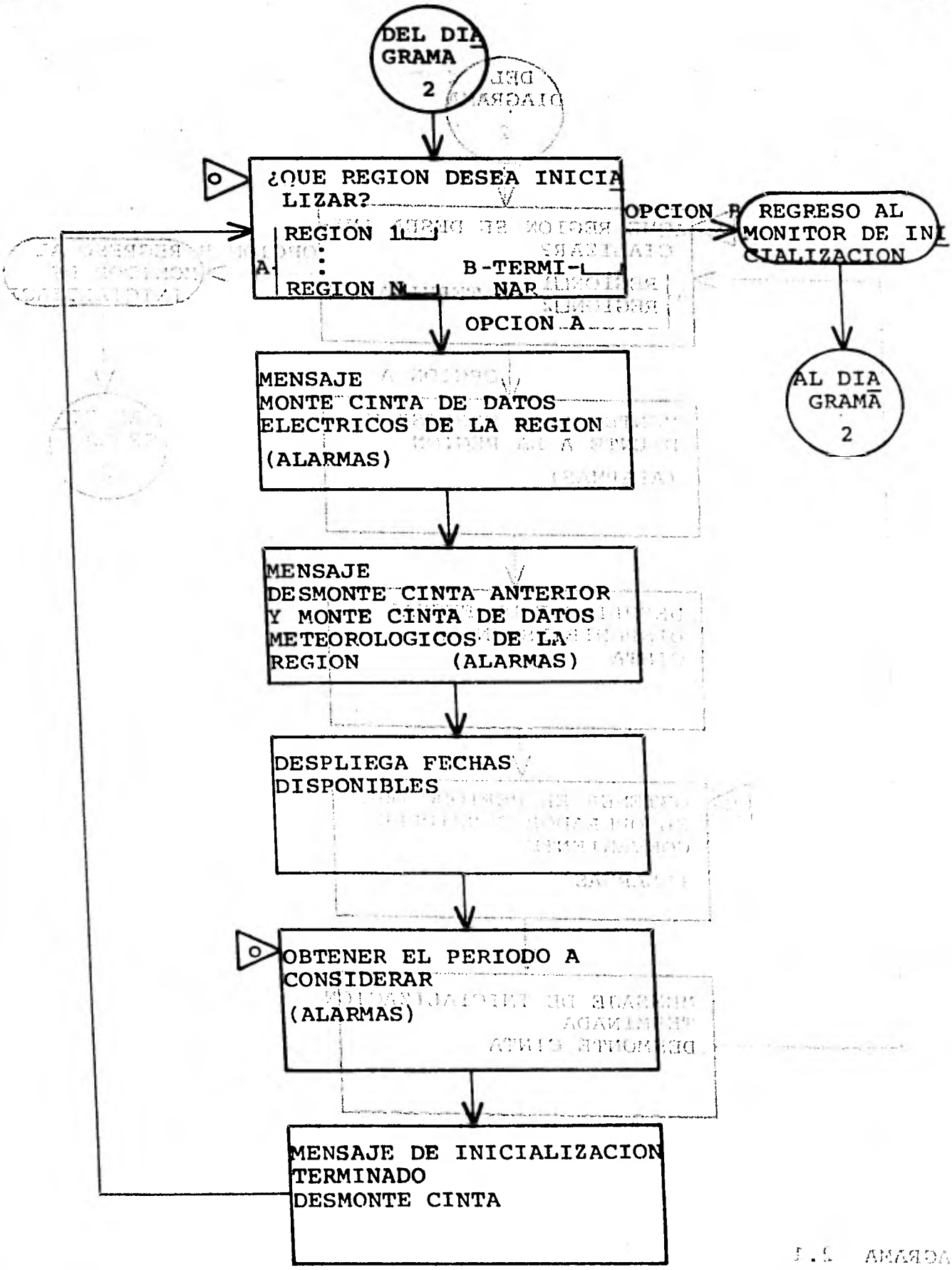

DEL DIA GRAMA 2
¿QUE REGION DESEA INICIALIZAR?
REGION 1
REGION N
A
B-TERMINAR
OPCION B
REGRESO AL MONITOR DE INICIALIZACION
AL DIA GRAMA 2
OPCION A
MENSAJE MONTE CINTA DE DATOS ELECTRICOS DE LA REGION (ALARMAS)
MENSAJE DESMONTE CINTA ANTERIOR Y MONTE CINTA DE DATOS METEOROLOGICOS DE LA REGION (ALARMAS)
DESPLIEGA FECHAS DISPONIBLES
OBTENER EL PERIODO A CONSIDERAR (ALARMAS)
MENSAJE DE INICIALIZACION TERMINADO DESMONTE CINTA

DIAGRAMA 2.2

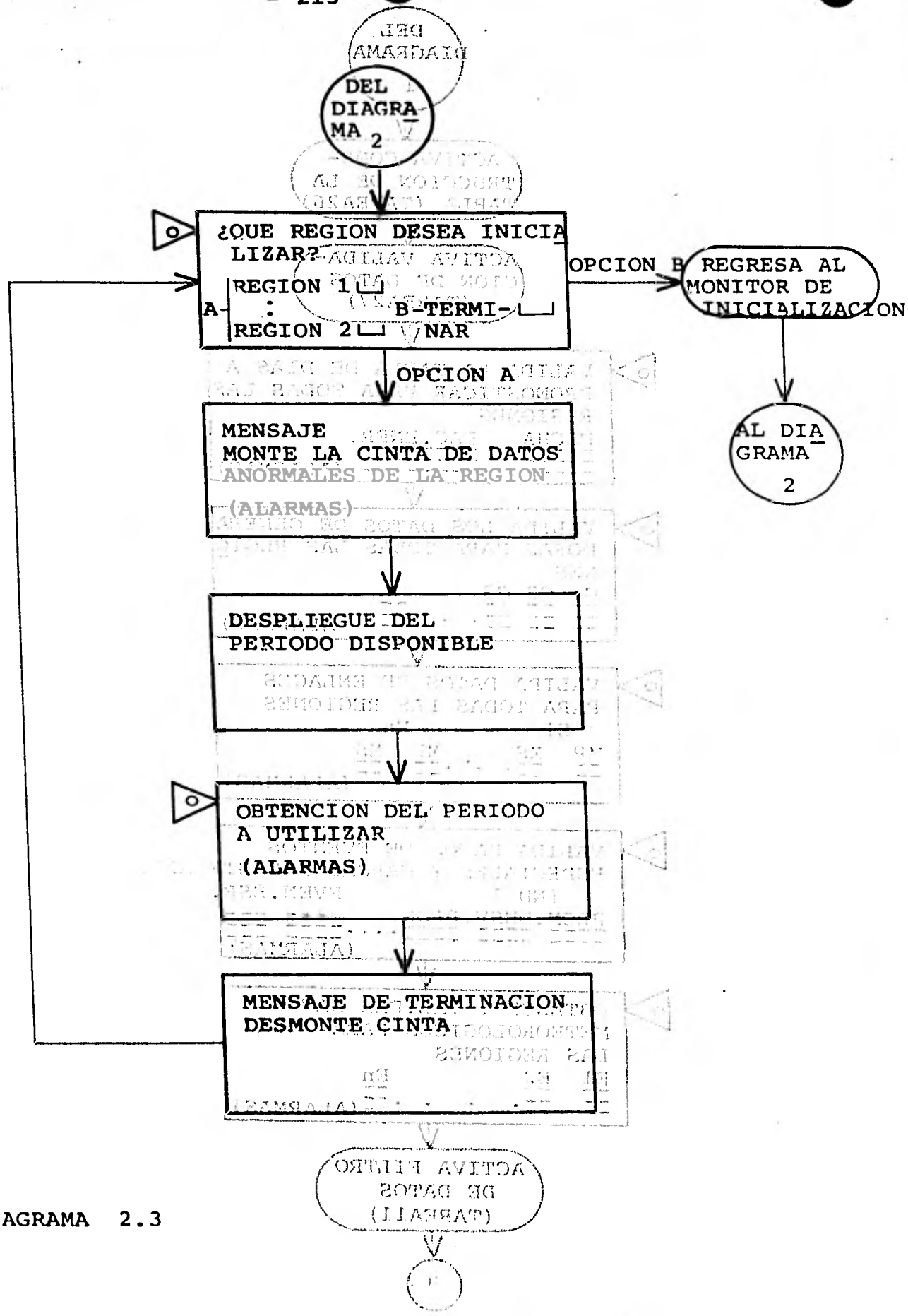

DEL DIAGRAMA 2
¿QUE REGION DESEA INICIALIZAR?
REGION 1
:
REGION 2
A-
B-TERMINAR
OPCION B
REGRESA AL MONITOR DE INICIALIZACION
AL DIAGRAMA 2
OPCION A
MENSAJE
MONTE LA CINTA DE DATOS
ANORMALES DE LA REGION
(ALARMAS)
DESPLIEGUE DEL
PERIODO DISPONIBLE
OBTENCION DEL PERIODO
A UTILIZAR
(ALARMAS)
MENSAJE DE TERMINACION
DESMONTE CINTA

DIAGRAMA 2.3

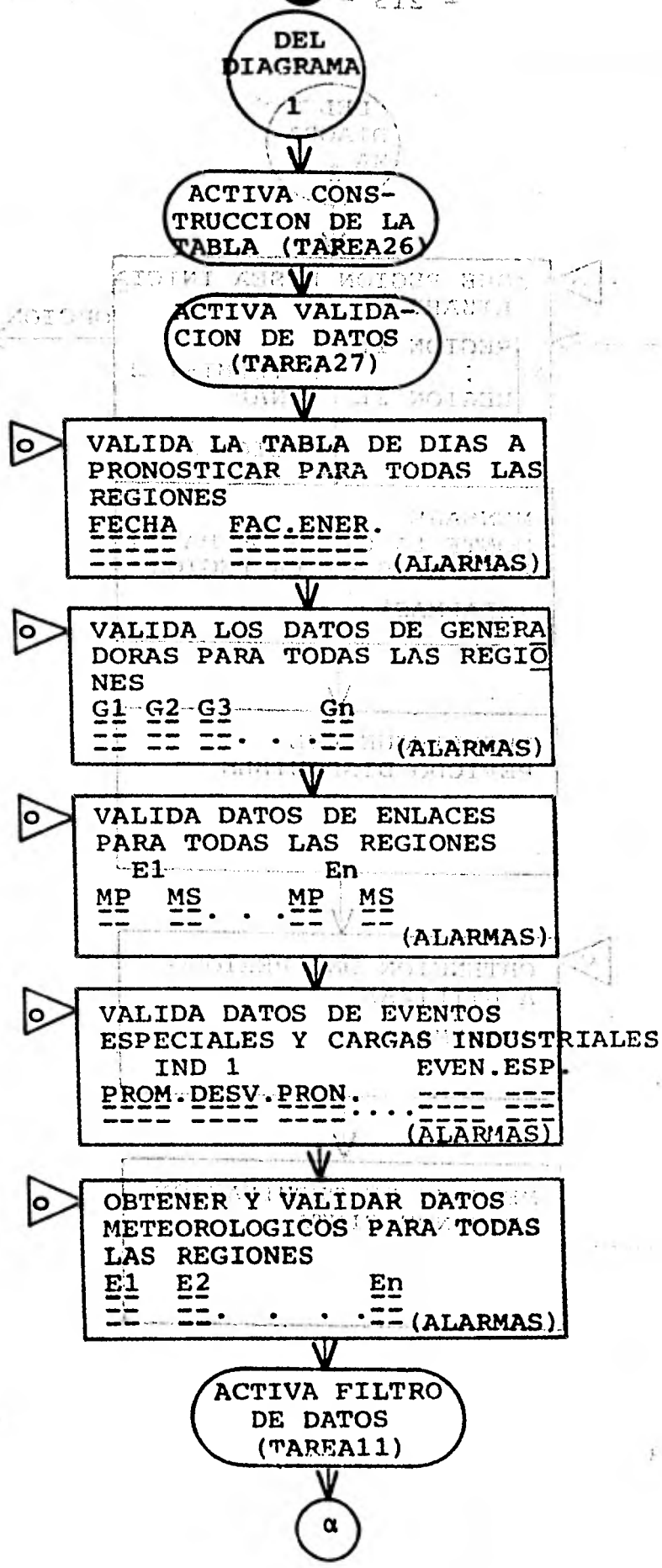

DEL DIAGRAMA 1
ACTIVA CONSTRUCCION DE LA TABLA (TAREA26)
ACTIVA VALIDACION DE DATOS (TAREA27)
VALIDA LA TABLA DE DIAS A PRONOSTICAR PARA TODAS LAS REGIONES
FECHA FAC.ENER.
(ALARMAS)
VALIDA LOS DATOS DE GENERADORAS PARA TODAS LAS REGIONES
G1 G2 G3 Gn
(ALARMAS)
VALIDA DATOS DE ENLACES PARA TODAS LAS REGIONES
E1 En
MP MS MP MS
(ALARMAS)
VALIDA DATOS DE EVENTOS ESPECIALES Y CARGAS INDUSTRIALES
IND 1 EVEN.ESP.
PROM.DESV.PRON.
(ALARMAS)
OBTENER Y VALIDAR DATOS METEOROLOGICOS PARA TODAS LAS REGIONES
E1 E2 En
(ALARMAS)
ACTIVA FILTRO DE DATOS (TAREA11)
α

DIAGRAMA 3(1)

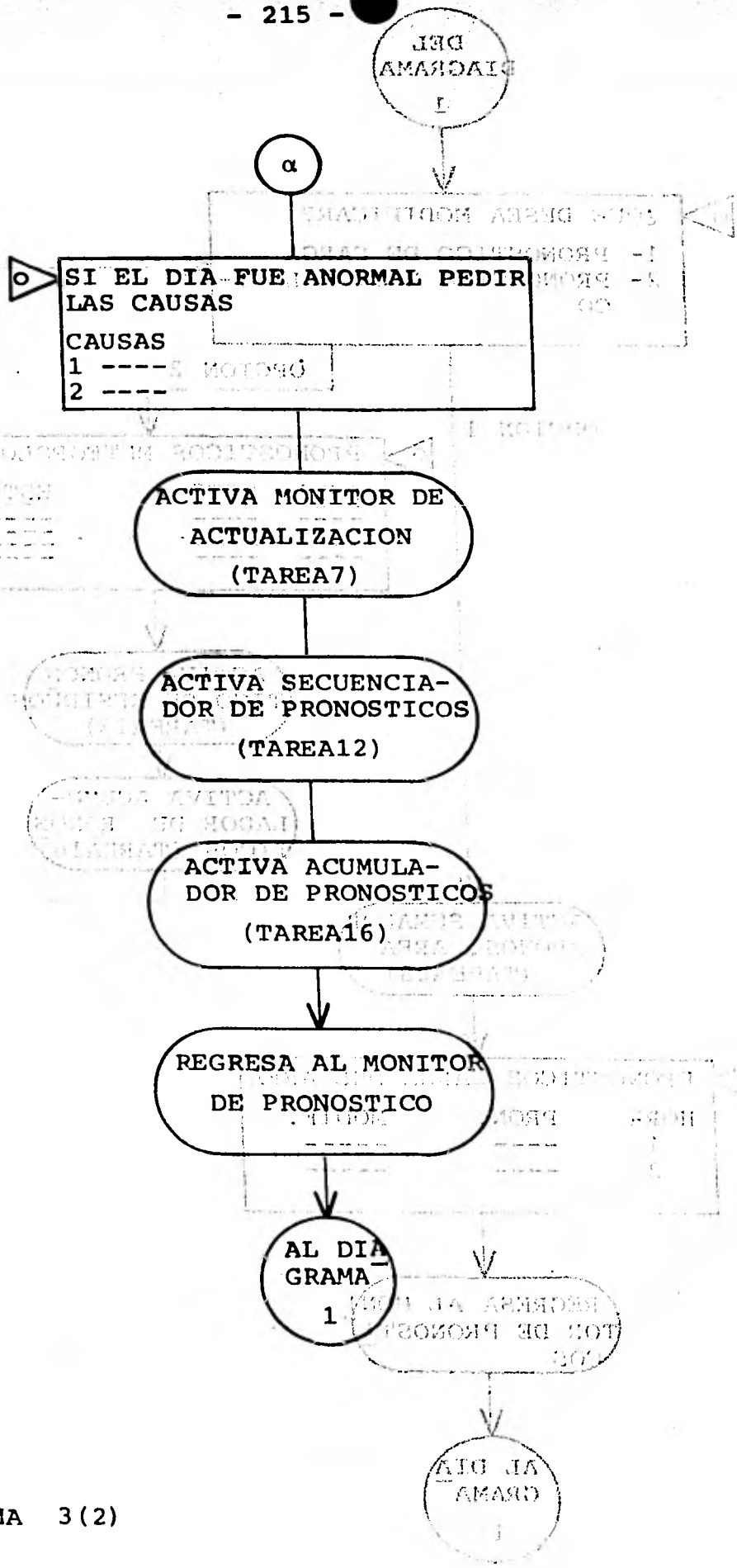

α
SI EL DIA FUE ANORMAL PEDIR LAS CAUSAS
CAUSAS
1 ----
2 ----
ACTIVA MONITOR DE ACTUALIZACION (TAREA7)
ACTIVA SECUENCIA-DOR DE PRONOSTICOS (TAREA12)
ACTIVA ACUMULA-DOR DE PRONOSTICOS (TAREA16)
REGRESA AL MONITOR DE PRONOSTICO
AL DIA-GRAMA 1

DIAGRAMA 3(2)

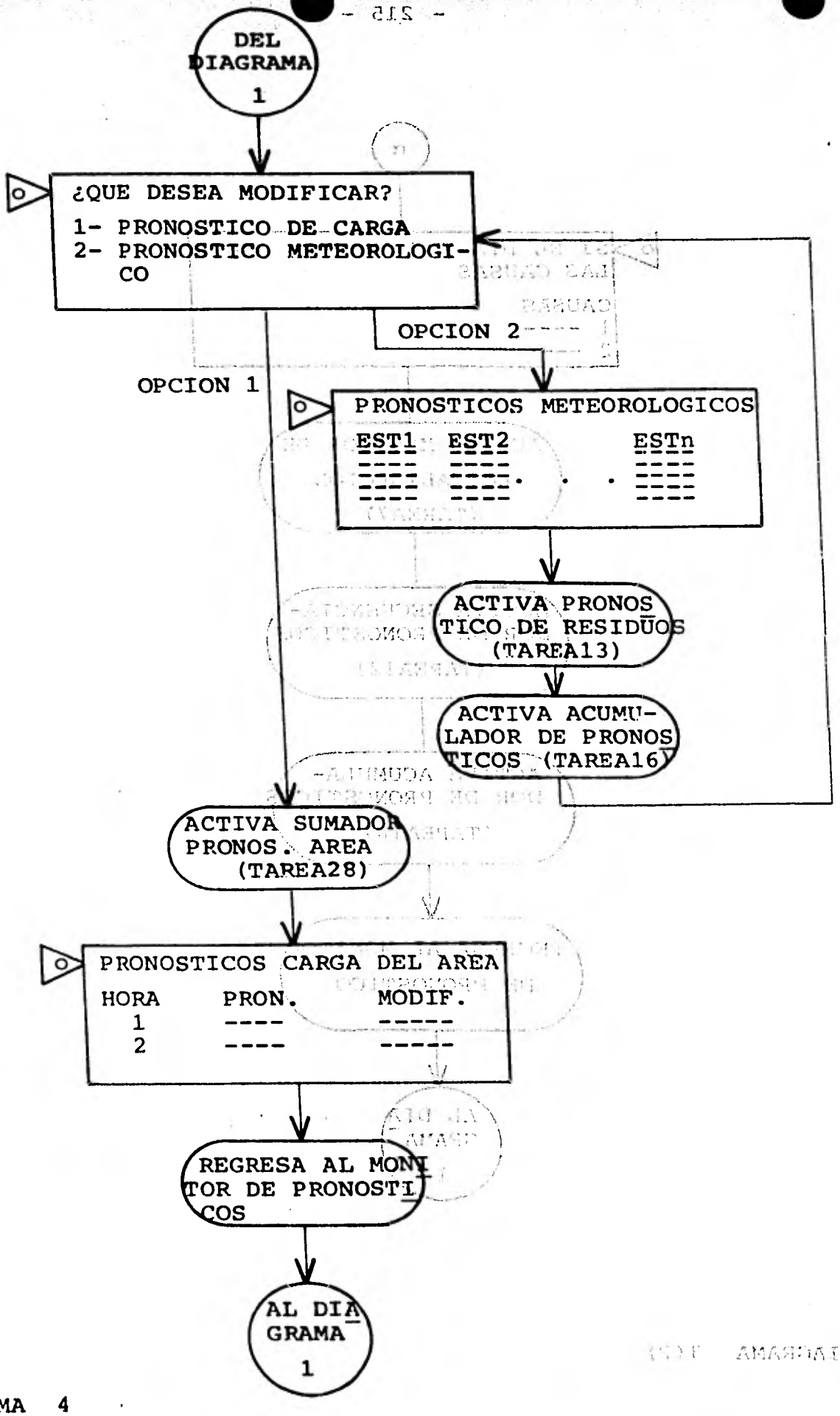

DEL DIAGRAMA 1
¿QUE DESEA MODIFICAR?
1- PRONOSTICO DE CARGA
2- PRONOSTICO METEOROLOGICO
OPCION 2
OPCION 1
PRONOSTICOS METEOROLOGICOS
EST1 EST2 ESTn
ACTIVA PRONOSTICO DE RESIDUOS (TAREA13)
ACTIVA ACUMULADOR DE PRONOSTICOS (TAREA16)
ACTIVA SUMADOR PRONOS. AREA (TAREA28)
PRONOSTICOS CARGA DEL AREA
HORA PRON. MODIF.
1 ---- -----
2 ---- -----
REGRESA AL MONITOR DE PRONOSTICOS
AL DIAGRAMA 1

DIAGRAMA 4

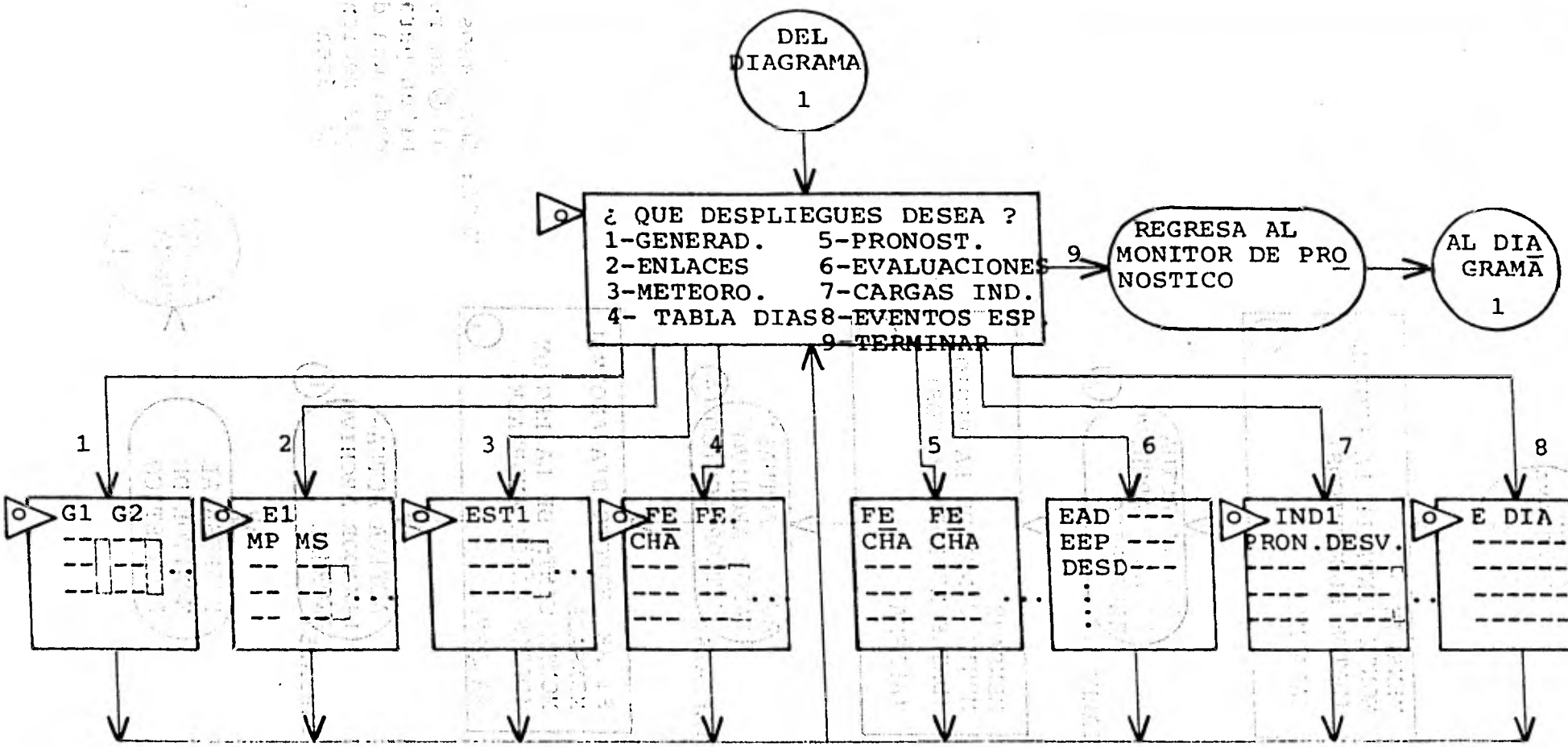
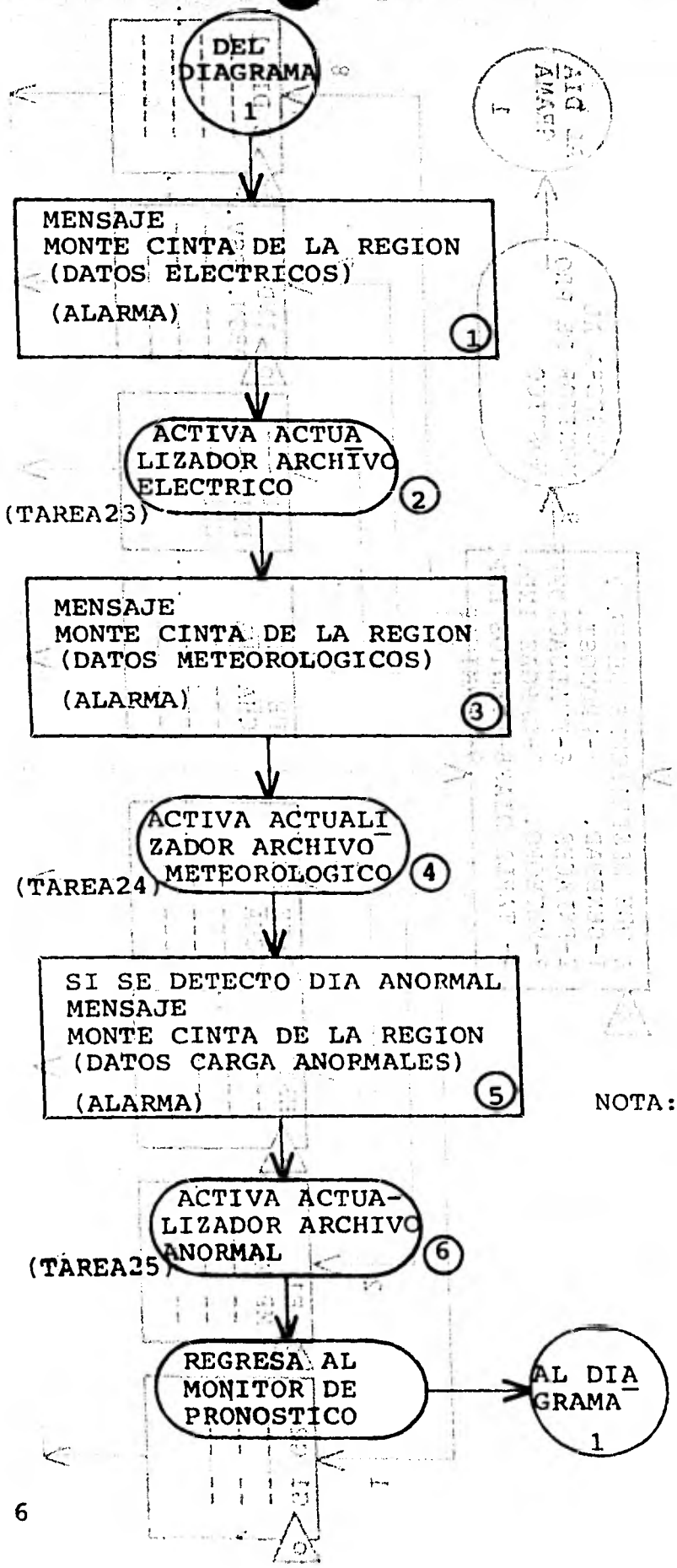

DEL DIAGRAMA 1
¿ QUE DESPLIEGUES DESEA ?
1-GENERAD. 5-PRONOST.
2-ENLACES 6-EVALUACIONES
3-METEORO. 7-CARGAS IND.
4- TABLA DIAS 8-EVENTOS ESP.
9-TERMINAR
9
REGRESA AL MONITOR DE PRONOSTICO
AL DIAGRAMA 1
1
2
3
4
5
6
7
8
G1 G2
E1
MP MS
EST1
FE FE.
CHA
FE FE
CHA CHA
EAD ---
EEP ---
DESD---
IND1
PRON.DESV.
E DIA 1

DIAGRAMA 5



NOTA: Los pasos ① al ⑥ se repiten para cada una de las regiones del área

DIAGRAMA 6

3.1.1.3 Diagrama de Interacción Funcional de Tareas, Bases de Datos y Archivos de Usuario.

El programa Pronóstico de carga, residirá en los Centros de Control de Areas. En el centro de Control Nacional, habrá un programa mucho más sencillo.

A continuación se muestran los diagramas de interacción funcional para ambos casos.

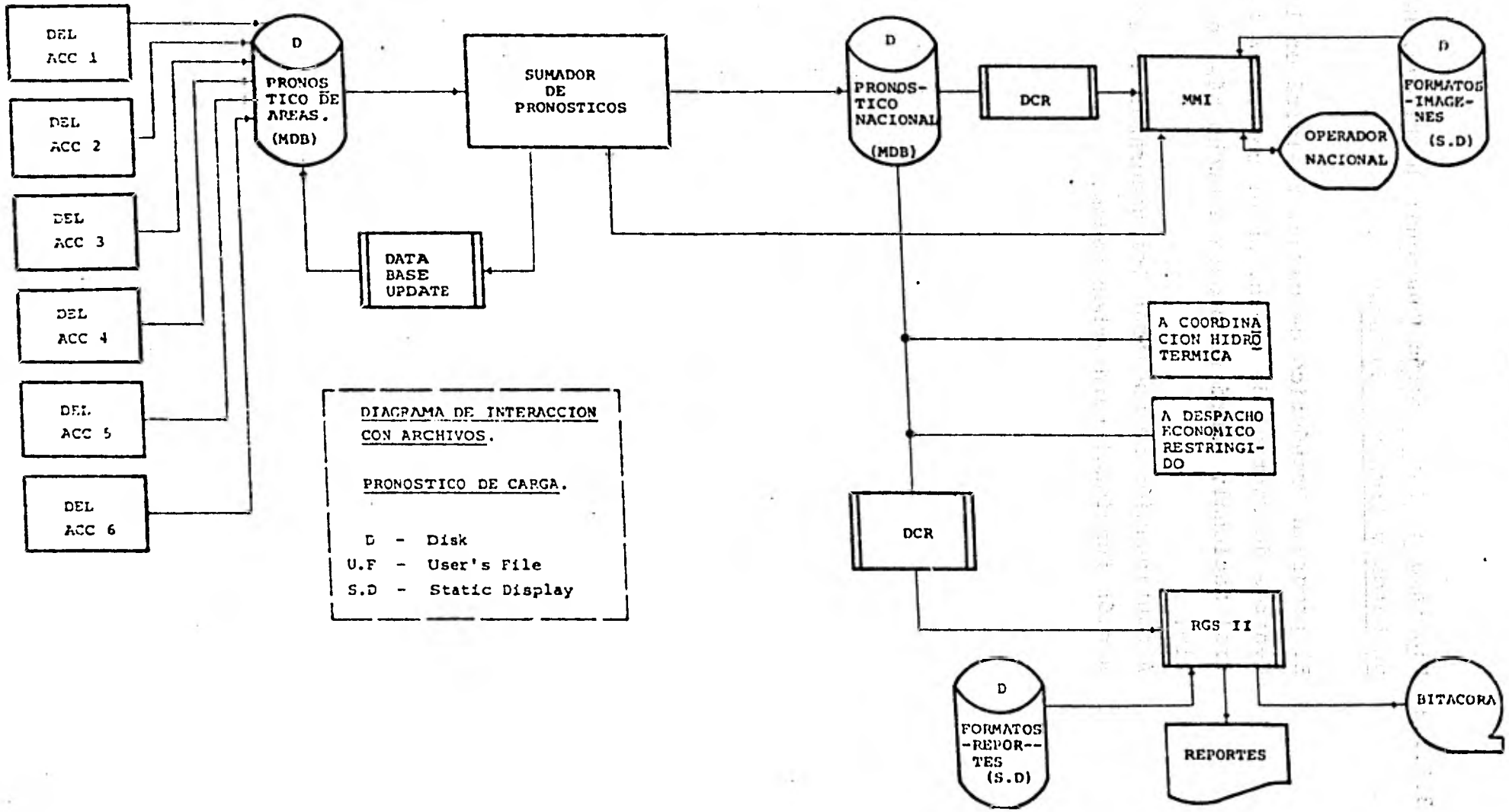


DIAGRAMA DE INTERACCION CON ARCHIVOS.

PRONOSTICO DE CARGA.

D - Disk
U.F - User's File
S.D - Static Display

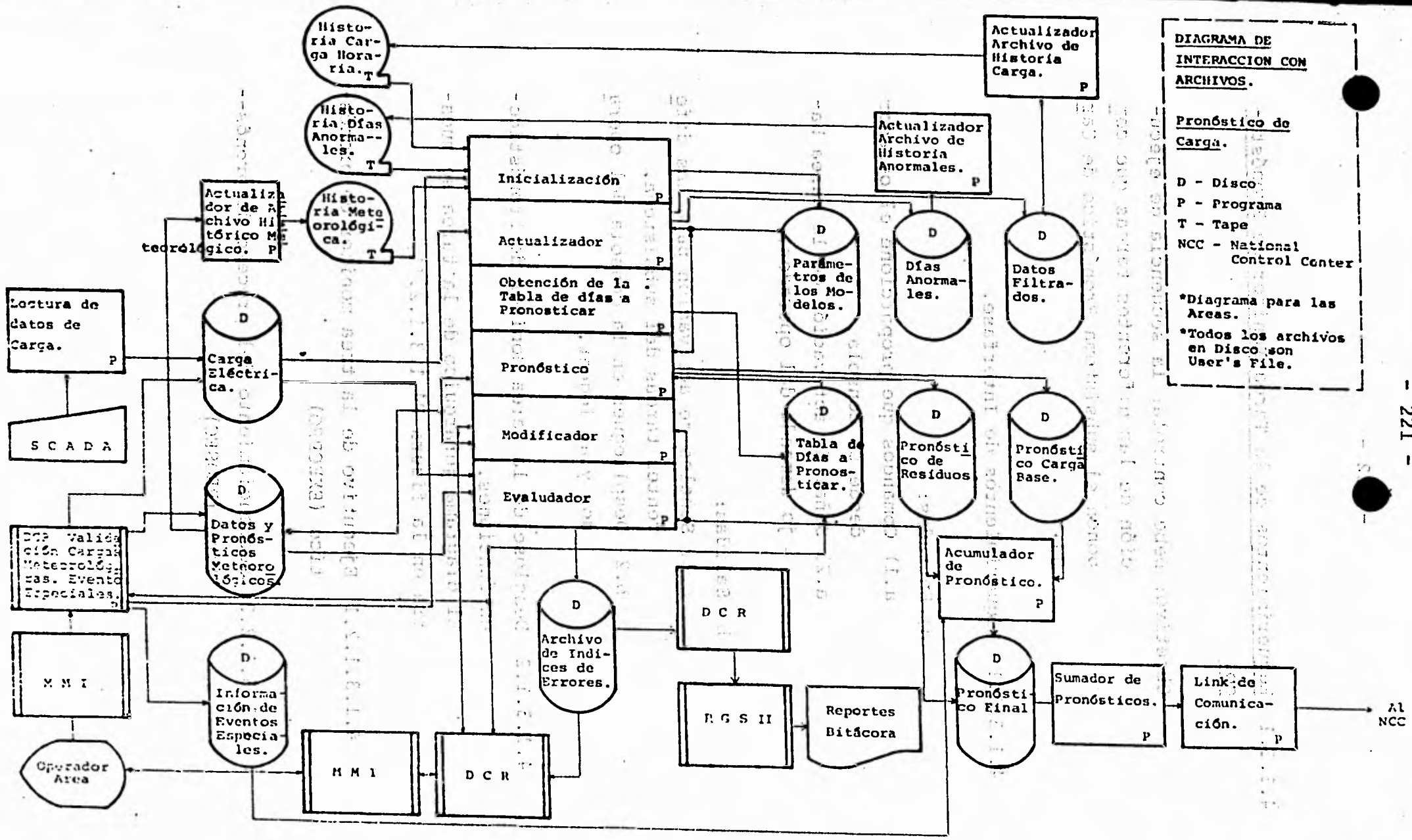


DIAGRAMA DE INTERACCION CON ARCHIVOS.
Pronóstico de Carga.
D - Disco
P - Programa
T - Tape
NCC - National Control Center
*Diagrama para las Areas.
*Todos los archivos en Disco son User's File.
Historia Carga Horaria. T
Historia Días Anormales. T
Historia Meteorológica. T
Actualizador de Archivo Histórico Meteorológico. P
Actualizador Archivo de Historia Carga. P
Actualizador Archivo de Historia Anormales. P
Inicialización P
Actualizador P
Obtención de la Tabla de días a Pronosticar P
Pronóstico P
Modificador P
Evaludador P
Parámetros de los Modelos.
Días Anormales.
Datos Filtrados.
Tabla de Días a Pronosticar.
Pronóstico de Residuos
Pronóstico Carga Base.
Lectura de datos de Carga. P
SCADA
Carga Eléctrica.
Datos y Pronósticos Meteorológicos.
Validación Cargas Meteorológicas. Eventos Especiales. P
MMI
Operador Area
Información de Eventos Especiales.
MMI
DCR
Archivo de Indices de Errores.
DCR
DGS II
Reportes Bitácora
Acumulador de Pronóstico. P
Pronóstico Final
Sumador de Pronósticos. P
Link de Comunicación. P
Al NCC

4.1.3.1 Requerimientos de la Tarea Monitor de Pronóstico.

Objetivo: Debe controlar la secuencia de ejecución de las diferentes tareas que componen el subsistema Pronóstico de Carga.

4.1.3.1.1 Requerimientos de Interfase.

a) Entradas:

a.1) Comandos que proporcione el oerador desde su consola.

a.2) Señal de activación de la tarea bajo demanda del operador.

b) Salidas:

b.1) Señales de activación para las diferentes tareas del subsistema.

b.2) Despliegues en la consola del operador y al logger.

4.1.3.1.2 Desglose de la tarea Monitor de Pronóstico en funciones.

El diagrama jerárquico de la tarea se muestra en la figura F4.1.3.1.2.(1).

4.1.3.1.2.1 Ejecutivo de la trea monitor de pronóstico (EXECPRO)

4.1.3.1.2.2 Secuenciamiento del proceso de pronóstico (PROSEC)

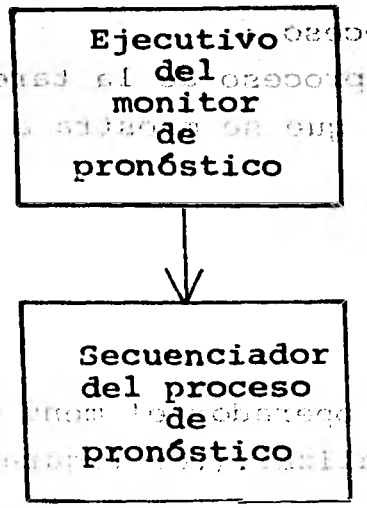


Diagrama jerárquico de estructuras de la tarea monitor de pronóstico

F4.1.3.1.2.(1)

4.1.3.1.3 Definición de funciones

4.1.3.1.3.1 EXECPRO

Tiene por finalidad controlar la secuencia de ejecuciones de las diferentes tareas del subsistema, dependiendo de las selecciones que realice el operador.

a) Entradas:

a.1) Señal de activación de la tarea proporcionado por el operador.

a.2) Señal de las tareas indicando el fin de ejecución de ellas.

a.3) Comandos que proporcione el operador para la selección de las actividades.

b) Proceso

El proceso de la tarea se muestra en el LDP que se muestra a continuación.

1. INICIO

2. REPITE

A. Despliega al operador el menú de actividades --- que puede realizar. (ver figura F4.1.3.1.3.1.(1))

B. Obtener la selección del operador (SEL)

C. SI (SEL es una selección válida) ENTONCES

1. CASO (a,b,c,d) SEL

a. Activa la tarea Monitor de Inicializa--- ción y desactiva la tarea actual hasta-- que termine su ejecución la tarea activa da.

b. LLAMA a PROSEC

c. Activa la tarea Modificador de Pronóstico y desactiva la tarea actual hasta que termine la ejecución de la tarea activada.

d. Activa la tarea Despliegues y validación de datos y desactiva la tarea actual hasta que termine su ejecución la tarea activada.

2. FIN-CASO

3. SI (SEL = actividad 6) ENTONCES

a. Activa la tarea Monitor de actualiza--- ción de los archivos históricos y desac tiva la tarea actual hasta que termine- su ejecución la tarea activada.

4. FINSI

D. SINO

1. Despliega mensaje de error al operador. "SELECCION INVALIDA, SELECCIONE DE NUEVO"

E. FINSI

3. HASTA (SEL = actividad 5)

4. FIN

c) Salidas:

c.1) Señal de activación para las diferen-- tes tareas.

c.2) Despliegues a la consola del operador.

MENU DE ACTIVIDADES

1 - INICIALIZACION DE LOS MODELOS DE PRONOSTICO

2 - PRONOSTICO

3 - MODIFICACION DE LOS PRONOSTICOS

4 - DESPLIEGUES Y VALIDACIONES DE DATOS

5 - NINGUNO

*** NECESARIO REALIZAR LA ACTUALIZACION DE LOS
ARCHIVOS HISTORICOS DE INFORMACION.
TECLEE SELECCION 6 PARA REALIZARLO ***
(este mensaje aparece cuando
sea necesario)

TECLEE AQUI SU SELECCION |____| ←(input)

Propuesta del despliegue
de menú de actividades.
F4.1.3.1.3.1.(1)

4.1.3.1.3.2 PROSEC

Para realizar el pronóntico de carga en forma correcta, es necesario activar en una secuencia lógica diferentes tareas del subsistema. El objetivo de esta función es, controlar dicha secuencia.

a) Entradas:

a.1) Señales de terminación de la ejecución de las tareas involucradas en la secuencia.

b) Proceso:

El proceso se muestra en el siguiente LDP.

1. INICIO

2. Activa tarea de construcción de tabla de días a pronosticar y desactiva la tarea actual hasta que termine la ejecución de dicha tarea.

3. Activa la tarea de despliegues y validación de datos y desactiva la tarea actual hasta que termine la ejecución de dicha tarea.

4. Activa el monitor de actualización y desactiva la tarea actual hasta que termine la ejecución de dicha tarea.

5. Activa el monitor de evaluación y desactiva la tarea actual hasta que termine la ejecución de dicha tarea

6. Activa la tarea secuenciador de pronósticos y desactiva la tarea actual hasta que termine de ejecutar la tarea activada.

7. FIN

c) Salidas:

c.1) Señales de activación para las diferentes tareas involucradas en el proceso de pronóstico.

4.1.3.1.4 Requerimientos de almacenamiento de información para la tarea Monitor de Pronóstico.

Todas las salidas y entradas a esta tarea, son señales de activación programa-programa por lo que no existe requerimientos de almacenamiento de información para esta tarea.

4.1.3.1.5 Restricciones de utilización de la tarea Monitor de Pronóstico.

a) Modos de uso

Será activado bajo demanda del operador por lo que no existe restricción en el modo de uso de esta tarea.

b) Precisión de cálculo

Esta tarea no realiza cálculo alguno por lo que no existe restricción en la precisión de cálculo.

c) Frecuencia de ejecución

Se debe ejecutar por lo menos una vez al día.

d) Tiempo de ejecución

Como es un programa que se ejecutará bajo demanda, no existe restricción en el tiempo de ejecución.

4.1.3.1.6 Restricciones de operación de la tarea Monitor de pronóstico.

a) Equipos de interfase

Básicamente es necesario contar con una pantalla de video donde puedan realizarse los despliegues.

b) Software de interfase

Es necesario contar un programa interfase Hombre-Máquina para realizar despliegues a la consola y obtener información de ella. También es necesario contar con interfase de activación programa-programa.

c) Interacción con el resto del sistema

No existe interacción alguna con otros subsistemas.

d) Interfase Hombre-Máquina

d.1) Desplegados

Sólo se realizará un despliegue de menú de actividades.

d.2) Reportes

Ninguno

d.3) Alarmas y mensajes

Se presentan mensajes de error e informativos como los mostrados en la fig. F4.1.3.1.3.1.(1)

4.1.3.1.7 Restricciones de desarrollo de la tarea --- Monitor de Pronóstico

a) Equipo de cómputo

Es necesario contar un equipo con interfase hombre-máquina para realizar des--- pliegues en la consola y obtener datos - desde ella. Asímismo debe tener posibilidades de activar un programa desde otra, y con capacidades que permitan el manejo de alarmas y mensajes. Un sistema que lo cumple es el sitema M9200 de Harris.

b) Sistema operativo

Debe permitir la interacción entre pro-- gramas además de poder conversar de alguna manera con el operador. Si el equipo- de cómputo necesario es el sistema M9200 de harris, será conveniente utilizar el- sistema operativo VULCAN RTOS (Real Time- Operating System)

c) Lenguajes

Utilizar FORTRAN 77

d) Programas disponibles

Ninguno

4.1.3.1.8 Requerimientos de mantenimientos y modificación para la tarea Monitor de Pronóstico.

Ninguno.

IV CONCLUSIONES.

En el presente trabajo se realiza un análisis de una metodología para el desarrollo de proyectos de programación (capítulo II), y una aplicación de dicha metodología a una problema real(capítulo III).

Es imperativo que en los organismos que realicen proyectos de programación,como lo son los centros de cómputo y de procesamiento de información en las empresas, adopten una metodología y estandaricen la documentación para producción de código.

La implantación de un método formal para realizar proyectos de programación trae beneficios como :

- entrega de productos a tiempo
- evitar sobre-costos en el desarrrollo del producto
- obtención de productos eficientes y confiables
- inmunidad al movimiento de personal
- etc.

Cabe mencionar que la metodología descrita en el presente trabajo, no es una solución infalible a los problemas existentes en el desarrollo de programación, pero es una buena base de la cual podemos partir para solucionar diferentes problemas que puedan surgir en el período de vida de un proyecto de programación.

También es necesario indicar que estas técnicas deben tener una mayor divulgación para realizar modificaciones y mejoras para obtener productos con mayor calidad y así satisfacer las necesidades de una sociedad en constante crecimiento.

V BIBLIOGRAFIA

1) Richardson G, Butler Ch. and Tomilson J.
"A Primer on Structured Program Design"
New York, Petrocelli Book, 1980.

2) Ralston A.; Editor.
"Encyclopedia of Computer Science"
New York, Van Nostrand Reinhold Co., 1976.

3) Cougar J Daniel.
"Evolution of Business System Analysis Techniques"
Computing Surveys, pp. 167-198, September, 1973.

4) Bohm C and Jacopini G.
"Flow Diagram, Tuning Machine and Languages with Two Formation Rules"
Communications of the ACM, pp. 366-371, May, 1966.

5) Dahl O.J, Dijkstra E.W and Hoare C.A.R.
"Structured Programming"
New York, Academic Press, 1972.

6) Mills H.D.
"How to Write Correct Programs and Know It"
Gaitherburg, Md.; IBM Corp. FSC73-5008, 1973.

7) Stevens P, Myers G and Constantine L.
"Structured Design"
IBM Systems Journal, Vol. 13, No. 2, pp. 115-139
1974.

8) Orr, Kenneth T.
"Structured Systems Design"
Kansas City, Langston-Kitch, 1978.

9) Van Leer P.
"Top-Down Development Using a PDL"
IBM Systems Journal, Vol 15, No. 2, pp. 155-170
1976.

10) McGowan, Clement L and Kelly John R.
"Top-Down Structured Programming Techniques"
New York, Petrocelli/Charter, 1975.

11) Katzan Harry.
"Systems Design and Documentation: An Introducction to the HIPO Method"
New York, Van Nostrand Reinhold Co., 1976.

12) Dahl O.J., Dijkstra E.W and Hoare C.A.R.
"Structured Programming"
New York, Academic Press, 1972.

13) Rogers J.G.
"Structured Programming for Virtual Storage Systems"
IBM Systems Journal, Vol. 14, No. 4, pp. 385-406,
1975.

14) Buckley F.
"Software Quality Assurance"
New York, IEEE Standard Seminar, 1981.

15) Yourdon E.
"How to Manage Structured Programming"
New York, Yourdon Inc., 1976.

16) Warnier J.D.
"Logical Construction of Programs"
Leiden, Steinfert Kroese, 1974.

17) De Marco T.
"Structured Analysis and Systems Specification"
New Jersey, Prentice Hall, 1977.

18) Tausworthe R.C.
"Standardized Development of Computer Software"
New Jersey, Prentice Hall, 1977.

19) USAF Rome Air Development Center.
"Structured Programming Series"
Vol, 1 - 15, July, 1977.

20) Torres A y Mandujano R.
"Normas de Codificación en FORTRAN 77 para la División de Sistemas de Potencia del Instituto De Investigaciones Eléctricas"
México D.F., a 1981.

21) Coyote H, Mandujano R y Valadez V.
"Checador Estático de Código"
México D.F., División de Sistemas de Potencia, Instituto de Investigaciones Eléctricas, 1981.

22) Hansen, P.B.
"Testing a Multiprogramming System"
Software-Practice and Experience,
3:2, pp. 145, April, 1973.

23) Boehm, B., et, al.,
"Characteristics of Software Quality, TRW Series on Software Technology"
New York: North-Holland Publishers, Inc., 1978.

24) Brooks, F.P., Jr.
"The Mythical Man-Month"
Reading Massachusetts: Addison-Wesley Publishing, 1975.

25) Foster, R.A.
"Introduction to Software Quality Assurance"
San Jose, California: R.A. Foster, 1975.

26) Mier, M. y Mandujano, R.
"Normas para un Lenguaje para Diseñar Programas(LDP)"
México,D.F.: División de Sistemas de Potencia,
Instituto de Investigaciones Eléctricas, 1981.

27) Wooldridge, S.
"Systems and Programming Standards"
New York: Petrocelli/Charter, 1977.

28) Fagen, M.E.
"Design and Code Inspections to Reduce Errors
in Program Development"
IBM System Journal, No. 3, pp. 182-211, 1976.

29) Harris Corporation.
A Natinal Energy Control Center for the
Republic of Mexico, 1980.

30) Documento de trabajo.
Comisión Federal de Electricidad-Instituto de
Investigaciones Eléctricas, Anexos Técnicos,
Programación Avanzada para el Centro Nacional de
Control de Energía.
México, D.F., 1980.

31) Romano R.; Quintana V.; López R. and Valadez V.
Constrained Economic Dispatch of Mutli-area Systems
using Dantzing-Wolfe Decomposition Principle.
Minneapolis: IEEE Power System Engineering Committee,
July, 1980.

32) Arriola E.; Mier, M. y López R.
"Un Método Eficiente para Eliminar Errores Anormales de Medición en el Proceso de Estimación de Estado"
México, D.F. : IEEE, Julio, 1978.

33) Arriola, E.; Mier M. y López R.
"Modificación al Método de Estimación de Estado en la American Electric Power Co. para el Procesamiento de Nodos de Inyección Cero"
México, D.F. : IEEE, Julio, 1978.

34) Quintana, V.H.; Nieva, R.; Santos, M.; Martínez, J. and Gerez, V.
"Automatic Tie-Line Power Control of Mutli-Area Power Systems"
Alburquerque, N.M.: 24th. Midwest Symposium on Circuits and Systems, June, 1981.

35) Romano, R.; Quintana, V. y Valadez, V.
"Un algoritmo de Programación Lineal para el Despacho Económico Restringido"
México, D.F. : IEEE, 1980.

36) Mier,M.; Inoue, M.
"Informe de los Procedimientos de Recopilación de Información Eléctrica y Análisis Estadístico de la Información Eléctrica y Meteorológica"
México D.F.; División de Sistemas de Potencia, Instituto de Investigaciones Eléctricas, 1978.

37) Galiana, F.D. and Schweppe, F.C.
"A Weather Dependent Probabilistic Model for Short Term Load Forecasting"
IEEE, C72, 171-2.

38) Lijesen, D.P; Rosing, J. and Lunberger, D.G.
"Adaptive Forecasting of Hourly Load Based on
Load Measurements and Weather Information"
New York; IEEE Winter Power Meeting, 1971.

39) Gupta, P.C. and Yamada, K.
"Adaptive Short Term Forecasting of Hourly Load
Using Weather Information"
IEEE, T72, 170-4.

40) Sharma, K.L. and Mahalanabis, A.K.
"Recursive Short Term Load Forecasting Algorithm"
Proc. IEEE, Vol. 121, No. 1, Jan. 1974.

41) Christiaanse, W.R.
"Short Term Load Forecasting Using General Exponential
Smoothing"
IEEE PAS-90, No. 2, March/April 1971.

42) Bonissone, P.; López, R.; Mier, M. y Orvañanos, E.
"Patrones de Reconocimiento en la Predicción de
Carga Eléctrica"
Revista de Ingeniería, Septiembre 1976.

43) Health Sciences Computing Facility
"Biomedical Computer Programs, P-Series(BMDP77)"
Department of Biomathematics, School of Medicine,
UCLA, Los Angeles, 1977.

44) Torres, A. y Czitrom, V.
"Métodos para la Solución de Problemas con Computadora
Digital"
México D.F., Representaciones y Servicios de Ingenie-
ría S.A., 1980.

44) Mier,M; Ruiz,M; Inoue,M y Castellanos,M.
"Estimación de Consumos Horarios para días Anormales"
IEEE, Sección México, Guadalajara,Jal., Octubre 1981

45) Gerez,V; Mier,M e Inoue,M.
"Ingeniería de Programación en la División de Sistemas de Potencia del Instituto de Investigaciones -- Eléctricas"
Instituto de Investigaciones Eléctricas, División - de Sistemas de Potencia, México D.F., Julio 1981

APENDICE A

PROGRAMACION ESTRUCTURADA

1. INTRODUCCION

OBJETIVO

Desarrollar programas más eficientes (legibilidad, tiempo y memoria), que sean fáciles de mantener y modificar.

TECNICAS A EMPLEAR

- Programación estructurada
- Desarrollo de programas de arriba hacia abajo (TOP-DOWN)
- Diagramas de árbol
- Diagramas HIPO (Entrada-Proceso-Salida jerarquizada)
- Diagramas de bloque
- Documentación usuario
- Corrección y prueba estructurada de los programas
- Bibliotecas (módulos útiles para diferentes programas)
- Pláticas estructuradas para la revisión de los programas

Documentación del programa

BENEFICIOS ESPERADOS

- Economía en tiempo
- Reducción de costos
- Mayor confiabilidad
- Mayor eficiencia
- Mayor productividad del grupo de trabajo
- Fácil detección de errores
- Fácil modificación y actualización (mantenimiento) de los programas

2. PROGRAMACION ESTRUCTURADA

2.1 Propósito.

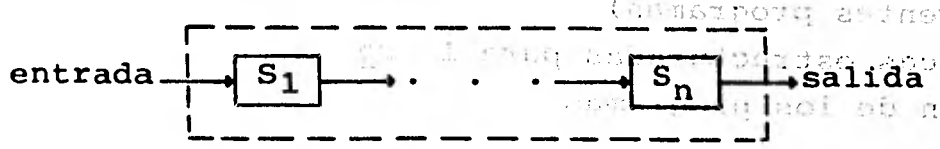
Producir programas :

- altamente confiables
- fáciles de entender
- fáciles de probar y corregir
- fáciles de mantener
- más eficientes a nivel memoria, tiempo y usuario

2.1 Teoría.

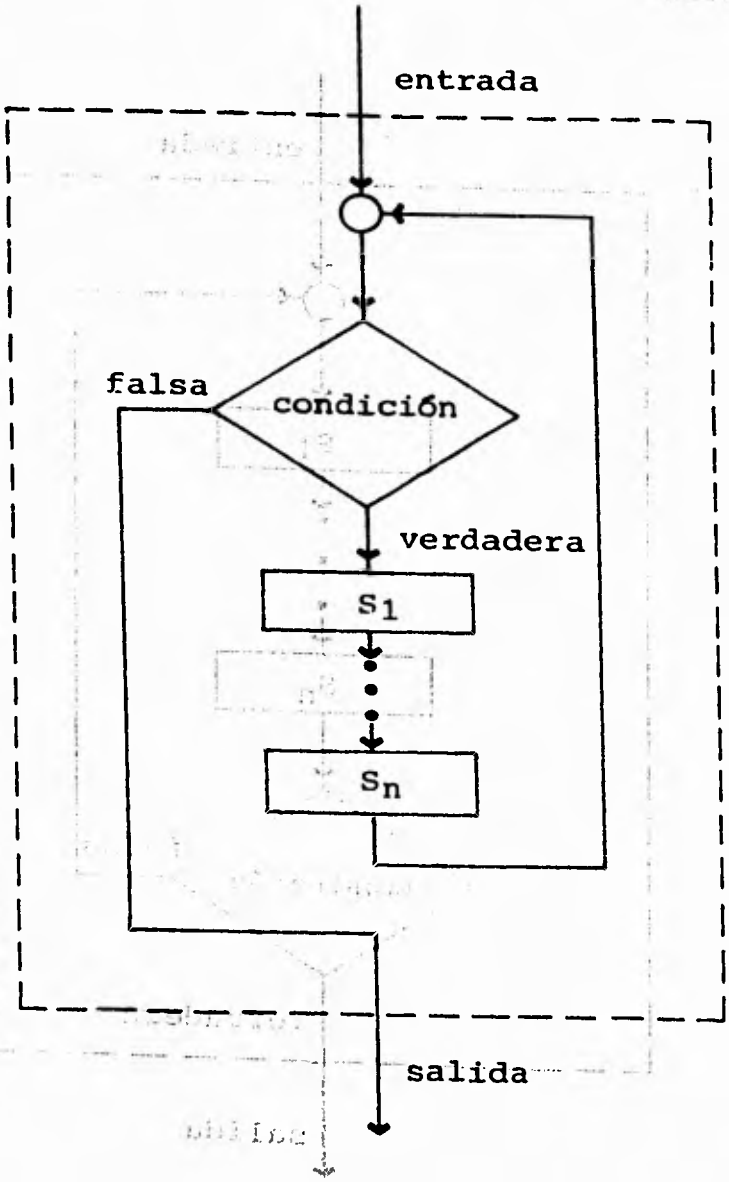
Cualquier programa puede elaborarse con las siguientes -- tres unidades lógicas (de una entrada y una salida) :

a) Secuencia

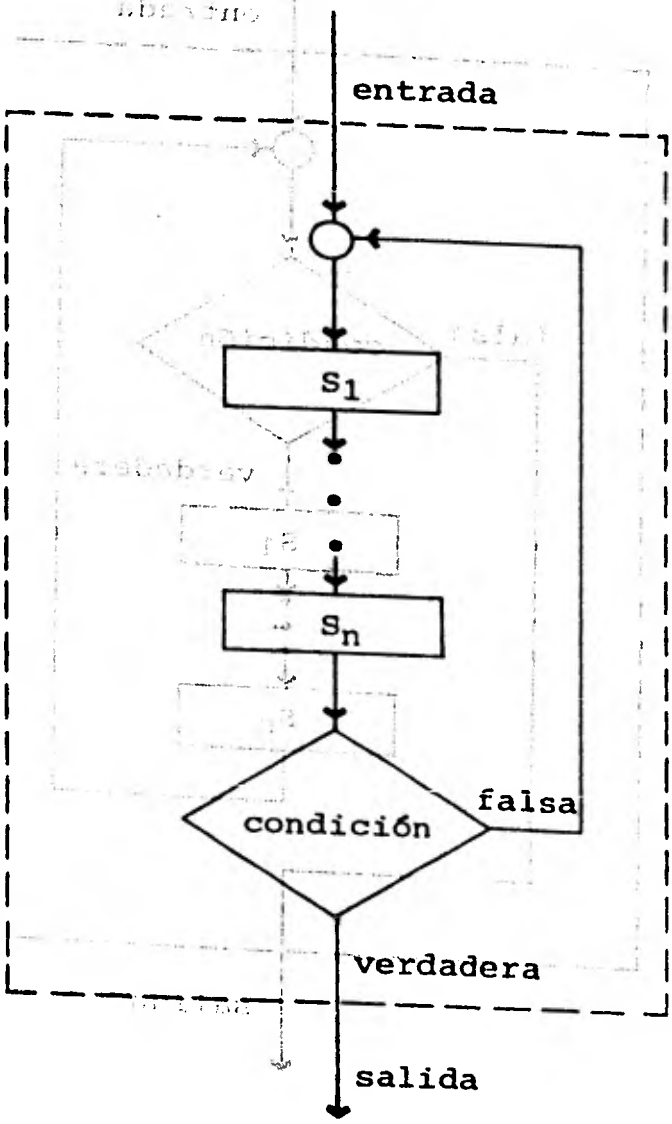


b) Repetición

- DO WHILE

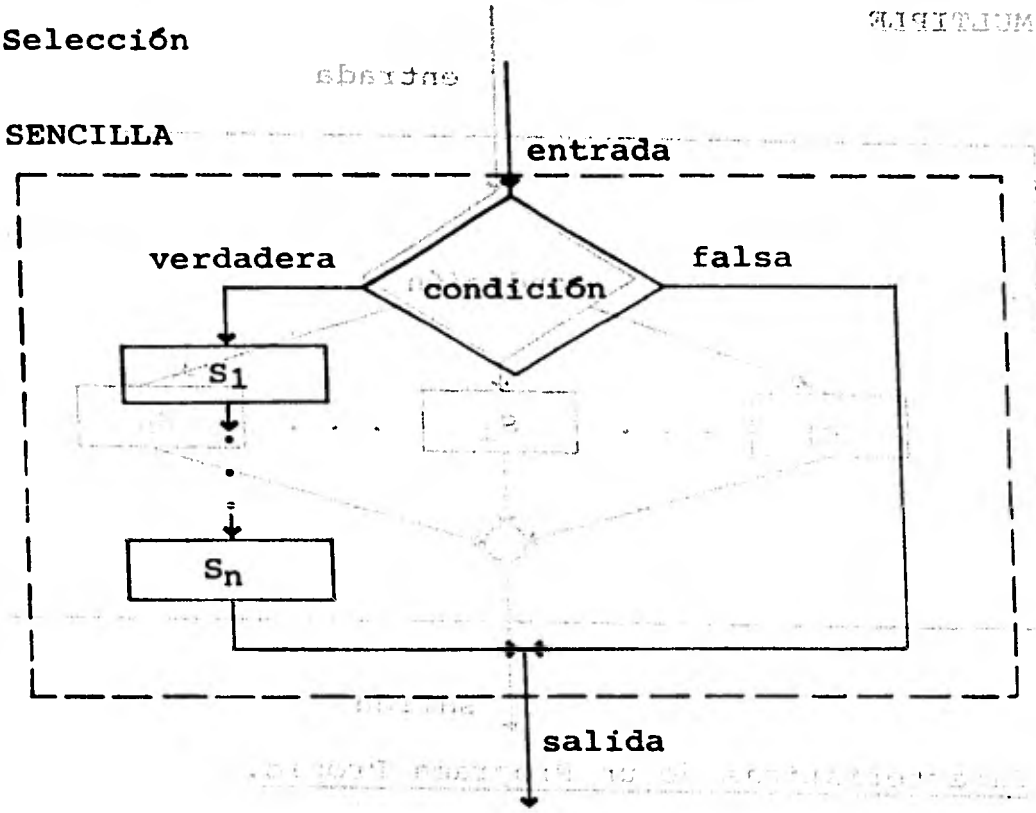


- DO UNTIL

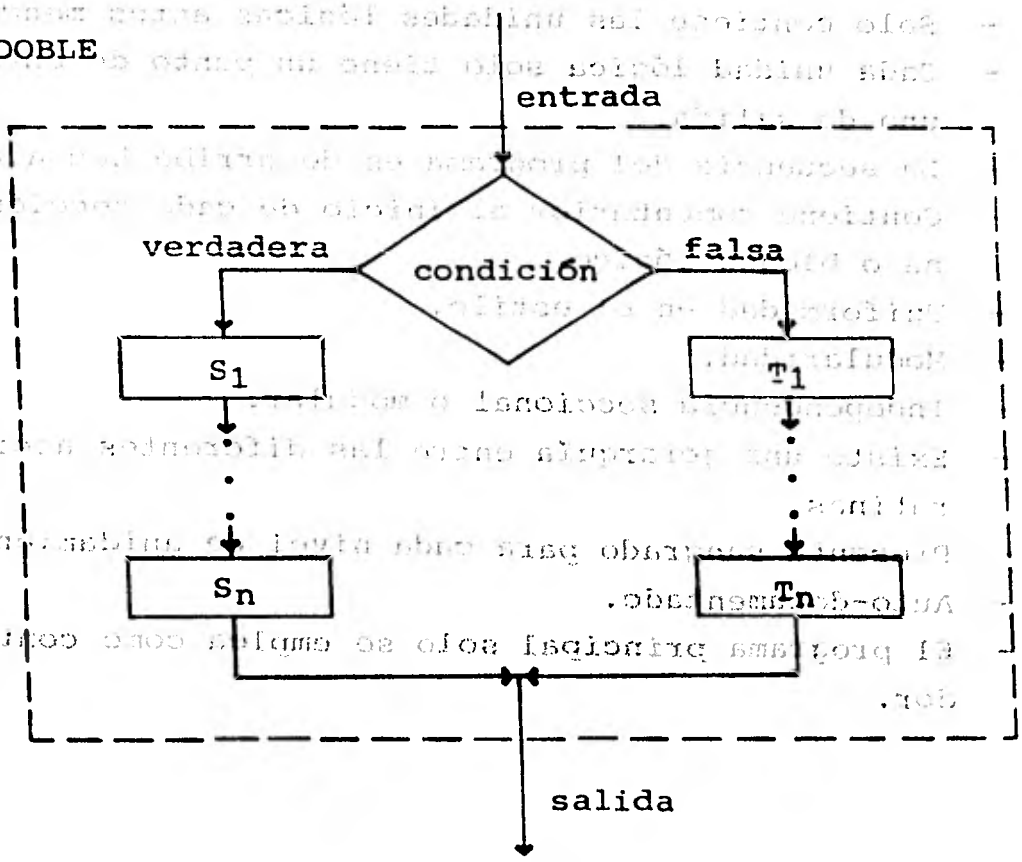


c) Selección

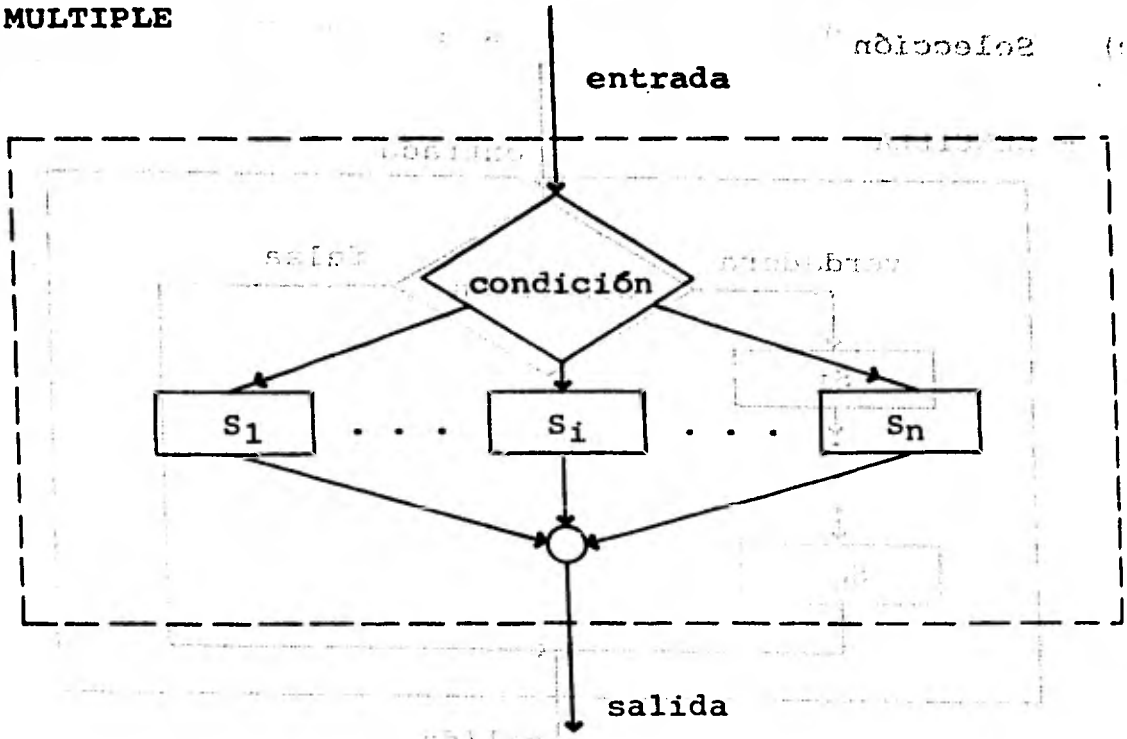
- SENCILLA



- DOBLE



- MULTIPLE



2.3 Características de un Programa Propio.

- Solo contiene las unidades lógicas antes mencionadas.
- Cada unidad lógica solo tiene un punto de entrada y uno de salida.
- La secuencia del programa es de arriba hacia abajo.
- Contiene comentarios al inicio de cada sección, rutina o bloque lógico.
- Uniformidad en el estilo.
- Modularidad.
- Independencia seccional o modular.
- Existe una jerarquía entre las diferentes acciones y rutinas.
- Presenta sangrado para cada nivel de anidamiento.
- Auto-documentado.
- El programa principal solo se emplea como controlador.

2.4 Recomendaciones para programas elaborados en FORTRAN .

a) Unidades lógicas en FORTRAN.

- Secuencia

S_1

S_2

.
.
.

S_n

- Repetición

DO WHILE

```
210  IF  (Condición.NE.TRUE) GO TO 220
                 ==========
                 ==========
                 ==========
                 GO TO 210

220  CONTINUE
```

DO UNTIL

```
      DO 200 I = ALFA, BETA
                 ____________
                 ____________
                 ____________
                 ____________
                 ____________
                 ____________
                 ____________

200   CONTINUE


200   CONTINUE
         ____________
         ____________
         ____________
         ____________
         ____________
         ____________
         ____________
      IF (condición.EQ.FALSE) GO TO 200
```

se ejecuta una vez
a pesar de que
BETA < ALFA

- Selección

SENCILLA

```
IF ( condición.EQ. TRUE ) ____________
```

DOBLE

```
     IF (condición.not.true) GO TO 210
                    ==========
                    ==========
                    ==========
                    ==========
                    GO TO 220
210  CONTINUE
                    ==========
                    ==========
                    ==========
                    ==========
220  CONTINUE
```

MULTIPLE

i) asignado

```
     ASSIGN m TO SWITCH
     ==========
     ==========
     ==========
     ==========
     GO TO switch,(α1,α2,α3)  <-- posibles valores de switch

                    GO TO 200
α1                  ==========
                    ==========
                    GO TO 200
α2                  ==========
                    ==========
                    GO TO 200
α3                  ==========
                    ==========
200  CONTINUE
```

ii) computado

```
      GO TO (α1,α2), e
α1          GO TO 200
α2          GO TO 200
200   CONTINUE
      e = 1 ó 2
```

b) Cada unidad lógica debe tener una entrada y una salida, es decir, evitar el uso de GO TO.

c) Emplear sangrado para cada nivel de anidamiento. En cada -- nivel de anidamiento se da un nuevo sangrado a la derecha.- Todas las unidades lógicas con igual nivel de anidamiento - tienen el mismo margen.

Ejemplo:

```
c ----------------------------
c              LEER Y ESCRIBIR
C - - - - - - - - - - - - - - -
          DO 10  I = 1 , N
             READ . . .
             IF (P.E.Q.0) F = 1
       10 CONTINUE
```

d) Autodocumentar el programa mediante comentarios claros y concisos.

i. Comentar el propósito de cada unidad lógica o de una sección indicando, si es necesario, la(s) variable(s) involucrada(s).

ii. Al inicio de un programa o rutina porporcionar :

- Objeto.
- Variables de entrada.
- Variables de salida.
- Subrutinas empleadas.
- Variables locales empleadas.
- Limitaciones y parámetros globales que modifica.
- Nombre del autor.
- Fecha de la última revisión.
- Nombre del revisor.
- Indicar técnica o método de empleado.

e) Modularidad: escribir rutinas que realicen acciones específicas y procurar que cada módulo no exceda de dos páginas.

f) Emplear el programa principal exclusivamente como controlador de las rutinas y de la secuencia.

g) Jerarquizar los módulos.

h) El anidamiento de unidades lógicas no debe exceder 4 niveles.

i) Emplear nombres de variables que den información sobre el contenido.

j) Declarar en la parte superior de cada módulo todos aquellos parámetros que sean constantes, es decir, evitar el empleo de números mágicos dentro del código, Vgr. : - PI para π
- EPS para criterio de convergencia épsilon

k) Crear tipos variables propios, para manejar la información en lugar de emplear números, Vgr. :

```
   INTEGER LU,MA,MIER,JUE,VIE,SA,DO
   DATA LU,MA,MIER,JUE,VIE,SA,DO/1,2,3,4,5,6.7/
   ================================
   ================================
   DO 20 I = LU, DO
   ================================
   ================================
20 CONTINUE
```

l) Emplear etiquetas en orden numérico ascendente.

m) Localizar los formatos al final de cada módulo.

n) Unformidad en el estilo.

o) Evitar el uso innecesario de GO TO's e IF's.

p) Definir valores constantes mediante la proposición DATA.

q) Pasar parámetros globales a través de COMMON.

r) Hacer programas eficientes. El factor humano tiene preferencia sobfe la máquina, por lo que :

i) claridad y legibilidad tienen preferencia sobre memoria y tiempo de ejecución.

s) Si se requiere de gran cantidad de memoria, tratar de emplear archivos en lugar de arreglos para guardar y procesar la información.

t) Sugerencias para reducir el tiempo de ejecución.

i. Las operaciones que más tiempo consumen (en orden creciente) son :

- multiplicación
- división
- exponenciación
- evaluación de funciones trigonométricas

ii. Evitar cálculos dentro de las unidades de repetición. Vgr :

NO

```
      DIMENSION A(20)
      DO 10 I = 1,20
         A(I) = A(I)*B*C
10    CONTINUE
```

SI

```
      DIMENSION A(20)
      TEMP = B*C
      DO 10 I = 1,20
         A(I) = A(I)*TEMP
10    CONTINUE
```

iii. Emplear parámetro de incremento del DO en lugar de operaciones dentro de la unidad de repetición. Vgr :

NO

```
      DIMENSION A(20)
      DO 10 I = 1,20
         A (2*I) = B
10    CONTINUE
```

SI

```
      DIMENSION A(20)
      DO 10 I = 2,20,2
         A(I) = B
10    CONTINUE
```

iv. Es más eficiente utilizar arreglos unidimensiona

les que multidimensionales. Vgr :

NO	SI
INTEGER A(20,100)	INTEGER A(20,100),AT(2000)
DO 20 I = 1,20	EQUIVALENCE (A,AT)
DO 20 J = 1,100	DO 20 I = 1,2000
A(I,J) = 0	AT(I) = 0
10 CONTINUE	10 CONTINUE

v. Evitar el uso de la división cuando sea posible. Vgr :

NO	SI
A = B/2	A = B*0.5

vi. La eficiencia en la manipulación de valores numéricos decrece en la siguiente forma :

- enteros
- reales
- complejos
- doble precisión

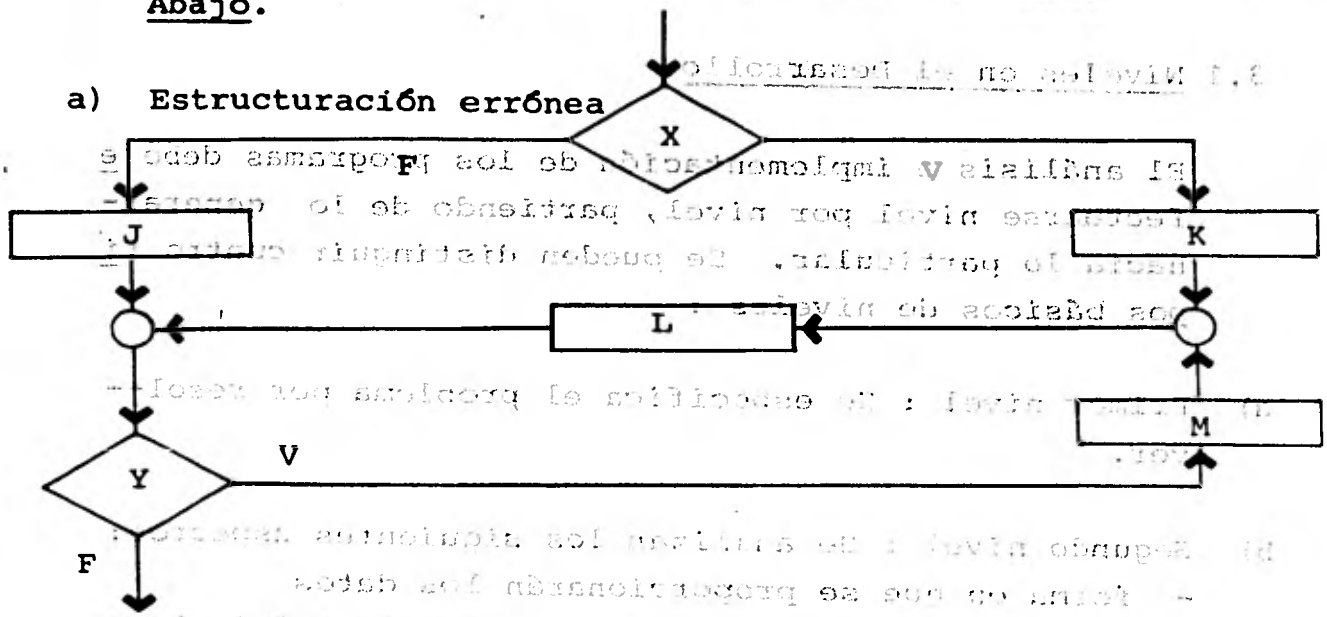
↓ menor eficiencia

vii. Evitar la conversión implícita de datos. Vgr :

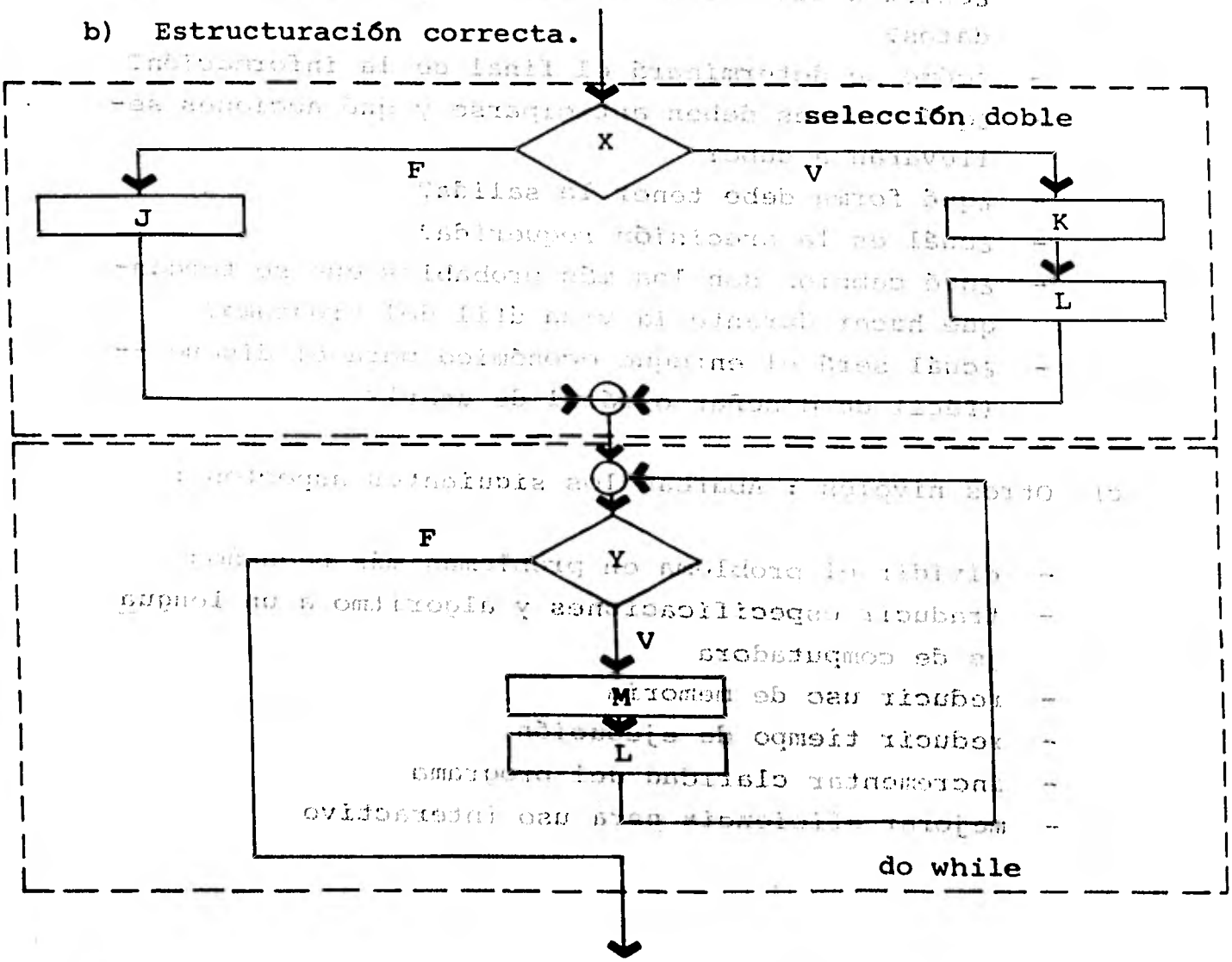
NO	SI
REAL A	REAL A
DOUBLE PRECISION D	DOUBLE PRECISION D
4 A = A + 1	4 A = A + 1.0
IF (A.EQ.0) GO TO 4	IF (A.EQ.0.0) GO TO 4
D = 3.14159	D = 3.14159 D 0

2.5 Ejemplo de un Programa Desarrollado de Arriba hacia-Abajo.

a) Estructuración errónea



b) Estructuración correcta.



3. DESARROLLO DE PROGRAMAS.

3.1 Niveles en el Desarrollo.

El análisis e implementación de los programas debe efectuarse nivel por nivel, partiendo de lo general hacia lo particular. Se pueden distinguir cuatro tipos básicos de niveles :

a) Primer nivel : Se especifica el problema por resolver.

b) Segundo nivel : Se analizan los siguientes aspectos:
- forma en que se proporcionarán los datos
- ¿existen límites en el volumen y magnitud de los datos?
- ¿cómo se determinará el final de la información?
- ¿qué errores deben anticiparse y qué acciones se llevarán a cabo?
- ¿qué forma debe tener la salida?
- ¿cuál es la precisión requerida?
- ¿qué cambios son los más probables que se tengan que hacer durante la vida útil del programa?
- ¿cuál será el enfoque económico para el diseño -- (fácil de diseñar o fácil de usar)?

c) Otros niveles : Abarcan los siguientes aspectos :

- dividir el problema en problemas más pequeños
- traducir especificaciones y algoritmo a un lenguaje de computadora
- reducir uso de memoria
- reducir tiempo de ejecución
- incrementar claridad del programa
- mejorar eficiencia para uso interactivo

d) **Ultimo nivel : Programa Completo.**

3.2 Fases en el Desarrollo.

a) **Especificar y aclarar requisitos y limitaciones del problema.**

- **Entrada: formato, orden y límites tanto en volumen como en magnitud.**
- **Salida: contenido, formato, orden y límites tanto en volumen como en magnitud.**
- **Errores: determinar probables tipos de error tanto en la entrada como en el proceso, así como las acciones a ser tomadas en dicho caso.**

b) **Diseñar estrategia de solución.**

- **cómo resolver el problema en escritorio**
- **determinar el algoritmo de solución**
- **desarrollar técnica de solución de lo general a lo particular, descomponiendo el problema en subproblemas.**
- **construir árbol de jerarquización**

c) **Seleccionar estructura de los datos.**

- **especificar variables requeridas así como su nombre**
- **especificar el tipo de las variables**
- **especificar relación entre las variables**

d) **Escribir el programa.**

- **traducir a un lenguaje de alto nivel las especificaciones del algoritmo**

3.3 Desarrollo del Programa de Arriba hacia Abajo (TOP-DOWN).

3.3.1 Propósito.

El objetivo de desarrollar los programas de arriba-hacia abajo o de afuera hacia adentro es :

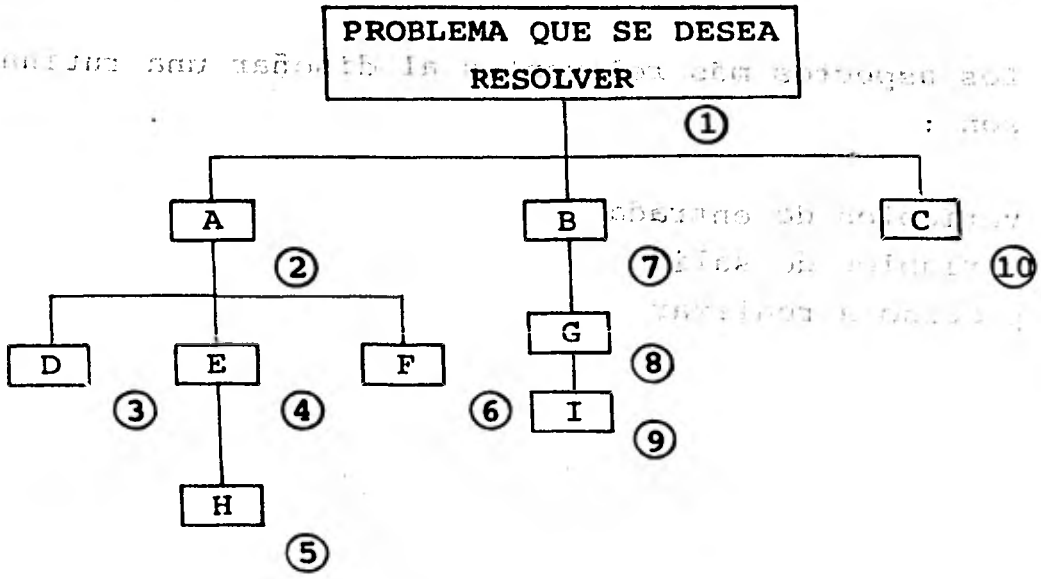
- mantener la visión global del problema así como de la estructura general del programa
- poder probar los módulos conforme se elaboran sin-requerir de módulos temporales de prueba
- planificar el avance y desarrollo del programa
- jerarquizar actividades
- distribuir el trabajo entre los miembros del grupo de trabajo
- estructurar el programa en módulos seccionalmente-independientes
- poder generar datos de prueba conforme se avanza - en la elaboración del proyecto
- en programas interactivos permite determinar y diseñar rutinas, que operan con el usuario, con anticipación para iniciar el entrenamiento
- mejorar la calidad y confiabilidad de los programas
- permite concentrarse en lo que se tiene que hacer- y no en cómo hacerlo

3.3.2 Método.

El desarrollo del programa se debe efectuar en forma jerárquica partiendo de los general hacia lo -- particular.

Para ello se emplea un diagrama de árbol o jerár-- quico en el cual se detalla cada actividad a tra-- vés de las ramificaciones; en dicho árbol la jerar

quía de las actividades se da de arriba hacia abajo y, en un mismo nivel, de la izquierda a la derecha como se muestra a continuación :



Los números encerrados en círculo indican la secuencia en que se deben ejecutar las actividades.

Cada bloque del árbol a su vez puede detallarse mediante otro árbol o diagramas HIPO.

El detallar cada bloque lleva primero a los diagramas de bloque y posteriormente al conjunto de estructuras o unidades lógicas que implementan la acción que se desea efectuar.

Al rediseñar una rutina o bloque, tal vez sea necesario modificar otras rutinas y/o estructuras de datos. Para realizar tal labor se busca un nivel del árbol jerárquico (backing up) en el cual los cambios no tengan efecto y a partir de ese punto se procede a corregir o modificar los bloques afectados de arriba hacia abajo (top-down).

Emplear subrutinas y funciones seccionalmente independientes para implementar cada bloque del árbol, esto permite obtener un programa mejor estructurado y más fácil de verificar.

Los aspectos más relevantes al diseñar una rutina son :

- **variables de entrada**
- **variables de salida**
- **proceso a realizar**

4. DIAGRAMAS HIPO.

4.1 Introducción.

Se denominan diagramas HIPO porque las siglas corresponden a "hierarchical input-process-output", es decir, "entrada-proceso-salida jerarquizados".

4.2 Propósito.

Es una forma gráfica de diseño que permite :

- tener documentación externa del programa
- estructurar las acciones del programa
- definir propósito de cada acción así como la información que requiere y la salida que proporciona
- reducir ambigüedad y la cantidad de documentación verbal
- identificar acciones que se deben realizar

4.3 Método.

Se procede de lo general a lo particular y usualmente un paquete HIPO consiste de :

- uno o más diagramas que dan una visión general del conjunto
- diagrama detallado
- tabla visual de contenido

Cada diagrama consiste de 3 partes (ver figura 4.1) :

a) entrada : files, records, campos, parámetros de control, etc.

b) proceso : pasos a seguir para realizar una función ó actividad

c) salidas del proceso

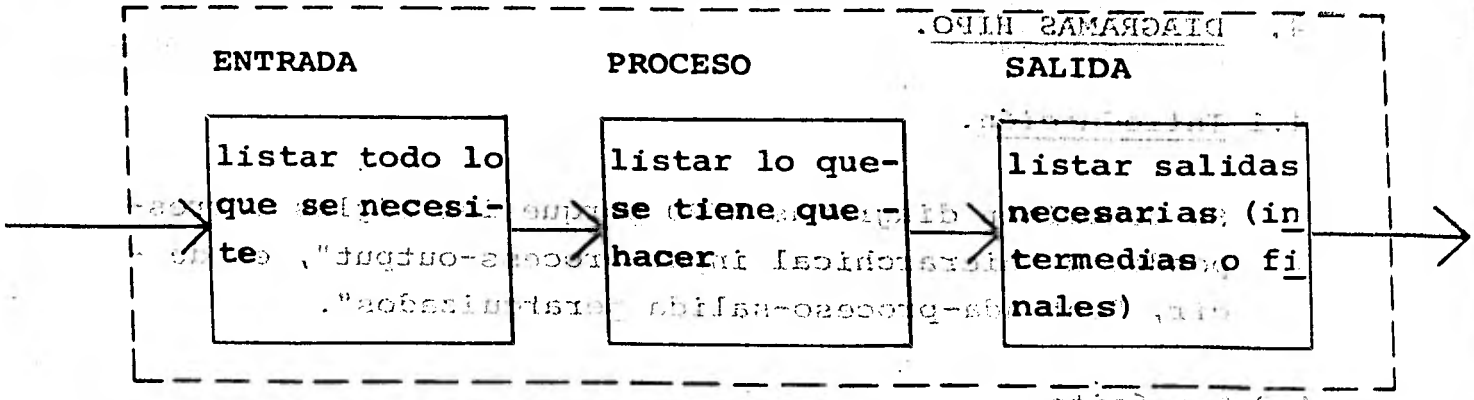


Fig. 4.1 Diagrama HIPO

En cada diagrama se incluye una tabla, en caso de ser necesaria, donde se da una descripción ampliada de los pasos a seguir así como de las rutinas necesarias en cada uno de ellos.

Tipo de paquetes HIPO :

a) Diseño inicial : preparado por el grupo de diseño al inicio del proyecto, da una visión global del mismo.

diseño funcional a alto nivel

b) Diseño detallado : preparado por el grupo de desarrollo (analistas y programadores).- Se emplea el diseño inicial como base.

diseño funcional detallado

c) Mantenimiento : documentación final del sistema.- Se identifican las funciones que deben ser modificadas y los módulos que los ejecutan.

producto, soporte, mantenimiento

5. DIAGRAMAS DE BLOQUES.

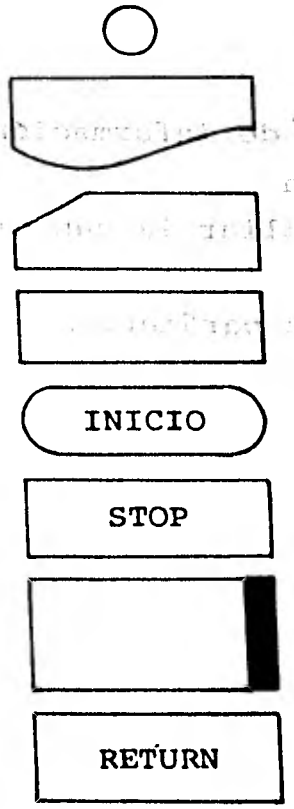
5.1 Propósito.

Presentar gráficamente y en forma sintetizada el contenido de cada rutina o módulo del programa, indicando la secuencia entre las diferentes acciones.

5.2 Método.

Todo diagrama de bloques contiene un inicio y un final (principio y fin del programa principal o de la subrutina).

Los bloques o figuras a emplear para la construcción del diagrama son :



conector

impresión

lectura

acción o conjunto de acciones

inicio de programa principal

fin de programa principal

llamado de subrutina o principio de subrutina

final de subrutina

6. DOCUMENTACION PARA EL USUARIO.

6.1 Propósito.

Permitir al usuario o comprador utilizar eficientemente el programa, indicarle las limitaciones del mismo y mostrarle la forma de operarlo.

6.2 Método.

La documentación para el usuario :

- lenguaje de programación empleado
- información que debe proporcionar el usuario para efectuar una corrida del programa
- archivos (temporales o permanentes) de los que haga uso el programa
- unidades de entrada/salida
- formato para los datos de entrada
- restricciones tanto en el volumen de información como en el rango de valores de la misma
- modificaciones a efectuar para ampliar la capacidad del programa
- tiempo de ejecución con diferentes parámetros
- resultados que se obtienen
- listado del programa
- muestra de una corrida
- mantenimiento del programa

7. PRUEBA Y CORRECCION DE PROGRAMAS.

7.1 Introducción.

Aproximadamente el 50% del esfuerzo y tiempo total, dedicado a la elaboración de un programa, se consume en la prueba del mismo.

La verificación debe hacerse sistemáticamente conforme se va desarrollando cada módulo, a fin de reducir el tiempo dedicado a este aspecto e incrementar la confiabilidad de los programas.

7.2 Significado de Programa Correcto.

Existen diferentes niveles de confiabilidad en un programa y por orden creciente de dificultad, en su obtención, son :

1. No errores de sintaxis.
2. Ejecución sin errores, es decir, sin operaciones inválidas.
3. Existe algún conjunto de datos para los cuales la respuesta es correcta.
4. Para un cierto tipo de datos la respuesta es siempre válida.
5. Para conjuntos de datos difíciles (deliberadamente escogidos) la respuesta es siempre válida.
6. Para cualquier conjunto de datos, válido con respecto a las especificaciones, la respuesta es correcta.
7. Para todos los posibles errores en la información de entrada proporciona una respuesta correcta o razonable.

Para un cliente el nivel mínimo de aceptación es 6.

7.3 Tipos de Errores.

1. Error en entender las especificaciones y requisitos del problema.
2. Error en entender el lenguaje de computación empleado así como sus casos especiales.
3. Falla del Algoritmo.
4. Accidentes por omisión :
   - en las condiciones para terminar un ciclo de repetición
   - criterios de convergencia
   - límite de iteraciones
   - etcétera

Remedios :

1. Dedicar más tiempo a entender el problema.
2. Experiencia con el lenguaje y sistema de cómputo empleado.
3. Desarrollo sistemático y estructurado del programa.
4. Revisar errores de sintaxis y considerar criterios para terminar toda unidad lógica. Pedir su opinión a otra persona que no ha tenido contacto con el programa y que no tiene prejuicios sobre el código.

7.4 Diseño de los Casos de Prueba.

Lo más razonable es empezar con los casos fáciles y seguir con los difíciles.

El diseño de estos casos depende en gran parte de la-

experiencia pero, en términos generales, deberán incluir:

a) Límites en el volumen de información (máximo, mínimo, muy pocos o demasiados).
b) Valores extremos en la información (máximo y mínimo-legal así como valores excesivos : demasiado grande-y pequeño, negativo, positivo).
c) Valores especiales (cero, blanco, etc., dependiendo-del problemas).
d) Valores no enteros, cuando se permitan.
e) Valores próximos a/ó que rebasen los límites establecidos.
f) Valores repetidos y conjutnos de valores de diferentes tipos.
g) Valores completamente anormales.

7.5 Cómo Tratar los Errores en la Información de Entrada.

- Checar siempre que los datos de entrada sean válidos.
- Cuando exista error imprimir un mensaje y decidir si se continúa la ejecución o no, dependiendo de la gravedad del error.

7.6 Verificación Modular de los Programas.

- Probar cada módulo conforme se va implementando.
- Si se implementa top-down y se verifica top-down, cada módulo es escrito hasta que el módulo que lo llama ha sido desarrollado y probado. Para probar cada módulo se utilizan rutinas o módulos mudos, es decir, rutinas que no realizan acción alguna.
- Si se verifica Bottom-up; diseñar los drivers que permitan correr el módulo o rutina independientemente - del resto del programa y, que permitan pruebas múlti

ples en cada corrida.

- Probar los módulos con entradas cuyas salidas sean conocidas.
- Reemplazar las subrutinas mudas con las reales conforme se disponga de ellas.

8. BIBLIOTECAS DE SOPORTE.

8.1 Propósito.

- Mejorar eficiencia y productividad en el desarrollo de programas.
- Evitar duplicidad de esfuerzos.
- Uso más efectivo del tiempo de trabajo.

8.2 Elementos Básicos.

a) Técnicos.

- Módulos : rutinas para acciones específicas.
- Bancos de datos.
- Documentación de los módulos.

b) Administrativos.

- Biblioteca interna : glosario de archivos, bancos de datos, usuarios y claves.
- Biblioteca externa : refleja el status de la interna.
- Procedimientos de máquina : incluye los comandos o rutinas necesarias para :

  i) crear y actualizar las bibliotecas
  ii) compilar y correr programas
  iii) obtener impresiones
  iv) ligar trabajos
  v) restauración de archivos y back-up
  vi) obtener listados del status de los archivos o directorios

- Procedimientos de oficina: reglas y estatutos para :

  i) aceptar estipulaciones y recomendaciones de los programadores
  ii) uso de los procedimientos de máquina
  iii) actualizar la biblioteca externa e interna
  iv) asignación de archivos y claves

c) Mantenimiento.

- Checar que los programas y módulos de la biblioteca estén estructurados y con sangrado.
- Checar que las instrucciones de los programas se adhieran al lenguaje de la instalación empleada.
- Desarrollar tracers o debugers para probar programas y checar tanto el tiempo como la memoria utilizada.
- Crear varias versiones (correspondientes a las fases de desarrollo) de un programa.
- Obtener listados de control estadístico tanto de claves como de archivos.

9. PLATICAS ESTRUCTURADAS.

9.1 Propósito.

Detectar y remover errores tan pronto como sea posible en el diseño de un programa.

9.2 En qué consisten :

Es una revisión del trabajo desarrollado por una persona para implementar un programa (diseño, código, documentación, etc.), por otros miembros del proyecto invitados por el interesado. La sesión la conduce la persona interesada y no participa el jefe del proyecto.

9.3 Beneficios.

- permite al programador detectar errores u omisiones
- permite al resto del grupo aprender nuevos enfoques
- permite que los miembros del grupo se enteren del trabajo desarrollado por otros

9.4 Fases en las que pueden emplearse.

- planeación y establecimiento del calendario del proyecto
- especificaciones del sistema o método de solución
- especificaciones funcionales del programa
- diseño de programa (estructura)
- diseño detallado del programa (información, módulos, documentación)
- documentación final (manual de usuario y de mantenimiento

- producto terminado

9.5 Características Básicas.

- arreglada y programada por la persona cuyo producto será revisado
- el jefe del proyecto no interviene
- el interesado selecciona a los revisores y el jefe del proyecto revisa la lista para checar que todos los relacionados con el tema han sido invitados
- se invitan 4 - 6 personas
- cada plática debe tener un conjunto definido de objetivos
- se entrega a los participantes el material a ser revisado con una semana de anticipación
- se elige un moderador por sesión, mismo que lleva una lista de errores, inconsistencias y discrepancias
- la plática solo sirve para detectar errores, no para solucionarlos
- la duración de la plática no debe exceder 2 horas

10. EQUIPOS DE TRABAJO.

10.1 Propósito.

- Coordinar esfuerzos de programación.
- Reducir esfuerzos y multiplicidad de trabajos.
- Aumentar productividad.
- Producir programas estructurados de enfoque general.
- Planear producción de programas y tiempo de entrega.
- Facilitar la asignación de trabajo y la segmentación de proyectos grandes.
- Preservar la estructura del programa a través de todas las fases.
- Hacer mejor uso de las facilidades existentes.
- Reducir costos.

10.2 Elementos.

- Jefe del grupo : responsable del diseño, rutinas principales, código crítico, interfaces, asignación de trabajo, integración de los módulos, reportes a la gerencia y contrato de personal.
- Programador de respaldo : se encarga de realizar pruebas, planear integración, ejecutar tareas principales y substituir al jefe cuando sea necesario.
- Bibliotecario : maneja los procedimientos de máquina, los de oficina, mantiene estadísticas del proyecto, se encarga de correr programas, etc.
- Técnicos familiarizados con :
  a) El problema que se desea resolver.
  b) El software y hardware a ser empleados.
  c) Técnicas especiales de programación y codificación.

11. BIBLIOGRAFIA.

1) Knuth, F.
"Notes on Avoiding "GO TO" statements'
Technical Report CS148, Stanford University,
January 1970.

2) Kosaraju, S.
"Analysis of Structured Programs"
Proc. of 5th Annual ACM Symp. on Theory Computing
May 1973.

3) Mills, H.
"Debugging Techniques in Large Systems"
(Ed. Rustin Randall) Prentice Hall, Englewood Cliffs,
N.J. 1971.

4) Wegner, E.
"Tree-Structured programs"
CACM, 16, 11,
November 1973.

5) Yourdon, E.
"Techniques of Program Structure and Design"
Prentice Hall, Englewood Cliffs,
N.J. 1975.

APENDICE B

NORMAS Y ESTANDARES

APENDICE B.1

ESTANDARES DE PROGRAMACION

CONTENIDO PAGINA

CONTENIDO **PAGINA**

CONTENIDO **PAGINA**

CONTENIDO PAGINA

1. INTRODUCCION.

El estándar o guía de programación en FORTRAN es un conjunto de normas, que permite que los programas y módulos sean más fáciles de leer, revisar, modificar y mantener.

Se pretende que esta guía sea un documento corto, conciso y sobre todo práctico sobre reglas de programación.

Para ello se establecerán las normas generales de programación y posteriormente se indicará la forma de implementarlas, junto con un ejemplo ilustrativo, para los siguientes compiladores :

- FORTRAN 66 (PDP 11-34 y UNIVAC)
- FORTRAN 77 (HARRIS/8)

Cabe mencionar que, para poder aprovechar y emplear todas las características de programación estructurada, es primordial y necesario que las etapas de Análisis y Diseño del programa se desarrollen también en forma estructurada. El antecedente inmediato a la etapa de programación deberá ser un diagrama de estructura que muestre la jerarquización de los módulos y el flujo de información entre ellos.

2. OBJETIVOS.

2.1 ESTILO

Se intenta unificar el estilo de programación sin limitar la creatividad del programador. Se entiende por estilo, la selección adecuada de hábitos o técnicas de programación. El empleo de estas técnicas ayudará en la tarea de producir programas correctos, eficientes, modificables, comprensibles y homogéneos.

2.2 PROPIEDAD

La unificación de las técnicas de programación hace que, como consecuencia del empleo de estructuras estándares y del entendimiento general de las bases sobre las que se desarrollan los módulos, los programas no serán propiedad exclusivamente del programador que desarrolló el programa. Esto es, no se dependerá del programador para el uso y mantenimiento del programa.

2.3 ESTRUCTURA

Los programas que se codifiquen siguiendo estas normas, mostrarán una estructura coherente, esto es, la apariencia física del código (característica estática) será un reflejo de las acciones o flujo de control dentro del programa (característica dinámica). Con lo anterior se facilitará e incrementará la comprensión de la lógica del programa, lo cual será de gran utilidad en la prueba y depuración del mismo.

2.4 CLARIDAD

La adopción de una disciplina de programción basada en el empleo de determinadas estructuras lógicas, así como de reglas de formateo que comprenden : el uso de líneas en blanco, espacios en blanco entre identificadores y el empleo de buenos comentarios, hace que los -- programas sean claros y legibles.

2.5 USO DEL FORTRAN

Se desea, además, enfatizar el uso de las buenas características del FORTRAN y tratar de evitar el uso de aquella características indeseables que oscurecen la lógica del programa y que, por lo tanto, hacen que los - programas sean más difíciles de entender, mantener y - modificar. Las facilidades básicas a emplear son : -- sangrado, longitud y nombre de los identificadores, modularidad y comentarios.

2.5.1 Sangrado

FORTRAN permite el uso de sangrado entre las columnas- 7 y 72, lo cual aumenta considerablemente la claridad- del programa y hace más obvio el flujo del control dentro del mismo. Lo anterior se debe a que, el uso del- sangrado, permite identificar claramente las estructuras de control así como su nivel de anidamiento.

2.5.2 Longitud y Nombre de las Variables

FORTRAN permite el uso de identificadores (nombres de- variables, archivos y módulos en general) de longitud- variable, entre 6 y 10 caracteres dependiendo del compilador empleado.

Se recomienda que los nombres de variables y módulos tengan una longitud máxima de 10 caracteres, cuando el compilador empleado así lo permite.

Los nombres de las variables debe representar el contenido de las mismas y los nombres de los módulos deben reflejar su función u objetivo.

Todas las variables deben ser explícitamente declaradas (empleando los declaradores REAL,INTEGER,COMPLEX,LOGICAL, CHARACTER). No dejar que el sistema asuma que las variables que empiezan con las letras A-H y O-Z sean de tipo real y, las que comienzan con I-N sean de tipo entero. Esta práctica, además de que puede alterar el significado del contenido de la variable, reduce en uno el número de caracteres disponibles para el identificador.

2.5.3 Modularidad

FORTRAN permite el empleo de módulos denominados funciones y subrutinas. El empleo de estos módulos para el desarrollo del programa es obligatorio, ya que la práctica de realizar todas las actividades en un solo módulo es obsoleta y debe ser evitada.

2.6 DOCUMENTACION DEL PROGRAMA

Todo programa debe contener comentarios que expliquen qué hace, cómo lo hace, etc.

La documentación se puede englobar en dos grandes bloques :

- Comentarios de prólogo ó encabezado
- Comentarios explicativos del cuerpo del programa

El contenido y uso de cada uno de estos se presenta en la sección 5.3 de este apéndice.

3. MODULARIDAD.

3.1 INTRODUCCION

Por modularidad se entiende el hecho de cumplir los objetivos de un programa mediante su descomposición en módulos organizados con diferentes niveles de jerarquía.

La organización modular de un proyecto se origina y lleva a cabo en la etapa de Diseño más que en la de Programación; sin embargo, amerita comentarse puesto que en la etapa de programción se puede requerir un mayor detalle del diagrama de estructura.

Esta técnica de desarrollo modular es una condición necesaria para la correcta implementación de un programa, así como para controlar la complejidad del mismo.

El descomponer y organizar un programa en forma modular no es equivalente a fragmentar en forma anárquica el programa. Para que la organización modular del programa sea correcta, se requiere que los módulos tengan :

- Una función específica.
- Bajo acoplamiento.
- Alta cohesión.

A continuación se describe cada uno de estos aspectos.

3.2 FUNCION Y SUBORDINACION

Cada módulo debe tener una función específica y el nombre del módulo debe reflejar dicha función.

La representación gráfica de la jerarquización de los módulos (diagrama de estructura) es semejante al organigrama de una empresa y, al igual que en una empresa, las funciones de los jefes (módulos de nivel alto) son-

exclusivamente de control y coordinación.

El número de módulos subordinados a otro módulo de mayor jerarquía no debe ser mayor de 7.

3.3 ACOPLAMIENTO

El acoplamiento es una medida de la independencia que existe entre los módulos. Se dice que un módulo es independiente cuando la función que ejecuta no depende de lo que se hizo en otro módulo.

Siempre debe buscarse que los módulos sean completamente independientes, es decir, que tengan bajo acoplamiento. Para ello se debe evitar el uso del COMMON no etiquetado y minimizar el uso de banderas como parámetros de salida. Nunca deben pasar banderas de un módulo de mayor jerarquía a otro de menor jerarquía.

3.4 COHESION

La cohesión es una medida de la solidez interna del módulo. Se dice que un módulo es sólido cuando sólo tiene una función que cumplir y todas las variables y actividades que maneja están encaminadas a dicho fin.

Entre más sólido sea un módulo, más independiente será de los otros módulos. Es por ello que, en términos generales, se cumplirá que:

- alta cohesión < = > bajo acoplamiento
- baja cohesión < = > alto acoplamiento

4. ESTRUCTURA DE LOS MODULOS.

4.1 TAMAÑO

El cuerpo de los módulos (programa principal, subrutinas y funciones) no debe exceder de 2 páginas (100-120 líneas) incluyendo los comentarios explicativos.
Puede haber excepciones para el caso de tablas muy grandes o comandos de decisión múltiple (CASE) muy extensos.

4.2 APARIENCIA FISICA

El código debe estar distribuido (empleando estructuras lógicas, sangrado y comentarios) en forma tal que refleje claramente la estructura y el flujo de control de la rutina. La presentación de un módulo se discute y ejemplifica en los apéndices para cada tipo de compilador.

4.3 ETIQUETAS Y FORMATOS

Los únicos comandos o instrucciones que pueden tener etiquetas son "CONTINUE" y "FORMAT".
Todas las etiquetas del primero (CONTINUE) deben aparecer numeradas de 100 en 100 y en orden ascendente.
Todos los formatos deben aparecer al final de la rutina con sus respectivas etiquetas numeradas en orden ascendente. La numeración de las mismas deberá ser de 100 en 100 a partir de la etiqueta 5000.

4.4 SANGRADO

Cada nuevo nivel de anidamiento en una estructura de control debe estar sangrado 3 espacios a la derecha.

Lo anterior es con el objeto de hacer visible la estructura global del módulo así como el rango de control que tiene cada estructura lógica. Menos de --- tres espacios reduce la legibilidad del módulo y más de tres es un desperdicio de espacio.

4.5 ESTRUCTURAS DE CONTROL VALIDAS

Existen tres tipos básicos de estructuras de control: secuencia, decisión y repetición. Estas estructuras pueden ser anidadas para obtener estructuras más complejas.

La característica fundamental de dichas estructuras-lógicas es que sólo tienen una entrada y una salida.

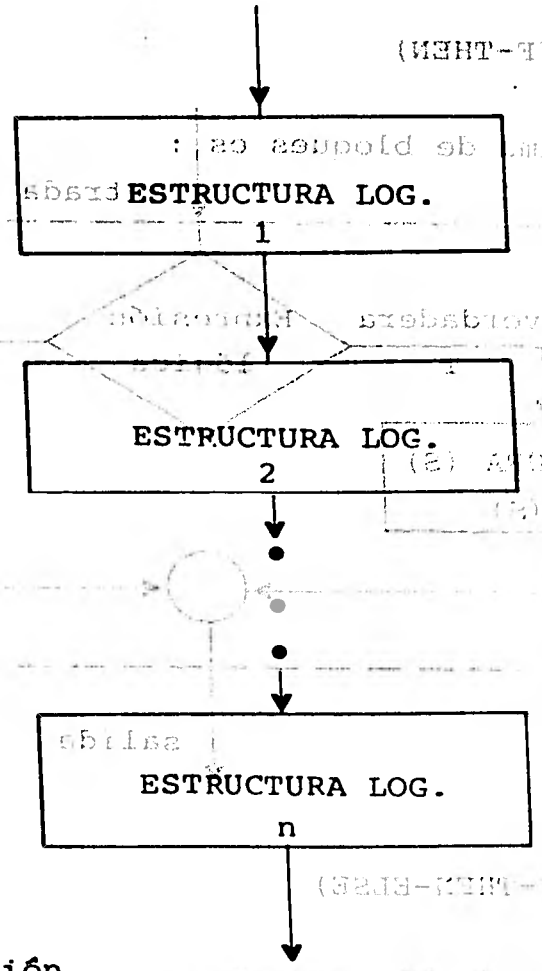
A continuación se ilustra la representación gráfica-de cada estructura lógica. La manera de implementar la en cada compilador se indica en el ejemplo correspondiente.

4.5.1 Secuencia

Es la estructura lógica o de control fundamental. Su existencia justifica y garantiza que el flujo de control dentro de los módulos sea de arriba hacia abajo (Top-Down).

Su diagrama de bloques es :

- Cada estructura lógica puede constar de uno o más-comandos.



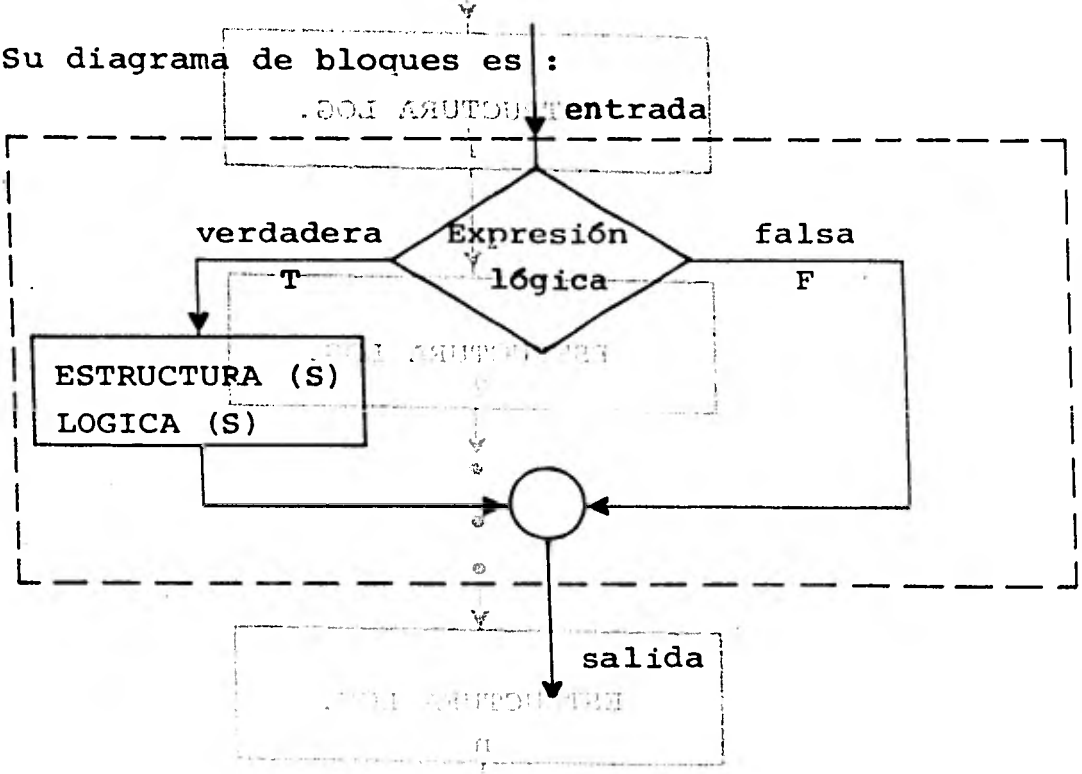
4.5.2 <u>Decisión</u>

Se emplean para transferir el control del módulo a otra localidad de acuerdo con ciertas condiciones. Existen 3 tipos :

- Simple.
- Doble.
- Múltiple.

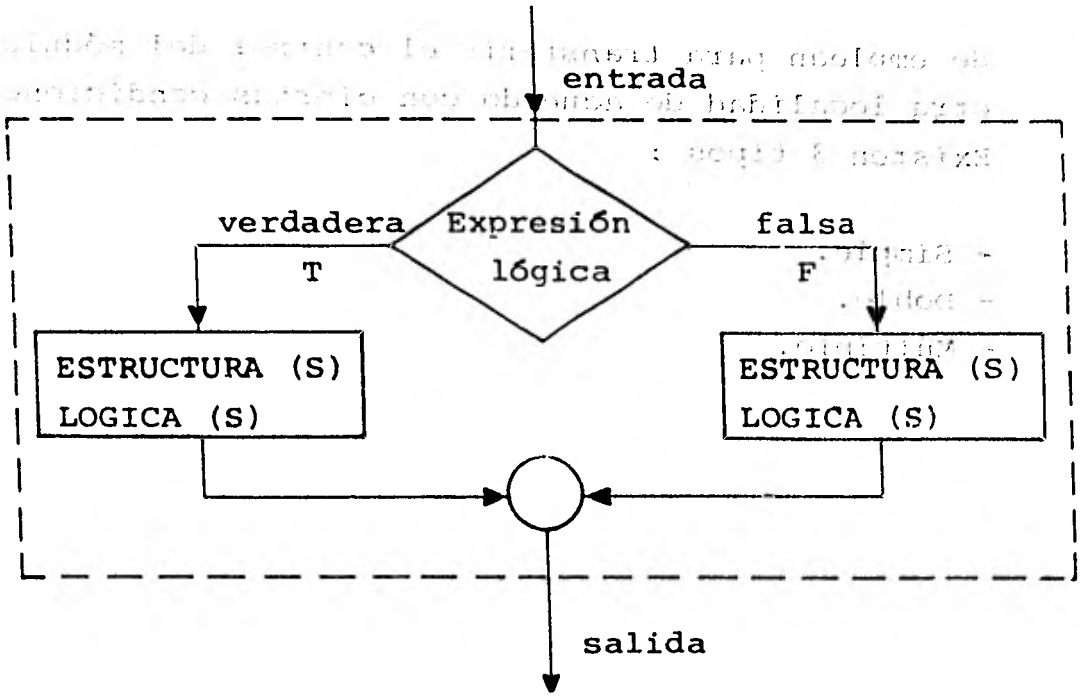
a) Simple (IF-THEN)

Su diagrama de bloques es :



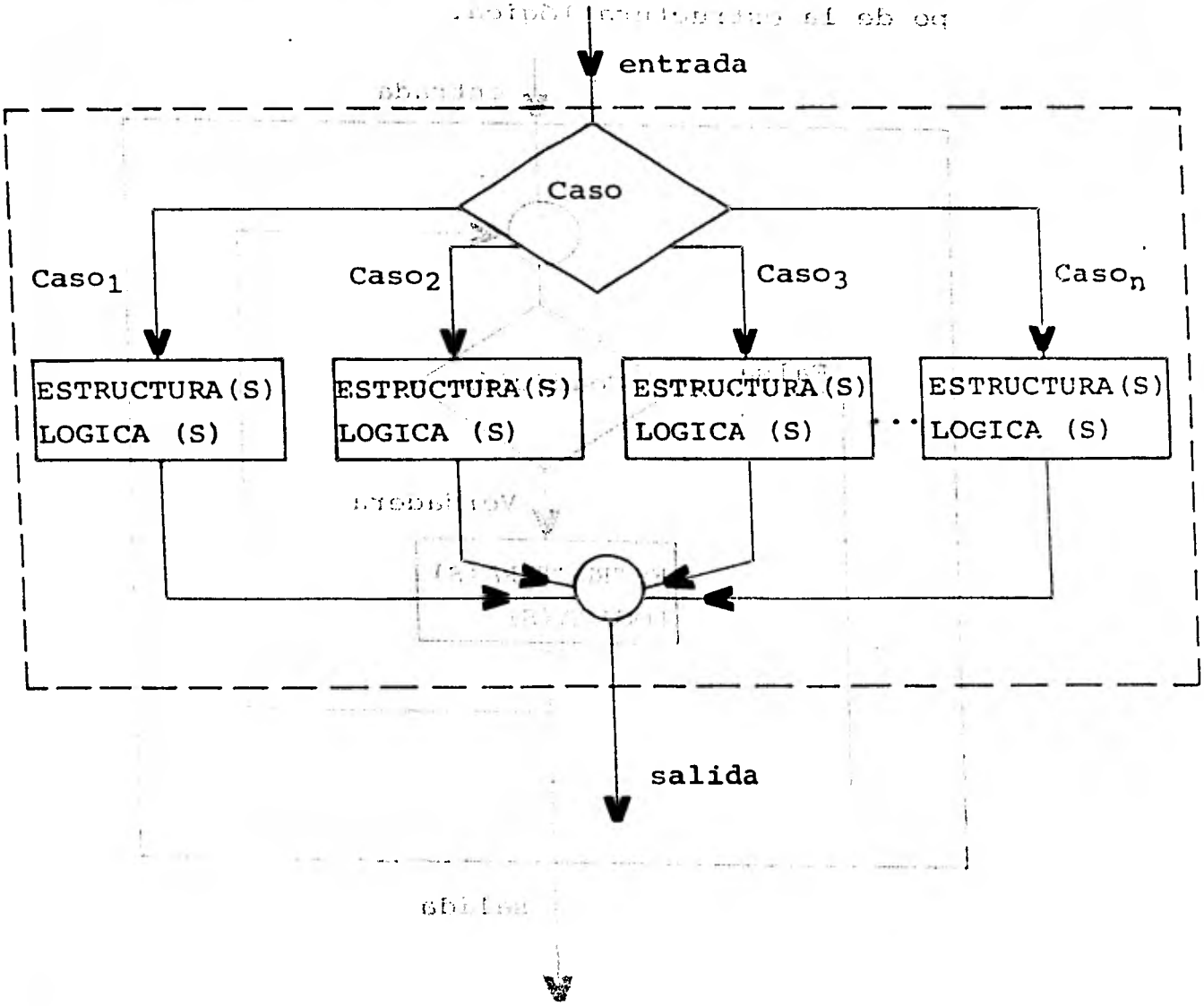
b) Doble (IF-THEN-ELSE)

Su diagrama de bloques es :



c) Múltiple (CASE)

Se emplea cuando dependiendo del valor de la variable (Caso) es la acción que se lleva a cabo.

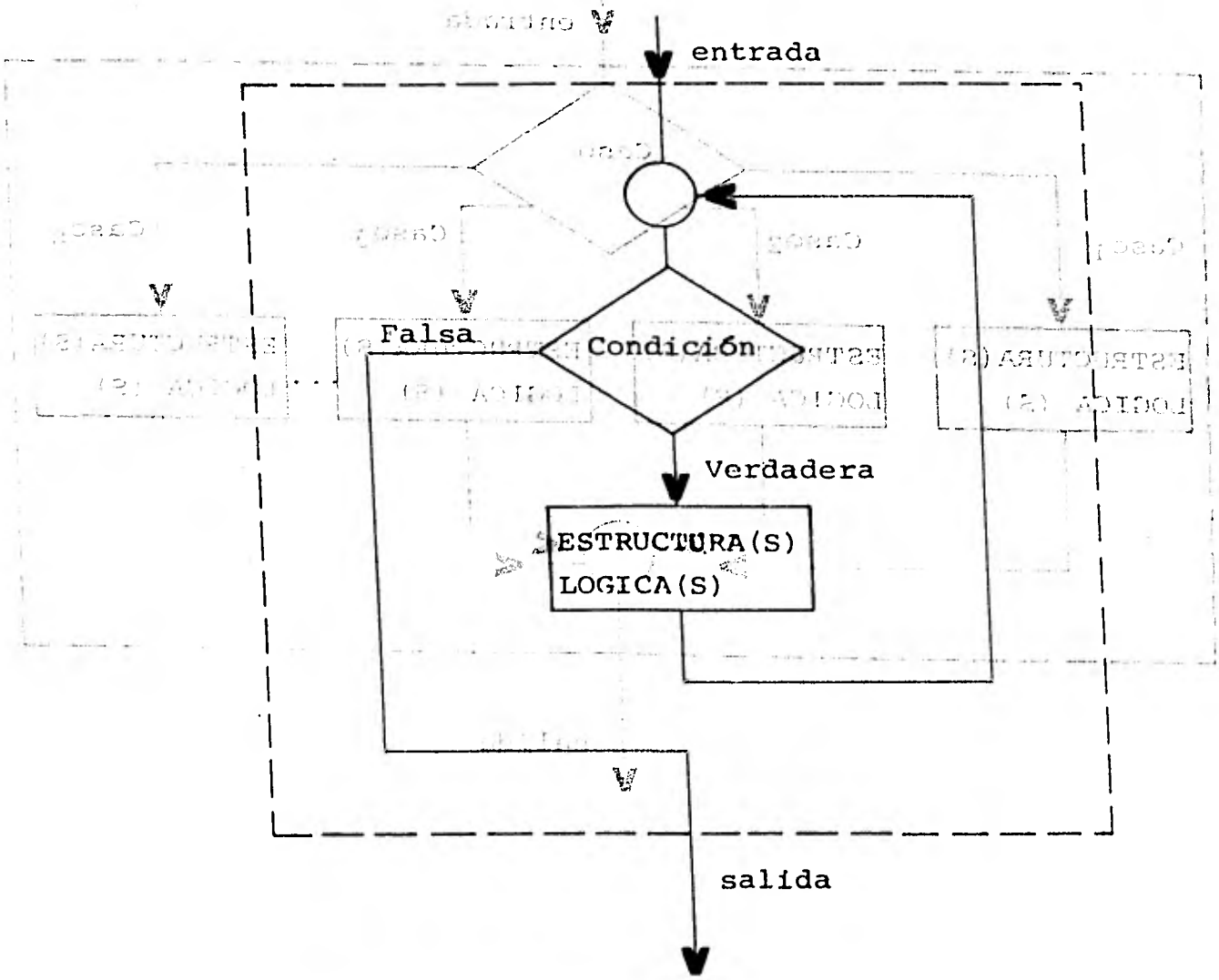


4.5.3 Repetición

Se emplean para ejecutar en forma repetitiva, hasta que se cumple una cierta condición, una serie de comandos y/o estructuras lógicas.

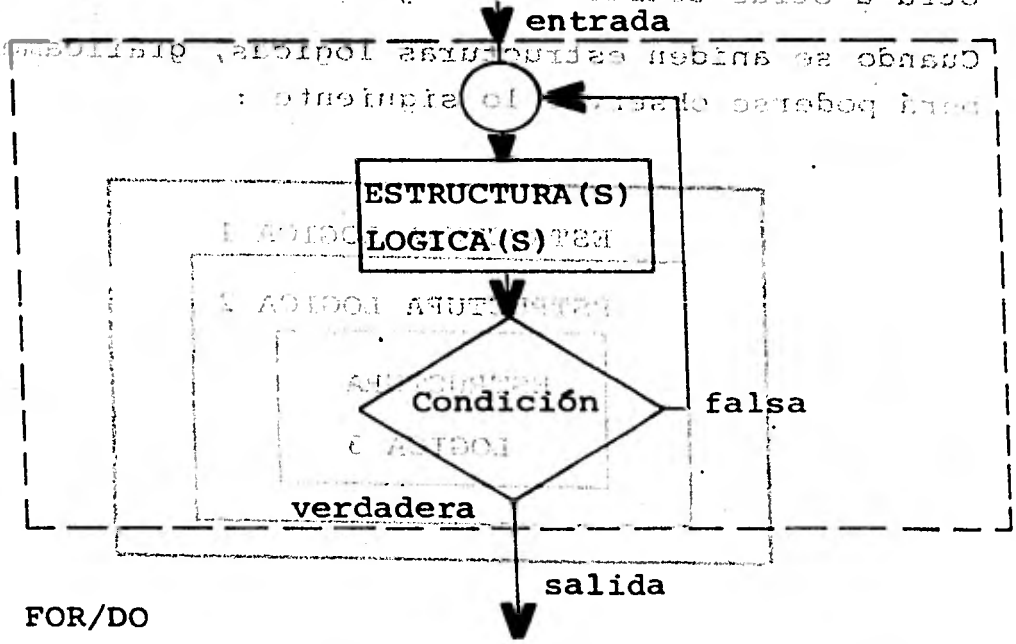
a) WHILE

Se checa la condición antes de ejecutar el cuerpo de la estructura lógica.



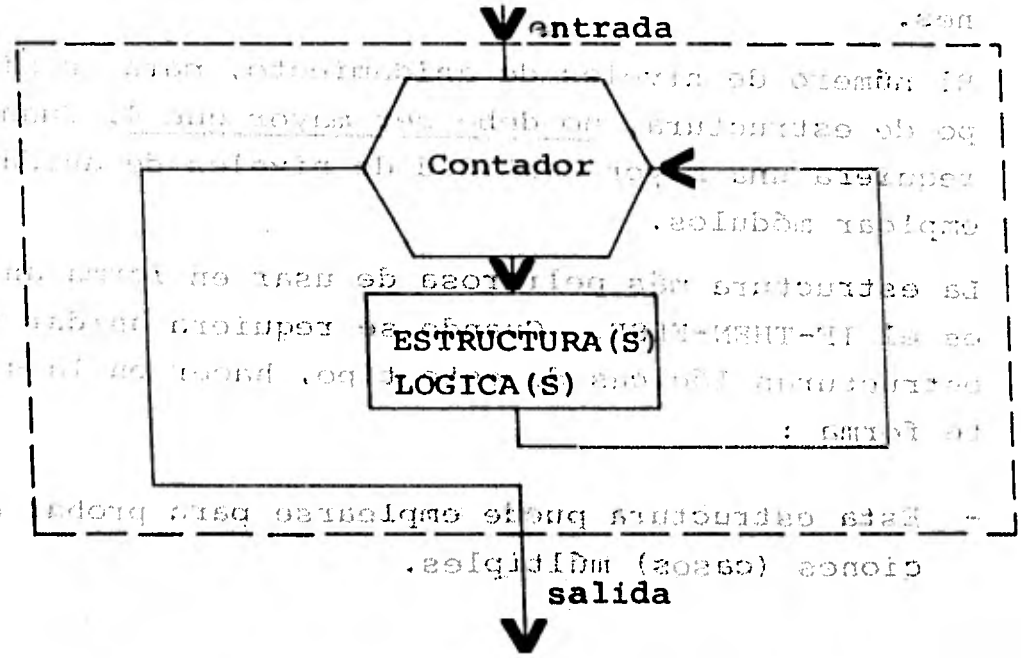
b) UNTIL

Se ejecuta el cuerpo de la estructura lógica y después se checa la condición.



c) FOR/DO

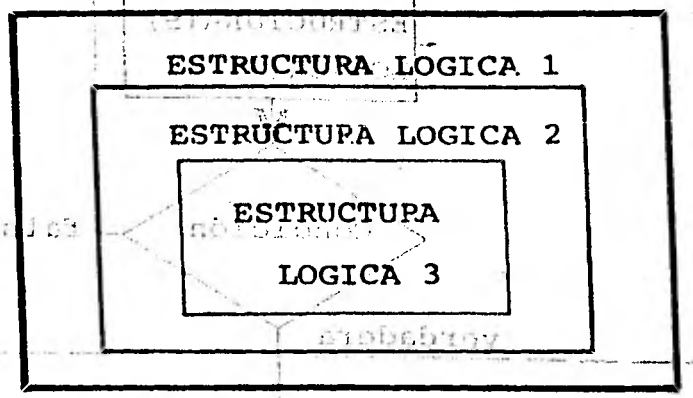
Se ejecuta el cuerpo de la estructura un número finito de veces conocido de antemano.



4.5.4 Anidamiento

Cualquier estructura lógica puede contener a su vez otra u otras estructuras lógicas.

Cuando se aniden estructuras lógicas, gráficamente deberá poderse observar lo siguiente :

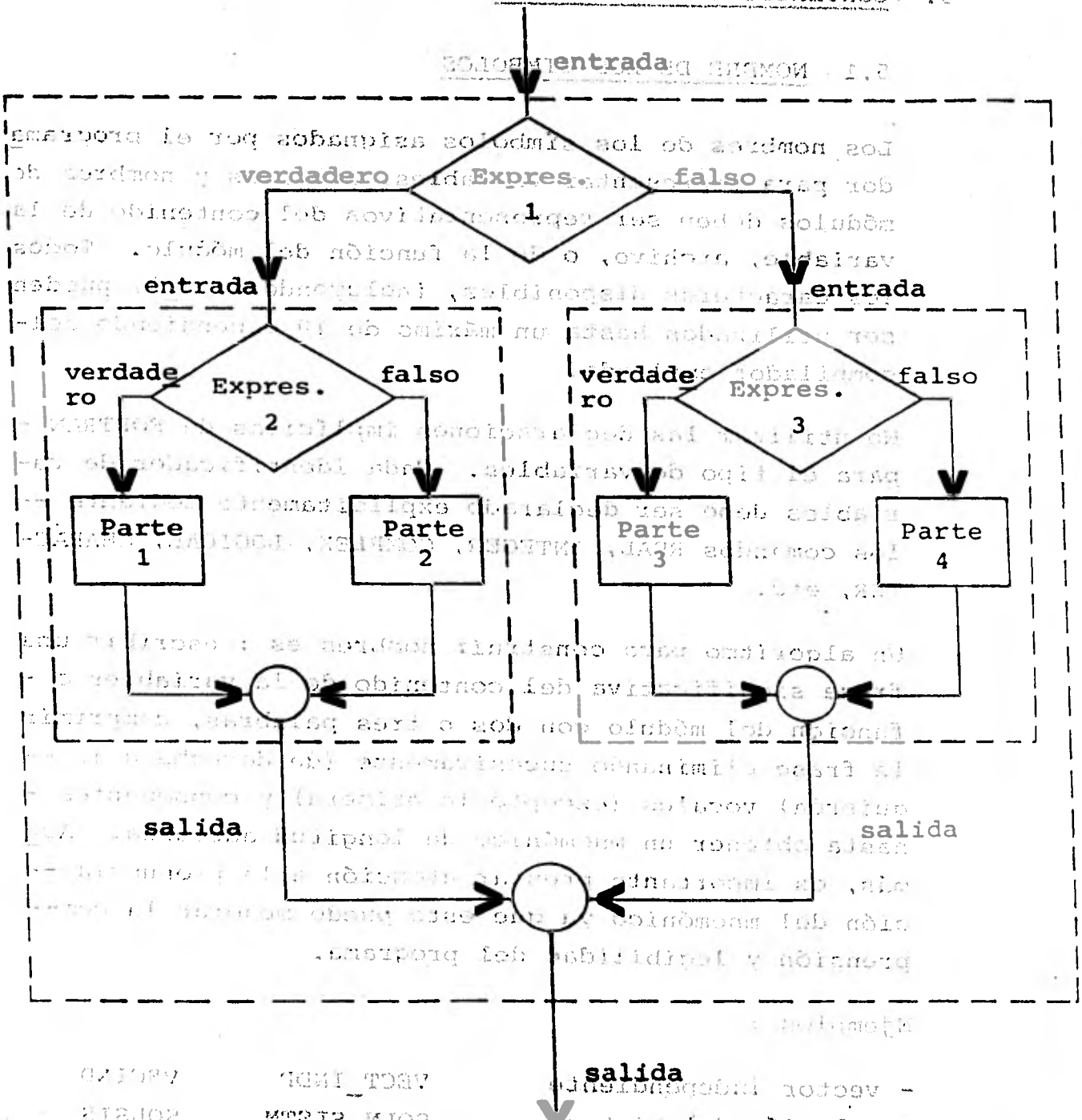


Es decir, debe quedar bien delimitado el campo de acción de cada estructura y no usar delimitadores comunes.

El número de niveles de anidamiento, para un mismo tipo de estructura, no debe ser mayor que 4. Cuando se requiera una mayor cantidad de niveles de anidamiento, emplear módulos.

La estructura más peligrosa de usar en forma anidada es el IF-THEN-ELSE. Cuando se requiera anidar varias estructuras lógicas de este tipo, hacer en la siguiente forma :

- Esta estructura puede emplearse para probar condiciones (casos) múltiples.


entrada
verdadero
Expres. 1
falso
entrada
entrada
verdade ro
Expres. 2
falso
verdade ro
Expres. 3
falso
Parte 1
Parte 2
Parte 3
Parte 4
salida
salida
salida

5. CONTENIDO DE LOS MODULOS.

5.1 NOMBRE DE LOS SIMBOLOS

Los nombres de los símbolos asignados por el programador para representar variables, archivos y nombres de módulos deben ser representativos del contenido de la variable, archivo, o de la función del módulo. Todos los caracteres disponibles, incluyendo el "_", pueden ser utilizados hasta un máximo de 10 dependiendo del compilador empleado.

No utilizar las declaraciones implícitas de FORTRAN para el tipo de variables. Cada identificador de variables debe ser declarado explícitamente mediante los comandos REAL, INTEGER, COMPLEX, LOGICAL, CHARACTER, etc.

Un algoritmo para construir nombres es : escribir una frase significativa del contenido de la variables o función del módulo con dos o tres palabras, comprimir la frase eliminando sucesivamente (de derecha a izquierda) vocales (excepto la primera) y consonantes hasta obtener un mnemónico de longitud adecuada. Además, es importante prestar atención a la pronunciación del mnemónico ya que esto puede mejorar la comprensión y legibilidad del programa.

Ejemplos :

- vector independiente	VECT_INDP	VECIND
- solución del sistema	SOLN_SISTM	SOLSIS
- apuntador	APUNTADOR	APNTDR

5.2 USOS PROHIBIDOS

Todo programa debe ser escrito con comandos secuenciales y las estructuras lógicas de control mencionadas en la sección 4.5, ya que el uso de otro tipo de comandos y estructuras solo oscurece el contenido y degrada el producto.

Es por ello que queda absolutamente prohibido el uso de las estructuras que a continuación se mencionan, excepto cuando se indique lo contrario.

5.2.1 Go to

Es el comando más dañino de todos; debe ser evitado en todas sus acepciones, excepto cuando se indique lo contrario en el ejemplo correspondiente al compilador que se desee emplear.

5.2.2 If Aritmético

Su uso queda prohibido.

5.2.3 Entry

El comando ENTRY no debe ser utilizado en las subrutinas. Subrutinas adicionales deberán ser escritas.

5.2.4 Equivalence

El uso del EQUIVALENCE se justifica solamente cuando se desea ahorrar espacio en memoria.

Solamente se debe emplear esta técnica como último recurso para resolver el problema, ya que emplear dos variables diferentes para representar la misma información, es muy confuso y oscurece la lógica del programa.

5.2.5 Stop

Nunca se debe utilizar.

5.2.6 Call Exit

Debe aparecer exclusivamente en el programa principal, es decir, no se permite finalizar la ejecución de un programa dentro de un módulo que no sea el programa principal.

5.2.7 Números Mágicos

Queda terminantemente prohibido el empleo de números mágicos dentro del código del módulo.

Todos aquellos parámetros que sean constantes deberán declararse en la parte superior del módulo mediante el comando 'DATA' o 'PARAMETER' y, además, deberán representarse mediante un identificador que refleje la función de dicho parámetro. Por ejemplo :

- PI para 3.1416
- EPS para un criterio de convergencia

5.2.8 Common

El uso del COMMON no etiquetado queda terminantemente prohibido.

El uso de COMMON etiquetado se deberá minimizar y emplear de preferencia sólo para transferir arreglos y/o nombres de archivos.

El (los) COMMON etiquetado(s) que se emplee(n) en un módulo sólo deberá contener identficadores de variables y/o archivos que se empleen en dicho módulo. En caso de conflicto, declarar un COMMON etiquetado para

cada variable.

5.3 FINALIZACION DE MODULOS (SUBRUTINAS Y FUNCIONES)

Debe existir una sola salida del módulo y esta deberá efectuarse mediante el comando 'RETURN' antes del 'END' del módulo. Se debe evitar el abandonar la ejecución del módulo en otros sitios del mismo.

5.4 DOCUMENTACION

Los comentarios sirven para explicar y/o aclarar qué sucede en el flujo de control del programa. Existen dos tipos de comentarios : comentarios de prólogo o encabezado del módulo y comentarios explicativos del cuerpo del módulo.

5.4.1 Comentarios de Prólogo

Todo módulo (programa, subrutina o función) debe contener una descripción breve de su función, uso, variables que maneja y aclaraciones pertinentes sobre el uso del mismo.

Estos comentarios deben aparecer inmediatamente después del nombre del módulo y deberán estar delimitados por dos líneas de asteriscos.

Los comentarios que se deberán incluir, por orden de aparición, son :

a) Función u Objetivo:

Describir en 1 ó 2 líneas la función que realiza el módulo y, en caso de ser relevante, indicar el método empleado.

b) Realizador :
Identificar a la Institución y Departamento.

c) Programador :
Identificar al programador del módulo.

d) Revisor :
Identificar a la persona que efectuó la última - revisión del módulo.

e) Fecha de la última revisión.

f) Descripción de los identificadores empleados :
Esta información brinda al programador y usua--- rios del módulo una gran ayuda en la comprensión, análisis y depuración del módulo.
Dicha descripción deberá estar estructurada en 4 secciones, a saber :

- Entradas
Consta del nombre y descripción de variables, constantes y archivos empleados como paráme-- tros de entrada al módulo. Se entiende por - parámetro de entrada cualquier dato que sien- do generado por otro módulo sea empleado por- éste.

- Salidas
Consta del nombre y descripción de variables- y archivos cuyos valores o contenido se gene- ran en el módulo y que son empleados por o-- tros módulos.

- Identificadores locales
Consta del nombre y descripción de variables, constantes y archivos que sean empleados por- el módulo para realizar su función. La infor mación contenida por estos identificadores --

sólo es empleada por el módulo.

- Módulos empleados
  Se incluye el nombre y descripción breve de las subrutinas y funciones que emplea el módulo.

Todos los identificadores en cada sección deberán listarse en orden alfabético, y tener un sangrado con respecto al nombre de la sección.

g) Notas y aclaraciones
Esta sección es opcional y sirve para indicar en forma breve :

- Variables, regiones COMMON y/o registro físicos que son modoficados en su contenido, por el módulo.
- Restricciones en volumen de información para los arreglos.
- Rango de valores permisibles para ciertas variables.

5.4.2 Comentarios explicativos

Estos comentarios se insertan en el código a fin de indicar cuál es el propósito de cada sección, conjunto de comandos o estructuras lógicas dentro del cuerpo del módulo.
El comentario deberá tener un sangrado igual al del nivel de anidamiento del código que describa. Deberá estar delimitado por líneas en blanco y comenzar con un asterisco.

5.4.3 Ejemplo

Consultar el apéndice del compilador que se desee utilizar.

6. SUGERENCIAS PARA MEJORAR LA EFICIENCIA DE LOS MODULOS EN FORTRAN.

6.1 PROPOSITO

Presentar y familiarizar al usuario con ciertos hábitos de estilo que pueden contribuir tanto a incrementar la legibilidad del módulo como a reducir el tiempo de ejecución del mismo.

6.2 METODOS

Para lograr lo anterior, al desarrollar el código de un módulo se deberá tomar en cuenta lo siguiente :

i) Crear tipos de variables propios del usuario para manejar la información, en lugar de emplear números.

Por ejemplo :

```
      INTEGER LU,MA,MIE,JUE,VIE,SAB,DOM
      DATA LU,MA,MIE,JUE,VIE,SAB,DOM/1,2,3,4,5,6,7/
      ==================================
      ==================================
      DO 200 I = LU,DOM
      ==================================
      ==================================
  200 CONTINUE
```

ii) La claridad y legibilidad del programa tienen preferencia sobre la cantidad de memoria empleada y el tiempo de ejecución, siempre y cuando no exista escasez de recursos con respecto a memoria, o sea crítico el tiempo de ejecución.

iii) Si se requiere de gran cantidad de memoria, y el-

tiempo de ejecución no es crítico, emplear ar-- chivos en lugar de arreglos para guardar y procesar la información.

iv) La jerarquía de las operaciones artiméticas, -- por orden creciente en su tiempo de ejecución,- es :

- Suma
- Resta
- Multiplicación
- División
- Exponenciación
- Funciones logarítmicas y trigonométricas

mayor tiempo de ejecución

Debido a lo anterior, es recomendable reducir - al mínimo el uso de la multiplicación, división, etc., sobre todo dentro de las estructuras lógicas de repetición.

v) Evitar cálculos dentro de las unidades de repetición. Por ejemplo :

NO

```
      REAL A(20)
      DO 100 I = 1,20
         A(I) = A(I)*B*C
100   CONTINUE
```

SI

```
      REAL A(20)
      TEMP = B*C
      DO 100 I = 1,20
         A(I) = A(I)*TEMP
100   CONTINUE
```

vi) Evitar el uso de la división cuando sea posi-- ble. Por ejemplo :

NO

```
      A = B/2
```

SI

```
      A = B*0.5
```

vii) Emplear el parámetro de incremento del DO en lugar de operaciones, cuando sea posible, -- dentro de la unidad de repetición. Por ejemplo :

NO	SI
REAL A(20)	REAL A(20)
DO 100 I = 1,10	DO 100 I = 2,20,2
A(2*I) = B	A(I) = B
100 CONTINUE	100 CONTINUE

viii) La eficiencia en la manipulación de valores-numéricos decrece, en términos generales, en la siguiente forma :

- Enteros
- Reales
- Complejos
- Doble precisión

menor eficiencia

ix) No usar la conversión implícita de datos, debido a que esto incrementa el tiempo de ejecución y, en ocasiones, la máquina no efectúa dicha conversión. Por ejemplo :

NO	SI
REAL A	REAL A
INTEGER D	INTEGER D
400 CONTINUE	400 CONTINUE
A = A+1	A = A+1.0
IF(A.EQ.0) GO TO 400	IF(A.EQ.0.0)
D = 3	GO TO 400
	D = 3

7. BIBLIOGRAFIA.

1) Van Tassel D.
"Program Style, Design,Efficiency, Debugging and Testing"
Prentice Hall, Englewood Cliffs NJ, 1978

2) Yourdon and Constantine
"Structured Design"
Prentice Hall, Englewood Cliffs NJ, 1979

EJEMPLOS

EJEMPLO A.

EJEMPLO A. IMPLEMENTACION PARA FORTRAN 66 (PDP 11-34 y- UNIVAC)

A.1 ESTRUCTURAS DE CONTROL

Cada estructura tiene un comando arriba y otro abajo que la delimitan. El cuerpo de la estructura va sangrado 3 espacios a la derecha de estos comandos.

A.1.1 Secuencia

```
| |S1
| |S2
| | :
| |Sn
```

Si : comando y/o estructura lógica.

No es válido poner dos comandos en una misma línea.

A.1.2 Decisión

a) Simple (IF- THEN)

i) Para ejecutar un comando

```
| |IF (expresión) comando
```

se ejecuta cuando expresión = TRUE

ii) Para ejecutar dos o más comandos.

```
| |IF(.NOT.(expresión)) GO TO N1
| |==========================
N1| |CONTINUE
```

se ejecuta cuando expresión = TRUE

b) Doble (IF-THEN-ELSE)

```
      IF(.NOT.(expresión)) GO TO N1
      ____________________
      ____________________  se ejecuta cuando
      ____________________  expresión = TRUE
      ____________________
      GO TO N2
N1    CONTINUE
      ____________________
      ____________________  se ejecuta cuando
      ____________________  expresión = FALSE
      ____________________
N2    CONTINUE
```

c) Múltiple (CASE)

Se implementa con el GO TO computado en la si---- guiente forma :

```
      GO TO (N1,N2,N3,...Nk), e ← expresión entera
      ____________________
      ____________________  el control pasa aquí
      ____________________  cuando e<1 ó e>k
      ____________________  * No es estándar
      GO TO Nk + 1
N1    CONTINUE
      ____________________
      ____________________  Comandos para e = 1
      ____________________
      GO TO Nk + 1
N2    CONTINUE
      ____________________
      ____________________  Comandos para e = 2
      ____________________
      GO TO Nk + 1
N3    CONTINUE
      •
      •
Nk    CONTINUE
      ____________________
      ____________________  Comandos para e = k
      ____________________
Nk+1  CONTINUE
```

A.1.3 Repetición

a) WHILE

```
N1 | CONTINUE
   | IF (.NOT.(expresión)) GO TO N2
   |
   |                    \ se ejecuta cuando
   |                     \ expresión = TRUE
   | GO TO N1
N2 | CONTINUE
```

b) UNTIL

```
N1 | CONTINUE
   |                    \ se ejecuta por lo menos una
   |                     \ vez y cuando expresión = -
   |                     \ FALSE
   | IF (.NOT.(expresión)) GO TO N1
```

c) FOR/DO

```
   | DO N1  V = e1, e2 (,e3)
   |                    incremento
   |                    valor final de V
   |                    valor inicial de V
N1 | CONTINUE
```

El cuerpo de loop se ejecuta por lo menos una --- vez y hasta que :

$V > e_2$	si	$e_3>0$	$e_3 \neq 0$
$V < e_2$	si	$e_3<0$	$e_3=1$ por default

A.1.4 Anidamientos

El anidamiento de estructuras lógicas y de control no debe exceder de 4 niveles.

Para el caso del IF-THEN-ELSE anidado, la estructura debe ser :

```
      IF(.NOT.(expresión1))GO TO N3
          IF(.NOT.(expresión2)) GO TO N1
          ____________________
          ____________________   se ejecuta cuando
          ____________________   expresión1=TRUE
          GO TO N2               expresión2=TRUE   Parte 1

N1    CONTINUE
          ____________________
          ____________________   se ejecuta cuando
          ____________________   expresión1=TRUE
          ____________________   expresión2=FALSE  Parte 2
          ____________________
          ____________________
N2    CONTINUE
      GO TO N6

N3    CONTINUE

      IF (.NOT.(expresión3)) GO TO N4
          ____________________
          ____________________   se ejecuta cuando
          ____________________   expresión1=FALSE y
          ____________________   expresión3=TRUE   Parte 3
          ____________________
          ____________________
            GO TO N5

N4    CONTINUE
          ____________________
          ____________________   se ejecuta cuando
          ____________________   expresión1=FALSE y
          ____________________   expresión3=TRUE   Parte 4
          ____________________
          ____________________
N5    CONTINUE

N6    CONTINUE
```

Si se requieren más de 4 niveles de anidamiento, emplear módulos.

EJEMPLO A.2

```
C     PROGRAMA PRINCIPAL
C**********************************************************************
C*
C*    OBJETO
C*    EFECTUAR PRONOSTICO DE CARGA MEDIANTE SUAVIZAMIENTO EXPO-*
C*    NENCIAL (FFTOES) EMPLEANDO EL METODO DE CHRISTIAANSE.   *
C*    SE EMPLEA UN MODELO RESIDUAL PARA CADA DIA DE LA SEMANA.   *
C*
C*    REALIZADO EN EL IIE/DIVISION DE SISTEMAS DE POTENCIA
C*
C*    IMPLEMENTADO POR JOSE ARMANDO TORRES FENTANES
C*    FECHA ULTIMA REVISION     AGOSTO   11,1980
C*    REVISADO POR MOTOKI INOUE
C*
C*    ENTRADAS
C*     FECHA - FECHA CORRESPONDIENTE A UNA MEDICION
C*     FORM - FORMATO PARA LECTURA DE PUNTOS MUESTRALES
C*     IM - ARCHIVO CON VALORES REALES(MEDIDOS) DE CARGA
C*     INIFEC - FECHA DEL PRIMER DIA DE PRONOSTICOS
C*     IPAR - ARCHIVO CON PARAMETROS DE ENTRADA
C*     LONG - LONGITUD DE LA VENTANA
C*     MAD - MAXIMO ADELANTO EN PRONOSTICOS
C*     NARM - NUMERO DE ARMONICAS (INCLUYENDO LA FUNDAMENTAL)
C*     NP - NUMERO DE PUNTOS A PRONOSTICAR
C*     NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS
C*     TF - PERIODO FUNDAMENTAL
C*     TITULO - NOMBRE DE LA SERIE
C*     XR - VALORES REALES(MEDIDOS) DE LA CARGA
C*
C*    SALIDAS
C*     ALFA - VELOCIDAD ANGULAR FUNDAMENTAL
C*     IP - ARCHIVO CON VALORES PRONOSTICADOS
C*     IU - ARCHIVO DE IMPRESION CON RESULTADOS Y ESTADISTICAS
C*     NDIA - NUMERO DEL DIA PARA EL PRONOSTICO CON 1 HORA DE AD-*
C*
C*    MODULOS EMPLEADOS
C*     ACTUAL - ACTUALIZA PARAMETROS DEL MODELO
C*     ESTAD - OBTIENE ESTADISTICAS DE LOS PRONOSTICOS
C*     INICIA - OBTIENE PARAMETROS INICIALES DEL MODELO
C*     [illegible] RESULTADOS DE LOS PRONOSTICOS
C*     NVOPRO - OBTIENE NUEVO PRONOSTICO
C*
C*    IDENTIFICADORES LOCALES
C*     ENCONT - TRUE SI SE ENCONTRO LA FECHA INICIAL EN ARCH. IM*
C*     FF - FRECUENCIA FUNDAMENTAL (1/TF)
C*     HORA - HORA DEL DIA
C*     I - CONTADOR Y NUMERO DEL PUNTO DE PRONOSTICO
C*     J - CONTADOR
C*     NDSEM - NUMERO DE DIAS EN LA SEMANA
C*     NHSDIA - NUMERO DE HORAS EN UN DIA
C*
C*    NOTAS.- CONDICION PARA FINALIZAR EJECUCION ES NPDIA<=0
C*     LA DIMENSION DE LOS ARREGLOS DEBE SATISFACER
C*              XP(MAD)
C*     LOS DIAS DE LA SEMANA SE NUMERAN DEL 1-7 A PARTIR
C*     DEL LUNES
C*
C**********************************************************************
```

```
C     PROGRAMA PRINCIPAL
C*****************************************************************
C                                                                 *
C     OBJETO                                                      *
C           EFECTUAR PRONOSTICO DE CARGA MEDIANTE SUAVIZAMIENTO EXPO-*
C           NENCIAL (FFTGES) EMPLEANDO EL METODO DE CHRISTIAANSE.  *
C           SE EMPLEA UN MODELO RESIDUAL PARA CADA DIA DE LA SEMANA. *
C                                                                 *
C     REALIZADO EN EL IIE/DIVISION DE SISTEMAS DE POTENCIA        *
C                                                                 *
C     IMPLEMENTADO POR JOSE ARMANDO TORRES FENTANES               *
C     FECHA ULTIMA REVISION    AGOSTO  11,1980                    *
C     REVISADO POR MOTOKI INOUE                                   *
C                                                                 *
C     ENTRADAS                                                    *
C           FECHA - FECHA CORRESPONDIENTE A UNA MEDICION          *
C           FORM - FORMATO PARA LECTURA DE PUNTOS MUESTRALES      *
C           IM - ARCHIVO CON VALORES REALES(MEDIDOS) DE CARGA     *
C           INIFEC - FECHA DEL PRIMER DIA DE PRONOSTICOS          *
C           IPAR - ARCHIVO CON PARAMETROS DE ENTRADA              *
C           LONG - LONGITUD DE LA VENTANA                         *
C           MAD - MAXIMO ADELANTO EN PRONOSTICOS                  *
C           NARM - NUMERO DE ARMONICAS (INCLUYENDO LA FUNDAMENTAL) *
C           NP - NUMERO DE PUNTOS A PRONOSTICAR                   *
C           NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS          *
C           TF - PERIODO FUNDAMENTAL                              *
C           TITULO - NOMBRE DE LA SERIE                           *
C           XR - VALORES REALES(MEDIDOS) DE LA CARGA              *
C                                                                 *
C     SALIDAS                                                     *
C           ALFA - VELOCIDAD ANGULAR FUNDAMENTAL                  *
C           IP - ARCHIVO CON VALORES PRONOSTICADOS                *
C           IW - ARCHIVO DE IMPRESION CON RESULTADOS Y ESTADISTICAS *
C           NDIA - NUMERO DEL DIA PARA EL PRONOSTICO CON 1 HORA DE AD*
C                                                                 *
C     MODULOS EMPLEADOS                                           *
C           ACTUAL - ACTUALIZA PARAMETROS DEL MODELO              *
C           ESTAD - OBTIENE ESTADISTICAS DE LOS PRONOSTICOS       *
C           INICIA - OBTIENE PARAMETROS INICIALES DEL MODELO      *
C           LISTA - LISTA RESULTADOS DE LOS PRONOSTICOS           *
C           NVOPRO - OBTIENE NUEVO PRONOSTICO                     *
C                                                                 *
C     IDENTIFICADORES LOCALES                                     *
C           ENCONT - TRUE SI SE ENCONTRO LA FECHA INICIAL EN ARCH. IM*
C           FF - FRECUENCIA FUNDAMENTAL (1/TF)                    *
C           HORA - HORA DEL DIA                                   *
C           I - CONTADOR Y NUMERO DEL PUNTO DE PRONOSTICO         *
C           J - CONTADOR                                          *
C           NDSEM - NUMERO DE DIAS EN LA SEMANA                   *
C           NHSDIA - NUMERO DE HORAS EN UN DIA                    *
C                                                                 *
C     NOTAS.- CONDICION PARA FINALIZAR EJECUCION ES NPDIA<=0      *
C             LA DIMENSION DE LOS ARREGLOS DEBE SATISFACER        *
C                       XP(MAD)                                   *
C             LOS DIAS DE LA SEMANA SE NUMERAN DEL 1-7 A PARTIR   *
C             DEL LUNES                                           *
C                                                                 *
C*****************************************************************
```

```
C
      CHARACTER*4 FORM,TITULO
      INTEGER FECHA,HORA,I,IM,IMPTR,INIFEC,IP,IPAR,IPPTR,IW,J,
     *        LONG,MAD,NARM,NDIA,NDSEM,NHSDIA,NP,NPDIA
      LOGICAL ENCONT
      REAL ALFA,FF,PI,TF,XR,XP
C
      DIMENSION FORM(20),TITULO(4)
      COMMON/FILE1/IW,IPAR
      COMMON/FILE2/IM,IP
      COMMON/PRON/XP(24)
      COMMON/REAL/FECHA(3),XR
C     ----------------------------------------
C     *PARAMETROS CONSTANTES
C     ----------------------------------------
      DATA IM/16/,IP/17/,IPAR/14/,IW/6/,PI/3.141592/
      DATA NDSEM/7/,NHSDIA/24/
C     ----------------
C     *ABRIR ARCHIVOS
C     ----------------
      REWIND IM
      REWIND IP
      REWIND IPAR
C     ----------------------------------------
C     *LEER VARIABLE DE CONTROL Y DATOS INICIALES
C     ----------------------------------------
      READ(IPAR,5000) NPDIA,INIFEC,TITULO
100   CONTINUE
      IF(.NOT.(NPDIA.GT.0)) GO TO 400
C     ----------------------------------------
C     *LEER PARAMETROS Y EVALUAR FRECUENCIA FUNDAMENTAL
C     ----------------------------------------
      READ(IPAR,5100) NP,NARM,MAD,LONG,TF
      READ(IPAR,5200) FORM
      FF=1./TF
      ALFA=2.*PI*FF
C     ----------------
C     *IMPRIMIR DATOS
C     ----------------
      WRITE(IW,5300) TITULO,FORM
      WRITE(IW,5400) NPDIA,NP,NARM,MAD,LONG,FF
C     ----------------------------------------
C     *OBTENER PARAMETROS INICIALES DEL MODELO
C     ----------------------------------------
      CALL INICIA(NARM,ALFA)
C     ----------------------------------------
C     *POSICIONAR APUNTADOR DEL ARCHIVO DE VALORES REALES
C     ----------------------------------------
      REWIND IM
200   CONTINUE
      READ(IM,FORM) FECHA,XR
      ENCONT=(FECHA(1).EQ.INIFEC(1)).AND.(FECHA(2).EQ.
     *        INIFEC(2)).AND.(FECHA(3).EQ.INIFEC(3))
      IF(.NOT.ENCONT) GO TO 200
```

```
C        ------------------------------------
C        *OBTENER PRONOSTICOS DEL PUNTO I
C        ------------------------------------
         NDIA=NPDIA-1
         DO 300 I=1,NP
C        ------------------------------------
C        *OBTENER DIA DEL PRONOSTICO
C        ------------------------------------
         HORA=MOD(I,NHSDIA)
         IF(HORA.EQ.1) NDIA=NDIA+1
C        ------------------------------------
C        *OBTENER PRONOSTICO
C        ------------------------------------
         CALL NVOPRO(I,MAD,NARM,ALFA,NDIA)
C        ------------------------------------
C        *GUARDAR PRONOSTICOS EN ARCHIVO IP
C        ------------------------------------
         WRITE(IP,5500) (XP(J),J=1,MAD)
C        ------------------------------------
C        *ACTUALIZAR PARAMETROS DEL MODELO
C        ------------------------------------
         CALL ACTUAL(NARM,NDIA,I)
C        ------------------------------------
C        *LEER VALOR REAL DE LA CARGA
C        ------------------------------------
         READ(IM,FORM) FECHA,XR
300    CONTINUE
C        ------------------------------------
C        *IMPRIMIR RESULTADOS
C        ------------------------------------
       CALL LISTA(NPDIA,NP,MAD,LONG)
C        ------------------------------------
C        *OBTENER ESTADISTICAS DE LOS PRONOSTICOS
C        ------------------------------------
       CALL ESTAD(NPDIA,NP,MAD,LONG)
C        ------------------------------------
C        *VERIFICAR SI SE DESEA PRONOSTICAR OTRA SERIE
C        ------------------------------------
       READ(IPAR,5000) NPDIA,INIFEC,TITULO
       GO TO 100
400    CONTINUE
       CALL EXIT
C        ------------------------------------
C        *FORMATOS DE LECTURA E IMPRESION
C        ------------------------------------
5000   FORMAT(I2,3I2,4A4)
5100   FORMAT(4I5,F10.0)
5200   FORMAT(20A4)
5300   FORMAT(1H1,//,5X,'SIMULACION DE PRONOSTICO DE CARGA',//,5X,
      *'METODO=FFTES',///,5X,'TITULO=',4A4,///,5X,'FORMATO= ',20A4)
5400   FORMAT(///,5X,'PRIMER DIA DE PRONOSTICOS=',
      *I5,//,5X,'CANTIDAD DE PUNTOS PARA EL PRONOSTICO=',I5,//,5X,
      *'CANTIDAD DE MULTIPLOS DE LA FREC.FUNDAMENTAL=',I5,//,5X,
      *'MAXIMO ADELANTO (HRS) PARA PRONOSTICO=',I5,//,5X,
      *'CANTIDAD DE PUNTOS PARA EL INTERVALO BASE DE PRONOSTICO=',I5,/
      *,5X,'FRECUENCIA FUNDAMENTAL=',1PE12.5)
5500   FORMAT(20F6.0)
       END
```

```
      SUBROUTINE INICIA(NARM,ALFA)
C*********************************************************************
C                                                                    *
C     OBJETO                                                         *
C          OBTENER PARAMETROS DE LOS MODELOS RESIDUALES PARA CADA    *
C          DIA DE LA SEMANA, LA MATRIZ DE TRANSICION Y LOS VECTO-    *
C          RES DE SUAVIZAMIENTO EXPONENCIAL                          *
C                                                                    *
C     IMPLEMENTADA POR JOSE ARMANDO TORRES FENTANES                  *
C     FECHA ULTIMA REVISION OCTUBRE 26,1979                          *
C     REVISADA POR MOTOKI INOUE                                      *
C                                                                    *
C     ENTRADAS                                                       *
C          ALFA - VELOCIDAD ANGULAR FUNDAMENTAL                      *
C          IPAR - ARCHIVO CON PARAMETROS                             *
C          NARM - NUMERO DE ARMONICAS                                *
C                                                                    *
C     SALIDAS                                                        *
C          ARM - ARMONICAS A EMPLEAR                                 *
C          BETA - CONSTANTES DE SUAVIZAMIENTO EXPONENCIAL            *
C          C Y S - ELEMENTOS DE LA MATRIZ DE TRANSICION              *
C          CARHOR - CARGA HORA-SEMANA PROMEDIO                       *
C          PAR - PARAMETROS DEL MODELO PARA CADA DIA DE LA SEMANA    *
C          SH - VECTORES DE SUAVIZAMIENTO EXPONENCIAL                *
C          VARHOR - VARIANCIA DE LA CARGA HORA-SEMANA                *
C                                                                    *
C     MODULOS REQUERIDOS                                             *
C          NINGUNO                                                   *
C                                                                    *
C     IDENTIFICADORES LOCALES                                        *
C          DOM - DIA DOMINGO                                         *
C          I - CONTADOR                                              *
C          J - CONTADOR                                              *
C          LUN - DIA LUNES                                           *
C          NHSDIA - NUMERO DE HORA EN UN DIA                         *
C          NPAR - NUMERO DE PARAMETROS PARA EL MODELO DIARIO         *
C          TEMP - VARIABLE TEMPORAL                                  *
C                                                                    *
C     NOTAS.-LA DIMENSION DE LOS ARREGLOS DEBE SATISFACER LO SIG.:   *
C                    ARM(NF)                                         *
C                    BETA(7)                                         *
C                    C(NF)                                           *
C                    PAR(2NF+1,7)                                    *
C                    S(NF)                                           *
C                    SH(2NF+1,7)                                     *
C          LOS DIAS DE LA SEMANA SE NUMERAN DEL 1 AL 7 A PARTIR      *
C          DEL LUNES                                                 *
C                                                                    *
C*********************************************************************
C
      INTEGER ARM,DOM,I,IPAR,J,LUN,NARM,NHSDIA,NPAR
      REAL ALFA,BETA,C,CARHOR,PAR,S,SH,TEMP,VARHOR
```

```
C*****************************************************************
      COMMON/FILE1/IW,IPAR
      COMMON/PAR/PAR(30,7)
      COMMON/SHBETA/SH(30,7),BETA(7)
      COMMON/STAT/CARHOR(24,7),VARHOR(24,7)
      COMMON/TRANS/C(15),S(15)
C     ------------------------------------------------------
C     *PARAMETROS CONSTANTES
C     ------------------------------------------------------
      DATA LUN/1/,DOM/7/,NHSDIA/24/
C     ------------------------------------------------------
C     *LEER CONSTANTES DE SUAVIZAMIENTO EXPONENCIAL
C     ------------------------------------------------------
      READ(IPAR,5000) BETA
      WRITE(IW,5100) BETA
C     ------------------------------------------------------
C     *LEER ARMONICAS
C     ------------------------------------------------------
      READ(IPAR,5200) (ARM(I),I=1,NARM)
      WRITE(IW,5300) (ARM(I),I=1,NARM)
C     ------------------------------------------------------
C     *CALCULAR NUMERO DE PARAMETROS DEL MODELO DIARIO
C     ------------------------------------------------------
      NPAR=2*NARM+1
C     ------------------------------------------------------
C     *OBTENER PARAMETROS DE LOS MODELOS DIARIOS
C     ------------------------------------------------------
      WRITE(IW,5400)
      DO 100 I=1,NPAR
         READ(IPAR,5500) (PAR(I,J),J=LUN,DOM)
         WRITE(IW,5600) (PAR(I,J),J=LUN,DOM)
100   CONTINUE
C     ------------------------------------------------------
C     *GENERAR COMPONENTES DE LA MATRIZ DE TRANSICION
C        (SENOS Y COSENOS DE LAS ARMONICAS)
C     ------------------------------------------------------
      DO 200 I=1,NARM
         TEMP=ALFA*FLOAT(ARM(I))
         C(I)=COS(TEMP)
         S(I)=SIN(TEMP)
200   CONTINUE
C     ------------------------------------------------------
C     *OBTENER VECTORES DE SUAVIZAMIENTO EXPONENCIAL SH
C     ------------------------------------------------------
      WRITE(IW,5700)
      DO 300 I=1,NPAR
         READ(IPAR,5500) (SH(I,J),J=LUN,DOM)
         WRITE(IPAR,5600) (SH(I,J),J=LUN,DOM)
300   CONTINUE
*****************************************************************
```

```
C     ------------------------------------------------------------
C     *OBTENER CARGA PROMEDIO Y VARIANCIA PARA CADA HORA
C      DE LA SEMANA
C     ------------------------------------------------------------
      WRITE(IW,5800)
      DO 400 I=1,NHSDIA
         READ(IPAR,5500) (CARHOR(I,J),J=LUN,DOM)
         WRITE(IW,5600) (CARHOR(I,J),J=LUN,DOM)
400   CONTINUE
      WRITE(IW,5900)
      DO 500 I=1,NHSDIA
         READ(IPAR,5500) (VARHOR(I,J),J=LUN,DOM)
         WRITE(IW,5600) (VARHOR(I,J),J=LUN,DOM)
500   CONTINUE
      WRITE(IW,6000)
      DO 600 I=1,NHSDIA
         WRITE(IW,5600) (SQRT(VARHOR(I,J)),J=LUN,DOM)
600   CONTINUE
      RETURN
C     ------------------------------------------------------------
C     *FORMATOS DE LECTURA E IMPRESION
C     ------------------------------------------------------------
5000  FORMAT(7F6.0)
5400  FORMAT(/,5X,'CONSTANTES DE SUAVIZAMIENTO EXPONENCIAL'
     *'(LUN-DOM)',/,7(/,10X,1PE12.5))
5200  FORMAT(20I4)
5300  FORMAT(/,5X,'MULTIPLOS DE LA FRECUENCIA FUNDAMENTAL'
     *20(/,10,I4))
5400  FORMAT(/,5X,'PARAMETROS AR',/)
5500  FORMAT(7(1PE11.4))
5600  FORMAT(9(2X,1PE12.5))
5700  FORMAT(/,5X,'VECTORES SH',/)
5800  FORMAT(////,5X,'CARGAS PROMEDIO HORARIAS(LUN-DOM)',/)
5900  FORMAT(////,5X,'VARIANCIA DE LA CARGA HORARIA(LUN-DOM)',/)
6000  FORMAT(////,5X,'DESV.ESTANDAR DE LA CARGA HORARIA(LUN-DOM)',/)
      END
```

```
      SUBROUTINE NVOPROC(IPUNTO,MAD,NARM,ALFA,NDIA)
C*****************************************************************
C                                                                 *
C     OBJETO                                                      *
C          OBTENER LOS PRONOSTICOS DE IPUNTO HASTA CON MAD        *
C          UNIDADES DE ADELANTO. SE EMPLEA UN MODELO PARA CADA DIA *
C          DE LA SEMANA.                                          *
C                                                                 *
C     IMPLEMENTADA POR JOSE ARMANDO TORRES FENTANES               *
C     FECHA ULTIMA REVISION  OCTUBRE 26, 1979                     *
C     REVISADA POR MOTOKI INOUE                                   *
C                                                                 *
C     ENTRADAS                                                    *
C          ALFA - VELOCIDAD ANGULAR FUNDAMENTAL                   *
C          ARM - MULTIPLOS DE LA FRECUENCIA FUNDAMENTAL(ARMONICAS) *
C          IPUNTO - PUNTO EN EL CUAL SE DESEAN LOS PRONOSTICOS    *
C          MAD - MAXIMO ADELANTO EN LOS PRONOSTICOS               *
C          NARM - NUMERO DE ARMONICAS                             *
C          NDIA - NUMERO DEL DIA DE PRONOSTICO PARA UNA HORA DE AD. *
C          PAR - PARAMETROS DEL MODELO                            *
C                                                                 *
C     SALIDAS                                                     *
C          XP - VALORES PRONOSTICADOS DE CARGA                    *
C                                                                 *
C     MODULOS EMPLEADOS                                           *
C          NINGUNO                                                *
C                                                                 *
C     IDENTIFICADORES LOCALES                                     *
C          DOM - DIA DOMINGO                                      *
C          I - CONTADOR                                           *
C          J - CONTADOR                                           *
C          L - CONTADOR                                           *
C          NDPRON - NUMERO DEL DIA DE LA HORA QUE SE PRONOSTICARA *
C          NDSEM - NUMERO DE DIAS DE LA SEMANA                    *
C          NHBASE - HORA PARA EL PRONOSTICO CON UNA HORA DE ADELANTO*
C          NHORA - HORA PARA EVALUAR EL PRONOSTICO CON RESPECTO A LA*
C                 ULTIMA ACTUALIZACION DEL MODELO DEL DIA          *
C          NHSDIA - NUMERO DE HORAS EN UN DIA                     *
C          NHTOT - HORA DEL TOTAL ACUMULADO QUE SE DESEA PRONOSTICAR*
C          TETA - VARIABLE TEMPORAL                               *
C                                                                 *
C     NOTAS.-LOS DIAS DE LA SEMANA SE NUMERAN DEL 1 AL 7 A PARTIR DEL*
C            LUNES                                                *
C*****************************************************************
C
      INTEGER DOM,I,IPUNTO,J,L,MAD,NARM,NDIA,NDPRON,NDSEM,NHBASE,
     *        NHORA,NHSDIA,NHTOT
      REAL ALFA,ARM,PAR,XP
C
      COMMON/ARM/ARM(15)
      COMMON/PAR/PAR(30,7)
      COMMON/PRON/XP(24)
```

```
C
C     *PARAMETROS CONSTANTES
C     ----------------------
      DATA DOM/7/,NDSEM/7/,NHSDIA/24/
C     -------------------------------------------------------
C     *OBTENER DIA Y HORA PARA PRONOSTICO CON UNA HORA
C      DE ADELANTO
C     -------------------------------------------------------
      NDPRON=NDIA
      NHBASE=MOD(IPUNTO,NHSDIA)
      IF(NHBASE.EQ.0) NHBASE=24
C     -------------------------------------------------------
C     *OBTENER PRONOSTICOS HASTA CON MAD UNIDADES DE ADELANTO
C     -------------------------------------------------------
      DO 400 J=1,MAD
C     -------------------------------------------------------
C     *OBTENER HORA Y DIA PARA EL PRONOSTICO CON J UNIDADES
C      DE ADELANTO
C     -------------------------------------------------------
      NHTOT=IPUNTO + J - 1
      NHORA=MOD(NHTOT,NHSDIA)
      IF(NHORA.EQ.0) NHORA=24
      IF((NHORA.EQ.1).AND.(J.NE.1)) NDPRON=NDPRON+1
      NDPRON=MOD(NDPRON,NDSEM)
      IF(NDPRON.EQ.0) NDPRON=DOM
C     -------------------------------------------------------
C     *OBTENER PUNTO DE EVALUACION DEL MODELO DEL DIA DE
C      ACUERDO CON SU ULTIMA EVALUACION
C     -------------------------------------------------------
      IF(.NOT.(NDPRON.EQ.NDIA)) GO TO 100
         NHORA=NHORA-NHBASE
         IF(NHORA.LT.0) NHORA=NHORA+24
         GO TO 200
100   CONTINUE
         NHORA=NHORA-1
200   CONTINUE
C     ------------------------
C     *OBTENER PRONOSTICO
C     ------------------------
      XP(J)=PAR(1,NDPRON)
      L=2
      DO 300 I=1,NARM
         TETA=ALFA*FLOAT(ARM(I))*FLOAT(NHORA)
         XP(J)=XP(J) + PAR(L,NDPRON)*SIN(TETA) +
               PAR(L+1,NDPRON)*COS(TETA)
         L=L+2
300   CONTINUE
400   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ESTAD(NPDIA,NP,MAD,LONG)
C*****************************************************************
C                                                                 *
C     OBJETO                                                      *
C          EVALUAR RESULTADOS DE LOS PRONOSTICOS                  *
C                                                                 *
C     IMPLEMENTADA POR JOSE ARMANDO TORRES FENTANES               *
C     FECHA ULTIMA REVISION OCTUBRE 26,1979                       *
C     REVISADA POR MOTOKI INOUE                                   *
C                                                                 *
C     ENTRADAS                                                    *
C          LONG - LONGITUD DE LA VENTANA(168 PARA SEMANA)         *
C          MAD - MAXIMO ADELANTO EN LOS PRONOSTICOS               *
C          NP - NUMERO DE PUNTOS PRONOSTICADOS                    *
C          NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS           *
C                                                                 *
C     SALIDAS                                                     *
C          LONG - LONGITUD DE LA VENTANA(168 PARA SEMANA)         *
C          MAD - MAXIMO ADELANTO EN LOS PRONOSTICOS               *
C          NHSDIA - NUMERO DE HORAS EN UN DIA                     *
C          NP - NUMERO DE PUNTOS PRONOSTICADOS                    *
C          NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS           *
C          ESTADISTICAS DIARIAS POR DIA,HORA Y TIEMPO DE ADELANTO *
C                                                                 *
C     MODULOS REQUERIDOS                                          *
C          DIA - ESTADISTICAS DIARIAS CON UNA HORA DE ADELANTO    *
C          DIACIA - ESTADISTICAS DIARIAS POR BLOQUES DE 24 HORAS  *
C          HORA - ESTADISTICAS POR HORA DE LA SEMAN               *
C          TOTAL - ESTADISTICAS PARA CADA TIEMPO DE ADELANTO      *
C                                                                 *
C     IDENTIFICADORES LOCALES                                     *
C          NINGUNO                                                *
C                                                                 *
C*****************************************************************
C
      INTEGER LONG,MAD,NHSDIA,NP,NPDIA
C     -------------------------
C     *PARAMETROS CONSTANTES
C     -------------------------
      DATA NHSDIA/24/
C     ---------------------------------------------------------
C     *OBTENER ESTADISTICAS PARA EL TOTAL DE LA SIMULACION
C      COMPARANDO ESTADISTICOS DE CADA TIEMPO DE ADELANTO
C     ---------------------------------------------------------
      CALL TOTAL(NP,MAD)
C     ---------------------------------------------------------
C     *OBTENER ESTADISTICAS DIARIAS PARA UNA HORA DE ADELANTO
C     ---------------------------------------------------------
      CALL DIA(NPDIA,NP,MAD,NHSDIA)
C     ---------------------------------------------------------
C     *OBTENER ESTADISTICAS PARA PRONOSTICOS HECHOS A LA
C      HORA UNO POR BLOQUES DE 24 HORAS
C     ---------------------------------------------------------
      CALL DIACIA(NPDIA,NP,MAD,NHSDIA)
C     ---------------------------------------------------------
C     *OBTENER ESTADISTICAS POR HORA DE LA SEMANA PARA CADA TIEMPO
C      DE ADELANTO
C     ---------------------------------------------------------
      CALL HORA(NP,MAD,LONG)
      RETURN
      END
```

```
      SUBROUTINE ESTDIA(NUMDIA,NHS)
C*****************************************************************
C                                                                *
C     OBJETO                                                     *
C          EVALUAR LAS ESTADISTICAS CORRESPONDIENTES A UN DIA    *
C                                                                *
C     IMPLEMENTADA POR JOSE A. TORRES FENTANES                   *
C     FECHA ULTIMA REVISION  ENERO 25 1980                       *
C     REVISADA POR MOTOKI INOUE                                  *
C                                                                *
C     ENTRADAS                                                   *
C          A(I,1) - HORA DEL DIA                                 *
C          A(I,2) - CARGA REAL                                   *
C          A(I,3) - CARGA PRONOSTICADA                           *
C          FECHA - FECHA DEL DIA                                 *
C          NHS - PERIODO DE EVALUACION (24 PARA DIA)             *
C          NUMDIA - NUMERO DEL DIA                               *
C          XR - VALOR REAL DE LA CARGA                           *
C                                                                *
C     SALIDAS                                                    *
C          BASE - CARGA MINIMA(MWH)                              *
C          DPICO - DIFERENCIA ENTRE PICOS(%)                     *
C          EABS - EROR ABSOLUTO PROMEDIO (%)                     *
C          ECARG - ERROR EN CARGA TOTAL DIARIA(%)                *
C          EMAXN - ERROR MAXIMO NEGATIVO DIARIO(%)               *
C          EMAXP - ERROR MAXIMO POSITIVO DIARIO(%)               *
C          ENERGP - ENERGIA PRONOSTICADA DEL DIA(MWH)            *
C          ENERGR - ENERGIA REAL DEL DIA(MWH)                    *
C          EPICO - ERROR EN EL PUNTO DEL PICO REAL(%)            *
C          HDIF - DIFERENCIA HORARIA ENTRE PICOS                 *
C          RMS - ERROR MEDIO CUADRATICO DIARIO(MWH)              *
C                                                                *
C     MODULOS EMPLEADOS                                          *
C          NINGUNO                                               *
C                                                                *
C     IDENTIFICADORES LOCALES                                    *
C          ERROR - ERRORES HORARIOS(%)                           *
C          I - CONTADOR                                          *
C          J - CONTADOR                                          *
C          HORA - HORA CONSIDERADA                               *
C          HPICOP - HORA DEL PICO PRONOSTICADO                   *
C          HPICOR - HORA DEL PICO REAL                           *
C          MES - NOMBRE DE LOS MESES                             *
C          NOMDIA - NOMBRE DE LOS DIAS                           *
C          XP(I) - PRONOSTICO DEL VALOR REAL CON I HORAS DE ADELANTO*
C          XPMAX - MAXIMO VALOR EN EL DIA DE LA CARGA PRONOSTICADA *
C          XRMAX - MAXIMO VALOR EN EL DIA DE LA CARGA REAL        *
C          Z - ERROR EN CARGA,NORMALIZADO CON RESPECTO AL VALOR REAL*
C                                                                *
C     NOTAS.-LOS DIAS DE LA SEMANA SE NUMERAN DEL 1 AL 7  A PARTIR *
C            DEL LUNES                                           *
C                                                                *
C*****************************************************************
```

```
      CHARACTER*3 MES,NOMDIA
      INTEGER FECHA,HDIF,HORA,HPICOP,HPICOR,I,J,NHS,NUMDIA
      REAL A,BASE,CARGP,CARGR,DPICO,EABS,ECARG,EMAXN,EMAXP,EPICO,
     *     ERROR,RMS,XR,XRMAX,XP,XPMAX,Z
C
      DIMENSION ERROR(24),MES(12),NOMDIA(7),XP(24)
      COMMON/DATOS/A(24,3)
      COMMON/FILE1/IW,IPAR
      COMMON/REAL/FECHA(3),XR
      COMMON/ESTDIA/BASE,CARGR,CARGP,EABS,ECARG,EPICO,RMS
C     -------------------------
C     *PARAMETROS CONSTANTES
C     -------------------------
      DATA MES/3HENE,3HFEB,3HMZO,3HABR,3HMAY,3HJUN,3HJUL,3HAGO,
     *     3HSEP,3HOCT,3HNOV,3HDIC/
      DATA NOMDIA/3HLUN,3HMAR,3HMIE,3HJUE,3HVIE,3HSAB,3HDOM/
C     ---------------------------------------
C     *INICIALIZAR CONTADORES DEL DIA
C     ---------------------------------------
      RMS=0.
      EABS=0.
      ECARG=0.
      CARGR=0.
      CARGP=0.
      EMAXP=-1.0E+20
      EMAXN=1.0E+20
      BASE=1.0E+20
C     ---------------------------------------
C     *OBTENER ESTADISTICAS DEL DIA
C     ---------------------------------------
      DO 500 HORA=1,NHS
         XR=A(HORA,2)
         XP(HORA)=A(HORA,3)
         Z=(XR-XP(HORA))/XR
         IF(EMAXP.LT.Z) EMAXP=Z
         IF(EMAXN.GT.Z) EMAXN=Z
         IF(BASE.GT.XR) BASE=XR
         CARGR=CARGR + XR
         CARGP=CARGP + XP(HORA)
         EABS=EABS + ABS(Z)
         RMS=RMS + (XR-XP(HORA))**2
         ERROR(I)=Z*100.
         IF(.NOT.(HORA.EQ.1)) GO TO 100
            HPICOP=1
            HPICOR=1
            XRMAX=XR
            XPMAX=XP(1)
            EPICO=Z
            GO TO 400
100      CONTINUE
            IF(.NOT.(XRMAX.LT.XR)) GO TO 200
               XRMAX=XR
               HPICOR=HORA
               EPICO=Z
200         CONTINUE
            IF(.NOT.(XPMAX.LT.XP(HORA)) GO TO 300
               XPMAX=XP(HORA)
               HPICOP=HORA
300         CONTINUE
400      CONTINUE
500   CONTINUE
```

```
C     ------------------------------------------------
C     *OBTENER ESTADISTICAS DEL DIA EN POR CIENTO
C     ------------------------------------------------
      ECARG=100.*ABS(CARGR-CARGP)/CARGR
      RMS=RMS/FLOAT(NHS)
      EABS=100.*EABS/FLOAT(NHS)
      EMAXP=100.*EMAXP
      EMAXN=100.*EMAXN
      EPICO=100.*EPICO
      DPICO=100.*(XRMAX-XPMAX)/XRMAX
      HDIF=HPICOR-HPICOP
C     ------------------------------------------------
C     *IMPRIMIR VALOR REAL, ESTIMADO Y ERROR HORARIO
C      PARA EL DIA
C     ------------------------------------------------
      WRITE(IW,5000) NOMDIA(NUMDIA),FECHA(1),MES(FECHA(2)),FECHA(3)
      DO 600 I=1,NHS
         WRITE(IW,5100) I,(A(I,J),J=2,3),ERROR(I)
600   CONTINUE
C     ------------------------------------
C     *IMPRIMIR ESTADISTICAS DEL DIA
C     ------------------------------------
      WRITE(IW,5200) EABS,SQRT(RMS),ECARG,EMAXP,EMAXN,EPICO,DPICO,
     *               HDIF,BASE
C     ----------------------------------------
C     *GRAFICAR VALORES REALES Y PRONOSTICADOS
C     ----------------------------------------
      WRITE(IW,5300)
      CALL GRAFI(NHS,3)
      RETURN
C     --------------------------
C     *FORMATOS DE IMPRESION
C     --------------------------
5000  FORMAT(///,5X,'PRONOSTICOS DEL DIA ',A3,' (',I2,' DE ',A3,
     *' DE 19',I2,')',//,5X,'HORA',6X,'CARGA REAL',7X,'PRONOSTICO',8X,
     *'ERROR(%)',/)
5100  FORMAT(5X,I4,3(5X,1PE12.5))
5200  FORMAT(///,5X,'**ESTADISTICAS DEL DIA**',//,5X,
     *'ERROR ABSOLUTO PROMEDIO(%)= ',1PE12.5,//,5X,
     *'DESVIACION ESTANDAR(MWH)= ',1PE12.5,//,5X,
     *'ERROR EN CARGA TOTAL(%)= ',1PE12.5,//,5X,
     *'ERROR MAXIMO POSITIVO(%)= ',1PE12.5,//,5X,
     *'ERROR MAXIMO NEGATIVO(%)= ',1PE12.5,//,5X,
     *'ERROR EN EL PICO REAL(%)= ',1PE12.5,//,5X,
     *'DIFERENCIA ENTRE PICOS(%)= ',1PE12.5,//,5X,
     *'DIFERENCIA HORARIA ENTRE PICOS= ',I3,//,5X,
     *'CARGA BASE(MWH)= ',1PE12.5)
5300  FORMAT(///,5X,'GRAFICA DE VALORES REALES(R) Y VALORES',
     *' ESTIMADOS(E)')
      END
```

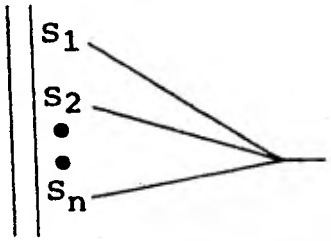
EJEMPLO B.

EJEMPLO B. IMPLEMENTACION PARA FORTRAN 77 (HARRIS).

B.1 ESTRUCTURAS DE CONTROL

Cada estructura tiene un comando arriba y otro abajo que la delimitan. El cuerpo de la estructura va sangrado 3 espacios a la derecha de estos comandos.

B.1.1 Secuencia



Si : comando y/o estructura lógica.

No es válido porner dos comandos en una misma línea.

B.1.2 Decisión

a) Simple (IF-THEN)

i) Para ejecutar un comando

```
IF (expresión) comando
```

se ejecuta cuando expresión = TRUE

ii) Para ejecutar varios comandos

```
IF (expresión) THEN
   ____________
   ____________
   ____________
   ____________
   ____________
   ____________
ENDIF
```

se ejecuta cuando expresión = TRUE

b) Doble (IF-THEN-ELSE)

```
IF (expresión) THEN
    ================
    ================   se ejecuta cuando
    ================   expresión = TRUE
    ================
    ================
ELSE
    ================
    ================   se ejecuta cuando
    ================   expresión = FALSE
    ================
    ================
ENDIF
```

c) Múltiple (CASE)

Se implementa con el GO TO computado en la siguiente forma :

```
      GO TO (N1,N2,N3,...,Nk), e←expresión entera
          ================
          ================   El control pasa aquí
          ================   cuando e<1 ó  e>k
          ================
            GO TO Nk+1
N1    CONTINUE
          ================
          ================   Comandos para e = 1
          ================
          ================
            GO TO Nk+1
N2    CONTINUE
          ================
          ================   Comandos para e = 2
          ================
          ================
            GO TO Nk+1
N3    CONTINUE
        •
        •
Nk    CONTINUE
          ================
          ================   Comandos para e = k
          ================
          ================
Nk+1  CONTINUE
```

B.1.3 Repetición

a) WHILE

```
WHILE (e)<------expresión lógica
_________
_________
_________
_________
_________
END WHILE
```

se ejecuta mientras expresión e = TRUE

b) UNTIL

```
DO
_________
_________
_________
_________
_________
UNTIL (e)<------expresión lógica
```

se ejecuta por lo menos una vez y hasta que e = TRUE

c) FOR/DO

```
FOR  V = e1,e2(,e3)
_________
_________
_________
_________
_________
END FOR
```

incremento
valor final de V
valor inicial de V

```
    DO  N1  V = e1,e2(,e3)
    _________
    _________
    _________
    _________
    _________
    _________
N1  CONTINUE
```

incremento
valor final de V
valor inicial de V

V, e_1, e_2 y e_3 pueden ser : enteros
reales
doble precisión
cuad precisión
positivos
negativos
expresiones

$e_3 \neq 0$ y $e_3 = 1$ por default
Si $e_1 > e_2$ y $e_3 > 0$
ó $e_1 < e_2$ y $e_3 < 0$

El loop no se ejecuta.

Al terminar de ejecutarse el loop, la variable - 'V' conserva su último valor.

B.1.4 Anidamientos

El anidamiento de estructuras lógicas y de control - no debe exceder de 4 niveles.
Para el caso del IF-THEN-ELSE anidado, la estructura debe ser :

En caso de requerirse más de 4 niveles de anidamiento, emplear módulos.

```
IF(expresión$_1$) THEN
    IF(expresión$_2$) THEN
        ______________
        ______________
        ______________    Se ejecuta cuando
        ______________    expresión$_1$=TRUE y       Parte 1
        ______________    expresión$_2$=TRUE
    ELSE
        ______________
        ______________
        ______________    Se ejecuta cuando
        ______________    expresión$_1$=TRUE y       Parte 2
        ______________    expresión$_2$=FALSE
    ENDIF
ELSE
    IF(expresión$_3$) THEN
        ______________
        ______________    Se ejecuta cuando
        ______________    expresión$_1$=FALSE y      Parte 3
        ______________    expresión$_3$=TRUE
        ______________
    ELSE
        ______________
        ______________    Se ejecuta cuando
        ______________    expresión$_1$=FALSE y      Parte 4
        ______________    expresión$_3$=FALSE
        ______________
    ENDIF
ENDIF
```

```
C     PROGRAMA PRINCIPAL
C*******************************************************************
C                                                                  *
C     OBJETO                                                       *
C       EFECTUAR PRONOSTICO DE CARGA MEDIANTE SUAVIZAMIENTO EXPO-  *
C       NENCIAL (EFTSES) EMPLEANDO EL METODO DE CHRISTIAANSE.      *
C       SE EMPLEA UN MODELO RESIDUAL PARA CADA DIA DE LA SEMANA.   *
C                                                                  *
C     REALIZADO EN EL IIE/DIVISION DE SISTEMAS DE POTENCIA         *
C                                                                  *
C     IMPLEMENTADO POR JOSE ARMANDO TORRES FENTANES                *
C     FECHA ULTIMA REVISION    AGOSTO  11,1980                     *
C     REVISADO POR HIROKI INOUE                                    *
C                                                                  *
C     ENTRADAS                                                     *
C       FECHA - FECHA CORRESPONDIENTE A UNA MEDICION               *
C       FORM - FORMATO PARA LECTURA DE PUNTOS MUESTRALES           *
C       IM - ARCHIVO CON VALORES REALES(MEDIDOS) DE CARGA          *
C       INIFEC - FECHA DEL PRIMER DIA DE PRONOSTICOS               *
C       IPAR - ARCHIVO CON PARAMETROS DE ENTRADA                   *
C       LONG - LONGITUD DE LA VENTANA                              *
C       MAD - MAXIMO ADELANTO EN PRONOSTICOS                       *
C       NARM - NUMERO DE ARMONICAS (INCLUYENDO LA FUNDAMENTAL)     *
C       NP - NUMERO DE PUNTOS A PRONOSTICAR                        *
C       NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS               *
C       TF - PERIODO FUNDAMENTAL                                   *
C       TITULO - NOMBRE DE LA SERIE                                *
C       XR - VALORES REALES(MEDIDOS) DE LA CARGA                   *
C                                                                  *
C     SALIDAS                                                      *
C       ALFA - VELOCIDAD ANGULAR FUNDAMENTAL                       *
C       IP - ARCHIVO CON VALORES PRONOSTICADOS                     *
C       IU - ARCHIVO DE IMPRESION CON RESULTADOS Y ESTADISTICAS    *
C       NDIA - NUMERO DEL DIA PARA EL PRONOSTICO CON 1 HORA DE AD*
C                                                                  *
C     MODULOS EMPLEADOS                                            *
C       ACTUAL - ACTUALIZA PARAMETROS DEL MODELO                   *
C       ESTAD - OBTIENE ESTADISTICAS DE LOS PRONOSTICOS            *
C       INICIA - OBTIENE PARAMETROS INICIALES DEL MODELO           *
C       LISTA - LISTA RESULTADOS DE LOS PRONOSTICOS                *
C       NVOPRO - OBTIENE NUEVO PRONOSTICO                          *
C                                                                  *
C     IDENTIFICADORES LOCALES                                      *
C       ENCONT - TRUE SI SE ENCONTRO LA FECHA INICIAL EN ARCH. IM*
C       FF - FRECUENCIA FUNDAMENTAL (1/TF)                         *
C       HORA - HORA DEL DIA                                        *
C       I - CONTADOR Y NUMERO DEL PUNTO DE PRONOSTICO              *
C       J - CONTADOR                                               *
C       NDSEM - NUMERO DE DIAS EN LA SEMANA                        *
C       NHSDIA - NUMERO DE HORAS EN UN DIA                         *
C                                                                  *
C     NOTAS.- CONDICION PARA FINALIZAR EJECUCION ES NPDIA<=0       *
C       LA DIMENSION DE LOS ARREGLOS DEBE SATISFACER               *
C                     XP(MAD)                                      *
C       LOS DIAS DE LA SEMANA SE NUMERAN DEL 1-7 A PARTIR          *
C       DEL LUNES                                                  *
C                                                                  *
C*******************************************************************
```

EJEMPLO B.2

```
C     PROGRAMA PRINCIPAL
C*********************************************************************
C                                                                    *
C     OBJETO                                                         *
C           EFECTUAR PRONOSTICO DE CARGA MEDIANTE SUAVIZAMIENTO EXPO-*
C           NENCIAL (FFTGES) EMPLEANDO EL METODO DE CHKISTIAANSE.    *
C           SE EMPLEA UN MODELO RESIDUAL PARA CADA DIA DE LA SEMANA. *
C                                                                    *
C     REALIZADO EN EL IIE/DIVISION DE SISTEMAS DE POTENCIA           *
C                                                                    *
C     IMPLEMENTADO POR JOSE ARMANDO TORRES FENTANES                  *
C     FECHA ULTIMA REVISION     AGOSTO  11,1980                      *
C     REVISADO POR MOTOKI INOUE                                      *
C                                                                    *
C     ENTRADAS                                                       *
C           FECHA - FECHA CORRESPONDIENTE A UNA MEDICION             *
C           FORM - FORMATO PARA LECTURA DE PUNTOS MUESTRALES         *
C           IM - ARCHIVO CON VALORES REALES(MEDIDOS) DE CARGA        *
C           INIFEC - FECHA DEL PRIMER DIA DE PRONOSTICOS             *
C           IPAR - ARCHIVO CON PARAMETROS DE ENTRADA                 *
C           LONG - LONGITUD DE LA VENTANA                            *
C           MAD - MAXIMO ADELANTO EN PRONOSTICOS                     *
C           NARM - NUMERO DE ARMONICAS (INCLUYENDO LA FUNDAMENTAL)   *
C           NP - NUMERO DE PUNTOS A PRONOSTICAR                      *
C           NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS             *
C           TF - PERIODO FUNDAMENTAL                                 *
C           TITULO - NOMBRE DE LA SERIE                              *
C           XR - VALORES REALES(MEDIDOS) DE LA CARGA                 *
C                                                                    *
C     SALIDAS                                                        *
C           ALFA - VELOCIDAD ANGULAR FUNDAMENTAL                     *
C           IP - ARCHIVO CON VALORES PRONOSTICADOS                   *
C           IW - ARCHIVO DE IMPRESION CON RESULTADOS Y ESTADISTICAS  *
C           NDIA - NUMERO DEL DIA PARA EL PRONOSTICO CON 1 HORA DE AD*
C                                                                    *
C     MODULOS EMPLEADOS                                              *
C           ACTUAL - ACTUALIZA PARAMETROS DEL MODELO                 *
C           ESTAD - OBTIENE ESTADISTICAS DE LOS PRONOSTICOS          *
C           INICIA - OBTIENE PARAMETROS INICIALES DEL MODELO         *
C           LISTA - LISTA RESULTADOS DE LOS PRONOSTICOS              *
C           NVOPRO - OBTIENE NUEVO PRONOSTICO                        *
C                                                                    *
C     IDENTIFICADORES LOCALES                                        *
C           ENCONT - TRUE SI SE ENCONTRO LA FECHA INICIAL EN ARCH. IM*
C           FF - FRECUENCIA FUNDAMENTAL (1/TF)                       *
C           HORA - HORA DEL DIA                                      *
C           I - CONTADOR Y NUMERO DEL PUNTO DE PRONOSTICO            *
C           J - CONTADOR                                             *
C           NDSEM - NUMERO DE DIAS EN LA SEMANA                      *
C           NHSDIA - NUMERO DE HORAS EN UN DIA                       *
C                                                                    *
C     NOTAS.- CONDICION PARA FINALIZAR EJECUCION ES NPDIA<=0         *
C             LA DIMENSION DE LOS ARREGLOS DEBE SATISFACER           *
C                       XP(MAD)                                      *
C             LOS DIAS DE LA SEMANA SE NUMERAN DEL 1-7 A PARTIR      *
C             DEL LUNES                                              *
C                                                                    *
C*********************************************************************
```

```
C
      CHARACTER*4 FORM,TITULO
      INTEGER FECHA,HORA,I,IM,IMPTR,INIFEC,IP,IPAR,IPPTR,IW,J,
     *        LONG,MAD,NARM,NDIA,NDSEM,NHSDIA,NP,NPDIA
      LOGICAL ENCONT
      REAL ALFA,FF,PI,TF,XR,XP
C
      DIMENSION FORM(20),TITULO(4)
      COMMON/FILE1/IW,IPAR
      COMMON/FILE2/IM,IP
      COMMON/PRON/XP(24)
      COMMON/REAL/FECHA(3),XR
C     ----------------------------
C     *PARAMETROS CONSTANTES
C     ----------------------------
      DATA IM/16/,IP/17/,IPAR/14/,IW/6/,PI/3.141592/
      DATA NDSEM/7/,NHSDIA/24/
C     ----------------------------
C     *ABRIR ARCHIVOS
C     ----------------------------
      REWIND IM
      REWIND IP
      REWIND IPAR
C     ----------------------------------------------------
C     *LEER VARIABLE DE CONTROL Y DATOS INICIALES
C     ----------------------------------------------------
      READ(IPAR,5000) NPDIA,INIFEC,TITULO
      WHILE (NPDIA.GT.0)
C        ----------------------------------------------------
C        *LEER PARAMETROS Y EVALUAR FRECUENCIA FUNDAMENTAL
C        ----------------------------------------------------
         READ(IPAR,5100) NP,NARM,MAD,LONG,TF
         READ(IPAR,5200) FORM
         FF=1./TF
         ALFA=2.*PI*FF
C        ----------------------------
C        *IMPRIMIR DATOS
C        ----------------------------
         WRITE(IW,5300) TITULO,FORM
         WRITE(IW,5400) NPDIA,NP,NARM,MAD,LONG,FF
C        ----------------------------------------------------
C        *OBTENER PARAMETROS INICIALES DEL MODELO
C        ----------------------------------------------------
         CALL INICIA(NARM,ALFA)
C        ----------------------------------------------------
C        *POSICIONAR APUNTADOR DEL ARCHIVO DE VALORES REALES
C        ----------------------------------------------------
         REWIND IM
         DO
            READ(IM,FORM) FECHA,XR
            ENCONT=(FECHA(1).EQ.INIFEC(1)).AND.(FECHA(2).EQ.
     *      INIFEC(2)).AND.(FECHA(3).EQ.INIFEC(3))
         UNTIL (ENCONT)
```

```
C         ------------------------------------
C         *OBTENER PRONOSTICOS DEL PUNTO I
C         ------------------------------------
          NDIA=NPDIA-1
          FOR I=1,NP
C
C            *OBTENER DIA DEL PRONOSTICO
C            ------------------------------
             HORA=MOD(I,NHSDIA)
             IF(HORA.EQ.1) NDIA=NDIA+1
C            ----------------------
C            *OBTENER PRONOSTICO
C            ----------------------
             CALL NVOPRO(I,MAD,NARM,ALFA,NDIA)
C            ------------------------------------
C            *GUARDAR PRONOSTICOS EN ARCHIVO IP
C            ------------------------------------
             WRITE(IP,5500) (XP(J),J=1,MAD)
C            ------------------------------------
C            *ACTUALIZAR PARAMETROS DEL MODELO
C            ------------------------------------
             CALL ACTUAL(NARM,NDIA,I)
C            ------------------------------
C            *LEER VALOR REAL DE LA CARGA
C            ------------------------------
             READ(IM,FORM) FECHA,XR
          ENDFOR
C         ------------------------
C         *IMPRIMIR RESULTADOS
C         ------------------------
          CALL LISTA(NPDIA,NP,MAD,LONG)
C         ------------------------------
C         *OBTENER ESTADISTICAS DE LOS PRONOSTICOS
C         ------------------------------
          CALL ESTAD(NPDIA,NP,MAD,LONG)
C         ------------------------------------------
C         *VERIFICAR SI SE DESEA PRONOSTICAR OTRA SERIE
C         ------------------------------------------
          READ(IPAR,5000) NPDIA,INIFEC,TITULO
        ENDWHILE
        CALL EXIT
C       ------------------------------
C       *FORMATOS DE LECTURA E IMPRESION
C       ------------------------------
5000    FORMAT(I2,3I2,4A4)
5100    FORMAT(4I5,F10.0)
5200    FORMAT(20A4)
5300    FORMAT(1H1,//,5X,'SIMULACION DE PRONOSTICO DE CARGA',//,5X,
       *'METODO=FFTES',///,5X,'TITULO=',4A4,///,5X,'FORMATO= ',20A4)
5400    FORMAT(///,5X,'PRIMER DIA DE PRONOSTICOS=',
       *I5,//,5X,'CANTIDAD DE PUNTOS PARA EL PRONOSTICO=',I5,//,5X,
       *'CANTIDAD DE MULTIPLOS DE LA FREC.FUNDAMENTAL=',I5,//,5X,
       *'MAXIMO ADELANTO (HRS) PARA PRONOSTICO=',I5,//,5X,
       *'CANTIDAD DE PUNTOS PARA EL INTERVALO BASE DE PRONOSTICO=',I5,//
       *,5X,'FRECUENCIA FUNDAMENTAL=',1PE12.5)
5500    FORMAT(20F6.0)
        END
```

```
      SUBROUTINE INICIA(NARM,ALFA)
C*****************************************************************
C                                                                 *
C     OBJETO                                                      *
C           OBTENER PARAMETROS DE LOS MODELOS RESIDUALES PARA CADA *
C           DIA DE LA SEMANA, LA MATRIZ DE TRANSICION Y LOS VECTO- *
C           RES DE SUAVIZAMIENTO EXPONENCIAL                      *
C                                                                 *
C     IMPLEMENTADA POR JOSE ARMANDO TORRES FENTANES               *
C     FECHA ULTIMA REVISION OCTUBRE 26,1979                       *
C     REVISADA POR JATF                                           *
C                                                                 *
C     ENTRADAS                                                    *
C           ALFA - VELOCIDAD ANGULAR FUNDAMENTAL                  *
C           IPAR - ARCHIVO CON PARAMETROS                         *
C           NARM - NUMERO DE ARMONICAS                            *
C                                                                 *
C     SALIDAS                                                     *
C           ARM - ARMONICAS A EMPLEAR                             *
C           BETA - CONSTANTES DE SUAVIZAMIENTO EXPONENCIAL        *
C           C Y S - ELEMENTOS DE LA MATRIZ DE TRANSICION          *
C           CARHOR - GARGA HORA-SEMANA PROMEDIO                   *
C           PAR - PARAMETROS DEL MODELO PARA CADA DIA DE LA SEMANA *
C           SH - VECTORES DE SUAVIZAMIENTO EXPONENCIAL            *
C           VARHOR - VARIANCIA DE LA CARGA HORA-SEMANA            *
C                                                                 *
C     MODULOS REQUERIDOS                                          *
C           NINGUNO                                               *
C                                                                 *
C     IDENTIFICADORES LOCALES                                     *
C           DOM - DIA DOMINGO                                     *
C           I - CONTADOR                                          *
C           J - CONTADOR                                          *
C           LUN - DIA LUNES                                       *
C           NHSDIA - NUMERO DE HORA EN UN DIA                     *
C           NPAR - NUMERO DE PARAMETROS PARA EL MODELO DIARIO     *
C           TEMP - VARIABLE TEMPORAL                              *
C                                                                 *
C     NOTAS.-LA DIMENSION DE LOS ARREGLOS DEBE SATISFACER LO SIG.: *
C                 ARM(NF)                                         *
C                 BETA(7)                                         *
C                 C(NF)                                           *
C                 PAR(2NF+1,7)                                    *
C                 S(NF)                                           *
C                 SH(2NF+1,7)                                     *
C                                                                 *
C           LOS DIAS DE LA SEMANA SE NUMERAN DEL 1 AL 7 A PARTIR  *
C           DEL LUNES                                             *
C                                                                 *
C*****************************************************************
```

```
C*****************************************************************
      INTEGER ARM,DOM,I,IPAR,J,LUN,NARM,NHSDIA,NPAR
      REAL ALFA,BETA,C,CARHOR,PAR,S,SH,TEMP,VARHOR
C
      COMMON/FILE1/IW,IPAR
      COMMON/PAR/PAR(30,7)
      COMMON/SHBETA/SH(30,7),BETA(7)
      COMMON/STAT/CARHOR(24,7),VARHOR(24,7)
      COMMON/TRANS/C(15),S(15)
C     ------------------------
C     *PARAMETROS CONSTANTES
C     ------------------------
      DATA LUN/1/,DOM/7/,NHSDIA/24/
C     ------------------------------------------
C     *LEER CONSTANTES DE SUAVIZAMIENTO EXPONENCIAL
C     ------------------------------------------
      READ(IPAR,5000) BETA
      WRITE(IW,5100) BETA
C     ------------------------
C     *LEER ARMONICAS
C     ------------------------
      READ(IPAR,5200) (ARM(I),I=1,NARM)
      WRITE(IW,5300) (ARM(I),I=1,NARM)
C     ------------------------------------------------------
C     *CALCULAR NUMERO DE PARAMETROS DEL MODELO DIARIO
C     ------------------------------------------------------
      NPAR=2*NARM+1
C     ------------------------------------------------------
C     *OBTENER PARAMETROS DE LOS MODELOS DIARIOS
C     ------------------------------------------------------
      WRITE(IW,5400)
      FOR I=I,NPAR
         READ(IPAR,5500) (PAR(I,J),J=LUN,DOM)
         WRITE(IW,5600) (PAR(I,J),J=LUN,DOM)
      ENDFOR
C     ------------------------------------------------------
C     *GENERAR COMPONENTES DE LA MATRIZ DE TRANSICION
C      (SENOS Y COSENOS DE LAS ARMONICAS)
C     ------------------------------------------------------
      FOR I=1,NARM
         TEMP=ALFA*FLOAT(ARM(I))
         C(I)=COS(TEMP)
         S(I)=SIN(TEMP)
      ENDFOR
C     ------------------------------------------------------
C     *OBTENER VECTORES DE SUAVIZAMIENTO EXPONENCIAL SH
C     ------------------------------------------------------
*****************WRITE(IW,5700)*****************************************
      FOR I=1,NPAR
         READ(IPAR,5500) (SH(I,J),J=LUN,DOM)
         WRITE(IPAR,5600) (SH(I,J),J=LUN,DOM)
      ENDFOR
```

```
C       ---------------------------------------------------------
C       *OBTENER CARGA PROMEDIO Y VARIANCIA PARA CADA HORA
C        DE LA SEMANA
C       ---------------------------------------------------------
        WRITE(IW,5800)
        FOR I=1,NHSDIA
           READ(IPAR,5500) (CARHOR(I,J),J=LUN,DOM)
           WRITE(IW,5600) (CARHOR(I,J),J=LUN,DOM)
        ENDFOR
        WRITE(IW,5900)
        FOR I=1,NHSDIA
           READ(IPAR,5500) (VARHOR(I,J),J=LUN,DOM)
           WRITE(IW,5600) (VARHOR(I,J),J=LUN,DOM)
        ENDFOR
        WRITE(IW,6000)
        FOR I=1,NHSDIA
           WRITE(IW,5600) (SQRT(VARHOR(I,J)),J=LUN,DOM)
        ENDFOR
        RETURN
C       ---------------------------------------
C       *FORMATOS DE LECTURA E IMPRESION
C       ---------------------------------------
 5000   FORMAT(7F6.0)
 5100   FORMAT(/,5X,'CONSTANTES DE SUAVIZAMIENTO EXPONENCIAL',
       *'(LUN-DOM)',/,7(/,10X,1PE12.5))
 5200   FORMAT(20I4)
 5300   FORMAT(/,5X,'MULTIPLOS DE LA FRECUENCIA FUNDAMENTAL',/,
       *20(/,10,I4))
 5400   FORMAT(/,5X,'PARAMETROS AR',/)
 5500   FORMAT(7(1PE11.4))
 5600   FORMAT(9(2X,1PE12.5))
 5700   FORMAT(/,5X,'VECTORES SH',/)
 5800   FORMAT(////,5X,'CARGAS PROMEDIO HORARIAS(LUN-DOM)',/)
 5900   FORMAT(////,5X,'VARIANCIA DE LA CARGA HORARIA(LUN-DOM)',/)
 6000   FORMAT(////,5X,'DESV.ESTANDAR DE LA CARGA HORARIA(LUN-DOM)',/)
        END
```

```
      SUBROUTINE NVOPRO(IPUNTO,MAD,NARM,ALFA,NDIA)
C*********************************************************************
C                                                                    *
C     OBJETO                                                         *
C           OBTENER LOS PRONOSTICOS DE IPUNTO HASTA CON MAD          *
C           UNIDADES DE ADELANTO. SE EMPLEA UN MODELO PARA CADA DIA  *
C           DE LA SEMANA.                                            *
C                                                                    *
C     IMPLEMENTADA POR JOSE ARMANDO TORRES FENTANES                  *
C     FECHA ULTIMA REVISION  OCTUBRE 26, 1979                        *
C     REVISADA POR MOTOKI INOUE                                      *
C                                                                    *
C     ENTRADAS                                                       *
C           ALFA - VELOCIDAD ANGULAR FUNDAMENTAL                     *
C           ARM - MULTIPLOS DE LA FRECUENCIA FUNDAMENTAL(ARMONICAS)  *
C           IPUNTO - PUNTO EN EL CUAL SE DESEAN LOS PRONOSTICOS      *
C           MAD - MAXIMO ADELANTO EN LOS PRONOSTICOS                 *
C           NARM - NUMERO DE ARMONICAS                               *
C           NDIA - NUMERO DEL DIA DE PRONOSTICO PARA UNA HORA DE AD. *
C           PAR - PARAMETROS DEL MODELO                              *
C                                                                    *
C     SALIDAS                                                        *
C           XP - VALORES PRONOSTICADOS DE CARGA                      *
C                                                                    *
C     MODULOS EMPLEADOS                                              *
C           NINGUNO                                                  *
C                                                                    *
C     IDENTIFICADORES LOCALES                                        *
C           DOM - DIA DOMINGO                                        *
C           I - CONTADOR                                             *
C           J - CONTADOR                                             *
C           L - CONTADOR                                             *
C           NDPRON - NUMERO DEL DIA DE LA HORA QUE SE PRONOSTICARA   *
C           NDSEM - NUMERO DE DIAS DE LA SEMANA                      *
C           NHBASE - HORA PARA EL PRONOSTICO CON UNA HORA DE ADELANTO*
C           NHORA - HORA PARA EVALUAR EL PRONOSTICO CON RESPECTO A LA*
C                 ULTIMA ACTUALIZACION DEL MODELO DEL DIA            *
C           NHSDIA - NUMERO DE HORAS EN UN DIA                       *
C           NHTOT - HORA DEL TOTAL ACUMULADO QUE SE DESEA PRONOSTICAR*
C           TETA - VARIABLE TEMPORAL                                 *
C                                                                    *
C     NOTAS.-LOS DIAS DE LA SEMANA SE NUMERAN DEL 1 AL 7 A PARTIR DEL*
C            LUNES                                                   *
C*********************************************************************
C
      INTEGER DOM,I,IPUNTO,J,L,MAD,NARM,NDIA,NDPRON,NDSEM,NHBASE,
     *        NHORA,NHSDIA,NHTOT
      REAL ALFA,ARM,PAR,XP
C
      COMMON/ARM/ARM(15)
      COMMON/PAR/PAR(30,7)
      COMMON/PRON/XP(24)
```

```
C     ------------------------------------------------------------
C     *PARAMETROS CONSTANTES
C     ------------------------------------------------------------
      DATA DOM/7/,NDSEM/7/,NHSDIA/24/
C     ------------------------------------------------------------
C     *OBTENER DIA Y HORA PARA PRONOSTICO CON UNA HORA
C      DE ADELANTO
C     ------------------------------------------------------------
      NDPRON=NDIA
      NHBASE=MOD(IPUNTO,NHSDIA)
      IF(NHBASE.EQ.0) NHBASE=24
C     ------------------------------------------------------------
C     *OBTENER PRONOSTICOS HASTA CON MAD UNIDADES DE ADELANTO
C     ------------------------------------------------------------
      FOR J=1,MAD
C     ------------------------------------------------------------
C        *OBTENER HORA Y DIA PARA EL PRONOSTICO CON J UNIDADES
C         DE ADELANTO
C     ------------------------------------------------------------
         NHTOT=IPUNTO + J - 1
         NHORA=MOD(NHTOT,NHSDIA)
         IF(NHORA.EQ.0) NHORA=24
         IF((NHORA.EQ.1).AND.(J.NE.1)) NDPRON=NDPRON+1
         NDPRON=MOD(NDPRON,NDSEM)
         IF(NDPRON.EQ.0) NDPRON=DOM
C     ------------------------------------------------------------
C        *OBTENER PUNTO DE EVALUACION DEL MODELO DEL DIA DE
C         ACUERDO CON SU ULTIMA EVALUACION
C     ------------------------------------------------------------
         IF(NDPRON.EQ.NDIA) THEN
            NHORA=NHORA-NHBASE
            IF(NHORA.LT.0) NHORA=NHORA+24
         ELSE
            NHORA=NHORA-1
         ENDIF
C     ------------------------------------------------------------
C        *OBTENER PRONOSTICO
C     ------------------------------------------------------------
         XP(J)=PAR(1,NDPRON)
         L=2
         FOR I=1,NARM
            TETA=ALFA*FLOAT(ARM(I))*FLOAT(NHORA)
            XP(J)=XP(J) + PAR(L,NDPRON)*SIN(TETA) +
                  PAR(L+1,NDPRON)*COS(TETA)
            L=L+2
         ENDFOR
      ENDFOR
      RETURN
      END
```

```
      SUBROUTINE ESTAD(NPDIA,NP,MAD,LONG)
C*****************************************************************
C                                                                 *
C     OBJETO                                                      *
C          EVALUAR RESULTADOS DE LOS PRONOSTICOS                  *
C                                                                 *
C     IMPLEMENTADA POR JOSE ARMANDO TORRES FENTANES               *
C     FECHA ULTIMA REVISION OCTUBRE 26,1979                       *
C     REVISADA POR JATF                                           *
C                                                                 *
C     ENTRADAS                                                    *
C          LONG - LONGITUD DE LA VENTANA(168 PARA SEMANA)         *
C          MAD - MAXIMO ADELANTO EN LOS PRONOSTICOS               *
C          NP - NUMERO DE PUNTOS PRONOSTICADOS                    *
C          NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS           *
C                                                                 *
C     SALIDAS                                                     *
C          LONG - LONGITUD DE LA VENTANA(168 PARA SEMANA)         *
C          MAD - MAXIMO ADELANTO EN LOS PRONOSTICOS               *
C          NHSDIA - NUMERO DE HORAS EN UN DIA                     *
C          NP - NUMERO DE PUNTOS PRONOSTICADOS                    *
C          NPDIA - NUMERO DEL PRIMER DIA DE PRONOSTICOS           *
C          ESTADISTICAS DIARIAS POR DIA,HORA Y TIEMPO DE ADELANTO *
C                                                                 *
C     MODULOS REQUERIDOS                                          *
C          DIA - ESTADISTICAS DIARIAS CON UNA HORA DE ADELANTO    *
C          DIACIA - ESTADISTICAS DIARIAS POR BLOQUES DE 24 HORAS  *
C          HORA - ESTADISTICAS POR HORA DE LA SEMAN               *
C          TOTAL - ESTADISTICAS PARA CADA TIEMPO DE ADELANTO      *
C                                                                 *
C     IDENTIFICADORES LOCALES                                     *
C          NINGUNO                                                *
C                                                                 *
C*****************************************************************
C
      INTEGER LONG,MAD,NHSDIA,NP,NPDIA
C     ------------------------
C     *PARAMETROS CONSTANTES
C     ------------------------
      DATA NHSDIA/24/
C     ------------------------------------------------------------
C     *OBTENER ESTADISTICAS PARA EL TOTAL DE LA SIMULACION
C      COMPARANDO ESTADISTICOS DE CADA TIEMPO DE ADELANTO
C     ------------------------------------------------------------
      CALL TOTAL(NP,MAD)
C     ------------------------------------------------------------
C     *OBTENER ESTADISTICAS DIARIAS PARA UNA HORA DE ADELANTO
C     ------------------------------------------------------------
      CALL DIA(NPDIA,NP,MAD,NHSDIA)
C     ------------------------------------------------------------
C     *OBTENER ESTADISTICAS PARA PRONOSTICOS HECHOS A LA
C      HORA UNO POR BLOQUES DE 24 HORAS
C     ------------------------------------------------------------
      CALL DIACIA(NPDIA,NP,MAD,NHSDIA)
C     ------------------------------------------------------------
C     *OBTENER ESTADISTICAS POR HORA DE LA SEMANA PARA CADA TIEMPO
C      DE ADELANTO
C     ------------------------------------------------------------
      CALL HORA(NP,MAD,LONG)
      RETURN
      END
```

```
      SUBROUTINE ESTDIA(NUMDIA,NHS)
C*****************************************************************
C                                                                 *
C     OBJETO                                                      *
C            EVALUAR LAS ESTADISTICAS CORRESPONDIENTES A UN DIA   *
C                                                                 *
C     IMPLEMENTADA POR JOSE A. TORRES FENTANES                    *
C     FECHA ULTIMA REVISION ENERO 25 1980                         *
C     REVISADA POR JATF                                           *
C                                                                 *
C     ENTRADAS                                                    *
C            A(I,1) - HORA DEL DIA                                *
C            A(I,2) - CARGA REAL                                  *
C            A(I,3) - CARGA PRONOSTICADA                          *
C            FECHA - FECHA DEL DIA                                *
C            NHS - PERIODO DE EVALUACION (24 PARA DIA)            *
C            NUMDIA - NUMERO DEL DIA                              *
C            XR - VALOR REAL DE LA CARGA                          *
C                                                                 *
C     SALIDAS                                                     *
C            BASE - CARGA MINIMA(MWH)                             *
C            DPICO - DIFERENCIA ENTRE PICOS(%)                    *
C            EABS - EROR ABSOLUTO PROMEDIO (%)                    *
C            ECARG - ERROR EN CARGA TOTAL DIARIA(%)               *
C            EMAXN - ERROR MAXIMO NEGATIVO DIARIO(%)              *
C            EMAXP - ERROR MAXIMO POSITIVO DIARIO(%)              *
C            ENERGP - ENERGIA PRONOSTICADA DEL DIA(MWH)           *
C            ENERGR - ENERGIA REAL DEL DIA(MWH)                   *
C            EPICO - ERROR EN EL PUNTO DEL PICO REAL(%)           *
C            HDIF - DIFERENCIA HORARIA ENTRE PICOS                *
C            RMS - ERROR MEDIO CUADRATICO DIARIO(MWH)             *
C                                                                 *
C     MODULOS EMPLEADOS                                           *
C            NINGUNO                                              *
C                                                                 *
C     IDENTIFICADORES LOCALES                                     *
C            ERROR - ERRORES HORARIOS(%)                          *
C            I - CONTADOR                                         *
C            J - CONTADOR                                         *
C            HORA - HORA CONSIDERADA                              *
C            HPICOP - HORA DEL PICO PRONOSTICADO                  *
C            HPICOR - HORA DEL PICO REAL                          *
C            MES - NOMBRE DE LOS MESES                            *
C            NOMDIA - NOMBRE DE LOS DIAS                          *
C            XP(I) - PRONOSTICO DEL VALOR REAL CON I HORAS DE ADELANTO*
C            XPMAX - MAXIMO VALOR EN EL DIA DE LA CARGA PRONOSTICADA *
C            XRMAX - MAXIMO VALOR EN EL DIA DE LA CARGA REAL      *
C            Z - ERROR EN CARGA,NORMALIZADO CON RESPECTO AL VALOR REAL*
C                                                                 *
C     NOTAS.-LOS DIAS DE LA SEMANA SE NUMERAN DEL 1 AL 7  A PARTIR *
C            DEL LUNES                                            *
C                                                                 *
C*****************************************************************
```

```
C
      CHARACTER*3 MES,NOMDIA
      INTEGER FECHA,HDIF,HORA,HPICOP,HPICOR,I,J,NHS,NUMDIA
      REAL A,BASE,CARGP,CARGR,DPICO,EABS,ECARG,EMAXN,EMAXP,EPICO,
     *     ERROR,RMS,XR,XRMAX,XP,XPMAX,Z
C
      DIMENSION ERROR(24),MES(12),NOMDIA(7),XP(24)
      COMMON/DATOS/A(24,3)
      COMMON/FILE1/IW,IPAR
      COMMON/REAL/FECHA(3),XR
      COMMON/ESTDIA/BASE,CARGR,CARGP,EABS,ECARG,EPICO,RMS
C     --------------------------
C     *PARAMETROS CONSTANTES
C     --------------------------
      DATA MES/3HENE,3HFEB,3HMZO,3HABR,3HMAY,3HJUN,3HJUL,3HAGO,
     *     3HSEP,3HOCT,3HNOV,3HDIC/
      DATA NOMDIA/3HLUN,3HMAR,3HMIE,3HJUE,3HVIE,3HSAB,3HDOM/
C     ------------------------------------------------------
C     *INICIALIZAR CONTADORES DEL DIA
C     ------------------------------------------------------
      RMS=0.
      EABS=0.
      ECARG=0.
      CARGR=0.
      CARGP=0.
      EMAXP=-1.0E+20
      EMAXN=1.0E+20
      BASE=1.0E+20
C     ------------------------------------------------------
C     *OBTENER ESTADISTICAS DEL DIA
C     ------------------------------------------------------
      FOR HORA=1,NHS
         XR=A(HORA,2)
         XP(HORA)=A(HORA,3)
         Z=(XR-XP(HORA))/XR
         IF(EMAXP.LT.Z) EMAXP=Z
         IF(EMAXN.GT.Z) EMAXN=Z
         IF(BASE.GT.XR) BASE=XR
         CARGR=CARGR + XR
         CARGP=CARGP + XP(HORA)
         EABS=EABS + ABS(Z)
         RMS=RMS + (XR-XP(HORA))**2
         ERROR(I)=Z*100.
         IF(HORA.EQ.1) THEN
            HPICOP=1
            HPICOR=1
            XRMAX=XR
            XPMAX=XP(1)
            EPICO=Z
         ELSE
            IF(XRMAX.LT.XR) THEN
               XRMAX=XR
               HPICOR=HORA
               EPICO=Z
            ENDIF
            IF(XPMAX.LT.XP(HORA)) THEN
               XPMAX=XP(HORA)
               HPICOP=HORA
            ENDIF
         ENDIF
      ENDFOR
```

```
C     ---------------------------------------------
C     *OBTENER ESTADISTICAS DEL DIA EN POR CIENTO
C     ---------------------------------------------
      ECARG=100.*ABS(CARGR-CARGP)/CARGR
      RMS=RMS/FLOAT(NHS)
      EABS=100.*EABS/FLOAT(NHS)
      EMAXP=100.*EMAXP
      EMAXN=100.*EMAXN
      EPICO=100.*EPICO
      DPICO=100.*(XRMAX-XPMAX)/XRMAX
      HDIF=HPICOR-HPICOP
C     ------------------------------------------------
C     *IMPRIMIR VALOR REAL, ESTIMADO Y ERROR HORARIO
C      PARA EL DIA
C     ------------------------------------------------
      WRITE(IW,5000) NOMDIA(NUMDIA),FECHA(1),MES(FECHA(2)),FECHA(3)
      FOR I=1,NHS
         WRITE(IW,5100) I,(A(I,J),J=2,3),ERROR(I)
      ENDFOR
C     ------------------------------------
C     *IMPRIMIR ESTADISTICAS DEL DIA
C     ------------------------------------
      WRITE(IW,5200) EABS,SQRT(RMS),ECARG,EMAXP,EMAXN,EPICO,DPICO,
     *               HDIF,BASE
C     ----------------------------------------
C     *GRAFICAR VALORES REALES Y PRONOSTICADOS
C     ----------------------------------------
      WRITE(IW,5300)
      CALL GRAFI(NHS,3)
      RETURN
C     ---------------------------
C     *FORMATOS DE IMPRESION
C     ---------------------------
5000  FORMAT(///,5X,'PRONOSTICOS DEL DIA ',A3,' (',I2,' DE ',A3,
     *' DE 19',I2,')',//,5X,'HORA',6X,'CARGA REAL',7X,'PRONOSTICO',8X,
     *'ERROR(%)',/)
5100  FORMAT(5X,I4,3(5X,1PE12.5))
5200  FORMAT(///,5X,'**ESTADISTICAS DEL DIA**',//,5X,
     *'ERROR ABSOLUTO PROMEDIO(%)= ',1PE12.5,//,5X,
     *'DESVIACION ESTANDAR(MWH)= ',1PE12.5,//,5X,
     *'ERROR EN CARGA TOTAL(%)= ',1PE12.5,//,5X,
     *'ERROR MAXIMO POSITIVO(%)= ',1PE12.5,//,5X,
     *'ERROR MAXIMO NEGATIVO(%)= ',1PE12.5,//,5X,
     *'ERROR EN EL PICO REAL(%)= ',1PE12.5,//,5X,
     *'DIFERENCIA ENTRE PICOS(%)= ',1PE12.5,//,5X,
     *'DIFERENCIA HORARIA ENTRE PICOS= ',I3,//,5X,
     *'CARGA BASE(MWH)= ',1PE12.5)
5300  FORMAT(///,5X,'GRAFICA DE VALORES REALES(R) Y VALORES',
     *' ESTIMADOS(E)')
      END
```

APENDICE B.2

NORMAS DEL LENGUAJE PARA DISEÑAR PROGRAMAS (LDP)

CONTENIDO PAGINA

I OBJETIVO.

El propósito de este documento es definir las normas - para el uso de un "Lenguaje para Diseñar Programas" -- (LDP).

El LDP es fácil de seguir y más facil de programar -- que los diagramas de flujos, debido a que "es casi -- código". Este "pseudo código", describe la estructura, control, iteraciones, flujo, etc., de un programa de computadora, con la ventaja sobre los diagramas de flujo y otro tipos de diagramas, que puede actualizarse continuamente en un procesador de textos.

El LDP deberá incluirse en los siguientes documentos :

- RPC : Requerimientos del Programa de Computadora.

- DPC : Diseño del Programa de Computadora.

- CODIGO

II INDENTACION Y ETIQUETAS.

Indentación y Etiquetas : La subordinación se lleva a-cabo indentando las líneas del Pseudo-código.

Ejemplo :

```
1.  LINEA A
2.  LINEA B
   A. LINEA C
   B. LINEA D
      1. LINEA E
      2. LINEA F
   C. LINEA G
3.  LINEA H
   A. LINEA I
      1. LINEA J
         A. LINEA K
         B. LINEA L
4.  CONTINUA CON LDP, REGRESA O TERMINA
```

Solamente se utilizan dos conjuntos de caracteres : - números y letras; se utilizan alternativamente en los niveles de subordinación.

Y la Identación se hace con cuatro espacios.

III ESTRUCTURAS DE CONTROL.

Las únicas estructuras de control son de selección, - repetición y llamadas a módulos.

a) Selección.
Las formas de Selección son :

A.1 SI(IF)
A.2 SI-ENTONCES-SINO (IF-THEN-ELSE)
A.3 CASO (CASE)

b) Repetición.
Las formas de Repetición :

B.1 EJECUTA MIENTRAS (WHILE)
B.2 EJECUTA-HASTA (DO-UNTIL)
B.3 REPITE (FOR,DO)

c) Llamadas a Módulos.
La forma de llamar a Módulos es el LLAMA (CALL).

IV CONSTRUCCIONES.

a) Selección.

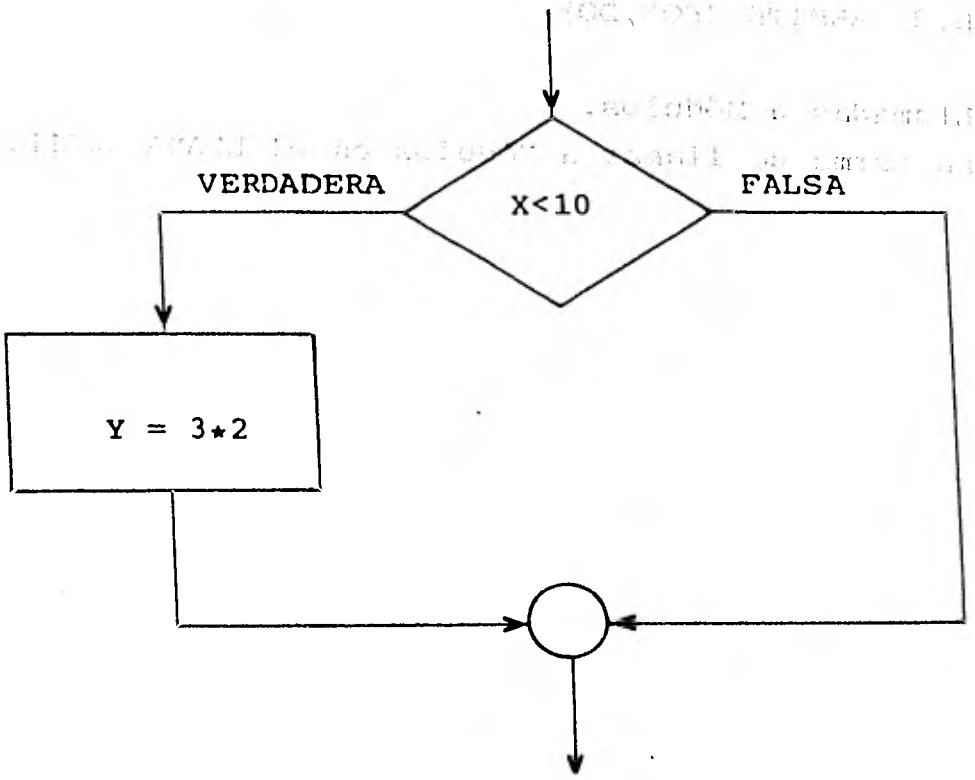
A.1 Construcción SI (IF)

El formato del SI es :

1. SI condición
   A. bloque verdadero

Donde condición es una expresión lógica y el bloque -- verdadero es un conjunto de proposiciones en Pseudo-código.

Ejemplo : 1. SI X < 10
A. Y = 3*2



A.2 Construcción SI-ENTONCES-SINO
(IF-THEN-ELSE)

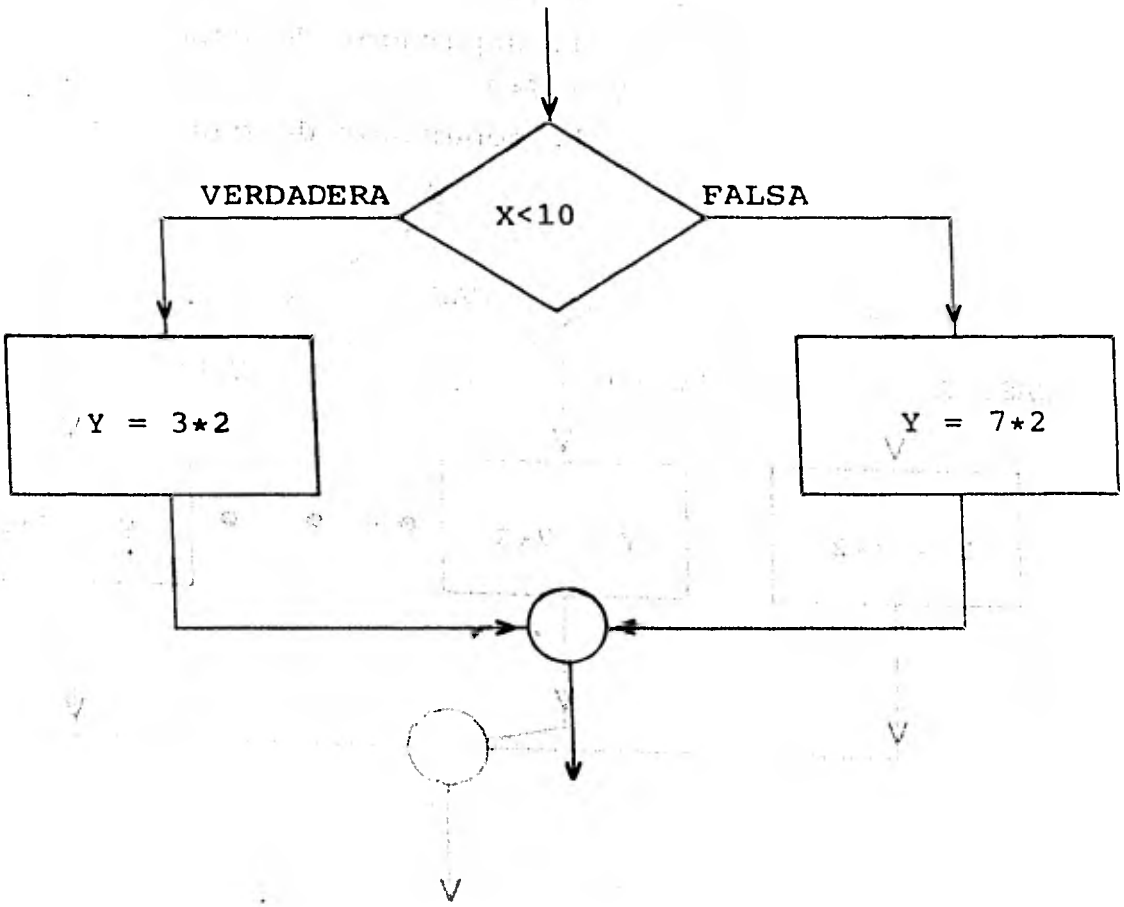
El formato del SI-ENTONCES-SINO es :

1. SI condición ENTONCES
   A. bloque verdadero
2. SINO
   A. bloque falso

Donde condición es una expresión lógica y los bloques verdadero y falso son proposiciones en Pseudo-código.

Ejemplo : 1. SI X < 10 ENTONCES
A. Y = 3*2

2. SINO
B. Y = 7*2



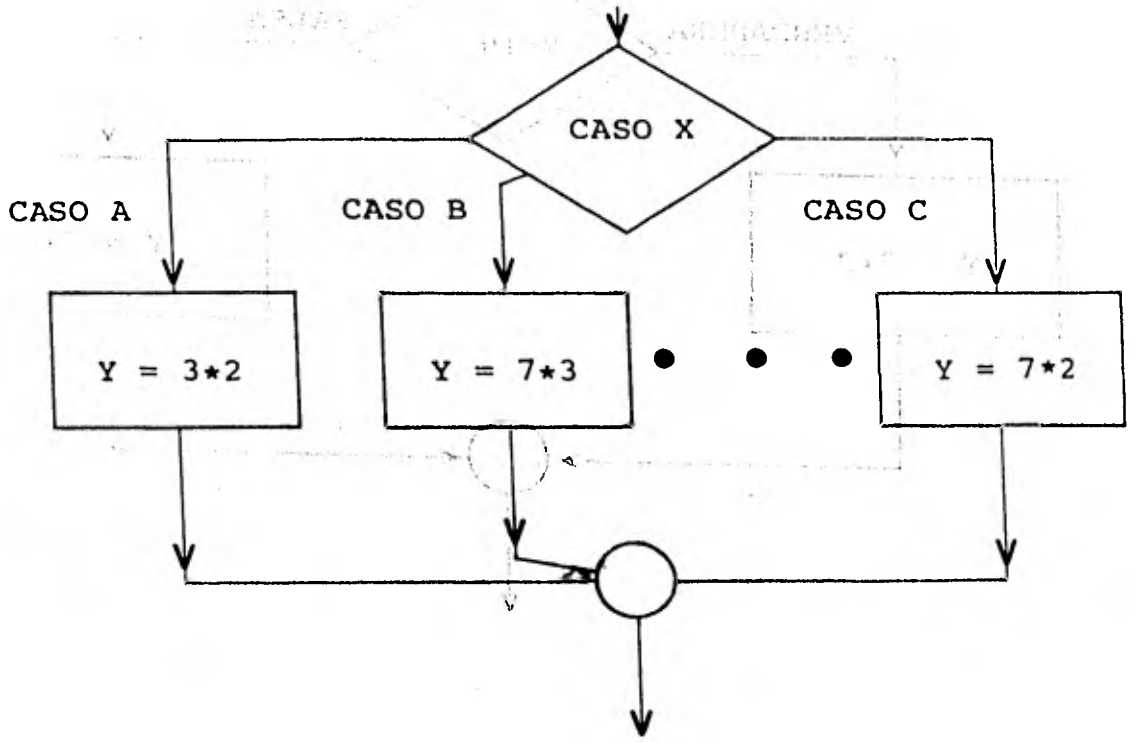
A.3 Construcción CASO (CASE)

El formato del CASE es :

1. CASO (A,B,C,...) condición
    A. bloque A
        1. separador de caso
    B. bloque B
        1. separador de caso
    C. bloque C
        1. separador de caso

Donde condición puede tomar varios valores y los bloques A,B,C,... son proposiciones en Pseudo-código.

Ejemplo : 1. CASO (A,B,C)X
    A. Y = 3*2
        1. separador de caso
    B. Y = 7*3
        1. separador de caso
    C. Y = 7*2
        1. separador de caso



b) Repetición.

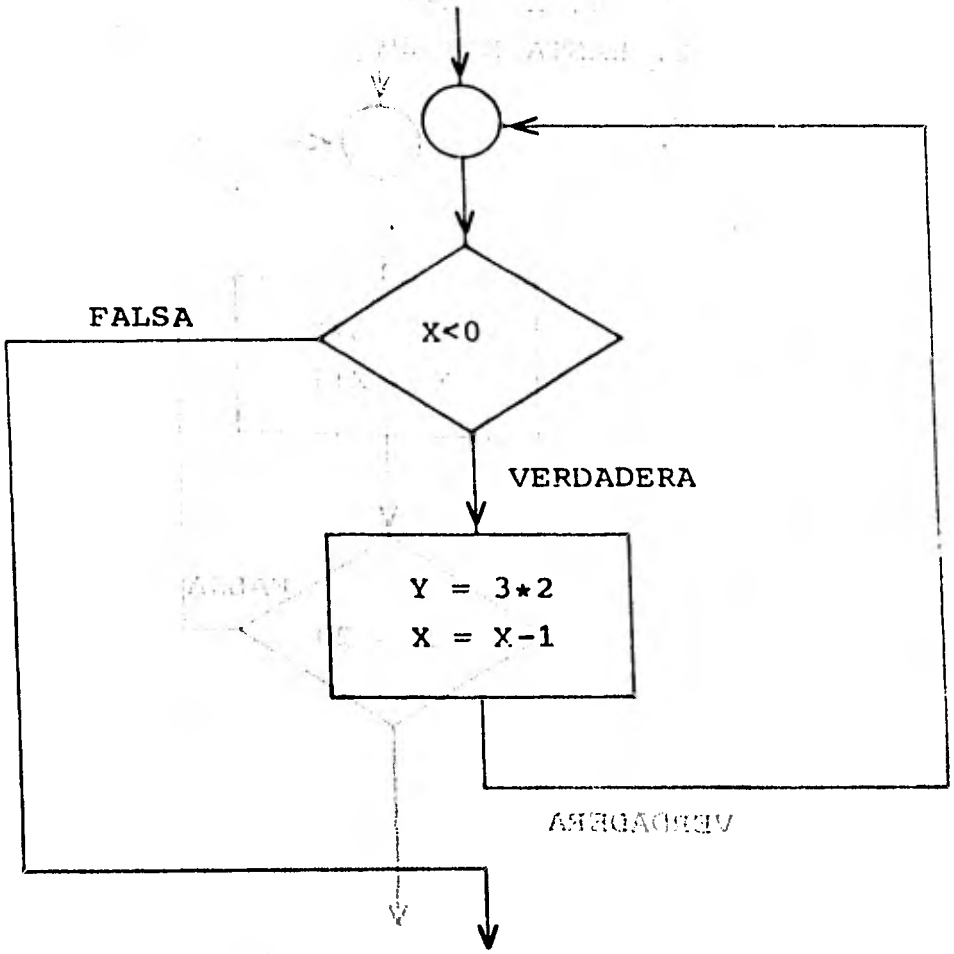
B.1 Construcción EJECUTA MIENTRAS (WHILE)

El formato del MIENTRAS es :

1. EJECUTA MIENTRAS condición
   A. bloque

Donde condición es una expresión lógica y el bloque A. son proposiciones en Pseudo-código.
En este caso se checa primero la condición, y si se - cumple, se ejecuta el bloque A.

Ejemplo : 1. MIENTRAS X < 0
A. Y = 3*2
B. X = X-1



B.2 Construcción EJECUTA-HASTA (DO-UNTIL)

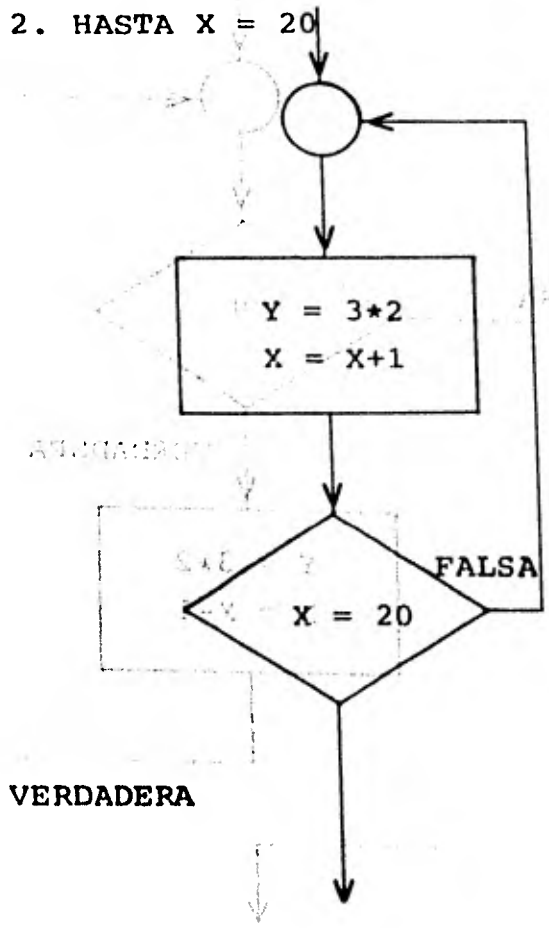
El formato del EJECUTA-HASTA es :

1. EJECUTA
   A. bloque
2. HASTA condición

Donde condición es una expresión lógica y el bloque A son proposiciones en Pseudo-código.

En este caso se ejecuta el bloque A y luego se checa la condición, ya sea para continuar ejecutando o salirse del bloque, de tal manera que siempre se ejecuta al menos una vez el bloque A.

Ejemplo : 1. EJECUTA
A. Y = 3*2
B. X = X+1
2. HASTA X = 20



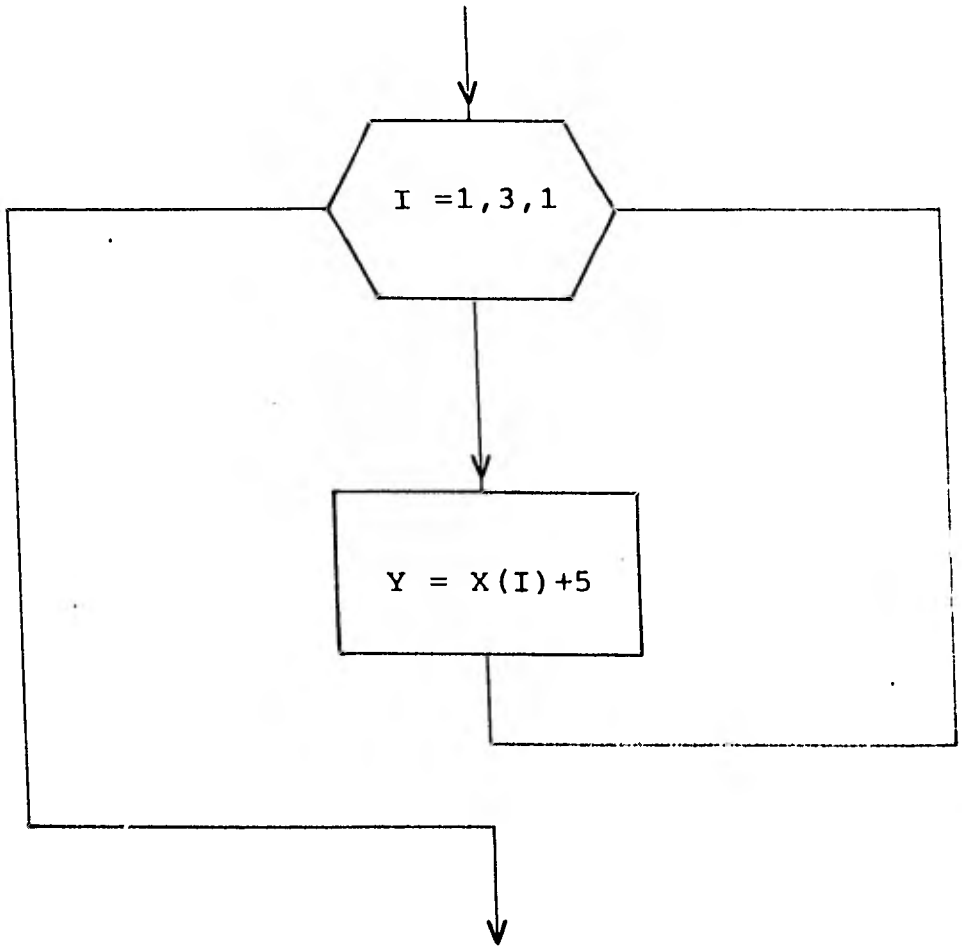
B.3 REPITE (FOR/DO)

El formato del REPITE (FOR) es :

1. REPITE I = N,M,(S)
   A. bloque

Donde N y M son enteros; N < M y el número de iteraciones será igual a M-N+1, S(opcional) es el incremento.- El bloque es un conjunto de proposiciones en Pseudo-código.

Ejemplo : REPITE I = 1,3,1
A. Y = X(I)+5



c) Llamadas a Módulos.

El formato de LLAMA (CALL) es :

LLAMA módulo

Donde módulo es el nombre de una rutina completa a - ejecutar.

Ejemplo : LLAMA rutina 1

V EJEMPLOS.

A continuación se presentan varios ejemplos de 'LDP', - para programción no numérica.

Ejemplo no.1

PROCEDIMIENTO 'selección de candidatos'

1. inicializa la selección de candidatos
2. EJECUTA
   A. selecciona al siguiente candidato
   B. SI se acepta ENTONCES
      1. registrarlo
   C. SI NO lo acepta
      1. intenta con el siguiente candidato
3. HASTA que haya éxito o no haya más candidatos

Los siguientes ejemplos fueron tomados de 'Software - Quality Assurance' del 'IEEE Standards Seminar'.

Ejemplo no. 2

```
3278 * CIFM - INVOKE THE FILE MANAGER (CALL LEVEL>4)
3279 *
3280 * INPUT    1. SRCRD - THE COMMAND
3281 *           2. CMTP - COMMAND TYPE
3282 *
3283 * OUTPUT   1. ? ON ERROR
3284 *
3285 * SUBROUTINES
3286 *           1. FMLD - LIST THE DIRECTORY
3287 *           2. FMPK - PACK THE DISK
3288 *           3. CIFT - GET FILE NAME AND TYPE
3289 *           4. FMRT - READ THE TAPE
3290 *           5. FMDF - DELETE THE FILE
3291 *           6. FMLE - LIST THE DIRECTORY ENTRY
3292 *           7. ISPR - PRINT A LINE
3293 *           8. FMCF - CREATE A FILE
3294 *           9. CIDC - EVALUATE A DECIMAL INTEGER
3295 *
3296 * LOGIC    1. IF COMMAND=LD (CMTP=6)
3297 *              A. CALL FMLD
3298 *           2. IF COMMAND=PK (CMTP=8)
3299 *              A. CALL FMPK
3300 *           3. IF COMMAND=CF,DF,RT,LE
3301 *              A. GET FILE NAME AND TYPE
3302 *              B. IF NO ERROR
3303 *                 1. IF COMMAND=RT
3304 *                    A. CALL FMRT
3305 *                 2. IF COMMAND=DF
3306 *                    A. CALL FMDF
3307 *                 3. IF COMMAND=LE
3308 *                    A. CALL FMLE
3309 *                 4. IF COMMAND=CF
3310 *                    A. IF NEXT CHARACTER NOT COMMA
3311 *                       1. PRINT ?
3312 *                    B. ELSE
3313 *                       1. MOVE PAST COMMA
3314 *                       2. EVALUATE THE DECIMAL INTEGER
3315 *                       3. IF NO ERROR
3316 *                          A. CALL FMCF
3317 *           4. RETURN
3318 *
3319 CIFMRA     DS        1                    RETURN ADDRESS SAV
3320 CIFMI1     DS        1                    IAR1 SAVE
3321 CIFMI2     DS        1                    IAR2 SAVE
3322 CIFMFLAR   DS        5                    FILE ARGUMENT
3323 CIFMG1     DS        1                    GR1 SAVE
3324 CIFM       STORE     1;CIFMG1;            SAVE GR1
3325             CPIG      1,,1;;;              SAVE IAR1
3326            STORE     1;CIFMI1;
3327            ILDI      1;CIFMRA;            SAVE RETURN ADDRES
3328            PUPM      ,,,1;;;
3329            CPIG      2,,1;;;              SAVE IAR2
3330            STORE     1;CIFMI2;
3331            LOAD      5;CMTP;              CMTP=6?
3332            CPIM      5;6;
3333            BUF       1;CIFM2;             SKIP IF NOT LD
3334            BRAN      0;FMLD;              LIST THE DIRECTORY
3335            BOF       0;CIFM4;             RETURN
```

LDP incluido dentro de los comentarios de un programa

```
D187F
D1880
D1881
D1882
D1887
P050D   1A 1 1887
P050E   09 1 0 1 0 7 0
P050F   1A 1 1880
P0510   1C 1 187F
P0511   06 0 0 0 1 7 0
P0512   09 2 0 1 0 7 0
P0513   1A 1 1881
P0514   13 5 187A
P0515   18 5 0006
P0516   11 1 0519
P0517   12 0 0724
P0518   11 0 0539
```

Ejemplo no. 3

```
                    3086 * CIXC - EXECUTE THE  ..MAND (CALL LEVEL >4)
                    3087 *
                    3088 * INPUT    1. CMIP - COMMAND TYPE
                    3089 *
                    3090 * OUTPUT   1. ? FOR COMMAND ERROR
                    3091 *
                    3092 * SUBROUTINES
                    3093 *          1. CICH - IDENTIFY THE COMMAND
                    3094 *          2. ISPR - PRINT A LINE
                    3095 *          3. CIFM - INVOKE THE FILE MANAGER
                    3096 *          4. CIMA - INVOKE THE NATIVE ASSEMBLER
                    3097 *          5. CILL - INVOKE THE LINKING LOADER
                    3098 *          6. CIDO - EXECUTE A DO COMMAND
                    3099 *
                    3100 * LOGIC    1. IDENTIFY THE COMMAND
                    3101 *          2. IF THE COMMAND IS UNDEFINED
                    3102 *             A. PRINT A QUESTION MARK
                    3103 *          3. IF THE COMMAND IS CF,RI,DF,LD,LE,PK
                    3104 *             A. INVOKE THE FILE MANAGER
                    3105 *          4. IF THE COMMAND IS ASM
                    3106 *             A. INVOKE THE NATIVE ASSEMBLER
                    3107 *          5. IF THE COMMAND IS LK
                    3108 *             A. INVOKE THE LINKING LOADER
                    3109 *          6. IF THE COMMAND IS DO
                    3110 *             A. EXECUTE A DO COMMAND
                    3111 *          7. RETURN
                    3112 *
D1B79               3113 CIXCRA     DS        1                    RETURN ADDRESS SAVE
D1B7A               3114 CMIP       DS        1                    COMMAND TYPE
                    3115 *                                         0=UNIDENTIFIED
                    3116 *                                         1=CF
                    3117 *                                         2=RI
                    3118 *                                         3=ASM
                    3119 *                                         4=LK
                    3120 *                                         5=DF
                    3121 *                                         6=LD
                    3122 *                                         7=LE
                    3123 *                                         8=PK
                    3124 *                                         9=DO
D1B7B               3125 CIXCII     DS        1                    IARI SAVE
D1B7C        0002   3126 CIXCQM     DC        2;                   QUESTION MARK STRING
D1B7D        203F   3127            DC        X'203F'
D1B7E               3128 CIXCGI     DS        1                    GRI SAVE
P0498   14 1 1B7E   3129 CIXC       STORE     1;CIXCGI; SAVE GRI
P0499   09 1 0 1 0 7 0 3130         CPIG      1,,1;;; SAVE IARI
P049A   1A 1 1B7B   3131            STORE     1;CIXCII;
P049B   1C 1 1B79   3132            ILDI      1;CIXCRA;            SAVE
P049C   06 0 0 0 1 7 0 3133         PDPM      ,,1;;;               RETURN ADDRESS
P049D   12 0 04C0   3134            BRAN      0;CICH;              IDENTIFY THE COMMAND
P049E   13 1 1B7A   3135            LOAD      1;CMIP;              COMMAND TYPE
P049F   18 1 0000   3136            CMPI      1;0;                 CMIP=0? (UNDEFINED)
P04A0   11 1 04A4   3137            BOF       1;CIXC3;             SKIP IF NOT 0
P04A1   10 1 203F   3138      LDI   1;X'203F';  PRINT A QUESTION MARK
P04A2   12 0 0180   3139      BRAN  0;ISP7;
P04A3   11 0 04BA   3140      BOF 0;CIXC7;  RETURN
P04A4   18 1 0001   3141 CIXC3      CMPI      1;1;                 CMIP=1? (CF)
P04A5   10 1 04B0   3142            BOT       1;CIXCFM;            IN IF CF
P04A6   18 1 0002   3143            CMPI      1;2;                 CMIP=2? (RI)
P04A7   10 1 04B0   3144            BOT       1;CIXCFM;            IN IF RI
```

LDP incluido en los comentarios de un programa

Ejemplo no. 4

```
2986 *
2987 * CIGC -  GET THE NEXT COMMAND (CALL LEVEL 1)
2988 *
2989 * INPUT     1. DOTOP - DO FILE STACK TOP
2990 *           2. COMMAND FILES
2991 *           3. IOST - END-OF-FILE
2992 *           4. DOST - DO FILE STACK
2993 *
2994 * OUTPUT    1. SRCRD - COMMAND STRING
2995 *           2. DOTOP - DO FILE STACK IS POPPED ON END-OF-FILE
2996 *           3. PRINTER PROMPTED WITH -
2997 *           4. SRCH - FIRST CHARACTER = SRCRD(1)
2998 *           5. SRIX - SOURCE INDEX = 1
2999 *           6. SRLH = SRCRD(0) = LENGTH OF COMMAND
3000 *
3001 * SUBROUTINES
3002 *           1. ISRD - READ A DISK RECORD
3003 *           2. ISCL - CLOSE A FILE
3004 *           3. ISPR - PRINT A LINE
3005 *           4. ISKB - READ THE KEYBOARD
3006 *
3007 * LOCAL DATA
3008 *           1. FCMD - FOUND A COMMAND (0=NO,1=YES)
3009 * LOGIC     1. FCMD=0 NO COMMAND YET
3010 *           2. WHILE DOTOP>= DOST AND FCMD=0  ACTIVE DO FILE AND NO
3011 *              A. READ A RECORD FROM THE FILE ON TOP OF THE STACK
3012 *              B. IF END-OF-FILE
3013 *                 1. CLOSE THE FILE
3014 *                 2. POP THE STACK
3015 *              C. ELSE
3016 *                 1. FCMD=1  FOUND A COMMAND
3017 *                 2. PRINT THE COMMAND
3018 *           3. IF FCMD=0  (NO COMMAND YET)
3019 *              A. PROMPT THE KEYBOARD
3020 *              B. READ THE KEYBOARD
3021 *           4. SRIX=1  WORD INDEX
3022 *           5. SRCH=SRCRD(1)  SCAN CHARACTER
3023 *           6. SRLH=SRCRD(0)  LENGTH
3024 *           7. RETURN
3025 *
3026 DOTOP     DS        1                    TOP OF DO FILE STACK
3027 DOMX      $EQU      50                   SIZE OF DO FILE STACK (10
3028 DOST      DS        DOMX                 DO FILE STACK  EACH 5-WORD
3029 *                                        A FILE ARGUMENT
3030 DOEND     $EQU      DOST+DOMX-1          LAST ADDRESS OF DO FILE ST
3031 CIGCFCMD  DS        1                    FOUND A COMMAND (0=NO,1=YE
3032 CIGCI1    DS        1                    IAR1 SAVE
3033 CIGCI2    DS        1                    IAR2 SAVE
3034 CIGCG1    DS        1                    GR1 SAVE
3035 CIGCPRCH  DC        2                    PROMPT STRING
3036           DC        X'202D';              MINUS SIGN
3037 CIGC      STORE     1;CIGCG1;            SAVE GR1
3038           CPIG      1,1,;;               SAVE IAR1
3039           STORE     1;CIGCI1;
3040           CPIG      2,1,;;               SAVE IAR2
3041           STORE     1;CIGCI2;
3042           ZR        5;;                  FCMD=0
3043           STORE     5;CIGCFCMD;
3044 CIGC2     LOAD      5;DOTOP;             DOTOP>=DOST?
```

LDP incluido en los comentarios de un programa

```
DIB40
        50
DIB41
      7026
DIB73
DIB74
DIB75
DIB76
DIB77          0002
DIB78          202D
P0468   1A 1 1B76
P0469   09 1 0 1 0 7 0
P046A   1A 1 1B74
P046B   09 2 0 1 0 7 0
P046C   1A 1 1B75
P046D   0A 5 0 5 0 0 0
P046E   1A 5 1B73
P046F   13 5 1B40
```

APENDICE B.3

AUDITOR DE CODIGO

CONTENIDO PAGINA

1. INTRODUCCION.

En virtud de que los programas que se desarrollen serán utilizados por personas ajenas al desarrollo, es necesario que la programación mantenga un estilo uniforme guiándose por ciertas normas de programación, de tal forma que sean fáciles de seguir, revisar, modificar y mantener por aquellas personas directamente interesadas en su aplicación.

Se presenta un "auditor" o checador de código cuyo objeto es controlar la calidad de los programas desarrollados, siguiendo como base las Normas de Programación, agregando algunos requerimientos y modificaciones que se han considerado necesarios para el mejor desarrollo de la programación.

El programa auditor verifica cada una de las normas de programación detectando los errores en los programas. El presente documento presenta, además de las normas de programación, una lista de los errores que detecta el auditor.

2. OBJETIVO.

El programa auditor de código, tiene la función de revisar los programas elaborados, identificando los aspectos de programación que se encuentren fuera de las normas establecidas. De esta forma se intenta unificar el estilo de programación de los analistas.

El objetivo del auditor no es optimizar, ni limitar la creatividad del programador, sólo se desea, mediante la unificación del estilo, obtener programas que presenten una estructura coherente y una mayor claridad.

Las normas de programación que permiten alcanzar tal objetivo pueden ser agrupadas en dos bloques principales :

- Normas de documentación en los encabezados.
- Normas de código.

Las primeras, normas de documentación en los encabezados, proporcionan a los programas una mayor claridad; mientras que las normas de código ayudan al programador en la elaboración de programas con estructura coherente.

3. NORMAS DE PROGRAMACION.

3.1 Normas de documentación en los encabezados.

Todo módulo (programa, subrutina o función) debe contener una descripción breve de su función, uso, variables que maneja y restricciones pertinentes sobre el uso del módulo.

Estos comentarios deben aparecer inmediatamente después del nombre del módulo y deberán estar delimitadas por dos líneas de asteriscos.

Los comentarios de encabezados deberán incluir, por orden de aparición los siguientes puntos :

a) Objetivo o propósito :
Describir en dos o tres líneas la función que realiza el módulo y, en caso de ser relevante, indicar el método empleado.

b) Entradas :
Proporcionar la siguiente información sobre las variables y constantes empleadas como entradas a un módulo (se entiende por entrada, cualquier cato que siendo generado por otro módulo, sea empleado por éste) :

Nombre : tipo = descripción

donde : el nombre es el nombre mnemónico de la variable. La descripción consiste en unas cuantas palabras que indican el significado físico de la variable.

c) Salidas :
Consiste en proporcionar para las variables de salida la misma información indicada para las variables de entrada, entendiéndose por salidas aquellos datos que se generan en el módulo y son empleados por otros módulos.

d) Restricciones :
Aquellas restricciones que existan para el módulo, dado el diseño del mismo.

e) Módulos :
Consiste en proporcionar el nombre y la descripción de todos los módulos, subrutinas o funciones que son utilizados por el presente módulo.- El nombre y descripción se presentará en el siguiente formato :

Nombre = Descripción

f) Identificadores locales :
Consiste en proporcionar para los identificadores locales la misma información indicada para las entradas, entendiéndose por identificadores locales aquellos datos que sean generados y usados internamente por el módulo.

g) Implementado por :
Identificar al realizador del módulo.

h) Fecha del último cambio :
Indicar la fecha, en forma extendida, del último cambio realizado al módulo.

i) Autor del cambio :

Identificar al autor del último cambio efectuado al módulo.

En la siguiente figura se presenta el formato general de los comentarios de Encabezado de un Programa.

```
C*******************************************************
C
C     PROPOSITO:
C
C
C     RPC: 0001
C     DDPC: 0001
C     RCPC: NUMERO DEL REQUERIMIENTO DE CAMBIO
C
C*    REFERENCIAS:
C        LISTA DE DOCUMENTOS QUE HAN SIDO UTILIZADOS PARA
C        ESTUDIO DEL ALGORITMO EMPLEADO EN EL PROGRAMA
C
C     PROCESO:
C        DESCRIPCION DEL ALGORITMO DEL PROGRAMA UTILIZANDO EL
C        LENGUJE LDP (LENGUJE PARA DISENO DE PROGRAMAS)
C
C
C
C     ENTRADAS:
C        MNEMONICO,I,J,K,L:ENTERA=DESCRIPCION;ADIM; R[1,100]
C        MNEMONICOA: REAL = DESCRIPCIONA; UNIDADESA; RANGOA(I,J]
C        MNEMONICOB: REAL = DESCRIPCIONB; UNIDADESB; 10-20
C        MNEMONICOC: REAL = DESCRIPCIONC; UNIDADESC; 100-200
C
C     SALIDAS:
C        MNEMONICO: LOGICA= DESCRIPCION; UNIDADES; 1-10
C        MNEMONICOX: ENTERA= DESCRIPCIONX; UNIDADESX; RANGOX [1,10]
C        MNEMONICOY: REAL = DESCRIPCIONY; UNIDADESY; RANGOY (0,1)
C        MNEMONICOZ: COMPLEJA= DESCRIPCIONZ; UNIDADESZ; 1-10
C
C
C     RESTRICCIONES :
C        LAS RESTRICCIONES DEL PROGRAMA DEBIDAS AL ALGORITMO EMPLEADO
C
C     SUBRUTINAS:
C        NOMBRE= DESCRIPCION
C        NOMBRE1= DESCRIPCION1
C        NOMBRE2= DESCRIPCION2
C
C     IDENTIFICADORES LOCALES:
C        MNEMONICO: REAL= DESCRIPCION; UNIDADES; 1-10
C        MNEMONICO1: ENTERA= DESCRIPCION1; UNIDADES1; RANGO1 [1,10]
C        MNEMONICO2: LOGICA= DESCRIPCION2; UNIDADES2; RANGO2 (0,1)
C
C
C
C     IMPLEMENTADO POR: AUTOR DEL PROGRAMA
C     FECHA DE ULTIMO CAMBIO: DIA DE  MES DE  ANO
C     AUTOR DEL CAMBIO: NOMBRE DEL AUTOR DEL CAMBIO
C
C*******************************************************
```

3.2 Normas de Código.

Los programas que se codifiquen siguiendo éstas normas, mostrarán una estructura coherente, lo que facilitará la comprensión de la lógica del programa, y será más sencillo el mantenimiento del mismo.

Se enumeran a continuación las diferentes normas decódigo indispensables para una programación coherente :

a) Longitud y nombre de las variables.
El nombre de los identificadores no debe de exceder de 10 caracteres y debe de representar la función que desempeñe.

b) Longitud y nombre de las subrutinas o módulos.
El nombre de los identificadores no debe de exceder de 10 caracteres y debe de representar la función que desempeñe, para el caso del compilador Harris, se debe cuidar que los 6 primeros caracteres no sean iguales en varios módulos.

c) Las variables deben ser explícitamente declarados.
No dejar que el sistema asuma el tipo de variable en función de la letra con la que empieza el nombre del identificador.

d) Tamaño del módulo o subrutina.
El tamaño de los módulos no debe de exceder de 120 líneas, incluyendo encabezados, código y comentarios adicionales.

e) Localización de los 'DATAS'.
Estos deben aparecer inmediatamente después de las declaraciones.

f) Localización de los formatos.
Los formatos deben aparecer al final de la subrutina, esto es al finalizar las líneas de código y antes del comando 'RETURN'.

g) Secuencia de la etiqueta.
Las etiquetas dentro de las líneas de código deben emepzar a partir de la etiqueta 100 y seguir consecutivamente de 100 en 100.
La etiqueta de los formatos deben empezar a partir de la etiqueta 5000 y seguir en orden ascendente de 100 en 100.

h) Uso de las etiquetas.
Todas las etiquetas deberán ser usadas, esto es, no se permite etiquetar una línea de código cuando ésta no la requiera.

i) Restricciones.
Queda absolutamente prohibido el uso de las siguientes estructuras y comandos :

1. Go to.
   Su uso queda prohibido puesto que se trata de un lenguaje Fortran estructurado.
2. Implicit.
3. Entry.
4. Equivalence.
5. Stop.
6. Call exit en subrutinas o módulos.

7. Common no etiquetado.
8. Do etiquetado.
9. If convencional o aritmético.
10. Múltiples returns.
11. Read y write no formateados.

j) Tamaño de los arreglos.
Los arreglos no deben de ser mayores de 3 dimensiones.

k) Comentarios adicionales.
Los programas deberán tener un mínimo de 25% de comentarios adicionales dentro del código.

l) Separación entre el encabezado y el código.
Es necesario separar por una línea de comentario en blanco el encabezado y las líneas de código.

m) Empleo de estructuras lógicas y de control apropiados.
Aprovechando el lenguaje Fortran 77 del sistema Harris, es conveniente usar la estructura lógica y de control que brinda :

- If - then - else
- Do - until
- While
- Repeat
- For
- etc.

3.3 Aspectos de Programación que detecta el auditor de código.

El programa auditor de código, se compara con el compilador de un sistema de cómputo, el cual detecta -- los errores de sintaxis en el lenguaje de los programas. El programa auditor de código checa los errores en el estilo y propiedad de la programación.

Presentamos a continuación una lista de los avisos - de error que manda el programa auditor :

- MENSAJES DE ERROR -

a) **ERROR : NO EXISTE ENCABEZADO EN LA SUBRUTINA**

El mensaje aparece cuando la subrutina no tiene- encabezado, falta documentarse.

b) **ERROR : HAY ERROR EN LA SECUENCIA DEL ENCABEZADO**

Este mensaje aparece cuando los comentarios del- encabezado no se presentan en el orden apropiado.

c) **ERROR : FALTA EL ARGUMENTO <IDENTIFICADOR> **

El mensaje aparece cuando falta un comentario en el encabezado.

d) **ERROR : EL ENCABEZADO DE LA SUBRUTINA NO LLEGO A BUEN FIN**

El mensaje se presenta cuando existió algunos de los errores anteriores, y aparece al final de la subrutina.

e) **ERROR : LA VARIABLE <IDENTIFICADOR> NO HA SIDO COMENTADA EN EL PROGRAMA O SUBRUTINA QUE LA UTILIZA**

El mensaje anterior aparece cuando una variable no ha sido declarada o comentada dentro del encabezado, o bien, cuando el nombre de la variable excede los 10 caracteres, de tal forma que no puede ser identificada.

f) **ERROR : LA SUBRUTINA <IDENTIFICADOR> NO HA SIDO COMENTADA EN EL PROGRAMA O SUBRUTINA QUE LA LLAMA**

Este mensaje aparece cuando una subrutina no ha sido comentada dentro del encabezado, o bien, cuando el nombre de la subrutina excede los 10 caracteres, de tal forma que no puede ser identificada.

g) **ERROR : HAY ERROR EN LA SECUENCIA DE LAS ETIQUETAS**

El mensaje aparece cuando las etiquetas no se encuentran numeradas de 100 en 100.

h) **ERROR : HAY ERROR EN LA SECUENCIA DE LAS ETIQUETAS DE FORMATO**

El mensaje aparece cuando las etiquetas de formato no se encuentran numeradas de 100 en 100, o no empiezan a partir de la etiqueta 5000.

i) **ERROR : LOS FORMATOS NO SE ENCUENTRAN AL FINAL DE LA SUBRUTINA**

El mensaje anterior aparece cuando los FORMATOS-

no se encuentran al finalizar las líneas de código (deben aparecer antes del RETURN).

j) **ERROR : EL NUMERO DE LINEAS EN LA SUBRUTINA ES MAYOR DE 120**

Este mensaje aparece cuando el número de líneas en la subrutina, incluyendo comentarios y encabezado, es mayor de 120.

k) **ERROR : -USO PROHIBIDO- HAY UN COMMON NO ETIQUETADO**

El mensaje aparece cuando en las declaraciones hay commons no etiquetados (COMMON/Nombre/Identificadores).

l) **ERROR : -USO PROHIBIDO- HAY UN GO TO EN SUBRUTINA**

El mensaje aparece cuando en el código hay GO TO's, es conveneiente usar Estructuras Lógicas y de Control apropiados.

m) **ERROR : -USO PROHIBIDO- HAY UN STOP EN SUBRUTINA**

El mensaje aparece cuando en el código existe un STOP, que ocasiona la interrupción en la comida del programa.

n) **ERROR : -USO PROHIBIDO- HAY UN CALL EXIT EN SUBRUTINA**

El mensaje aparece cuando en el código existe un CALL EXIT lo que ocasiona una salida del programa sin haber completado la corrida.

o) **ERROR : -USO PROHIBIDO- HAY UN ENTRY EN SUBRUTINA**

Este mensaje aparece cuando en el código existe un ENTRY que no es una forma apropiada de entrar a una subrutina.

p) **ERROR : -USO PROHIBIDO- HAY UN IF ARITMETICO**

El mensaje aparece cuando en el código hay un IF aritmético, es conveniente usar estructuras lógicas, de control apropiado.

q) **ERROR : -USO PROHIBIDO- HAY UN IMPLICIT EN SUBRUTINA**

El mensaje aparece cuando en las declaraciones hay variables declaradas como IMPLICIT.

r) **ERROR : -USO PROHIBIDO- HAY UN EQUIVALENCE EN SUBRUTINA**

El mensaje aparece cuando en las declaraciones hay variables declaradas como EQUIVALENCE.

s) **ERROR : -USO PROHIBIDO- HAY UN DO ETIQUETADO**

El mensaje aparece cuando en el código existe un DO etiquetado, es conveniente usar estructuras lógicas y de control apropiadas.

t) **ERROR : -USO PROHIBIDO- HAY READ O WRITE NO FORMATEADO**

El mensaje aparece cuando en el código existe un READ o WRITE no formateado, es decir, con formato libre.

u) **ERROR : -USO PROHIBIDO- HAY ARREGLOS DE MAS DE TRES DIMENSIONES**

Este mensaje aparece cuando existen arreglos de más de 3 dimensiones en alguna subrutina.

v) **ERROR : -USO PROHIBIDO- HAY ETIQUETAS NO REQUERIDAS**

El mensaje aparece cuando en el código existen etiquetas no requeridas.

w) **ERROR : LOS DATAS NO APARECEN DESPUES DE LAS DECLARACIONES**

El mensaje aparece cuando en las declaraciones los DATA's se encuentran antes de las declaraciones de variables.

x) **ERROR : EXISTEN MULTIPLES RETURNS**

El mensaje aparece cuando en el código de la subrutina existen más de 1 RETURN.

y) **ERROR : FALTA SEPARAR EL ENCABEZADO DE LAS LINEAS DE CODIGO**

Este mensaje aparece cuando no existe la línea de comentario en blanco entre el encabezado y las líneas de código.

z) **ERROR : EL PORCENTAJE DE COMENTARIOS EN EL PROGRAMA ES MENOR DEL 25%**

El programa auditor proporciona el NUMERO DE LINEAS DE COMENTARIOS adicionales y el NUMERO DE LINEAS DE CODIGO si el porcentaje de líneas de comentario es menor del 25%.

APENDICE C.

P R O G R A M A S

APENDICE C.1

CODIGO DEL PROGRAMA PROTOTIPO DE ESTIMACION DE CARGA.

"METODO DE COMPONENTES"

```
16 JUL 81 10:18:42   HARRIS FORTRAN 77  OPTIMIZING COMPILER     01411
 MODULE NAME:   *MAIN*
   1: C          PROGRAMA PRINCIPAL
   2: C*****************************************************************
   3: C       OBJETO
   4: C          SIMULAR EL PRONOSTICO DE CARGA MEDIANTE EL METODO DE
   5: C          COMPONENTES VERSION SIN RUIDO
   6: C          SE UTILIZAN MODELOS CON GRUPOS DE DIAS DE LA SEMANA
   7: C
   8: C       IMPLEMENTADA POR MOTOKI INOUE KUBO
   9: C       FECHA DE ULTIMA REVISION  2 DE OCTUBRE DE 1980
  10: C       REVISADA POR  MOTOKI INOUE
  11: C
  12: C       ENTRADAS
  13: C             ID = IDENTIFICACION DEL DIA CON QUE SE INICIA LA
  14: C                  SIMULACION
  15: C         LOTIME = NUMERO DE DIAS POR PRONOSTICAR
  16: C            TOL = TOLERANCIA PARA LA BANDA DEL FILTRO
  17: C
  18: C       SALIDAS
  19: C          RESULTADOS DE PRONOSTICOS Y SU RESPECTIVA EVALUACION
  20: C          EN EL ARCHIVO DE IMPRESION
  21: C
  22: C       IDENTIFICADORES LOCALES
  23: C           IENT = ARCHIVO DE CARGAS REALES PARA LA INICIALIZACION
  24: C                  DEL MODELO DE PRONOSTICO
  25: C           IRES = ARCHIVOS CON LOS RESIDUOS DEL PRONOSTICO
  26: C          IENT2 = ARCHIVO DE CARGAS REALES (MEDICIONES)
  27: C           IEVA = ARCHIVO DE LAS ESTADISTICAS DE LA EVALUACION
  28: C           ISAL = ARCHIVO CON LOS PARAMETROS DE LOS MODELOS DE
  29: C                  PRONOSTICO
  30: C          ISAL2 = ARCHIVOS CON LOS VALORES PRONOSTICADOS
  31: C             IR = ARCHIVO DE LECTURA (UNIDAD 5)
  32: C             IW = ARCHIVO DE IMPRESION (UNIDAD 6)
  33: C           IERR = INDICADOR DE DATOS RECHAZADOS
  34: C            IPR = CONTADOR DE LOS PRONOSTICOS REALIZADOS
  35: C              J = CONTADOR
  36: C
  37: C       MODULOS EMPLEADOS
  38: C          INICIO = INICIALIZACION DEL MODELO DE PRONOSTICO
  39: C          EVALUA = EVALUACION DE PRONOSTICOS
  40: C          FCAST  = REALIZACION DE LOS PRONOSTICOS
  41: C          UPDATE = ACTUALIZACION DE LOS PARAMETROS DE LOS MODELOS
  42: C                   DE PRONOSTICO
  43: C
  44: C       NOTAS
  45: C          ESTA VERSION AUN NO CONTEMPLA EFECTOS METEOROLOGICOS
  46: C
  47: C*****************************************************************
  48:        COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
  49:       *  BETA(7,24)
  50:        COMMON/TRES/IERR
  51:        COMMON/DAY/IDAY,MONTH,IYEAR
  52:        COMMON/CARGA1/ZP(24)
  53:        COMMON/CARGA2/Z(24)
```

```
81 10:18:43   HARRIS FORTRAN 77   OPTIMIZING COMPILER     01411                RE
E NAME:  *MAIN*
      COMMON/PARAM/ALFA,M,TD,IPR,IHORA,LDTIME,INIH,ND,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
C     ------------------------------------------------------
C     *ASIGNACION DE ARCHIVOS
C     ------------------------------------------------------
      IR=25
      IW=6
      IENT=10
      IENT2=11
      ISAL=15
      ISAL2=16
      IEVA=17
      IRES=12
      REWIND IENT
      REWIND IRES
      REWIND ISAL
      REWIND ISAL2
      REWIND IEVA
      IPR=0
C     ------------------------------------------------------
C     *LLAMADA A LA SUBRUTINA DE INICIALIZACION
C     ------------------------------------------------------
      CALL INICIO
C     ------------------------------------------------------
C     *TERMINA INICIALIZACION
C      ESCRIBE LETRERO
C     ------------------------------------------------------
      WRITE(IW,5000)
C     ------------------------------------------------------
C     *LEER PERIODO DE PRONOSTICO
C     ------------------------------------------------------
      READ(IR,5100)LDTIME,TD,TOL
      READ(IR,5200)IDAY,MONTH,IYEAR
C     ------------------------------------------------------
C     *COMIENZA LA SIMULACION
C     ------------------------------------------------------
      FOR J=1,LDTIME
C        ------------------------------------------------
C        *REALIZA EL PRONOSTICO
C        ------------------------------------------------
       .  CALL ECAST
C        ------------------------------------------------
C        *ACTUALIZA LOS PARAMETROS
C        ------------------------------------------------
       .  CALL UPDATE
C        ------------------------------------------------
C        *SI EL DIA FUE NORMAL EVALUA LOS ERRORES
C        ------------------------------------------------
       .  IF(IERR.NE.1)CALL EVALUA
      ENDFOR
C     ------------------------------------------------------
C     *FIN DE LA SIMULACION
C     ------------------------------------------------------
```

```
JUL 81 10:18:45   HARRIS FORTRAN 77  OPTIMIZING COMPILER   01011
ODULE NAME:  *MAIN*
07:          CALL EXIT
08: C        ---------------------------------
09: C        *FORMATOS DE LECTURA E IMPRESION
10: C        ---------------------------------
11: 5000     FORMAT(1H0,10X,'LA INICIALIZACION HA CONCLUIDO NORMALMENTE')
12: 5100     FORMAT(2I5,F10.4)
13: 5200     FORMAT(3I2)
14:          END
```

```
91 10:18:46   HARRIS FORTRAN 77  OPTIMIZING COMPILER     01411                RE
E NAME:  *MAIN*
      SUBROUTINE INICIO
C*********************************************************************
C     OBJETO
C        INICIALIZAR LOS PARAMETROS DEL MODELO DE PRONOSTICO DE CARGA
C        MEDIANTE EL METODO DE COMPONENTES SINCRUIDO
C
C        SE EMPLEAN MODELOS PARA CADA GRUPO DE DIAS DE LA SEMANA
C
C     IMPLEMENTADA POR  MOTOKI INOUE KURO
C     FECHA DE LA ULTIMA REVISION  30 DE SEPTIEMBRE DE 1980
C     REVISADA POR MOTOKI INOUE
C
C     ENTRADAS
C        ALFA = FACTOR DE PESO PARA LA ACTUALIZACION DEL COMPONENTE
C               SEMANAL.
C        IDAY = CONJUNTAMENTE CON MONTH Y YEAR COMPONEN LA FECHA
C               DE LOS DATOS DE CARGAS REALES
C        IDIA = IDENTIFICACION DEL DIA CON QUE COMIENZA EL ARCHIVO DE
C               DATOS REALES
C           M = LONGITUD DE VENTANA DESLIZANTE, DEBE SER UN MULTIPLO
C               DE 7
C          ND = NUMERO DE DIAS DISPONIBLES PARA LA INICIALIZACION DE
C               LOS PARAMETROS
C         TOL = FACTOR DE TOLERANCIA DE LA BANDA DEL FILTRO
C           Z = CARGAS REALES (MEDICIONES)
C
C     SALIDAS
C        BETA = ARREGLO CON LOS FACTORES DE PESO HORARIO PARA LA COMPO-
C               NENTE SEMANAL
C        PROM = ARREGLO DE PROMEDIOS DE ENERGIA DIARIA PARA EL FILTRO
C        PRON = ARREGLO DESTINADO A CONTENER LAS ULTIMAS CUATRO SEMANAS
C               DE PRONOSTICO. SE INICIALIZA EN CEROS PARA SU POSTERIOR
C               USO EN EL PROGRAMA DE PRONOSTICO
C           S = COMPONENTE SEMANAL INICIALIZADO
C           T = COMPONENTE DE TENDENCIA INICIALIZADO
C          VE = ARREGLO CONTENIENDO LOS DATOS REALES DE LOS ULTIMOS
C               M DIAS
C
C     IDENTIFICADORES LOCALES
C        ALF2 = FACTOR DE ACTUALIZACION (ALF2=1.0-ALFA)
C        GROWF = FACTOR DE CRECIMIENTO
C           I = CONTADOR
C          ID = IDENTIFICADOR DEL DIA QUE SE PROCESA
C         ID4 = VARIABLE AUXILIAR PARA LA DETERMINACION DEL IDENTIFI-
C               CADOR DEL DIA Y DEL GRUPO
C         ID5 = VARIABLE AUXILIAR PARA LA DETERMINACION DEL IDENTIFI-
C               CADOR DEL DIA Y DEL GRUPO
C         ID6 = VARIABLE AUXILIAR PARA LA DETERMINACION DEL IDENTIFI-
C               CADOR DEL DIA Y DEL GRUPO
C         ID7 = IDENTIFICADOR AUXILIAR DEL GRUPO DE DIA DE LA SEMANA
C        IERR = INDICADOR DE DIA RECHAZADO
C         IPR = CONTADOR DE PRONOSTICOS, SOLO SE INICIALIZA Y NO ES
```

```
C*****************UTILIZADO DENTRO DE ESTE MODULO*********************
C            IK = CONTADOR
C          IRECH = CONTADOR DE DIAS RECHAZADOS
C             J = CONTADOR
C            JI = CONTADOR
C            AK = CONTADOR
C            KJ = CONTADOR
C            KK = CONTADOR
C          KONT = VARIABLE AUXILIAR PARA LA IDENTIFICACION DEL DIA
C             L = CONTADOR
C            LL = VARIABLE AUXILIAR CONTENIENDO LA IDENTIFICACION DEL
C                 DIA EN LA ETAPA DE INICIALIZACION DEL FILTRO
C            LX = VARIABLE AUXILIAR PARA LA DETERMINACION DEL IDENTIFI
C                 CACION DEL DIA
C            M1 = VARIABLES CON LIMITES DE ITERACIONES
C            M2 = VARIABLES CON LIMITES DE ITERACIONES
C            M3 = VARIABLES CON LIMITES DE ITERACIONES
C           NNA = CONTADOR
C          NDIA = CONTADOR DE DIAS UTILIZADOS PARA LA INICIALIZACION
C          TEMP = VARIABLE AUXILIAR PARA EL CALCULO DE LA COMPONENTE
C                 SEMANAL
C
C      MODULOS EMPLEADOS
C        INTFIL = INICIALIZACION DE LOS PARAMETROS DEL FILTRO
C        HISDAT = LECTURA DE LOS DATOS REALES HISTORICOS
C        INITEN = INICIALIZACION DE LA COMPONENTE DE TENDENCIA
C        INICIC = INICIALIZACION DE LA COMPONENTE CICLICA
C        SEMDIA = DETERMINACION DEL DIA DE LA SEMANA
C         ANORM = SUAVIZAMIENTO DE DATOS ANORMALES
C        ACTTEN = ACTUALIZACION DE LA COMPONENTE DE TENDENCIA
C        ACTCI1 = ACTUALIZACION DE LA COMPONENTE CICLICA
C         ACTVE = ACTUALIZACION DE LA MATRIZ DE ENERGIAS
C         WRPAR = ESCRITURA DE LOS PARAMETROS DEL MODELO
C        FILTER = FILTRA LOS DATOS DIARIOS
C           ING = IDENTIFICA EL GRUPO DE LOS DIAS DE LA SEMANA
C
C      NOTAS - LOS DIAS DE LA SEMANA SE ENUMERAN DE 1 AL 7 A PARTIR DE
C              LUNES.
C              LOS DIAS DE LA SEMANA SE AGRUPAN EN LA SIGUIENTE FORMA:
C                GRUPO 1 --- LUNES
C                GRUPO 2 --- MARTES,MIERCOLES,JUEVES,VIERNES
C                GRUPO 3 --- SABADO
C                GRUPO 4 --- DOMINGO
C
C*****************************************************************
      COMMON/UNO/T(24),VF(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     * BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,TD,IPR,IHORA,LDTIME,INTH,ND,TOL
```

```
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      ---------------------------------------------------
      *INICIALIZACION DE VARIABLES Y VALORES CONSTANTES
      ---------------------------------------------------
      IRECH=0
      GROWF=0.0001242
      ---------------------------------------------------
      *LECTURA DE LOS PARAMETROS
      ---------------------------------------------------
      READ(IR,5000)ALFA,M,ND,IDIA,TOL
      LL=IDIA
      WRITE(IW,5100)ALFA,M
      ---------------------------------------------------
      *INICIALIZA LOS PARAMETROS DEL FILTRO
      ---------------------------------------------------
      CALL INIFIL(LL)
      REWIND IENT
      NN=IDIA-1
      IERR=0
      ---------------------------------------------------
      *LEER LOS DATOS ACUMULADOS DEL ARCHIVO
      ---------------------------------------------------
      FOR K=1,M
       . CALL HISDAT
       .  FOR J=1,24
       .  .   VE(K,J)=Z(J)
       .  ENDFOR
          ---------------------------------------------
          *INICIALIZA LA COMPONENTE DE TENDENCIA BASICA
          ---------------------------------------------
       . CALL INITEN
      ENDFOR
      ID4=6-M+IDIA
      ID5=M-6
      ---------------------------------------------------
      *IDENTIFICA EL DIA Y DETERMINA EL GRUPO A QUE PERTENECE
      ---------------------------------------------------
      FOR I=ID5,M
       . ID6=I+ID4
       . IF(ID6.GT.7)ID6=ID6-7
       . ID7=ING(ID6)
          ---------------------------------------------
          *CALCULO DE LA COMPONENTE CICLICA SEMANAL (INICIAL)
          ---------------------------------------------
       . CALL INICIC(ID7,I)
      ENDFOR
      M2=M+1
      ---------------------------
      *LEER NUEVOS DATOS
      ---------------------------
      NDIA=M
      FOR I=M2,ND
       . CALL HISDAT
```

```
C       ----------------------------------------------------------
C       *IDENTIFICACION DEL DIA-------------------------------------
C       ----------------------------------------------------------
      . CALL SEMDIA(IDAY,MONTH,IYEAR,ID)
      . LX=ID
      . CALL FILTER(IERR)
      . IF(IERR.NE.0) CALL ANORM(GROWE,IRECH)
C       ----------------------------------------------------------
C       *ACTUALIZA LA COMPONENTE DE TENDENCIA-----------------------
C       ----------------------------------------------------------
      . CALL ACTTEN
C       ----------------------------------------------------------
C       *ACTUALIZA LA MATRIZ DE ENERGIAS----------------------------
C       ----------------------------------------------------------
      . CALL ACTVF
C       ----------------------------------------------------------
C       *ACTUALIZA LA COMPONENTE CICLICA SEMANAL--------------------
C       ----------------------------------------------------------
      . CALL ACTCT1
      ENDFOR
C     ----------------------------------------------------------
C     *ESCRIBIR LOS PARAMETROS INICIALIZADOS EN EL
C      ARCHIVO DE PARAMETROS
C     ----------------------------------------------------------
      FOR I=1,7
      . FOR J=1,24
      . . BETA(I,J)=ALFA
      . . PRONO(I,J,1)=0.0
      . . PRONO(I,J,2)=0.0
      . ENDFOR
      ENDFOR
      IPR=0
      CALL WRPAR
      WRITE(IW,5500)IRECH,ND
C     ----------------------------------------------------------
C     *FORMATOS DE LECTURA E IMPRESION----------------------------
C     ----------------------------------------------------------
5000  FORMAT(F10.4,3I5,F10.4)
5100  FORMAT(1H0,20X,'FACTOR DE PESO=',F11.5,/,21X,'LONG. VENTANA=',
5500  FORMAT(//10X,'SE RECHAZARON ',I4,' DIAS DE LOS ',I4,
     * ' DIAS CON QUE SE DISPONE PARA INICIALIZAR')
      RETURN
      END
```

```
81 10:18:51    HARRIS FORTRAN 77 OPTIMIZING COMPILER    01411                REV
NAME:  *MAIN*
     FUNCTION ING(N)
*****************************************************************************
     OBJETO                                                                 *
        DETERMINR EL GRUPO A QUE PERTENECE EL DIA DE LA SEMANA              *
        CON IDENTIFICACION "N"                                              *
                                                                            *
     IMPLEMENTADA POR MOTOKI INOUE KUBO                                     *
     FECHA DE ULTIMA REVISION  30  DE SEPTIEMBRE 1980                       *
     REVISADA POR  MOTOKI INOUE K.                                          *
                                                                            *
     ENTRADAS                                                               *
        N = IDENTIFICACION DEL DIA                                          *
                                                                            *
     SALIDAS                                                                *
        ING = IDENTIFICACION DEL GRUPO DE DIAS DE LA SEMANA                 *
              ING=1   SI    N=1                                             *
              ING=2   SI    N=2,3,4,5                                       *
              ING=3   SI    N=6                                             *
              ING=4   SI    N=7                                             *
                                                                            *
     IDENTIFICADORES LOCALES                                                *
        NINGUNO                                                             *
                                                                            *
     MODULOS EMPLEADOS                                                      *
        NINGUNO                                                             *
                                                                            *
     NOTAS                                                                  *
        NINGUNO                                                             *
                                                                            *
*****************************************************************************
      IF(.NOT.(N.GE.2.AND.N.LE.5)) THEN
      .  IF(.NOT.(N.GE.2)) THEN
            ----------------
            *GRUPO NUMERO 1
            ----------------
      .  .  ING=1
      .  ELSE
      .  .  IF(.NOT.(N.LE.6)) THEN
               ----------------
               *GRUPO NUMERO 4
               ----------------
      .  .  .  ING=4
      .  .  ELSE
               ----------------
               *GRUPO NUMERO 3
               ----------------
      .  .  .  ING=3
      .  .  ENDIF
      .  ENDIF
      ELSE
         ----------------
```

```
16 JUL 81 10:18:52    HARRIS FORTRAN 77  OPTIMIZING COMPILER    01411
MODULE NAME:   TNG
370: C         *GRUPO NUMERO 2
371: C         ----------------
372:           . TNG=2
373:           ENDIF
374:           RETURN
375:           END
```

```
R1 10:18:52   HARRIS FORTRAN 77  OPTIMIZING COMPILER     01411                REV
NAME:  *MAIN*
      SUBROUTINE ECAST
C*****************************************************************************
C     OBJETO                                                                *
C        REALIZAR EL PRONOSTICO DE CARGA HORARIA CON 24 HORAS               *
C        DE ADELANTO                                                        *
C                                                                           *
C     IMPLEMENTADA POR MOTOKI INOUE KUBO                                    *
C     FECHA DE ULTIMA REVISION  2 DE OCTUBRE DE 1980                        *
C     REVISADA POR MOTOKI INOUE K.                                          *
C                                                                           *
C     ENTRADAS                                                              *
C           ID = IDENTIFICACION DEL DIA A PRONOSTICAR                       *
C         PRONO = ARREGLO CON LOS ULTIMOS PRONOSTICOS                       *
C             S = COMPONENTE CICLICA SEMANAL                                *
C             T = COMPONENTE DE TENDENCIA                                   *
C                                                                           *
C     SALIDAS                                                               *
C         PRONO = ARREGLO CON LOS ULTIMOS PRONOSTICOS ACTUALIZADO           *
C            ZP = PRONOSTICO DE CARGA HORARIO                               *
C        ZPHEX = PRONOSTICO HEXAHORARIO                                     *
C                                                                           *
C     IDENTIFICADORES LOCALES                                               *
C             I = CONTADOR                                                  *
C        IAUXIL = VARIABLE AUXILIAR PARA EL REDONDEO DEL PRONOSTICO         *
C           IPR = CONTADOR DE PRONOSTICOS REALIZADOS                        *
C             L = IDENTIFICADOR DE GRUPO DE DIAS                            *
C             K = CONTADOR                                                  *
C             J = CONTADOR                                                  *
C            KK = VARIABLE AUXILIAR PARA CALCULOS                           *
C            LX = INDICADOR AUXILIAR DEL DIA DE LA SEMANA                   *
C         INTEG = CONTADOR                                                  *
C             R = COMPONENTE RESIDUAL                                       *
C                                                                           *
C     MODULOS EMPLEADOS                                                     *
C      RDPAR = LECTURA DE LOS PARAMETROS DEL MODELO                         *
C      SIGFE = DETERMINACION DE LA FECHA DEL SIGUIENTE DIA                  *
C     SEMDIA = DETERMINACION DEL DIA DE LA SEMANA                           *
C      ECRES = PRONOSTICO DE RESIDUOS (COMPONENTE METEOROLOGICA)            *
C      WRPAR = ESCRITURA DE LOS PARAMETROS DEL MODELO                       *
C        TUG = IDENTIFICACION DEL GRUPO DE DIAS                             *
C                                                                           *
C     NOTAS                                                                 *
C        EL ARCHIVO 20 ES UN ARCHIVO DE ALMACENAMIENTO AUXILIAR             *
C        DE USO EXCLUSIVO DE ESTE MODULO (CONTIENE PRONOSTICOS)             *
C                                                                           *
C*****************************************************************************
C
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/TERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
```

```
6 JUL 81 10:18:52    HARRIS FORTRAN 77  OPTIMIZING COMPILER    01411
MODULE NAME:  FCAST
429:      ***COMMON/PARAM/ALFA,M,TD,IPR,IHORA,LOTTME,INTH,ND,TOL
430:      *   COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
431:      *   DIMENSION R(24),ZPHEX(4)
432:      *   REWIND ISAL2
433:  C   *   ---------------------------------------------------
434:  C   *   *LEE DEL ARCHIVO DE PARAMETROS (ISAL) LOS PARAMETROS
435:  C   *    NECESARIOS PARA EL PRONOSTICO.
436:  C   *   ---------------------------------------------------
437:      *   CALL RDPAR
438:  C   *   ---------------------------------------------------
439:  C   *   *DETERMINA A QUE GRUPO PERTENECE EL DIA POR PREDECIR
440:  C   *   ---------------------------------------------------
441:      *    CALL SIGFE(IDAY,MONTH,IYEAR,NDAY,NMONTH,NYEAR)
442:      *    CALL SEMDIA(NDAY,NMONTH,NYEAR,ID)
443:      *   LX=ID
444:      *   FOR J=1,7
445:      *   .  FOR K=1,4
446:      *   .  .  ZPHEX(K)=0.0
447:      *   .  ENDFOR
448:      *   .  L=ING(LX)
449:      *   .  CALL FCRFS(R)
450:      *   .  FOR I=1,24
451:  C   *   .  ---------------------------
452:  C   *   .  *REALIZA LA PREDICCION
453:  C   *   .  ---------------------------
454:      *   .  .  ZP(I)=T(I)/FLOAT(M)+S(L,I)+0.5+R(I)
455:      *   .  .  IAUXIL=ZP(I)
456:      *   .  .  ZP(I)=FLOAT(IAUXIL)
457:      *   .  ENDFOR
458:      *   .  INTEG=0
459:      *   .  FOR K=1,4
460:      *   .  .  KK=INTEG*6
461:      *   .  .  ZPHEX(K)=ZP(KK+1)+ZP(KK+2)+ZP(KK+3)+ZP(KK+4)+ZP(KK+5)+ZP(K
462:      *   .  .  INTEG=INTEG+1
463:      *   .  ENDFOR
464:      *   .  LX=LX+1
465:      *   .  IF(LX.GT.7) LX=1
466:  C   *   ---------------------------------------------------
467:  C   *   *ESCRIBE EL EL ARCHIVO DEL PRONOSTICO (ISAL2)
468:  C   *    EL RESULTADO DE LA PREDICCION
469:  C   *   ---------------------------------------------------
470:      *   .  WRITE(ISAL2,5000)(ZP(I),I=1,12)
471:      *   .  WRITE(ISAL2,5000)(ZP(I),I=13,24)
472:      *   .  WRITE(ISAL2,5200)(ZPHEX(I),I=1,4)
473:      ***   .  IF(J.EQ.1)THEN*****************************************
474:      *   .  .  WRITE(20,5100)NDAY,NMONTH,NYEAR,(ZP(I),I=1,12)
475:      *   .  .  WRITE(20,5100)NDAY,NMONTH,NYEAR,(ZP(I),I=13,24)
476:  C   *   WRITE(20,5000)(ZPHEX(I),I=1,4)
477:      *   .  ENDIF
478:      *   ENDFOR
479:      *   IPR=IPR+1
480:      *   REWIND ISAL2
481:      *   READ(ISAL2,5000)(ZP(I),I=1,12)
```

```
E NAME:  FCAST
        READ(ISAL2,5000)(ZP(I),I=13,24)
        IF(IPR.LE.7) THEN
            FOR I=1,24
        .   .   PRONO(ID,I,1)=ZP(I)
        .   ENDFOR
        ELSE
C           ------------------------------------
C           *ACTUALIZA LA MATRIZ DE PRONOSTICOS
C           ------------------------------------
        .   FOR I=1,24
        .   .   PRONO(ID,I,1)=PRONO(ID,I,2)
        .   .   PRONO(ID,I,2)=ZP(I)
        .   ENDFOR
        ENDIF
C       ---------------------------------------------
C       *ESCRIBE EN EL ARCHIVO DE PARAMETROS, LA MATRIZ
C        ACTUALIZADA DE PRONOSTICOS
C       ---------------------------------------------
        CALL WRPAR
        RETURN
C       ------------------------------------
C       *FORMATOS DE LECTURA E IMPRESION
C       ------------------------------------
5000    FORMAT(8E10.4)
5100    FORMAT(3I2,12F6.0)
5200    FORMAT(20F6.0)
        END
```

```
18 JUL 81 10:14:57   HARRIS FORTRAN 77 OPTIMIZING COMPILER   01411
MODULE NAME:  *MAIN*
509:        SUBROUTINE UPDATE
510: C
511: C*****************************************************************
512: C      OBJETO                                                    *
513: C         ACTUALIZAR LOS PARAMETROS DE LOS MODELOS DE PRONOSTICO *
514: C                                                                *
515: C                                                                *
516: C      IMPLEMENTADA POR MOTOKI INOUE KURO                        *
517: C      FECHA DE ULTIMA REVISION  2 DE OCTUBRE DE 1980            *
518: C      REVISADA POR  MOTOKI INOUE K.                             *
519: C                                                                *
520: C      ENTRADAS                                                  *
521: C           ALFA = FACTOR DE PESO PARA ACTUALIZAR LA 'BETA'      *
522: C           BETA = MATRIZ DE FACTORES DE PESO PARA LA ACTUALIZACION *
523: C                  DE LA COMPONENTE CICLICA                      *
524: C             ID = IDENTIFICACION DEL DIA                        *
525: C            IPR = NUMERO DE PRONOSTICOS REALIZADOS              *
526: C              M = LONGITUD DE LA VENTANA DESLIZANTE             *
527: C             ND = NUMERO DE DIAS UTILIZADOS PARA LA ACTUALIZACION *
528: C                  E INICIALIZACION                              *
529: C           PROM = PROMEDIO DE ENERGIAS DIARIAS PARA EL FILTRO   *
530: C          PRONO = MATRIZ DE PRONOSTICOS ANTERIORES              *
531: C              S = COMPONENTE CICLICA SEMANAL ANTERIOR           *
532: C              T = COMPONENTE DE TENDENCIA ANTERIOR              *
533: C             VE = DATOS DE CARGA HORARIA REAL DE LOS M ULTIMOS DIAS *
534: C              Z = DATOS DE CARGA HORARIA MAS RECIENTE           *
535: C                                                                *
536: C      SALIDAS                                                   *
537: C           BETA = MATRIZ DE FACTORES DE PESO PARA LA COMPONENTE *
538: C                  CICLICA ACTUALIZADO                           *
539: C             ND = NUMERO DE DIAS UTILIZADOS PARA LA ACTUALIZACION *
540: C           PROM = PROMEDIO DE ENERGIA DIARIA ACTUALIZADO        *
541: C              S = COMPONENTE CICLICA ACTUALIZADO                *
542: C              T = COMPONENTE DE TENDENCIA ACTUALIZADO           *
543: C             VE = DATOS DE CARGA HORARIA DE LOS M ULTIMOS DIAS  *
544: C                  ACTUALIZADO                                   *
545: C                                                                *
546: C      IDENTIFICADORES LOCALES                                   *
547: C           ALF2 = FACTOR DE PESO PARA LA ACTUALIZACION DE LA    *
548: C                  COMPONENTE CICLICA                            *
549: C          GROWF = FACTOR DE CRECIMIENTO                         *
550: C              I = CONTADOR                                      *
551: C           IERR = INDICADOR DE DATOS RECHAZADOS                 *
552: C              J = CONTADOR                                      *
553: C             JJ = CONTADOR                                      *
554: C              L = IDENTIFICADOR DEL GRUPO DE DIAS               *
555: C             M1 = LIMITE DE ITERACIONES                         *
556: C           RES1 = VARIABLE AUXILIAR PARA LA ACTUALIZACION DE 'BETA' *
557: C           RES2 = VARIABLE AUXILIAR PARA LA ACTUALIZACION DE 'BETA' *
558: C             X1 = FACTOR DE ACTUALIZACION PARA 'BETA'           *
559: C             Y1 = VARIABLE AUXILIAR PARA LA ACTUALIZACION DE LA *
560: C                  COMPONENTE CICLICA                            *
561: C             Y2 = VARIABLE AUXILIAR PARA LA ACTUALIZACION DE LA *
```

```
81 10:18:58    HARRIS FORTRAN 77 OPTIMIZING COMPILER    01411
E NAME:   *MAIN*
C                 COMPONENTE CICLICA
C            Y3 = VARIABLE AUXILIAR PARA LA ACTUALIZACION DE LA
C                 COMPONENTE CICLICA
C            Y4 = VARIABLE AUXILIAR PARA LA ACTUALIZACION DE LA
C                 COMPONENTE CICLICA
C            Y5 = VARIABLE AUXILIAR PARA LA ACTUALIZACION DE LA
C                 COMPONENTE CICLICA
C
C      MODULOS EMPLEADOS
C         RDPAR = LECTURA DE LOS PARAMETROS DEL MODELO
C        CARDAT = LECTURA DE LOS DATOS REALES (MEDICIONES)
C        SEMDIA = DETERMINACION DEL DIA DE LA SEMANA
C         ANORM = SUAVIZAMIENTO DE LOS DATOS ANORMALES
C        ACTTEN = ACTUALIZACION DE LA COMPONENTE DE TENDENCIA
C         ACTVE = ACTUALIZACION DE LA MATRIZ DE ENERGIAS
C        ACTCIC = ACTUALIZACION DE LA COMPONENTE CICLICA
C         WRPAR = ESCRITURA DE LOS PARAMETROS DEL FILTRO
C        FILTER = FILTRADO DE DATOS
C           TNG = IDENTIFICACION DE GRUPOS DE DIAS
C
C      NOTAS
C         NINGUNO
C
C*****************************************************************
C
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,ID,IPR
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      GRONF=0.0001242
      IRECH=0
C     ---------------------------------------------------
C     *LEE LOS PARAMETROS DEL ARCHIVO DE PARAMETROS (ISAL)
C     ---------------------------------------------------
      CALL RDPAR
C     ----------------------------------
C     *LEE LOS DATOS DE CARGA DE UN DIA
C     ----------------------------------
      CALL CARDAT
C     ------------------------------
C     *CHECA SI LOS DATOS SON BUENOS
C     ------------------------------
      CALL SEMDIA(IDAY,MONTH,IYEAR,ID)
      M1=M-1
      CALL FILTER(IERR)
      IF(IERR.NE.0)CALL ANORM(GRONF,IRECH)
C     ---------------------------------------------
C     *ACTUALIZA LA COMPONENTE DE TENDENCIA BASICA
C     ---------------------------------------------
      CALL ACTTEN
```

```
C     ------------------------------------------------------------
C     *ACTUALIZA LA MATRIZ DE ENERGIAS
C     ------------------------------------------------------------
      CALL ACTVE
C     ------------------------------------------------------------
C     *DETERMINA A QUE GRUPO PERTENECE EL NUEVO DATO
C     ------------------------------------------------------------
      L=ING(ID)
C     ------------------------------------------------------------
C     *SI SE TIENEN LOS ULTIMOS PRONOSTICOS,REALIZA LA
C      ACTUALIZACION DE LOS FACTORES DE PESO (BETA(I,J))
C     ------------------------------------------------------------
      IF(IPR.GT.7) CALL ACBETA
C     ------------------------------------------------------------
C     *ACTUALIZA LA COMPONENTE CICLICA SEMANAL
C     ------------------------------------------------------------
      CALL ACTCIC(L)
      ND=ND+1
C     ------------------------------------------------------------
C     *ESCRIBE EN EL ARCHIVO DE PARAMETROS (ISAL) LOS
C      PARAMETROS ACTUALIZADOS
C     ------------------------------------------------------------
      CALL WRPAR
      RETURN
      END
```

```
81 10:18:59   HARRIS FORTRAN 77   OPTIMIZING COMPILER     01411                RE
E NAME:  *MAIN*
      SUBROUTINE EVALUA
C*****************************************************************************
C*****************************************************************************
C     OBJETO                                                                  *
C        CALCULAR VARIOS INDICES DE ERROR PARA EVALUAR LA CALIDAD             *
C        DEL PRONOSTICO                                                       *
C                                                                             *
C     IMPLEMENTADA POR MOTOKI INOUE KUBO                                      *
C     FECHA DE ULTIMA REVISION  2 DE OCTUBRE DE 1980                          *
C     REVISADA POR  MOTOKI INOUE K.                                           *
C                                                                             *
C     ENTRADAS                                                                *
C         EPICO = ERROR EN PICO PROMEDIO ANTERIOR                             *
C        EPROAB = ERROR ABSOLUTO PROMEDIO ANTERIOR                            *
C        EPROEN = ERROR EN ENERGIA PROMEDIO ANTERIOR                          *
C            ID = IDENTIFICACION DEL DIA POR EVALUAR                          *
C          IDAY = JUNTO CON 'MONTH' E 'IYEAR' COMPONEN LA FECHA               *
C                 DEL DIA POR EVALUAR                                         *
C           NDE = NUMERO DE DIAS EVALUADOS                                    *
C                                                                             *
C     SALIDAS                                                                 *
C         EPICO = ERROR EN PICO PROMEDIO ACTUALIZADO                          *
C        EPROAB = ERROR ABSOLUTO PROMEDIO ACTUALIZADO                         *
C        EPROEN = ERROR EN ENERGIA PROMEDIO ACTUALIZADO                       *
C           NDE = NUMERO DE DIAS EVALUADOS ACTUALIZADO                        *
C                                                                             *
C        RESULTADOS DE LA EVALUACION DEL DIA Y LOS ERRORES PROMEDIOS          *
C        EN EL ARCHIVO DE IMPRESION (IW)                                      *
C                                                                             *
C     IDENTIFICADORES LOCALES                                                 *
C           AUX = VARIABLE AUXILIAR PARA EL CALCULO DE ERRORES                *
C          AUX1 = VARIABLE AUXILIAR PARA EL CALCULO DE ERRORES                *
C          AUX2 = VARIABLE AUXILIAR PARA EL CALCULO DE ERRORES                *
C          AUX3 = VARIABLE AUXILIAR PARA EL CALCULO DE ERRORES                *
C          AUX4 = VARIABLE AUXILIAR PARA EL CALCULO DE ERRORES                *
C          AUX5 = VARIABLE AUXILIAR PARA EL CALCULO DE ERRORES                *
C           DIF = DIFERENCIA HORARIA ENTRE LA CARGA REAL Y LA                 *
C                 PRONOSTICADA EN MWH                                         *
C             E = VECTOR CON LOS ERRORES HORARIOS Y DEL DIA                   *
C            EP = ENERGIA PREDICHA                                            *
C            ER = ENERGIA REAL                                                *
C            EX = ERROR EN ENERGIA DEL DIA                                    *
C             I = CONTADOR                                                    *
C          IMES = ARREGLO CON LOS NOMBRES DE LOS MESES                        *
C            IP = HORA DEL PICO PRONOSTICADO                                  *
C            IR = HORA DEL PICO REAL                                          *
C                                                                             *
C     MODULOS EMPLEADOS                                                       *
C        NINGUNO                                                              *
C                                                                             *
C     NOTAS                                                                   *
C        NINGUNO                                                              *
C                                                                             *
```

```
C*****************************************************************
C*****************************************************************
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      COMMON/PARAM/ALFA,MAXD,IPR,IHORA,LDTIME,INTH,ND
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/DAY/IDAY,MONTH,IYEAR
      DIMENSION E(28),IMES(12),RES(24)
C     ----------------------------------
C     *ASIGNACION DE VALORES CONSTANTES
C     ----------------------------------
      DATA IMES/3HENE,3HFEB,3HMAR,3HABR,3HMAY,3HJUN,3HJUL,
     * 3HAGO,3HSEP,3HOCT,3HNOV,3HDIC/
C     ------------------------------------------------
C     *DETERMINACION DE LA HORA PICO REAL Y PREDICHA
C     ------------------------------------------------
      IR=17
      IP=17
      FOR I=18,22
      .  IF(Z(I).GT.Z(IR)) IR=I
      .  IF(ZP(I).GT.ZP(IP)) IP=I
      ENDFOR
C     ----------------------------------
C     *E(1) = DIFERENCIA DE HORAS PICO
C     ----------------------------------
      E(1)=IP-IR
C     ----------------------------------------
C     *INICIALIZACION DE VARIABLES AUXILIARES
C     ----------------------------------------
      AUX1=0.0
      AUX2=0.0
      AUX4=0.0
      AUX5=15.0
      ER=0.0
      EP=0.0
C     ----------------------------------------
C     *ESCRIBE EL ENCABEZADO DE LA EVALUACION
C     ----------------------------------------
      WRITE(IW,5700)
      FOR I=1,24
C        ----------------------------------------------
C        *DETERMINACION DE DIFERENCIA DE CARGAS HORARIAS
C        ----------------------------------------------
      .  DIF=ZP(I)-Z(I)
C        ----------------------------------------------
C        *DETERMINACION DE ERRORES HORARIOS EN POR CIENTO
C        ----------------------------------------------
      .  AUX3=(DIF/Z(I))*100.0
C        ----------------------------------------------
C        *ACUMULA EL CUADRADO DE LAS DIFERENCIAS HORARIAS
C         PARA EL CALCULO DE VARIANCIA
C        ----------------------------------------------
      .  AUX1=AUX1+DIF*DIF
```

```
C     ------------------------------------------------------------
C     *ESCRIBE LOS ERRORES HORARIOS
C     ------------------------------------------------------------
      WRITE(IW,5800)I,Z(I),ZP(I),AUX3
      RES(I)=DIF
C     ------------------------------------------------------------
C     *DETERMINA LOS ERRORES MAXIMOS Y MINIMOS
C     ------------------------------------------------------------
      IF(AUX3.GT.AUX4) AUX4=AUX3
      IF(AUX3.LT.AUX5) AUX5=AUX3
C     ------------------------------------------------------------
C     *ACUMULA LOS ERRORES HORARIOS
C     ------------------------------------------------------------
      AUX2=AUX2+ABS(AUX3)
C     ------------------------------------------------------------
C     *ACUMULA LAS CARGAS HORARIAS PARA OBTENER LA ENERGIA DIARIA
C     ------------------------------------------------------------
      EP=EP+ZP(I)
      ER=ER+Z(I)
C     ------------------------------------------------------------
C     *ALMACENA LOS ERRORES HORARIOS
C     ------------------------------------------------------------
      E(I+4)=AUX3
    ENDDOR
C     ------------------------------------------------------------
C     *GUARDA LOS RESIDUOS EN EL ARCHIVO DE RESIDUOS
C     ------------------------------------------------------------
      WRITE(IRES,6600)IDAY,MONTH,IYEAR,(RES(I),I=1,12)
      WRITE(IRES,6600)IDAY,MONTH,IYEAR,(RES(I),I=13,24)
C     ------------------------------------------------------------
C     *DETERMINACION DEL ERROR EN PICO
C     ------------------------------------------------------------
      E(2)=E(IR+4)
C     ------------------------------------------------------------
C     *DETERMINACION DEL ERROR EN ENERGIA
C     ------------------------------------------------------------
      EX=((EP-ER)/ER)*100.0
C     ------------------------------------------------------------
C     *DETERMINACION DE LA DESVIACION STANDARD
C     ------------------------------------------------------------
      E(3)=SQRT(AUX1/24.0)
C     ------------------------------------------------------------
C     *DETERMINACION DEL ERROR PROMEDIO DIARIO
C     ------------------------------------------------------------
      E(4)=AUX2/24.0
C     ------------------------------------------------------------
C     *DETERMINACION DEL RANGO MAXIMO DE ERROR
C     ------------------------------------------------------------
      RANGE=AUX4+ABS(AUX5)
C     ------------------------------------------------------------
C     *ESCRIBE ENCABEZADOS CORRESPONDIENTES A LOS DIAS
C     ------------------------------------------------------------
      IF(ID.EQ.7)WRITE(IW,5000) IDAY,IMES(MONTH),IYEAR
```

```
16 JUL 81 10:19:03   HARRIS FORTRAN 77 OPTIMIZING COMPILER
MODULE NAME:  EVALUA
799:          IF(ID.EQ.1)WRITE(IW,5100) IDAY,IMES(MONTH),IYEAR
800:          IF(ID.EQ.2)WRITE(IW,5200) IDAY,IMES(MONTH),IYEAR
801:          IF(ID.EQ.3)WRITE(IW,5300) IDAY,IMES(MONTH),IYEAR
802:          IF(ID.EQ.4)WRITE(IW,5400) IDAY,IMES(MONTH),IYEAR
803:          IF(ID.EQ.5)WRITE(IW,5500) IDAY,IMES(MONTH),IYEAR
804:          IF(ID.EQ.6)WRITE(IW,5600) IDAY,IMES(MONTH),IYEAR
805: C        ---------------------------------------------
806: C        *ESCRIBE LOS INDICES DE EVALUACION
807: C        ---------------------------------------------
808:          WRITE(IW,5900)E(1),E(2),E(3),E(4),EX
809: C        ---------------------------------------------
810: C        *ESCRIBE EL RANGO DE ERROR
811: C        ---------------------------------------------
812:          WRITE(IW,6400)RANGE
813:          REWIND IEVA
814: C        ---------------------------------------------
815: C        *LEER ARCHIVO DE ERRORES ACUMULADOS
816: C        ---------------------------------------------
817:          READ(IEVA,6000) NDE
818:          IF(NDE.EQ.0) THEN
819: C           ---------------------------------------------
820: C           *SI ES EL PRIMER DIA DE EVALUACIONES,INICIALIZA
821: C            LOS ERRORES PROMEDIOS
822: C           ---------------------------------------------
823:           .  NDE=1
824:           .  EPROAB=E(4)
825:           .  EPROST=E(3)
826:           .  EPROEN=ABS(EX)
827:           .  EPICO=ABS(E(2))
828:           ELSE
829: C           ---------------------------------------------
830: C           *SI NO ES EL PRIMER DIA DE EVALUACIONES LEE LOS
831: C            ERRORES PROMEDIOS ANTERIORES Y ACTUALIZA
832: C           ---------------------------------------------
833:           .  READ(IEVA,6100) EPROAB,EPROST,EPROEN,EPICO
834:           .  NDE=NDE+1
835: C           ---------------------------------------------
836: C           *ACTUALIZA LOS ERRORES PROMEDIO
837: C           ---------------------------------------------
838:           .  AUX=FLOAT(NDE-1)/FLOAT(NDE)
839:           .  EPROAB=AUX*EPROAB+E(4)/FLOAT(NDE)
840:           .  EPROST=SQRT(AUX*EPROST*EPROST+(E(3)*E(3))/FLOAT(NDE))
841:           .  EPROEN=AUX*EPROEN+ABS(EX)/FLOAT(NDE)
842:           .  EPICO=AUX*EPICO+ABS(E(2))/FLOAT(NDE)
843:           END IF
844:           REWIND IEVA
845: C        ---------------------------------------------
846: C        *ESCRIBE EN EL ARCHIVO DE ERRORES,LOS ERRORES PROMEDIO
847: C         ACTUALIZADOS
848: C        ---------------------------------------------
849:          WRITE(IEVA,6000)NDE
850:          WRITE(IEVA,6100)EPROAB,EPROST,EPROEN,EPICO
851: C        ---------------------------------------------
```

```
81 10:19:05   HARRIS FORTRAN 77  OPTIMIZING COMPILER   01411
NAME:  EVALUA
      *IMPRIME LOS ERRORES PROMEDIO EN PAREI
*****************************************************************
      WRITE(IW,6200)EPROAB,EPROST,EPROEN,EPICO
      WRITE(IW,6300)ER,EP
      RETURN
      ----------------------------------
      *FORMATOS DE LECTURA E IMPRESION
      ----------------------------------
5000  FORMAT(//15X,' ERRORES DEL DIA DOMINGO',I4,' DE ',A3,' DEL ',I3//)
5100  FORMAT(//15X,' ERRORES DEL DIA LUNES',I4,' DE ',A3,' DEL ',I3//)
5200  FORMAT(//15X,' ERRORES DEL DIA MARTES',I4,' DE ',A3,' DEL ',I3//)
5300  FORMAT(//15X,' ERRORES DEL DIA MIERCOLES',I4,' DE ',A3,
     *' DEL ',I3//)
5400  FORMAT(//15X,' ERRORES DEL DIA JUEVES',I4,' DE ',A3,' DEL ',I3//)
5500  FORMAT(//15X,' ERRORES DEL DIA VIERNES',I4,' DE ',A3,' DEL ',I3//)
5600  FORMAT(//15X,' ERRORES DEL DIA SABADO',I4,' DE ',A3,' DEL ',I3//)
5700  FORMAT(1H1,10X,'HORA',5X,'CARGA REAL',5X,'PRONOSTICO',5X,
     * 'ERROR')
5800  FORMAT(8X,I4,9X,F8.3,5X,F8.3,5X,F8.3,' %')
5900  FORMAT(15X,'DIFERENCIA DE HORAS PICO-----',F10.3,/,15X,
     *  'ERROR EN PICO----------------',F10.3,'%',/,15X,
     *  'DESVIACION STANDARD----------',F10.3,'MWH',
     */15X,'ERROR PROMEDIO DEL DIA-------',F10.3,' %',
     */15X,'ERROR EN ENERGIA DIARIA------',F10.3,' %')
6000  FORMAT(I5)
6100  FORMAT(5F10.5)
6200  FORMAT(15X,'ERROR ABSOLUTO PROMEDIO------',F10.3,' %',/15X,
     1 'DESVIACION STANDARD PROMEDIO ',F10.3,' MWH',/15X,
     2 'ERROR DE ENERGIA PROMEDIO----',F10.3,' %',/15X,
     3 'ERROR EN PICO PROMEDIO-------',F10.3,/)
6300  FORMAT(15X,'ENERGIA REAL-----------------',F10.3,' MWH',/15X,
     * 'ENERGIA PREDICHA-------------',F10.3,' MWH')
6400  FORMAT(15X,'RANGO MAXIMO DE ERROR--------',F10.3,'%',/)
6500  FORMAT(F15.5)
6600  FORMAT(3I2,12F6.0)
      END
*****************************************************************
```

```
6 JUL 81 10:19:08   HARRIS FORTRAN 77  OPTIMIZING COMPILER
MODULE NAME:  *MAIN*
888:        SUBROUTINE FILTER(JERR)
889: C*****************************************************************
890: C      OBJETO
891: C         DETERMINAR SI LOS DATOS CORRESPONDIENTES A UN DIA SON ACEP-
892: C         TABLES PARA REALIZAR LA ACTUALIZACION DE LOS PARAMETROS
893: C
894: C      IMPLEMENTADA POR  MOTOKI INOUE KURO
895: C      FECHA DE ULTIMA REVISION  30 DE SEPTIEMBRE DE 1980
896: C      REVISADA POR MOTOKI INOUE K.
897: C
898: C      ENTRADAS
899: C         ID = IDENTIFICACION DEL DIA A FILTRAR
900: C         PROM = PROMEDIOS DE ENERGIAS DIARIAS
901: C
902: C      SALIDAS
903: C         JERR = INDICADOR DE ACEPTACION DE LOS DATOS
904: C                JERR = 0      DATOS ACEPTADOS
905: C                JERR = 1      DATOS RECHAZADOS
906: C         PROM = PROMEDIOS DE ENERGIAS DIARIAS ACTUALIZADAS
907: C
908: C      IDENTIFICADORES LOCALES
909: C         CLIM = LIMITE DE TOLERANCIA PARA LA ENERGIA DIARIA
910: C         DESV = DESVIACION ESTANDAR PROMEDIO DE LAS ULTIMAS 4 SEMANAS
911: C         ENER = ENERGIA DEL DIA A FILTRAR
912: C            I = CONTADOR
913: C         IMES = ARREGLO CON LOS NOMBRES DE LOS MESES (ALFANUMERICO)
914: C         ISEM = ARREGLO CON LOS NOMBRES DE LOS DIAS DE LA SEMANA
915: C                (ALFANUMERICO)
916: C          PRO = ENERGIA DIARIA PROMEDIO DE LAS ULTIMAS 4 SEMANAS
917: C
918: C      MODULOS UTILIZADOS
919: C         NINGUNO
920: C
921: C      NOTAS
922: C         NINGUNO
923: C
924: C*****************************************************************
925:        COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
926:       *  BETA(7,24)
927:        COMMON/CARGA2/Z(24)
928:        COMMON/DAY/IDAY,MONTH,IYEAR
929:        COMMON/PARAM/ALFA,M,ID,IPR,IHORA,LDTIME,INTH,HD,TOL
930:        COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
931:        DIMENSION IMES(12)
932: C      ----------------------------------
933: C      *ASIGNACION DE VALORES CONSTANTES
934: C      ----------------------------------
935:        DATA IMES/3HENE,3HFEB,3HMAR,3HABR,3HMAY,3HJUN,3HJUL,
936:       * 3HAGO,3HSEP,3HOCT,3HNOV,3HDIC/
937: C      ----------------------------
938: C      *INICIALIZACION DE VARIABLES
939: C      ----------------------------
940:        JERR=0
```

```
      ENER=0.0
      PRO=0.0
      DESV=0.0
C     ----------------------------
C     *OBTENER LA ENERGIA DEL DIA
C     ----------------------------
      FOR I=1,24
       . ENER=ENER+Z(I)
      ENDFOR
C     ----------------------------------------
C     *OBTENER EL PROMEDIO DEL DIA DE LA SEMANA
C      CORRESPONDIENTE
C     ----------------------------------------
      FOR I=1,4
       . PRO=PRO+PROM(ID,I)
       . DESV=DESV+PROM(ID,I)*PROM(ID,I)
      ENDFOR
      DESV=(4.0*DESV-PRO*PRO)/16.0
      PRO=PRO/4.0
      DESV=SQRT(DESV)
      CLIM=PRO-(DESV*TOL)
C     ----------------------------
C     *CHECA SI LOS DATOS SON BUENOS
C     ----------------------------
      IF(ENER.LT.CLIM)JERR=1
      IF(JERR.NE.0)WRITE(IW,5000) IDAY,IMES(MONTH),IYEAR
C     ----------------------------------
C     *ACTUALIZA LA MATRIZ DE PROMEDIOS
C     ----------------------------------
      PROM(ID,1)=PROM(ID,2)
      PROM(ID,2)=PROM(ID,3)
      PROM(ID,3)=PROM(ID,4)
      IF(JERR.EQ.1)ENER=CLIM
      PROM(ID,4)=ENER
      RETURN
C     ----------------------
C     *FORMATO DE IMPRESION
C     ----------------------
5000  FORMAT(///,10X,'**** SE RECHAZARON LOS DATOS DEL DIA',
     *T4,'DE',A3,' DEL',I4,' ****')
      END
```

```
16 JUL 81 10:19:11    HARRIS FORTRAN 77  OPTIMIZING COMPILER
MODULE NAME:  *MAIN*
982:          SUBROUTINE SEMDIA(IDAY,MONTH,IYEAR,ID)
983: C*****************************************************************
984: C        SUBRUTINA PARA DETERMINAR CUALQUIER DIA DE LA SEMANA
985: C        DE LA FECHA DADA POR IDAY,MONTH,IYEAR
986: C        ESTA FECHA PUEDE VARIAR DE 1901 AL 1999
987: C
988: C            IMPLEMENTADA POR MOTOKI INOUE K.
989: C            FECHA DE ULTIMA REVISION  28 DE JULIO 1980
990: C            REVISADA POR M.I.K.
991: C
992: C                ENTRADAS
993: C                      FECHA DEL DIA
994: C                         IDAY    =    DIA (2 CIFRAS)
995: C                         MONTH   =    MES (2 CIFRAS)
996: C                         IYEAR   =    ANO (2 CIFRAS)
997: C
998: C                SALIDAS
999: C                        ID = IDENTIFICACION DEL DIA DE LA SEMANA
1000: C                          ID = 1,2,......,7
1001: C                          ID = LUN,MAR,.......,DOM
1002: C
1003: C                IDENTIFICADORES LOCALES
1004: C                     VARIABLES AUXILIARES PARA LOS CALCULOS
1005: C                        IAUX    =   AUXILIAR PARA LA DETERMINACION-
1006: C                                    DE LOS ANOS BISIESTOS
1007: C                NDMES    =    ARREGLO QUE CONTIENE LOS NUMEROS DE DIAS
1008: C                NDAY     =    NUMERO DE DIAS TRANSCURRIDOS
1009: C                NBIS     =    NUMERO DE ANOS BISIESTOS
1010: C                NMONTH   =    NUMERO DE MESES TRANSCURRIDOS EN EL-
1011: C                              ANO DE LA FECHA
1012: C                              DE LOS MESES DEL ANO.
1013: C
1014: C                MODULOS EMPLEADOS
1015: C                    NINGUNO
1016: C
1017: C        NOTAS
1018: C            PARA DETERMINAR LOS DIAS DE LA SEMANA, CON FECHAS MAS ALLA
1019: C            DE 1999, LA VARIABLE "IYEAR", TENDRA QUE DARSE EN 4 CIFRAS
1020: C            Y DESPUES DE LA PRIMERA INSTRUCCION EJECUTABLE, INSERTAR
1021: C            "NYEAR=IYEAR-1900", Y TODOS LOS CALCULOS QUE INVOLUCREN A LA
1022: C            VARIABLE "IYEAR" REALIZARLO CON "NYEAR".
1023: C
1024: C*****************************************************************
1025:          DIMENSION NDMES(12)
1026: C        ----------------------------------
1027: C        *ASIGNACION DE VALORES CONSTANTES
1028: C        ----------------------------------
1029:          DATA NDMES/31,28,31,30,31,30,31,31,30,31,30,31/
1030:          NDMES(2)=28
1031: C        ------------------------------------------------
1032: C        *CALCULO DE LAS VARIABLES AUXILIARES PARA DETERMINAR
1033: C         SI EL ANO ES BISIESTO
1034: C        ------------------------------------------------
```

```
NAME:  SEMDTA
      IAUX=MOD(IYEAR,4)
*****IF(IAUX.EQ.0)NDMES(2)=29*****************************************
      ----------------------------------------
      *LOGICA PARA LA DETERMINACION DEL DIA
      ----------------------------------------
      NBIS=IFIX((FLOAT(IYEAR)/4.0)+0.75)
      NDAY=(IYEAR-1)*365+NBIS
      NMONTH=MONTH-1
      IF(NMONTH.NE.0) THEN
      .  FOR I=1,NMONTH
      .  .  NDAY=NDAY+NDMES(I)
      .  ENDFOR
      ENDIF
      NDAY=NDAY+IDAY
      ID=MOD(NDAY,7)
      IF(ID.EQ.0)ID=7
      RETURN
      END
```

```
      SUBROUTINE SIGFE(IDAY,MONTH,IYEAR,NDAY,NMONTH,NYEAR)
C*****************************************************************
C     OBJETO
C     DETERMINAR LA FECHA DEL SIGUIENTE DIA
C
C     IMPLEMENTADA POR  MOTOKI INOUE KUNO
C     FECHA DE ULTIMA REVISION 2 DE DICIEMBRE DE 1980
C     REVISADA POR MIK
C
C     ENTRADAS
C        FECHA DEL DIA
C           IDAY  =  DIA
C           MONTH =  MES
C           IYEAR =  ANO
C
C     SALIDAS
C        FECHA DEL SIGUIENTE DIA
C           NDAY  =  DIA
C           NMONTH = MES
C           NYEAR =  ANO
C
C     IDENTIFICADORES LOCALES
C        IAUX     =  VARIABLE AUXILIAR PARA DETERMINAR EL NUMERO
C                    DE AÑOS BISIESTOS
C        NDMES    =   ARREGLO CON LOS NUMEROS DE DIAS DE LOS MESES
C
C     MODULOS EMPLEADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************
      DIMENSION NDMES(12)
C     ---------------------------------
C     *ASIGNACION DE VALORES CONSTANTES
C     ---------------------------------
      DATA NDMES/31,28,31,30,31,30,31,31,30,31,30,31/
      NDMES(2)=28
C     ---------------------------------
C     *DETERMINAR SI EL AÑO ES BISIESTO
C     ---------------------------------
      IAUX=MOD(IYEAR,4)
      IF(IAUX.EQ.0) NDMES(2)=29
C     ------------------------------------
C     *DETERMINACION DE LA SIGUIENTE FECHA
C     ------------------------------------
      NDAY=IDAY+1
      NMONTH=MONTH
      NYEAR=IYEAR
      IF(NDAY.GT.NDMES(MONTH)) NDAY=1
      IF(NDAY.EQ.1) NMONTH=MONTH+1
      IF(NMONTH.GT.12) THEN
```

```
NAME:  SIGFF
     .  NYEAR=IYEAR+1
     .  NMONTH=1
*****************************************************************
      ENDIF
      RETURN
      END
```

```
16 JUL 81 10:19:14    HARRIS FORTRAN 77 OPTIMIZING COMPILER    01411
 MODULE NAME:  *MAIN*
1111:          SUBROUTINE HISDAT
1112: C*****************************************************************
1113: C        OBJETO
1114: C        LEER EL ARCHIVO DE DATOS HISTORICOS
1115: C
1116: C        IMPLEMENTADA POR MOTOKI INOUE KUBO
1117: C        FECHA DE ULTIMA REVISION  3 DE DICIEMBRE DE 1980
1118: C        REVISADA POR MTK
1119: C
1120: C        ENTRADAS
1121: C            IDAY = DIA
1122: C            MONTH = MES
1123: C            IYEAR = ANO
1124: C            Z = DATOS REALES DE CARGA
1125: C
1126: C        SALIDAS
1127: C            IDAY = DIA
1128: C            MONTH = MES
1129: C            IYEAR = ANO
1130: C            Z = DATOS REALES DE CARGA
1131: C
1132: C        IDENTIFICADORES LOCALES
1133: C            I = CONTADOR
1134: C
1135: C        MODULOS EMPLEADOS
1136: C            NINGUNO
1137: C
1138: C        NOTAS
1139: C            NINGUNO
1140: C
1141: C*****************************************************************
1142:          COMMON/UNO/T(24),VF(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
1143:         1  BETA(7,24)
1144:          COMMON/TRES/IERR
1145:          COMMON/DAY/IDAY,MONTH,IYEAR
1146:          COMMON/CARGA1/ZP(24)
1147:          COMMON/CARGA2/Z(24)
1148:          COMMON/PARAM/ALFA,M,ID,IPR,IHORA,LDTIME,INTH,ND,TOL
1149:          COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
1150: C        ------------------------------
1151: C        *LECTURA DE LOS DATOS DE CARGA
1152: C        ------------------------------
1153:          READ(IENT,5000)IDAY,MONTH,IYEAR,(Z(J),J=1,12)
1154:          READ(IENT,5000)IDAY,MONTH,IYEAR,(Z(J),J=13,24)
1155: C        -------------------
1156: C        *FORMATO DE LECTURA
1157: C        -------------------
1158: 5000     FORMAT(3I2,12F6.0)
1159:          RETURN
1160:          END
```

```
81 10:19:16   HARRIS FORTRAN 1770 OPTIMIZING COMPILER   V01.41.1
E NAME:  *MAIN*
      SUBROUTINE INTFIL(LL)
C*********************************************************************
C     OBJETO                                                          *
C     INICIALIZAR PARAMETROS DEL FILTRO                               *
C                                                                     *
C     IMPLEMENTADA POR MOTOKI INOUE KUBO                              *
C     FECHA DE ULTIMA REVISION  3 DE DICIEMBRE DE 1980                *
C     REVISADA POR MIK                                                *
C                                                                     *
C     ENTRADAS                                                        *
C        M     =  LONGITUD DE LA VENTANA DESLIZANTE                   *
C        Z     =  CARGA HISTORICA                                     *
C                                                                     *
C     SALIDAS                                                         *
C        PROM =  PROMEDIO DE ENERGIAS PARA CADA DIA DE LA SEMANA      *
C                                                                     *
C     IDENTIFICADORES LOCALES                                         *
C        I     =  CONTADOR                                            *
C        J     =  CONTADOR                                            *
C        K     =  CONTADOR                                            *
C        LL    =  INDICADOR DEL DIA DE LA SEMANA                      *
C        PRO   =  VARIABLE AUXILIAR PARA CALCULAR PROMEDIOS DE ENERGIA *
C                                                                     *
C     MODULOS UTILIZADOS                                              *
C        HISDAT = LECTURA DE DATOS DE CARGA HISTORICA                 *
C                                                                     *
C     NOTAS                                                           *
C        NINGUNO                                                      *
C                                                                     *
C*********************************************************************
      COMMON/UNO/T(24),VF(24,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,ID,IPR,IHORA,LDTIME,IDTH,ND,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      REWIND IENT
C     ------------------------------------
C     *INICIALIZA LOS PARAMETROS DEL FILTRO
C     ------------------------------------
      FOR I=1,4
      .  FOR I=1,7
      .  .  CALL HISDAT
      .  .  FOR K=1,24
      .  .  .  PRO=PRO+Z(K)
      .  .  ENDFOR
      .  .  PROM(LL,J)=PRO
      .  .  LL=LL+1
      .  .  PRO=0.0
      .  .  IF(LL.GT.7)LL=1
      .  ENDFOR
```

```
16 JUL 81 10:19:17   HARRIS FORTRAN 77 OPTIMIZING COMPILER
MODULE NAME:  INTFTI.
1214:          ENDFOR
1215:          RETURN
1216:          END
```

```
81 10:19:18    HARRIS FORTRAN 77 OPTIMIZING COMPILER    20141101 IN THE ARE
E NAME:  *MAIN*
      SUBROUTINE INITEN
C*****************************************************************************
C     OBJETO
C     INICIALIZAR LA COMPONENTE DE TENDENCIA CON LOS M PRIMEROS DIAS
C
C     IMPLEMENTADA POR MOTOKI INOUE KUBO
C     FECHA DE ULTIMA REVISION: 3 DE DICIEMBRE DE 1980
C     REVISADA POR MIK
C
C     ENTRADAS
C        Z      =  CARGA HORARIA HISTORICA
C
C     SALIDA
C        T      =  COMPONENTE DE TENDENCIA INICIALIZADA
C
C     IDENTIFICADORES LOCALES
C        I      =  CONTADOR
C
C     MODULOS UTILIZADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************************
      COMMON/UNO/T(24),VF(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,TO,IPR,THORA,LDTIME,INITH,ND,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
C     ---------------------------------------------------
C     *INICIALIZA LA COMPONENTE DE TENDENCIA BASICA
C     ---------------------------------------------------
      FOR I=1,24
      .  T(I)=T(I)+Z(I)
      ENDFOR
      RETURN
      END
```

```
16 JUL 81 10:19:18   HARRIS FORTRAN 77 OPTIMIZING COMPILER   R01411 01 18
 MODULE NAME:  *MAIN*
1258:        SUBROUTINE INTCIC(IDENT,T)
1259: C*****************************************************************
1260: C      OBJETO
1261: C      INICIALIZAR LA COMPONENTE CICLICA SEMANAL
1262: C
1263: C      IMPLEMENTADA POR MOTOKI INOUE KUBO
1264: C      FECHA DE ULTIMA REVISION 3 DE DICIEMBRE DE 1980
1265: C      REVISADA POR MIK
1266: C
1267: C      ENTRADAS
1268: C         IDENT =  IDENTIFICACION DEL DIA
1269: C         M     =  LONGITUD DE VENTANA
1270: C         T     =  COMPONENTE DE TENDENCIA INICIALIZADA
1271: C         VE    =  MATRIZ DE ENERGIA DE LOS ULTIMOS M DIAS
1272: C
1273: C      SALIDA
1274: C         S     =  COMPONENTE CICLICA SEMANAL INICIALIZADA
1275: C
1276: C      IDENTIFICADORES LOCALES
1277: C         I     =  CONTADOR
1278: C         J     =  CONTADOR
1279: C
1280: C      MODULOS UTILIZADOS
1281: C         NINGUNO
1282: C*****************************************************************
1283: C      NOTAS
1284: C         NINGUNO
1285: C
1286: C*****************************************************************
1287:        COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PROND(7,24,2),
1288:       *  BETA(7,24)
1289:        COMMON/TRES/IERR
1290:        COMMON/DAY/IDAY,MONTH,IYEAR
1291:        COMMON/CARGA1/ZP(24)
1292:        COMMON/CARGA2/Z(24)
1293:        COMMON/PARAM/ALFA,M,TD,IPR,IHORA,LDTIME,INTH,ND,TOL
1294:        COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
1295: C      ---------------------------------------------------
1296: C      *CALCULO DE LA COMPONENTE CICLICA SEMANAL (INICIAL)
1297: C      ---------------------------------------------------
1298:        DO J=1,24
1299:        .  S(IDENT,J)=VE(I,J)-T(J)/FLOAT(M)
1300:        ENDDO
1301:        RETURN
1302:        END
```

```
E NAME:  *MATN*
      SUBROUTINE ANORM(GROWF,IRECH)
C*****************************************************************
C     OBJETO
C     SUAVIZAR LOS DATOS ANORMALES
C
C     IMPLEMENTADA POR MOTOKI INOUE KUBO
C     FECHA DE ULTIMA REVISION: 3 DE DICIEMBRE DE 1980
C     REVISADA POR MIK
C
C     ENTRADAS
C        GROWF =  FACTOR DE CRECIMIENTO  SEMANAL
C        VE    =  MATRIZ DE ENERGIAS
C
C     SALIDA
C        Z     =  DATOS SUAVIZADOS
C
C     IDENTIFICADORES LOCALES
C        I     =  CONTADOR
C        MX    =  INDICADOR DE LOS DATOS DE HACE UNA SEMANA
C
C     MODULOS EMPLEADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,TD,IPR,THORA,LDTIME,INTH,ND,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      IRECH=IRECH+1
      MX=M-6
C     ---------------------------------------------------
C     *AFECTA POR EL FACTOR DE CRECIMIENTO LOS DATOS DE
C      HACE UNA SEMANA
C     ---------------------------------------------------
      FOR J=1,24
      .  Z(J)=VE(MX,J)*(GROWF+1.0)
      ENDFOR
      RETURN
      END
```

```
16 JUL 81 10:19:21    HARRIS FORTRAN 77 OPTIMIZING COMPILER    04.11
MODULE NAME:  *MAIN*
1349:          SUBROUTINE ACTTEN
1350: C*****************************************************************
1351: C        OBJETO
1352: C        ACTUALIZAR LA COMPONENTE DE TENDENCIA DATOS NUEVOS
1353: C
1354: C        IMPLEMENTADA POR MOTOKI INOUE KUBO
1355: C        FECHA DE ULTIMA REVISION: 3 DE DICIEMBRE DE 1980
1356: C        REVISADA POR MTK
1357: C
1358: C        ENTRADAS
1359: C           T      =  COMPONENTE DE TENDENCIA ANTERIOR
1360: C           VE     =  MATRIZ DE ENERGIAS DE LOS ULTIMOS DIAS
1361: C           Z      =  DATOS DE CARGA HORARIA NUEVA
1362: C
1363: C        SALIDA
1364: C           T      =  COMPONENTE DE TENDENCIA ACTUALIZADA
1365: C
1366: C        IDENTIFICADORES LOCALES
1367: C           J      =  CONTADOR
1368: C
1369: C        MODULOS EMPLEADOS
1370: C           NINGUNO
1371: C
1372: C        NOTAS
1373: C           NINGUNO
1374: C
1375: C*****************************************************************
1376:          COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
1377:         *  BETA(7,24)
1378:          COMMON/TRES/TERR
1379:          COMMON/DAY/IDAY,MONTH,IYEAR
1380:          COMMON/CARGA1/ZP(24)
1381:          COMMON/CARGA2/Z(24)
1382:          COMMON/PARAM/ALFA,M,TD,IPR,THORA,LDTINE,INTH,ND,TOL
1383:          COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
1384: C        ----------------------------------------
1385: C        *ACTUALIZA LA COMPONENTE DE TENDENCIA
1386: C        ----------------------------------------
1387:          FOR J=1,24
1388:          .  T(J)=T(J)-VE(1,J)+Z(J)
1389:          ENDFOR
1390:          RETURN
1391:          END
```

```
E NAME:  *MAIN*
      SUBROUTINE ACTVE
C*****************************************************************
C     OBJETO
C     ACTUALIZAR LA MATRIZ DE ENERGIAS CON DATOS DE CARGA HORARIA
C     NUEVA
C
C     IMPLEMENTADA POR MOTOKI INOUE KUBO
C     FECHA DE ULTIMA REVISION 3 DE DICIEMBRE DE 1980
C     REVISADA POR MTK
C
C     ENTRADAS
C        M      =  LONGITUD DE VENTANA
C        VE     =  MATRIZ DE ENERGIAS ANTERIOR
C        Z      =  DATOS DE CARGA HORARIA NUEVA
C
C     SALIDA
C        VE     =  MATRIZ DE ENERGIAS ACTUALIZADA
C
C     IDENTIFICADORES LOCALES
C        I      =  CONTADOR
C        K      =  CONTADOR
C        MX     =  APUNTADOR DEL RENGLON DE ACTUALIZACION
C
C     MODULOS UTILIZADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,ID,IPR,IHORA,LDTIME,INIH,ND,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      MX=M-1
C     ---------------------------------
C     *ACTUALIZA LA MATRIZ DE ENERGIAS
C     ---------------------------------
      FOR I=1,MX
      .  FOR K=1,24
      .  .  VE(I,K)=VE(I+1,K)
      .  ENDFOR
      ENDFOR
      FOR I=1,24
      .  VE(M,I)=Z(I)
      ENDFOR
      RETURN
      END
```

```
1444:        SUBROUTINE ACTCI1
1445: C*****************************************************************
1446: C      OBJETO
1447: C      ACTUALIZAR LA COMPONENTE CICLICA SEMANAL (UTILIZADO EN EL
1448: C      MODULO INICIO)
1449: C
1450: C      IMPLEMENTADA POR MOTOKI INOUE KUBO
1451: C      FECHA DE ULTIMA REVISION  3 DE DICIEMBRE DE 1980
1452: C      REVISADA POR MIK
1453: C
1454: C      ENTRADAS
1455: C         ALFA   =  FACTOR DE ACTUALIZACION
1456: C         ID     =  IDENTIFICACION DEL DIA
1457: C         J      =  CONTADOR
1458: C         M      =  LONGITUD DE VENTANA
1459: C         S      =  COMPONENTE CICLICA SEMANAL ANTERIOR
1460: C
1461: C      SALIDA
1462: C         S      =  COMPONENTE CICLICA SEMANAL ACTUALIZADO
1463: C
1464: C      IDENTIFICADORES LOCALES
1465: C         ALF2   =  FACTOR DE ACTUALIZACION (1-ALFA)
1466: C         J      =  CONTADOR
1467: C         TEMP   =  VARIABLE AUXILIAR CONTENIENDO LA PARTE NUEVA DE
1468: C                   LA COMPONENTE CICLICA SEMANAL
1469: C
1470: C      MODULOS EMPLEADOS
1471: C         ING = IDENTIFICADOR DE GRUPO DE DIAS
1472: C
1473: C      NOTAS
1474: C         NINGUNO
1475: C
1476: C*****************************************************************
1477:        COMMON/UNO/T(24),VF(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
1478:       *  BETA(7,24)
1479:        COMMON/TRES/TERR
1480:        COMMON/DAY/IDAY,MONTH,IYEAR
1481:        COMMON/CARGA1/ZP(24)
1482:        COMMON/CARGA2/Z(24)
1483:        COMMON/PARAM/ALFA,M,ID,IPR,THORA,LDTIME,INTH,ND,TOL
1484:        COMMON/FILES/TR,TW,TENT,TSAL,TSAL2,TEVA,TENT2,TRES
1485:        LX=ING(ID)
1486:        ALF2=1.0-ALFA
1487: C      ----------------------------------------
1488: C      *ACTUALIZA LA COMPONENTE CICLICA SEMANAL
1489: C      ----------------------------------------
1490:        FOR J=1,24
1491:        .  TEMP=Z(J)-T(J)/FLOAT(M)
1492:        .  S(LX,J)=ALFA*TEMP+ALF2*S(LX,J)
1493:        ENDFOR
1494:        RETURN
1495:        END
```

```
E NAME:  *MAIN*
      SUBROUTINE WRPAR
C*****************************************************************
C     OBJETO
C     ESCRIBIR LOS PARAMETROS EN SU ARCHIVO CORRESPONDIENTE
C
C     IMPLEMENTADA POR MOTOKI INOUE KUBO
C     FECHA DE ULTIMA REVISION  3 DE DICIEMBRE DE 1980
C     REVISADA POR MIK
C
C     ENTRADAS
C        T = COMPONENTE DE TENDENCIA
C        S = COMPONENTE CICLICA
C        VE = MATRIZ DE ENERGIAS
C        Z = CARGAS REALES
C        PROM = PROMEDIO DE ENERGIAS
C        BETA = FACTORES DE PESO
C        PRONO = MATRIZ DE PRONOSTICOS REALIZADOS
C        ND = NUMERO DE DIAS EMPLEADOS
C        IPR = CONTADOR DE PRONOSTICOS
C        M = LONGITUD DE VENTANA
C        ALFA = FACTOR DE ACTUALIZACION
C
C     SALIDAS
C        T = COMPONENTE DE TENDENCIA
C        S = COMPONENTE CICLICA
C        VE = MATRIZ DE ENERGIAS
C        Z = CARGAS REALES
C        PROM = PROMEDIO DE ENERGIAS
C        BETA = FACTORES DE PESO
C        PRONO = MATRIZ DE PRONOSTICOS REALIZADOS
C        ND = NUMERO DE DIAS EMPLEADOS
C        IPR = CONTADOR DE PRONOSTICOS
C        M = LONGITUD DE VENTANA
C        ALFA = FACTOR DE ACTUALIZACION
C
C     IDENTIFICADORES LOCALES
C        I = CONTADOR
C        J = CONTADOR
C        K = CONTADOR
C
C     MODULOS EMPLEADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     * BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
```

```
      COMMON/PARAM/ALFA,M,ID,IPR,THORA,LDTIME,INTH,NDT,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      REWIND ISAL
C     ---------------------------
C     *ESCRITURA DE LOS PARAMETROS
C     ---------------------------
      WRITE(ISAL,5000)(T(J),J=1,24)
      WRITE(ISAL,5000)((S(I,J),J=1,24),I=1,4)
      WRITE(ISAL,5000)((VE(I,J),J=1,24),I=1,M)
      WRITE(ISAL,5000)(Z(J),J=1,24)
      WRITE(ISAL,5000)((PROM(I,J),J=1,4),I=1,7)
      WRITE(ISAL,5000)((BETA(I,J),J=1,24),I=1,7)
      FOR I=1,2
      .  WRITE(ISAL,5000)((PRONO(K,J,I),J=1,24),K=1,7)
      ENDFOR
      WRITE(ISAL,5100)NDTA,IPR,M,ALFA
      RETURN
C     ----------------------
C     *FORMATO DE ESCRITURA
C     ----------------------
5000  FORMAT(8E10.3)
5100  FORMAT(3I11,F6.3)
      END
```

```
81 10:19:29    HARRIS FORTRAN 77 OPTIMIZING COMPILER    01411
E NAME:  *MAIN*
      SUBROUTINE PDPAR
C*****************************************************************
C     OBJETO
C     LEER LOS PARAMETROS DEL MODELO DE PREDICCION DE SU ARCHIVO
C     CORRESPONDIENTE
C
C     IMPLEMENTADA POR MOTOKI INOUE KUBO
C     FECHA DE ULTIMA REVISION 3 DE DICIEMBRE DE 1980
C     REVISADA POR MIK
C
C     ENTRADAS
C        T = COMPONENTE DE TENDENCIA
C        S = COMPONENTE CICLICA
C        VE = MATRIZ DE ENERGIAS
C        Z = CARGAS REALES
C        PROM = PROMEDIO DE ENERGIAS
C        BETA = FACTORES DE PESO
C        PRONO = MATRIZ DE PRONOSTICOS REALIZADOS
C        ND = NUMERO DE DIAS EMPLEADOS
C        IPR = CONTADOR DE PRONOSTICOS
C        M = LONGITUD DE VENTANA
C        ALFA = FACTOR DE ACTUALIZACION
C
C     SALIDAS
C        T = COMPONENTE DE TENDENCIA
C        S = COMPONENTE CICLICA
C        VE = MATRIZ DE ENERGIAS
C        Z = CARGAS REALES
C        PROM = PROMEDIO DE ENERGIAS
C        BETA = FACTORES DE PESO
C        PRONO = MATRIZ DE PRONOSTICOS REALIZADOS
C        ND = NUMERO DE DIAS EMPLEADOS
C        IPR = CONTADOR DE PRONOSTICOS
C        M = LONGITUD DE VENTANA
C        ALFA = FACTOR DE ACTUALIZACION
C
C     IDENTIFICADORES LOCALES
C        I = CONTADOR
C
C     MODULOS EMPLEADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,ID,IPR,IHORA,LDTIME,INTH,ND,TOL
```

```
16 JUL 81 10:19:29    HARRIS FORTRAN 77 OPTIMIZING COMPILER    P01411
 MODULE NAME:  RDPAR
1625:          COMMON/FILES/TR,IW,IENT,ISAL,ISAL2,IFVA,IENT2,IRES
1626:          REWIND ISAL
1627:  C       ---------------------------------------------------
1628:  C       *LEE DEL ARCHIVO DE PARAMETROS (ISAL) LOS PARAMETROS
1629:  C        NECESARIOS PARA EL PRONOSTICO.
1630:  C       ---------------------------------------------------
1631:          READ(ISAL,5000)(T(J),J=1,24)
1632:          READ(ISAL,5000)((S(I,J),J=1,24),I=1,4)
1633:          READ(ISAL,5000)((VE(I,J),J=1,24),I=1,M)
1634:          READ(ISAL,5000)(Z(J),J=1,24)
1635:          READ(ISAL,5000)((PROM(I,J),J=1,4),I=1,7)
1636:          READ(ISAL,5000)((BETA(I,J),J=1,24),I=1,7)
1637:          FOR I=1,2
1638:          .  READ(ISAL,5000)((PRONO(K,J,I),J=1,24),K=1,7)
1639:          ENDFOR
1640:          READ(ISAL,5100)ND,TPR,M,ALFA
1641:          RETURN
1642:  C       ------------------------
1643:  C       *FORMATOS DE LECTURA
1644:  C       ------------------------
1645:  5000    FORMAT(8F10.3)
1646:  5100    FORMAT(3I11,F6.3)
1647:          END
```

```
81 10:19:32   HARRIS FORTRAN 77 - OPTIMIZING COMPILER   01411
F NAME:  *MAIN*
      SUBROUTINE FCRES(R)
C*****************************************************************
C     OBJETO
C     SUBRUTINA PARA SIMULAR EL PRONOSTICO DE LOS RESIDUOS
C     (ES LA COMPONENTE SENSIBLE A FACTORES METEOROLOGICOS)
C       *EN ESTA VERSION LA SUBRUTINA ES MUDA*
C
C     IMPLEMENTADA POR MOTOKI INOUE K
C     FECHA DE ULTIMA REVISION  3 DE DICIEMBRE DE 1980
C     REVISADA POR MOTOKI INOUE K
C
C     ENTRADAS
C        NINGUNO
C
C     SALIDAS
C        R = COMPONENTE RESIDUAL
C
C     IDENTIFICADORES LOCALES
C        I = CONTADOR
C
C     MODULOS EMPLEADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************
      DIMENSION R(24)
      FOR I=1,24
      .  R(I)=0.0
      ENDFOR
      RETURN
      END
```

```
6 JUL 81 10:19:33   HARRIS FORTRAN 77 OPTIMIZING COMPILER  01411
MODULE NAME:  *MAIN*
681:        SUBROUTINE CARDAT
682: C*****************************************************************
683: C      OBJETO
684: C      LEER DATOS DE CARGA DE UN DIA
685: C
686: C      IMPLEMENTADA POR MOTOKI INOUE KUBO
687: C      FECHA DE ULTIMA REVISION  3 DE DICIEMBRE DE 1980
688: C      REVISADA POR MIK
689: C
690: C      ENTRADAS
691: C         IDAY = DIA
692: C         MONTH = MES
693: C         IYEAR = ANO
694: C         Z = CARGA REAL
695: C
696: C      SALIDAS
697: C         IDAY = DIA
698: C         MONTH = MES
699: C         IYEAR = ANO
700: C         Z = CARGA REAL
701: C
702: C      IDENTIFICADORES LOCALES
703: C         JJ = CONTADOR
704: C
705: C      MODULOS UTILIZADOS
706: C         NINGUNO
707: C
708: C      NOTAS
709: C         NINGUNO
710: C
711: C*****************************************************************
712:        COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
713:       *  BETA(7,24)
714:        COMMON/TRES/IERR
715:        COMMON/DAY/IDAY,MONTH,IYEAR
716:        COMMON/CARGA1/ZP(24)
717:        COMMON/CARGA2/Z(24)
718:        COMMON/PARAM/ALFA,M,ID,IPR,IHORA,LDTIME,INTH,ND,TOL
719:        COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
720: C      -----------------
721: C      *LECTURA DE DATOS
722: C      -----------------
723:        READ(IENT2,5000)IDAY,MONTH,IYEAR,(Z(JJ),JJ=1,12)
724:        READ(IENT2,5000)IDAY,MONTH,IYEAR,(Z(JJ),JJ=13,24)
725:        RETURN
726: C      -------------------
727: C      *FORMATO DE LECTURA
728: C      -------------------
729: 5000   FORMAT(3I2,12F6.0)
730:        END
```

```
A1 10:19:35    HARRIS FORTRAN 77 OPTIMIZING COMPILER    2014
F NAME:  *MAIN*
      SUBROUTINE ACBETA
C*****************************************************************
C*(1) OBJETO
C     ACTUALIZAR LOS VALORES DE BETA HORARIOS(FACTORES DE PESO)
C     QUE SE UTILIZA EN LA ACTUALIZACION DE LA COMPONENTE CICLICA
C
C     IMPLEMENTADA POR MOTOKI INOUE KUBO
C     FECHA DE ULTIMA REVISION  3 DE DICIEMBRE DE 1980
C     REVISADA POR MIK
C
C     ENTRADAS
C        ALFA  =  FACTOR DE PESO PARA ACTUALIZAR BETAS
C        BETA  =  MATRIZ CON LOS VALORES ANTERIORES DE BETAS
C        PRONO =  MATRIZ CONTENIENDO LOS ULTIMOS PRONOSTICOS
C        VE    =  MATRIZ DE ENERGIA DE LOS ULTIMOS N DIAS
C        Z     =  DATOS DE CARGA HORARIA
C
C     SALIDA
C        BETA  =  FACTORE DE PESO ACTUALIZADOS
C
C     IDENTIFICADORES LOCALES
C        I     =  CONTADOR
C        RES1  =  VARIABLE AUXILIAR PARA CALCULOS
C        RES2  =  VARIABLE AUXILIAR PARA CALCULOS
C        X1    =  FACTOR DE ACTUALIZACION (1-ALFA)
C
C     MODULOS UTILIZADOS
C        NINGUNO
C
C     NOTAS
C        NINGUNO
C
C*****************************************************************
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,ID,IPR,IHORA,LDTIME,INTH,ND,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      FOR I=1,24
C        ---------------------------------
C        *CALCULO DE LOS VALORES AUXILIARES
C        ---------------------------------
       .  RES1=PRONO(ID,I,2)-Z(I)
       .  RES2=PRONO(ID,I,1)-VE(22,I)
       .  X1=1.0-ALFA
C        ------------------------------
C        *ACTUALIZA LOS FACTORES DE PESO
C        ------------------------------
       .  BETA(ID,I)=X1*BETA(ID,I)+ALFA*(RES1/(RES2+0.01))
C        ---------------------------------------------
```

```
16 JUL 81 10:19:35   HARRIS FORTRAN 77 OPTIMIZING COMPILER   20141101
 MODULE NAME:  ACBETA
1784: C        *SI ALGUN VALOR DE 'BETA' SE DISPARA,SATURARLO
1785: C*****************************************************************
1786:          IF((BETA(ID,T).LE.0.0).OR.(BETA(ID,T).GE.1.0))BETA(ID,T)=0.
1787:        ENDFOR
1788:        RETURN
1789:        END
```

```
81 10:19:36   HARRIS FORTRAN 77  OPTIMIZING COMPILER    01411                RE
LE NAME:  *MAIN*
      SUBROUTINE ACTCIC(L)
C*********************************************************************
C     OBJETO                                                        *
C     ACTUALIZAR LA COMPONENTE CICLICA CON NUEVOS DATOS             *
C                                                                   *
C     IMPLEMENTADA POR MOTOKI INOUE KUBO                            *
C     FECHA DE ULTIMA REVISION  4 DE DICIEMBRE DE 1980              *
C     REVISADA POR MIK                                              *
C                                                                   *
C     ENTRADAS                                                      *
C        ALFA   =  FACTOR DE ACTUALIZACION                          *
C        BETA   =  FACTORES DE PESO PARA LA ACTUALIZACION DE LA     *
C                  COMPONENTE CICLICA                               *
C        ID     =  IDENTIFICACION DEL DIA                           *
C        M      =  LONGITUD DE VENTANA                              *
C        ND     =  NUMERO DE DIAS UTILIZADOS HASTA LA FECHA         *
C        S      =  COMPONENETE CICLICA SEMANAL ANTERIOR             *
C                                                                   *
C     SALIDA                                                        *
C        S      =  COMPONENTE CICLICA SEMANAL ACTUALIZADO           *
C                                                                   *
C     IDENTIFICADORES LOCALES                                       *
C        ALF2   =  FACTOR DE ACTUALIZACION                          *
C        T      =  CONTADOR                                         *
C        Y1     =  VARIABLE AUXILIAR PARA CALCULOS                  *
C                                                                   *
C     MODULOS EMPLEADOS                                             *
C        NINGUNO                                                    *
C                                                                   *
C     NOTAS                                                         *
C        NINGUNO                                                    *
C                                                                   *
C*********************************************************************
      COMMON/UNO/T(24),VE(28,24),S(4,24),PROM(7,4),PRONO(7,24,2),
     *  BETA(7,24)
      COMMON/TRES/IERR
      COMMON/DAY/IDAY,MONTH,IYEAR
      COMMON/CARGA1/ZP(24)
      COMMON/CARGA2/Z(24)
      COMMON/PARAM/ALFA,M,ID,IPR,THORA,LDTIME,INTH,ND,TOL
      COMMON/FILES/IR,IW,IENT,ISAL,ISAL2,IEVA,IENT2,IRES
      FOR I=1,24
       .  ALF2=1.0-BETA(ID,I)
       .  Y1=Z(I)-T(I)/FLOAT(M)
C         ----------------------------------------
C         *ACTUALIZA LA COMPONENTE CICLICA SEMANAL
C         ----------------------------------------
       .  S(L,I)=BETA(ID,I)*Y1+ALF2*S(L,I)
      ENDFOR
      RETURN
      END
REQUIRED:       65 PAGES
```

APENDICE C.2

RESULTADOS DEL PROGRAMA PROTOTIPO DE ESTIMACION DE CARGA

"METODO DE COMPONENTES"

HORA	CARGA REAL	PRONOSTICO	ERROR
1	4809.000	4810.000	.021 %
2	4636.000	4665.000	.626 %
3	4552.000	4582.000	.659 %
4	4461.000	4556.000	2.130 %
5	4565.000	4566.000	.022 %
6	4710.000	4693.000	-.361 %
7	4833.000	4921.000	1.821 %
8	5414.000	5445.000	.573 %
9	5874.000	5879.000	.085 %
10	6235.000	6117.000	-1.893 %
11	6246.000	6223.000	-.368 %
12	6388.000	6316.000	-1.127 %
13	6350.000	6299.000	-.803 %
14	6356.000	6293.000	-.991 %
15	6302.000	6216.000	-1.365 %
16	6266.000	6336.000	1.117 %
17	6556.000	6550.000	-.092 %
18	6646.000	6733.000	1.309 %
19	6693.000	6750.000	.852 %
20	7403.000	7305.000	-1.324 %
21	7912.000	7831.000	-1.024 %
22	7528.000	7495.000	-.438 %
23	6655.000	6580.000	-1.127 %
24	5783.000	5891.000	1.868 %

ERRORES DEL DIA LUNES 3 DE JUL DEL 78

DIFERENCIA DE HORAS PICO-----	.000
ERROR EN PICO----------	-1.024%
DESVIACION STANDARD----------	65.729MWH
ERROR PROMEDIO DEL DIA-------	.916 %
ERROR EN ENERGIA DIARIA------	-.085 %
RANGO MAXIMO DE ERROR--------	4.022%
ERROR ABSOLUTO PROMEDIO------	.916 %
DESVIACION STANDARD PROMEDIO	65.729 MWH
ERROR DE ENERGIA PROMEDIO----	.085 %
ERROR EN PICO PROMEDIO-------	1.024
ENERGIA REAL----------	143173.000 MWH
ENERGIA PREDICHA----------	143052.000 MWH

HORA	CARGA REAL	PRONOSTICO	ERROR
1	5526.000	5461.000	-1.176 %
2	5251.000	5242.000	-.171 %
3	5042.000	5097.000	1.091 %
4	5076.000	5072.000	-.079 %
5	5162.000	5123.000	-.756 %
6	5203.000	5186.000	-.327 %
7	5305.000	5344.000	.735 %
8	5793.000	5773.000	-.345 %
9	6252.000	6075.000	-2.831 %
10	6324.000	6340.000	.253 %
11	6553.000	6423.000	-1.984 %
12	6500.000	6493.000	-.108 %
13	6533.000	6520.000	-.199 %
14	6483.000	6451.000	-.494 %
15	6505.000	6353.000	-2.337 %
16	6496.000	6420.000	-1.170 %
17	6622.000	6747.000	1.888 %
18	6782.000	6938.000	2.300 %
19	6683.000	6926.000	3.636 %
20	7390.000	7465.000	1.015 %
21	7991.000	7899.000	-1.151 %
22	7742.000	7556.000	-2.023 %
23	6787.000	6770.000	-.250 %
24	5999.000	5962.000	-.617 %

ERRORES DEL DIA MARTES 4 DE JUL DEL 78

DIFERENCIA DE HORAS PICO----- .000
ERROR EN PICO---------- -1.151%
DESVIACION STANDARD---------- 97.634MWH
ERROR PROMEDIO DEL DIA------ 1.122 %
ERROR EN ENERGIA DIARIA------ -.223 %
RANGO MAXIMO DE ERROR-------- 6.467%

ERROR ABSOLUTO PROMEDIO------ 1.019 %
DESVIACION STANDARD PROMEDIO 83.225 MWH
ERROR DE ENERGIA PROMEDIO---- .154 %
ERROR EN PICO PROMEDIO------ 1.088

ENERGIA REAL---------- 149970.000 MWH
ENERGIA PREDICHA---------- 149636.000 MWH

HORA	CARGA REAL	PRONOSTICO	ERROR
1	5601.000	5481.000	-2.142 %
2	5356.000	5250.000	-1.979 %
3	5124.000	5075.000	-.956 %
4	5169.000	5071.000	-1.896 %
5	5224.000	5136.000	-1.685 %
6	5253.000	5187.000	-1.256 %
7	5348.000	5328.000	-.374 %
8	5787.000	5787.000	.000 %
9	6191.000	6116.000	-1.211 %
10	6392.000	6337.000	-.860 %
11	6398.000	6468.000	1.094 %
12	6496.000	6490.000	-.092 %
13	6479.000	6536.000	.880 %
14	6510.000	6453.000	-.876 %
15	6472.000	6402.000	-1.082 %
16	6434.000	6454.000	.311 %
17	6621.000	6724.000	1.556 %
18	6628.000	6898.000	4.074 %
19	6777.000	6852.000	1.107 %
20	7399.000	7444.000	.608 %
21	8084.000	7927.000	-1.942 %
22	7664.000	7594.000	-.913 %
23	6898.000	6775.000	-1.783 %
24	6111.000	5965.000	-2.389 %

ERRORES DEL DIA MIERCOLES 5 DE JUL DEL 78

DIFERENCIA DE HORAS PICO---------- .000
ERROR EN PICO---------- -1.942%
DESVIACION STANDARD---------- 98.463MWH
ERROR PROMEDIO DEL DIA---------- 1.294 %
ERROR EN ENERGIA DIARIA---------- -.443 %
RANGO MAXIMO DE ERROR---------- 6.463%

ERROR ABSOLUTO PROMEDIO---------- 1.111 %
DESVIACION STANDARD PROMEDIO 88.596 MWH
ERROR DE ENERGIA PROMEDIO---------- .250 %
ERROR EN PICO PROMEDIO---------- 1.372

ENERGIA REAL---------- 150416.000 MWH
ENERGIA PREDICHA---------- 149750.000 MWH

HORA	CARGA REAL	PRONOSTICO	ERROR
1	5585.000	5516.000	-1.235 %
2	5201.000	5272.000	1.365 %
3	5250.000	5091.000	-3.029 %
4	5207.000	5096.000	-2.132 %
5	5221.000	5162.000	-1.130 %
6	5269.000	5204.000	-1.294 %
7	5262.000	5322.000	1.140 %
8	5741.000	5792.000	.888 %
9	6125.000	6135.000	.163 %
10	6398.000	6342.000	-.875 %
11	6411.000	6449.000	.593 %
12	7025.000	6489.000	-7.630 %
13	6528.000	6519.000	-.138 %
14	6512.000	6459.000	-.814 %
15	6343.000	6421.000	1.230 %
16	6502.000	6442.000	-.923 %
17	6755.000	6693.000	-.918 %
18	6715.000	6819.000	1.549 %
19	6756.000	6835.000	1.169 %
20	7433.000	7437.000	.054 %
21	8084.000	7976.000	-1.336 %
22	7650.000	7614.000	-.471 %
23	6841.000	6825.000	-.234 %
24	6117.000	6017.000	-1.635 %

ERRORES DEL DIA JUEVES 6 DE JUL DEL 78

DIFERENCIA DE HORAS PICO----- .000
ERROR EN PICO----------1.336%
DESVIACION STANDARD----------130.766MWH
ERROR PROMEDIO DEL DIA----------1.331 %
ERROR EN ENERGIA DIARIA----------.667 %
RANGO MAXIMO DE ERROR----------9.179%

ERROR ABSOLUTO PROMEDIO------ 1.166 %
DESVIACION STANDARD PROMEDIO 100.806 MWH
ERROR DE ENERGIA PROMEDIO---- 1.354 %
ERROR EN PICO PROMEDIO---------- 1.363

ENERGIA REAL----------150931.000 MWH
ENERGIA PREDICHA----------149924.000 MWH

HORA	CARGA REAL	PRONOSTICO	ERROR
1	5682.000	5544.000	-2.429 %
2	5343.000	5254.000	-1.666 %
3	5314.000	5137.000	-3.331 %
4	5202.000	5139.000	-1.211 %
5	5187.000	5190.000	.058 %
6	5396.000	5218.000	-3.299 %
7	5453.000	5305.000	-2.714 %
8	5729.000	5775.000	.803 %
9	6273.000	6161.000	-1.785 %
10	6519.000	6367.000	-2.332 %
11	6509.000	6447.000	-.953 %
12	6555.000	6651.000	1.465 %
13	6690.000	6519.000	-2.556 %
14	6621.000	6476.000	-2.190 %
15	6503.000	6405.000	-1.507 %
16	6504.000	6457.000	-.723 %
17	6839.000	6705.000	-1.959 %
18	6980.000	6776.000	-2.923 %
19	6897.000	6805.000	-1.334 %
20	7244.000	7424.000	2.485 %
21	7978.000	8014.000	.451 %
22	7707.000	7636.000	-.921 %
23	6859.000	6839.000	-.292 %
24	6103.000	6051.000	-.852 %

ERRORES DEL DIA VIERNES 7 DE JUL DEL 78

DIFERENCIA DE HORAS PICO---------- .000
ERROR EN PICO---------- .451%
DESVIACION STANDARD---------- 118.786MWH
ERROR PROMEDIO DEL DIA---------- 1.677 %
ERROR EN ENERGIA DIARIA---------- -1.178 %
RANGO MAXIMO DE ERROR---------- 5.816%
ERROR ABSOLUTO PROMEDIO---------- 1.268 %
DESVIACION STANDARD PROMEDIO---------- 104.652 MWH
ERROR DE ENERGIA PROMEDIO---------- .519 %
ERROR EN PICO PROMEDIO---------- 1.481
ENERGIA REAL---------- 152087.000 MWH
ENERGIA PREDICHA---------- 150295.000 MWH

HORA	CARGA REAL	PRONOSTICO	ERROR
1	5633.000	5591.000	-.746 %
2	5279.000	5369.000	1.705 %
3	5227.000	5186.000	-.784 %
4	5146.000	5169.000	.447 %
5	5218.000	5185.000	-.632 %
6	5234.000	5216.000	-.344 %
7	5283.000	5202.000	-1.533 %
8	5502.000	5478.000	-.436 %
9	5785.000	5806.000	.363 %
10	5999.000	6025.000	.433 %
11	5960.000	6129.000	2.836 %
12	6156.000	6248.000	1.494 %
13	6062.000	6153.000	1.501 %
14	6036.000	6057.000	.348 %
15	5767.000	5903.000	2.358 %
16	5869.000	5881.000	.204 %
17	6038.000	6003.000	-.580 %
18	5928.000	6019.000	1.535 %
19	6136.000	6183.000	.766 %
20	6813.000	6868.000	.807 %
21	7421.000	7421.000	.000 %
22	7482.000	7463.000	-.265 %
23	6575.000	6469.000	-1.612 %
24	5876.000	5809.000	-1.140 %

ERRORES DEL DIA SABADO 8 DE JUL DEL 78

DIFERENCIA DE HORAS PICO---------- .000
ERROR EN PICO---------- .000%
DESVIACION STANDARD---------- 69.807MWH
ERROR PROMEDIO DEL DIA---------- .953 %
ERROR EN ENERGIA DIARIA---------- .287 %
RANGO MAXIMO DE ERROR---------- 4.448%

ERROR ABSOLUTO PROMEDIO---------- 1.216 %
DESVIACION STANDARD PROMEDIO---------- 99.692 MWH
ERROR DE ENERGIA PROMEDIO---------- .480 %
ERROR EN PICO PROMEDIO---------- .984

ENERGIA REAL---------- 142425.000 MWH
ENERGIA PREDICHA---------- 142533.000 MWH

HORA	CARGA REAL	PRONOSTICO	ERROR
1	5415.000	5379.000	-.665 %
2	5134.000	5069.000	-1.266 %
3	5014.000	4918.000	-1.915 %
4	4850.000	4838.000	-.247 %
5	4914.000	4816.000	-1.994 %
6	4880.000	4812.000	-1.393 %
7	4712.000	4545.000	-3.544 %
8	4592.000	4554.000	-.828 %
9	4738.000	4746.000	.169 %
10	4843.000	4801.000	-.867 %
11	4793.000	4851.000	1.210 %
12	4905.000	5024.000	2.426 %
13	4948.000	4949.000	.020 %
14	4922.000	4992.000	1.422 %
15	4815.000	4895.000	1.661 %
16	4997.000	4862.000	-2.702 %
17	4924.000	4984.000	1.219 %
18	4987.000	5033.000	.922 %
19	5240.000	5165.000	-1.431 %
20	5857.000	5928.000	1.212 %
21	6628.000	6600.000	-.422 %
22	6518.000	6493.000	-.384 %
23	5927.000	5943.000	.270 %
24	5336.000	5291.000	-.843 %

ERRORES DEL DIA DOMINGO 9 DE JUL DEL 78

DIFERENCIA DE HORAS PICO----- .000
ERROR EN PICO----- .422%
DESVIACION STANDARD----------72.856MWH
ERROR PROMEDIO DEL DIA----- 1.240 %
ERROR EN ENERGIA DIARIA----- .324 %
RANGO MAXIMO DE ERROR----- 5.970%

ERROR ABSOLUTO PROMEDIO------ 1.215 %
DESVIACION STANDARD PROMEDIO 96.317 MWH
ERROR DE ENERGIA PROMEDIO----- .458 %
ERROR EN PICO PROMEDIO-------- .904

ENERGIA REAL----------123889.000 MWH
ENERGIA PREDICHA----------123488.000 MWH