



Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

INGENIERIA DE PROGRAMACION PARA EL DESARROLLO DE
UN PROGRAMA DE "MOVILIDAD" DE UNIDADES
GENERADORAS DE ENERGIA ELECTRICA.

TESIS PROFESIONAL

Que para obtener el Título de
INGENIERO EN COMPUTACION

P r e s e n t a

JUAN CARLOS PEREZ CASTAÑEDA



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

Resumen	1
Introducción	2
Capítulo I: Metodología	5
Capítulo II: Organización del Sistema Eléctrico Mexicano ..	41
Capítulo III: Fases de Requerimientos y de Diseño	58
Capítulo IV: Fase de Desarrollo	92
Conclusiones	115
Apéndice A: Estructuras Básicas	119
Apéndice B: Normas de Programación	123
Apéndice C: Normas de Planes de Pruebas	128
Apéndice D: Manual de Usuario	132
Bibliografía	147

Resumen.

Este trabajo muestra el desarrollo de un programa de computadora así como la metodología que se ha usado para desarrollarlo.

El programa forma parte de un sistema de programas de cómputo de aplicación técnica (en sistemas de potencia). La metodología ha sido usada en todos los programas que forman el sistema para llevar un control eficiente del desarrollo, tanto de los programas, como del sistema en general.

I N T R O D U C C I Ó N .

El objetivo del presente trabajo es mostrar la forma en que se ha aplicado una metodología para desarrollar sistemas de programas de cómputo. Se presenta como ejemplo un programa que forma parte del Sistema de Información y Control en Tiempo Real, que el Instituto de Investigaciones Eléctricas está desarrollando para el Centro Nacional de Control de Energía (CENACE) de la Comisión Federal de Electricidad.

Aun cuando la metodología no es completamente original, el desarrollo de este trabajo es importante por tres razones:

- 1) La metodología no ha sido completamente difundida.
- 2) Es una aplicación nueva.
- 3) Algunas recomendaciones surgidas de la experiencia de utilizar esta metodología son expuestas en esta tesis.

El desarrollo del presente trabajo tiene la siguiente secuencia:

En el capítulo uno se expone la metodología que se ha usado para desarrollar sistemas de programas de cómputo.

El capítulo dos trata del sistema de programas en el que se ha venido aplicando la metodología mencionada. Uno de los programas que forman parte del sistema (Programa ejemplo) es desarrollado aplicando esta metodología y se muestra en los capítulos tres y cuatro.

En el capítulo tres se presentan las etapas de requerimientos y de diseño del programa ejemplo; mientras que el capítulo cuatro trata la etapa de desarrollo.

En las conclusiones se muestran las principales ventajas de la metodología presentada en los capítulos anteriores; se dan algunas sugerencias en cuanto a la aplicación de la misma y se describen las experiencias obtenidas al desarrollar programas de esta manera.

El apéndice 'A' presenta un breve resumen de las estructuras básicas de control que comprende la programación estructurada.

En el apéndice 'B' se muestran algunas de las normas usadas para controlar la generación de código ejecutable durante la fase de desarrollo del programa ejemplo.

El apéndice 'C' contiene un resumen de las normas emitidas para controlar la elaboración del documento Planes de Prueba (ver capítulo IV).

En el apéndice 'D' se ha agregado el manual de usuario del programa ejemplo; esta forma original es presentada como propuesta para documentar los manuales de usuario.

C A P Í T U L O I .

Metodología.

1.0 METODOLOGÍA.

Este capítulo contiene la parte teórica con la que se desarrollará en los capítulos siguientes el diseño, desarrollo e integración del programa ejemplo (llamado 'Ajuste Interactivo de Generación' (AIG)).

El sistema de programas de cómputo sobre el que se aplicarán estos conceptos es grande, es para el Centro Nacional de Control de Energía (CENACE). Por lo tanto todas las especificaciones están referidas a sistemas de cómputo de gran magnitud.

1.1 La Programación En Sus Inicios.

En los inicios de los sistemas de cómputo los programadores realizaban su trabajo en una forma independiente. Al desarrollar un programa probaban que funcionara correctamente en la máquina; esto era suficiente para que aquellos programadores consideraran que su programa era válido. Posteriormente el usuario al utilizar el programa empezaba a tener problemas, debido a que en ocasiones la programación no cumplía cabalmente con los requerimientos del usuario; además cuando este deseaba integrar nuevas funciones al programa, resultaba excesivamente complicado. En otros casos, si deseaba integrar varios programas para formar un sistema resultaba casi imposible. En muchas ocasiones sucedía que se tenía que reprogramar casi todo el sistema. Esto ocasionaba que los costos de mantenimiento y de la programación en general se incrementaran muchísimo.

Es evidente que los programadores no estaban aplicando una metodología adecuada para desarrollar sus productos dado que los mayores esfuerzos los dedicaban a las fases de codificación y de pruebas, despreciando la mayoría de las veces la etapa de análisis.

Hay que notar, que el desarrollo del equipo de cómputo progresaba en una forma vertiginosa; aun así, muchas de las mejoras en la manufactura del equipo perdían parte de su utilidad al ser manejadas por una programación deficiente. De esta manera, resultaba que mientras los costos del equipo iban siempre disminuyendo, los costos de la programación iban siempre en aumento.

La clase de problemas de programación antes mencionados se llevaron a presentar en prácticamente todos los sistemas de programas de cómputo. La gravedad del asunto llegó a un extremo tan grande que se le dio en llamar la 'Crisis de la Programación' [FKR82]

En parte los programas no se desarrollaban de una manera satisfactoria debido a que el control administrativo había fallado.

Saltaba a la vista que era necesario elaborar una metodología para el desarrollo de programas de computadora, la cual fuera capaz de solucionar problemas tales como:

- Sobre costo en el desarrollo.
- Retraso en la entrega de los programas.
- Baja confiabilidad e inconformidad del usuario, por definición mala o incompleta de los requerimientos.
- El mantenimiento de programas.
- Desarrollo de programas dentro del contexto de un sistema.
- Administración efectiva de sistemas de programación.

Se llevaron a cabo una serie de investigaciones en los ámbitos científicos, para desarrollar metodologías de programación que evitaran los problemas mencionados.

Estas metodologías han ido evolucionando con los años, genéricamente se conocen como 'Técnicas de Diseño Estructurado' (diseño de arriba hacia abajo, diseño de abajo hacia arriba, programación estructurada, etc.).

En la actualidad se hace énfasis en contar con una definición adecuada del problema, un diseño estructurado, un análisis completo de los datos y de la lógica del programa antes de empezar su codificación.[GER81] Con base en estos conceptos, en el Instituto de Investigaciones Eléctricas se ha desarrollado una metodología para el desarrollo de programación, la parte teórica de la misma se presenta en lo que resta de este capítulo, la parte práctica se presenta en los capítulos 3 y 4.

1.2 El Enfoque Sistemico Para El Desarrollo De Programas.

Se desea que los programas tengan las siguientes características: ser útiles, eficientes, transferibles, mantenibles, confiables, uniformes y probables. [GER83]

Para lograr esto se puede llevar a cabo la programación con un Enfoque Sistemico. Este enfoque consiste en dar un tratamiento de sistema al problema que se trata de solucionar por medio de un programa de computadora. Antes de empezar a generar código para solucionar el problema se tiene que pasar por una etapa de análisis y definición de requerimientos.

El enfoque sistemico para el desarrollo de programas de computadora divide este proceso de desarrollo en cuatro fases: [GER83]

- 1.-Definición de requerimientos.
- 2.-Diseño.
- 3.-Desarrollo.

4.-Operación y mantenimiento.

Se planea que el sistema de programas sea integrado en el futuro a un sistema de información. A continuación se delinearán cada una de las fases enumeradas; además en la tabla 1-1 se muestra un resumen de las fases, las etapas, los documentos generados y las herramientas teóricas que comprende esta metodología. Cada fase está dividida en etapas; al trabajar en las etapas se generan los documentos que se mencionan en la tabla. Para trabajar en estas etapas se echa mano de las herramientas teóricas que se mencionan.

FASE	ETAPA	DOCUMENTO	HERRAMIENTA
Definición de Requerimientos	Definición de Objetivos	Documento de Objetivos (DO)	Modelo Conceptual (enfoque de usuario)
	Análisis	Reporte de Análisis (RA)	Modelo Conceptual (enfoque de analista)
	Especificación de Requerimientos	Requerimientos de Programa de Computadora (RPC)	
Diseño	Arquitectura	Documento de Arquitectura (ARQ)	Diagrama de Estructura del Sistema Total
	Diagrama de Estructura	Diseño de Programa de Computadora (DPC)	Diagrama de Estructura por Programa
	Detallado de Módulos		Pseudocódigo
Desarrollo y Pruebas	Codificación	Código	Lenguaje de Programación
Pruebas	Integración	Planes de Pruebas (PLP)	RPC
	Pruebas	Procedimientos de Pruebas (PRP)	PLP
Operación y Mantenimiento		Manuales de Usuario (MU)	RPC y DPC

Tabla 1-1.

1.2.1 Fase De Definición De Requerimientos. -

Para la definición de requerimientos, es necesario primero analizar cuales son los objetivos que se pretenden alcanzar con la elaboración del sistema. Se delinear los objetivos y se establecen las funciones generales que se deséa el sistema ejecute. Una vez planteados los objetivos del sistema y las funciones generales que se deséa que este lleve a cabo, se prosigue con la fase de especificación de requerimientos. En esta parte se especifica más claramente como se van a alcanzar los objetivos, es decir, se delinear con más detalle las funciones que ejecutará el sistema, trazando las características que debe tener cada una de ellas.

Para llevar a cabo la definición de requerimientos en una forma sistemática, se ha dividido este proceso en tres etapas, estas son: Definición de Objetivos, Análisis y Especificación de Requerimientos. A continuación se explica en que consiste cada una de ellas.

1.2.1.1 Definición De Objetivos. -

En esta etapa el analista tiene una fuerte interacción con el usuario para establecer cuales son las necesidades de este último. Una vez hecho esto, se realiza el planteamiento de los objetivos del sistema. El planteamiento de estos objetivos quedará en un documento llamado: Documento de Objetivos (DO) del Proyecto. En este documento se establecen los objetivos que el usuario deséa que sean cubiertos, se mencionan las funciones que se implementarán para cubrir dichos objetivos y las características que debe tener el usuario que trabaje

con estas funciones.

En la tabla 1-2 se muestra el índice que debe ser llenado para todos los Documentos de Objetivos que se generen en el subsistema Coordinación Hidro Térmica CHT.

1	Introducción.
2	Objetivos del Documento
2.1	Consideraciones de Utilización.
2.1.1	Cobertura.
2.1.2	Usuario.
2.2	Restricciones de desarrollo.
2.2.1	Modularidad.
2.2.2	Flexibilidad.
2.3	Criterios de Aceptación.
2.4	Tiempo de Desarrollo.
3	Referencias.

Tabla 1-2.

Las especificaciones se desplazan en los siguientes incisos.

i) Consideraciones de utilización. Abarca los aspectos de Cobertura y Usuario. En el aspecto cobertura se mencionan las funciones que debe cumplir el sistema para alcanzar los objetivos. En el aspecto usuario se menciona a que tipo de usuario va dirigido el sistema.

ii) Restricciones de desarrollo. Debe contener las restricciones que se imponen al desarrollo del sistema. Tanto en modularidad como en flexibilidad, para prevenir futuras expansiones y mantenimiento.

iii) Criterios de aceptación. Debe especificar quienes lo aceptan, bajo que condiciones será aceptado y que documentación debe llevar de

respaldo.

iv) Tiempo de desarrollo. Se establecerá una fecha de inicio y de terminación del sistema.

1.2.1.2 Análisis. -

En esta etapa, como su nombre lo indica, se hace un análisis para establecer la forma en que los objetivos se transformarán en especificaciones del sistema. Al hacer este análisis suelen surgir cuestiones como las que se mencionan a continuación:

- a) Conflictos que se pueden llegar a presentar al tratar de dar forma a objetivos diferentes.
- b) Establecer prioridades entre eventos comunes.
- c) Hacer un análisis de factibilidad de los objetivos.
- d) Proposición de procedimientos (si es el caso) para llevar a buen término los objetivos.

Hay que tener muy presente que el hacer un buen análisis es una condición necesaria para que el proyecto tenga éxito y que todo análisis debe quedar plasmado en un documento denominado: Reporte de Análisis (RA). En la elaboración de este documento deberán participar tanto el analista como el usuario; deberán especificar si es que se han establecido compromisos y/o prioridades entre las formas de llevar a cabo los objetivos (el como y el porque). Observemos que aun cuando la etapa de análisis es principalmente responsabilidad del analista, la participación del usuario es necesaria en gran medida.

En el Instituto de Investigaciones Eléctricas (IIE) se ha dado el caso de que a muchos de los problemas analizados (en relación a centros de control de energía) se les ha dado un enfoque novedoso, además de ser únicos en cuanto a sus características físicas (geográficas, de red, etc.), por esta particularidad se han reportado siguiendo la guía de publicaciones de la Sociedad de Ingenieros en Potencia del IEEE.[IEEE77] De esta forma se documenta esta actividad, tanto para el sistema, como para la comunidad científica en general. El índice de un documento de este tipo es mostrado en la tabla 1-3, este es, entonces, el Reporte de Análisis mencionado anteriormente.

a) Título.
b) Sumario.
c) Introducción.
d) Cuerpo del Reporte.
e) Conclusiones.
f) Apéndices.
g) Referencias.
Tabla 1-3

En forma breve el índice es:

- 1 Título. Debe indicar claramente el contenido del documento.
- 2 Sumario. Un grupo de oraciones que comprendan las cualidades esenciales del documento.
- 3 Introducción. Debe orientar al lector con respecto al problema. Incluyendo la naturaleza del problema, las bases de trabajos previos, el propósito y significado del documento y si es necesario, el método por el que el problema se atacará y la

organización del material en el documento.

4 Cuerpo. Contiene el mensaje completo del documento.

5 Conclusiones. Deben cubrir lo que muestra el trabajo y su significado, limitaciones y ventajas. Si es el caso, deberá incluir aplicaciones de los resultados y recomendaciones para futuros trabajos.

6 Apéndices. Desarrollos y pruebas matemáticas que son importantes en el documento.

7 Referencias. Publicaciones comunes disponibles.

Todo esto en un máximo de nueve páginas. Como se podrá observar, sea o no que se vaya a publicar el reporte, esta es una buena guía para elaborar los Reportes de Análisis en general

1.2.1.3 Especificación De Requerimientos. -

En esta etapa se establecen formalmente en un documento denominado Requerimientos de Programas de Computadora (RPC), las acciones que el sistema deberá ser capaz de ejecutar y en que medida se alcanzarán los objetivos; en general describe lo que será el sistema de programas de cómputo. Este documento debe contar con las siguientes características: [GER83]

- ser comprensible
- ser formal
- ser completo
- ser modificable

El RPC debe ser comprensible tanto para el usuario como para el analista y para el diseñador.

Ser formal implica que la redacción debe ser clara para cada uno de los requerimientos, identificándolos en forma explícita; para esto se debe utilizar una notación suficientemente rigurosa y a la vez entendible.

Deberá ser completo en cuanto a abarcar todos aquellos aspectos que no debe decidir el diseñador.

Modificable. Se refiere a que las modificaciones de la información se hagan de una manera rápida y al costo mínimo.

Tanto el DO como el RPC deben considerar los siguientes aspectos:

- El uso que se dará al sistema a desarrollar.
- La descripción del sistema y su ambiente por medio de uno o varios modelos conceptuales.
- El tiempo de entrega y el costo de desarrollo.
- Criterios de aceptación del producto.

Las diferencias entre los documentos están en:

- El grado de detalle.
- El enfoque de los modelos conceptuales. En el DO con enfoque de usuario y en el RPC con enfoque de diseñador y/o matemático.
- El DO señala características generales del sistema y el RPC señala características particulares de las partes que constituyen el sistema.

Se puede decir además que el DO es la iniciación de un proyecto, y que el RPC puede dar la alternativa de llevarlo a cabo o no. [GER83]

En la tabla 1-4 se muestra la estructura de un RPC [GER81] para el caso particular del sistema de programas del Centro Nacional de Control de Energía.

- | |
|---|
| 1. Introducción y propósito. |
| 2. Referencias. |
| 2.1 Especificación del sistema de información y control en tiempo real. |
| 2.2 Especificación funcional de interfase del programa. |
| 2.3 Estudios y reportes. |
| 2.4 Normas y convenciones. |
| 2.5 Glosarios, abreviaciones y símbolos. |
| 3. Definición del programa. |
| 3.1 Definición de subprogramas. |
| 3.2 Diagramas de interacción funcional de subprogramas. |
| 3.3 Resumen de subprogramas y módulos. |
| 4. Requerimientos del subprograma. |
| 4.X Requerimientos del subprograma X. |
| 5. Certificación de calidad. |
| 5.X Requerimientos de certificación de calidad del subprograma X. |
| 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 Definición de términos. |
| 6.5 Referencias. |

Tabla 1-4

Algunos de los puntos importantes que contiene el documento RPC son:

- i) Definición conceptual del sistema. Abarcando el entorno del sistema, su frontera y el tipo de usuario para el que se diseña.
- ii) Una reseña de todas las tareas que comprende el sistema.

- iii) Una explicación de la forma como se usará el sistema.
- iv) Una indicación de las restricciones de máxima cantidad de elementos que maneja el sistema.
- v) Un glosario de los términos que se manejen en las especificaciones.
- vi) El establecimiento de los requerimientos que se desean para el sistema.
- vii) La muestra de todos los mensajes que se desea que envíe el sistema.
- viii) El detalle de todos los desplegados que se desea que maneje el sistema.
- ix) Mencionar las normas de control de calidad a que debe ser sujeta la elaboración del sistema.
- x) Mencionar los criterios con que se aceptará el sistema y quienes serán los encargados de aplicarlos.
- xi) Finalmente, mostrar una matriz con los requerimientos establecidos, esto es para localizarlos más fácilmente.

Para hacer estas especificaciones se puede usar prosa y algún lenguaje gráfico (diagramas de flujo de proceso, de flujo de información, de flujo de control y notación matemática).

Hay que tener presente que el documento RFC será la base de desarrollo del sistema, por lo tanto, se debe mantener involucrado al usuario durante su especificación para evitar en lo posible modificaciones grandes en el futuro.

Para hacer las especificaciones se debe seguir una metodología y se debe tener en cuenta un conjunto de normas. La metodología sirve para la adquisición, análisis y presentación de la información concerniente al sistema. Las normas están constituidas por un conjunto de reglas que sirven para gobernar la elaboración de los documentos y también definen el criterio con el que se aprobarán los documentos terminados.

1.2.1.4 Modelo Conceptual. -

La herramienta principal que se usa a lo largo de la fase de definición de requerimientos es el modelo conceptual. Modelo conceptual de un sistema es un compendio de información sobre el mismo. [GER78] Dado que el propósito del estudio determina el tipo de información que se obtiene, no existe un modelo único por sistema. En el inciso 2.3.4 del capítulo 2 se puede observar un modelo conceptual. En general el modelo conceptual de un sistema comprende:

- Entidades y atributos, aquí entidad es un sinónimo de componente del sistema. Atributo es una propiedad o característica de las entidades.
- Proceso. Donde proceso es sinónimo de toda acción que origine cambios en el sistema.
- Frontera o ambiente, la frontera que separa al sistema del medio ambiente.

Los 'estados' en el sistema consisten de información sobre entidades relevantes y sus atributos en el medio ambiente.

Las 'acciones' del sistema son respuestas del sistema a eventos en el medio ambiente; estas acciones pueden, a su vez, originar otros eventos en el medio ambiente.

La información del estado puede obtenerse en una forma sistemática al observar que evento particular ocurre, que mensaje se envía al sistema y que acción se origina; finalmente, cómo la información contenida en el mensaje se deposita en los estados del sistema.

Esta secuencia de interacciones establece sistemáticamente las funciones principales del sistema (las acciones) así como el flujo de datos (mensajes). Similarmente, un analista puede establecer el flujo de control, por ejemplo la secuencia de funciones. Esto se hace al estudiar primero la secuencia de eventos en el medio ambiente.

Los estados, acciones, eventos y los flujos del sistema son el modelo.[YEHBO] En la figura 1-1 se ve la estructura básica de un modelo conceptual simple.

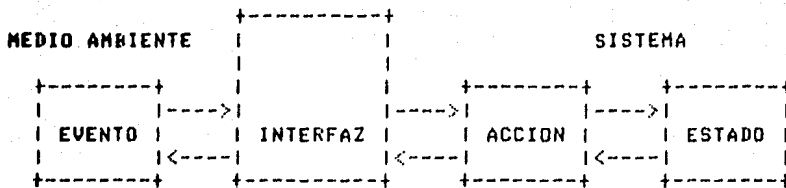


Figura 1-1

1.2.2 Fase De Diseño. -

Esta fase se ha dividido en tres etapas: Arquitectura, Diagrama de Estructura y Detallado de Módulos, mismas que se explican a continuación.

1.2.2.1 Arquitectura. -

En esta primera etapa se detectan las funciones más importantes que contiene el RPC para que sean convertidas en programas. Una vez hecho esto, se deberán definir las interrelaciones que existirán entre las funciones convertidas en programas. Finalmente se documentará esta etapa, en el documento de Arquitectura (ARQ), mostrando una lista de todos los programas con que contará el sistema y las interfaces que se establecerán entre ellos.

Para realizar las definiciones anteriores es de gran ayuda el modelo conceptual del RPC.

El índice que debe contener un documento de arquitectura se muestra en la tabla 1-5.

1	Introducción.
1.1	Identificación del documento.
1.2	Organización del documento.
1.3	Nomenclatura y convenciones.
2	Modelo conceptual.
2.1	Descripción.
2.2	Tareas.
3	Diagrama de estructura.
4	Relación programas-tareas.
5	Descripción de programas.
5.X	Programa X.
5.X.1	Objetivo.
5.X.2	Diagrama de flujo de información.
5.X.3	Características generales.
5.X.4	Interacción con la base de datos.
6	Bibliografía.

Tabla 1-5

Los puntos que incluye este índice son, en forma breve:

- i) **Introducción.** Se da la identificación del documento, la forma en que se organiza la información dentro del mismo y se especifica la nomenclatura y las convenciones que se usarán a lo largo del documento.
- ii) **Modelo conceptual del subsistema.** Se describe lo que es el subsistema y se mencionan las tareas (o funciones) que lo componen.
- iii) **Diagrama de estructura.** Muestra como se relacionan los

- Programas que comprende el subsistema con la base de datos y la interfaz hombre-máquina.
- iv) Relación programas-tareas. Se da una relación en forma de tabla del (o los) programa(s) que realiza(n) cada una de las tareas del subsistema.
- v) Descripción de los programas. Se describe en forma breve en que consiste cada uno de los programas del inciso (iv), incluyendo para cada uno: objetivo, diagrama de flujo de información, características generales e interacción con la base de datos.
- vi) Bibliografía. Todas la referencias que sirvieron para elaborar el documento.

En el inciso 3.2.1 del capítulo 3 se puede observar un diagrama de flujo de información. En este tipo de diagrama se muestra, para un programa dado, como fluye la información entre este, la base de datos, otros programas del sistema y la interface hombre-máquina. En el inciso 3.1.1 del capítulo 3 se puede ver el diagrama de estructura que dio como resultado la arquitectura del subsistema CHT. Para elaborar uno de estos diagramas será necesario definir:

- los conjuntos de información en forma de datos de entrada, de salida y de la base de datos.
- identificar los programas a desarrollar, documentar los programas por medio de diagramas de flujo de información que contemplen las entradas, salidas y la base de datos e incluir una reseña breve de lo que hace cada programa.
- definir el comportamiento dinámico del sistema por medio de los eventos y las condiciones que lo afectan. Donde un evento es un

hecho que ocurre durante la operación del sistema. Y una condición es un enunciado que al cambiar su estado (de falso a verdadero o viceversa) ocasiona la ocurrencia de un evento [GER83].

-definir las interacciones entre organizaciones, es decir, quien va a proporcionar la información de entrada y hacia donde ira la salida del sistema.

-definir los nombres de las actividades.

1.2.2.2 Diagrama De Estructura. -

Una vez hecha la especificación de los programas y las interrelaciones entre ellos, para cada programa se elabora una estructura Jerárquico-funcional en la que se consideran todos los módulos que constituyen el programa. Esta estructura Jerárquico-funcional queda plasmada en el diagrama de estructura. Para elaborar el diagrama de cada programa se sigue un 'Refinamiento a Pasos'.

En el diagrama de estructura se representan por medio de gráficos las interacciones de los diferentes programas del sistema (estos gráficos contienen solo dos elementos: i) bloques que representan módulos del sistema y ii) flechas que representan el flujo de datos entre los módulos). El diagrama de estructura es una representación gráfica que se puede elaborar para diferentes niveles de abstracción del sistema ('refinamiento a pasos'). En el nivel más general se pueden representar todas las partes que componen al sistema. Para

cada una de estas partes constitutivas del sistema se puede, a su vez, elaborar un diagrama de estructura que comprenda todas las funciones que la integran. También para estas funciones componentes se puede elaborar un diagrama de estructura, si es necesario.

La relación es jerárquica en el sentido de mostrar como es que se organizarán los módulos para controlar la secuencia de ejecución. Funcional para mostrar como se interrelacionarán unas con otras las funciones que desarrollan los módulos del programa. La relación jerárquica solo indica que módulo llamará a que módulo(s) y la relación funcional solo indica como se relacionan los módulos. Es decir, no se indica el orden en que un módulo llama a otros, ni el número de veces que lo llama. Sin embargo, esto se puede deducir fácilmente del pseudocódigo del módulo correspondiente, como se verá en el inciso 1.2.2.3.

Para hacer el diseño del diagrama de estructura se pueden usar tres pasos repetidamente. Estos pasos enuncian los principios básicos de lo que se conoce como "Descomposición Funcional" o "Refinamiento a Pasos".

- 1.- Redactar en lenguaje natural una instrucción que describa una de las funciones del programa.
- 2.- Si la función se puede desarrollar completamente en instrucciones en lenguaje natural en una página aproximadamente, ir al paso 3. En caso contrario descomponer la función en subfunciones.
- 3.- Repetir la descomposición funcional para todas las funciones y subfunciones.

El método enunciado propone pasos a seguir, pero en general depende de la capacidad del diseñador para identificar funciones independientes y decidir cuando es que una instrucción es fácilmente convertible a código.

Para evaluar si un diagrama ha sido bien diseñado se puede echar mano de los siguientes criterios (debe quedar claro que estos criterios son una forma cualitativa de calificar el diseño):

i) El tamaño de los módulos (aunque este queda casi siempre a criterio del diseñador).

ii) El acoplamiento entre módulos (ya que un módulo fuertemente acoplado no es transportable). Transportable (dentro de la misma máquina) quiere decir que se pueda usar el módulo en otro programa, sin necesidad de introducir modificaciones en el código del módulo. Acoplamiento es un reflejo de la interdependencia entre módulos. Si dos módulos están muy acoplados, cuando se modifica el código de uno hay que modificar el código del otro. Se ha hecho una clasificación del tipo de acoplamiento entre módulos [YOU75]. Acoplamiento por datos, el acoplamiento va en proporción directa al número de datos. Acoplamiento por estampado, cuando se usa como argumento un dato estructurado. Acoplamiento por control, un módulo influye en la ejecución de otro módulo a través de una bandera. Acoplamiento por datos externos, dos o más módulos usan la misma estructura de datos; uno para leer, otro para escribir y otro para modificar un archivo. Entonces un cambio en la estructura de los datos externos origina cambios en el código de los módulos [GER83]. Acoplamiento por datos globales, cuando se comparte una área global de memoria. Acoplamiento

por contenido, se presenta solo en lenguaje ensamblador, un módulo puede modificar la secuencia de instrucciones de otro módulo.

1.2.2.3 Detallado De Módulos. -

A partir del diagrama de estructura se hace la especificación de cada módulo en forma más concreta por medio de pseudocódigo estructurado. El pseudocódigo es un lenguaje intermedio entre el lenguaje natural y el código de programación. Son instrucciones en lenguaje natural enmarcadas por palabras clave del código de programación. Es estructurado por presentar las características de la programación estructurada. Se podría abundar mucho sobre este tema, pero no es el objetivo de este trabajo, por lo tanto solo se mencionarán a continuación algunos aspectos relevantes:

"La programación estructurada esta basada en un teorema matemático probado, dicho teorema establece que cualquier programa puede ser escrito utilizando solamente tres estructuras básicas de control simples: i) la secuencia, ii) la decisión o alternativa y iii) la repetición" [GER83]

Donde la secuencia consiste simplemente en enunciar una tras otra las instrucciones necesarias.

La decisión o alternativa permite la ejecución o no de una o un grupo de instrucciones dependiendo de alguna condición. La decisión se da en varias formas: simple, doble y múltiple.

La repetición, como su nombre lo indica, permite la ejecución de una o un grupo de instrucciones un número determinado de veces. Ver apéndice de estructuras básicas.

El detallado de cada módulo quedará especificado en un documento que se llama Diseño de Programa de Computadora (DPC), que contendrá también el diagrama de estructura del programa. En esta etapa es donde se especifican los datos que cada módulo usa, tanto los internos como los globales y los externos.

Para especificar los procesos se puede utilizar un Lenguaje para Diseño de Procesos (LDP) [RODBI], que es un pseudocódigo que cumple con todas las características mencionadas anteriormente (de programación estructurada). Ver apéndice de estructuras básicas.

Una forma cualitativa de evaluar la construcción de un módulo es a través de los siguientes criterios:

i) La cohesión de los módulos. Donde cohesión es la independencia de la función que realiza un módulo, con respecto a las funciones que realizan los demás módulos. Hay varios tipos de cohesión: Cohesión por función, cuando todos los elementos de un módulo contribuyen a ejecutar una y solo una función. Cohesión por secuencia, los elementos ejecutan tareas donde los datos que produce una función son la información que recibe otra. Cohesión por comunicación, cuando un módulo realiza varias funciones que usan la misma estructura de datos. Cohesión por procedimiento, donde varias funciones se han ensamblado en un módulo único porque el control fluye de una función a otra. Cohesión por coincidencia temporal, varias funciones aparecen juntas

porque sus elementos están relacionados en el tiempo. Cohesión por clase: varias funciones son realizadas en un módulo porque se cree que pertenecen a una misma clase o categoría. Cohesión por coincidencia: simplemente se encuentran varias funciones en un módulo sin justificación alguna.

ii) La parsimonia de los módulos. Esto es, los módulos deben ser hechos para cumplir de la mejor manera posible con los requerimientos que les corresponden, sin pretender hacer módulos que sobrepasen los requerimientos del usuario.

Dado que las especificaciones en pseudocódigo se pasarán a un lenguaje de programación, es necesario hacer una selección del lenguaje con el que se considere más adecuado desarrollar el sistema. Para realizar la selección se puede hacer un análisis formal de los lenguajes de programación disponibles en la máquina que se usará. El análisis contemplará dos criterios: uno administrativo y otro técnico. El criterio administrativo considerará tanto el costo como el tiempo de desarrollo; además, el mantenimiento durante la vida útil de los programas. El criterio técnico considerará primeramente: la representación de datos, las estructuras de control y la programación de sistemas. Considerará también lo siguiente: i) transportabilidad, ii) capacidad de documentación, iii) el compromiso entre tiempo y espacio y iv) la facilidad para ligar con módulos ensamblador y v) que los programas se puedan segmentar.

1.2.2.4 Índice De Un Diseño De Programa De Computadora. -

Siguiendo la metodología del enfoque sistémico se ha elaborado una guía denominada: 'Guía para el Diseño de Programas de Computadora' [ROD83]. La guía tiene como objetivo proporcionar una serie de normas y lineamientos que permitan producir un documento llamado 'Diseño de Programa de Computadora' (DPC), en donde se reportan los resultados de la etapa de diseño de un programa de computadora. [ROD83]

El índice de esta guía se muestra en la tabla 1-6. En el se puede observar cual debe ser el contenido de un DPC para cualquier programa que se elabore para el sistema de programas del Centro Nacional de Control de Energía, como se podrá observar, contiene rasgos particulares que surgieron de las mismas necesidades del sistema. En el capítulo 3 se muestra un ejemplo de desarrollo de un DPC y se explica en que consiste cada punto del índice.

EL INDICE DEL DFC DE UN PROGRAMA.

El índice de un DFC deberá contener los siguientes encabezados de sección:

1. PANORAMA GENERAL.
 - 1.1 INTRODUCCION.
 - 1.2 OBJETIVO.
 - 1.3 RESUMEN FUNCIONAL.
2. REFERENCIAS.
3. DIAGRAMA DE ESTRUCTURA.
4. DIAGRAMA DE FLUJO DE INFORMACION.
5. DESCRIPCION DEL PROCESO DEL PROGRAMA.
 - 5.1 (COMPONENTES DEL PROGRAMA).

PARA CADA COMPONENTE DEL PROGRAMA QUE APARECE EN EL DIAGRAMA DE ESTRUCTURA INCLUYA LO SIGUIENTE:

 - RESEÑA DEL PROCESO.
 - DETALLE DEL PROCESO.
 - VARIABLES EXTERNAS.
 - A) ENTRADA.
 - B) ENTRADA-SALIDA.
 - C) SALIDA.
 - D) PROGRAMAS EXTERNOS.
 - VARIABLES GLOBALES.
 - A) ENTRADA.
 - B) ENTRADA-SALIDA.
 - C) SALIDA.
 - VARIABLES LOCALES.
 - A) ARGUMENTOS.
 1. ENTRADA
 2. ENTRADA-SALIDA.
 3. SALIDA.
 - B) INTERNAS.
 - PARAMETROS.
 - A) GLOBALES.
 - B) LOCALES.
 - ESTRUCTURAS DE ALTO NIVEL DE LOS DATOS LOCALES.
 - 5.2 TOTAL DE LINEAS DE PSEUDOCODIGO.
6. INTERFASES EXTERNAS.
 - 6.1 VARIABLES EXTERNAS.
 - 6.2 MATRIZ DE VARIABLES EXTERNAS.
 - 6.3 PROGRAMAS EXTERNOS.
 - 6.4 MENSAJES Y ALARMAS.
 - 6.5 CONDICIONES DE ACTIVACION Y TERMINACION.
 - 6.6 ESTRUCTURAS DE ALTO NIVEL DE LOS DATOS EXTERNOS.

Tabla 1-6

7.	VARIABLES Y PARAMETROS GLOBALES.
7.1	VARIABLES GLOBALES.
7.2	PARAMETROS GLOBALES.
7.3	MATRIZ DE VARIABLES Y PARAMETROS GLOBALES.
7.4	ESTRUCTURAS DE ALTO NIVEL DE LOS DATOS GLOBALES.
8.	EQUIPO Y LENGUAJE.
9.	RECUPERACION DE FALLAS.
10.	CERTIFICACION DE CALIDAD.
10.1	MATRIZ DE REQUERIMIENTOS.
11.	GLOSARIO.

Tabla 1-6 (Continuación)

1.2.3 Fase De Desarrollo. -

En esta fase se hará la codificación de los módulos que forman los programas. Se verificará que todos y cada uno de los módulos funcionen en la forma correcta. Se hará la integración de los módulos para formar programas y la integración de los programas para dar forma al sistema. Esta fase se ha dividido en tres etapas para ser ejecutada; estas son: Codificación, Integración y Pruebas, mismas que se describen a continuación.

1.2.3.1 Codificación. -

En esta etapa se hará la traducción de las funciones especificadas en pseudo-código a código ejecutable. Cuando se hace la traducción de una instrucción en pseudocódigo estructurado a un lenguaje de alto nivel, hay que tener cuidado ya que no siempre existen las estructuras básicas en los lenguajes de alto nivel. Si se

da este caso se podrán considerar dos opciones: una es hacer la emulación de las estructuras básicas con las instrucciones disponibles, esto presenta la desventaja de que se hace más extenso el programa y más laboriosa la codificación. La otra opción es utilizar un preprocesador. El preprocesador es un programa que extiende la sintaxis del lenguaje de programación para que el usuario pueda usar las estructuras básicas. Ver figura 1-2. Presenta varias ventajas, entre ellas destacan:

- Se eliminan las violaciones que se cometían a las estructuras básicas con el lenguaje en su forma original, ya que con la ayuda del preprocesador se pueden usar las estructuras básicas.
- Hay menos diferencias entre el diseño y la codificación.
- Se hace más sencilla la traducción de pseudocódigo a código, con lo que aumenta la confiabilidad y se reduce la probabilidad de cometer errores de codificación.
- Se hace más sencilla la labor de mantenimiento.

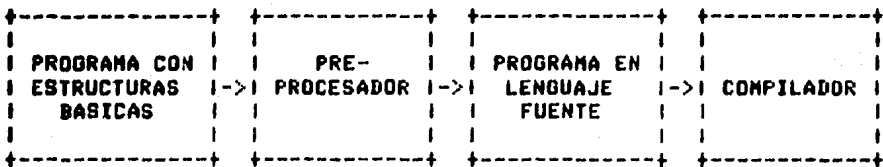


Figura 1-2

Para realizar la codificación se contemplan dos enfoques. Uno considerado como radical, en el que se diseña un programa, se codifica y se prueba enseguida. El otro enfoque, considerado más conservador, establece que se diseñen todos los programas y después se codifiquen y se prueben como sistema. Algunos puntos que pueden servir de guía para decidir cual enfoque se debe aplicar al codificar son: el tiempo con el que se cuenta para desarrollar el sistema; que tanto conocen los diseñadores el sistema; la labor que representa el elaborar los programas, etc. Una vez hecha la codificación, resta verificar que los módulos que han sido modificados y probados cumplan con las normas de programación establecidas para la codificación de los programas del sistema. Estas normas de codificación para el lenguaje de programación, son un conjunto de especificaciones que definen los encargados de Control de Calidad en el proyecto, para homogeneizar la forma de codificar. Las especificaciones sirven para indicarle al programador como es que debe estar presentado el programa, en cuanto a encabezados, declaración de variables, comentarios, etc. Las normas de ninguna manera pretenden coartar la libertad del programador cuando este desarrolle su trabajo, más bien, son guías para homogeneizar el estilo de programación en todos los productos terminados.

El Control de Calidad del proyecto tiene como objetivo verificar que el desarrollo de los productos vaya cumpliendo con los requerimientos que se han establecido. Para esto puede dictar una serie de normas sobre las diferentes fases de desarrollo del proyecto para ir supervisándolo. Las normas son una de las políticas del Control de Calidad a lo largo del proyecto. Se pueden dar otras, como

son: reuniones periódicas con el usuario para constatar que se vayan cumpliendo los requerimientos, acciones correctivas, etc.

Dado que a lo largo del proyecto se generan varios tipos de documentos y que las normas deben regir el desarrollo del proyecto, se dictan diferentes normas para la elaboración de cada tipo de documento y para la elaboración de código. Los criterios generales que contemplan las normas son:

- Estructura. Especifica claramente la estructura que debe tener el documento (o el código); esto es, la organización del material que contiene el documento, los títulos que debe llevar, los encabezados, donde y cuando van las descripciones, los diagramas que debe tener, si es necesario un glosario, etc.
- Estilo. Recomendaciones sobre el estilo general que se desea que tenga el documento.
- Claridad. Se desea que la información que contienen los documentos sea rápidamente accesada, que sea comprensible y que sea fácil de actualizar; para esto es necesario que la norma dirija la elaboración del documento en forma clara y que la persona encargada de elaborarlo, lo haga también en forma clara.
- Documentación. Mencionar la documentación que fue necesaria para elaborar este documento, las referencias, la bibliografía, las certificaciones del control de calidad, etc.
- Uso. Especificar que características técnicas debe tener la persona que usará el documento o programa, las condiciones que se deben cumplir antes de ejecutar el programa, sus interfaces con otros programas, bajo que condiciones de falla es posible

recuperar el programa, en que lenguaje se desarrollo el código, en que máquina y en que medio se almacena el documento o código.

- Usos prohibidos. Restringir el uso de determinadas palabras en los documentos y el uso de algunas estructuras de control en el código.

Ya que los módulos han sido entregados, se puede verificar que cumplan con las normas en una de las dos formas siguientes: mediante la revisión del código por parte del equipo de Control de Calidad o a través de un programa auditor de código. La segunda opción es más viable. Sobre todo cuando se van a auditar todos los programas de un sistema grande.

Ya que los módulos han sido auditados pasarán a formar parte de una Biblioteca de Soporte de Programas (BSP). Esta biblioteca es un sistema que sirve para controlar y mantener toda la programación y documentación relacionada con el proyecto.

1.2.3.2 Integración. -

Hay dos tipos de integración: incremental y no incremental. En la primera, la validación de los nuevos módulos entregados se hace agregándolos con módulos ya probados e integrados con anterioridad. En el segundo tipo de integración, la validación se hace para cada módulo, simulando los módulos con los que el primero interactúa. Este procedimiento resulta siempre más laborioso que el primero.

Para realizar la integración de los programas del sistema hay que tomar en cuenta la manera como se combinan los módulos y el diseño de pruebas para detectar errores. Una clasificación de errores puede ser como sigue: error de codificación, de análisis, de especificación y de diseño. En esta etapa básicamente se trata de encontrar errores de codificación; aunque no disminuye la probabilidad de detectar cualquiera de los otros tipos de errores.

Finalmente lo que hay que tener presente es que el hecho de detectar los errores lo más pronto que se pueda hace que disminuya el costo de la programación.

1.2.3.3 Pruebas De Alto Nivel. -

En esta etapa se desea detectar errores del tipo: de análisis, de especificación o de diseño. Para detectar los errores se hacen varios tipos de pruebas; entre estos están: las pruebas funcionales, de implantación, de sistema y de aceptación.

Las pruebas funcionales son para encontrar errores de análisis y de especificación que pudieran haberse cometido en cualquiera de las etapas de análisis y/o especificación de requerimientos. Se trata de encontrar errores sobre la forma de trabajar del programa; es decir, sobre la función que debe cumplir según lo especifican los requerimientos.

Estas pruebas se generan a partir de los documentos de Reporte de Análisis y Requerimientos de Programas de Computadora. De las pruebas funcionales se emiten los documentos Planes de Pruebas (PLP) y

Procedimientos de Pruebas (PRP). En la tabla 1-7 se muestra el índice de un PLP y en la tabla 1-8 se muestra el índice de un PRP, en el capítulo 4 de esta tesis se muestran ejemplos de estos documentos.

- 1) Identificación.
- 2) Análisis.
- 3) Caracterización de salidas.
- 4) Caracterización de entradas.
- 5) Reseña de la prueba.
- 6) Casos de prueba.

Tabla 1-7

1. Panoramas general.
2. Referencias.
3. Requerimientos de operación.
 - 3.1 Equipo.
 - 3.2 Programas.
 - 3.3 Bases de datos.
 - 3.4 Sistema operativo.
 - 3.5 Entradas.
 - 3.6 Salidas.
 - 3.7 Métodos.
 - 3.8 Personal.
4. Procedimientos de Pruebas de Aceptación.

Esta sección se divide en subsecciones, una para cada prueba y cada prueba contendrá las siguientes subsecciones:

 - Nombre de la prueba.
 - Reseña.
 - Condiciones.
 - Requerimientos satisfechos.
 - Procedimientos de prueba.

Tabla 1-8

Las pruebas de implantación, se diseñan con el fin de encontrar errores de especificación, de concepto y de diseño en el ambiente real de trabajo del sistema; esto es, con la máquina y su equipo físico en general (el sistema operativo, las interfases, etc.).

Las pruebas de sistema se hacen para comprobar que se ha cumplido con los objetivos que se plantearon al inicio del sistema en el Documento de Objetivos. Además se debe hacer una revisión exhaustiva de los manuales del sistema: el del usuario, el de operación y el de procedimientos. Para aclarar con el usuario todas las dudas y dejar los documentos en una forma completamente clara para él.

Las pruebas de aceptación son las que se deben de realizar para comprobar que el producto terminado cumple con las características del que fue contratado. La ejecución de estas pruebas, en general, es bueno encargarlas a terceras personas para que haya imparcialidad absoluta.

1.2.4 Fase De Operación Y Mantenimiento. -

En esta fase se encuentra ya el sistema operando. Es entonces, en el trabajo diario, cuando se notarán los aciertos o los errores que se hayan cometido en cualquiera de las fases precedentes.

Por lo que toca al mantenimiento de sistemas de cómputo, suele no quedar muy claro a veces que es lo que se entiende por mantenimiento. Y es que con el sentido tradicional que se ha usado esta palabra, se asocia de inmediato con partes electrónicas o mecánicas. Entonces

mantener significa cambiar partes desgastadas de un sistema, o ajustar partes que se han deteriorado por el uso frecuente, e inclusive sustituir partes completas por otras partes nuevas que realizan exactamente la misma función. Pero un programa no sufre este tipo de desgaste, mecánico o electrónico, ni necesita que se le ajuste, etc. Más bien el mantenimiento en programas de computadora va relacionado con tres acciones que sirven para conservar un programa en uso. Dichas acciones son: i) corrección de problemas latentes; ii) perfeccionamiento de programas y mejoras a la documentación y iii) cambios adaptivos al medio ambiente de la programación.

Estas tres acciones también se pueden empezar a aplicar desde las fases de Definición de Requerimientos y Desarrollo de Programas para evitar en lo posible errores que se llegarían a detectar solo hasta la fase de pruebas.

El operador en su trabajo diario de interacción con el sistema será el que de pautas para llevar a cabo las acciones mencionadas.

C A P I T U L O I I .

Organización del Sistema Eléctrico Mexicano.

2.0 ORGANIZACIÓN DEL SISTEMA ELÉCTRICO MEXICANO.

A lo largo de este capítulo se hablará en forma breve de los centros de control de energía, de como está estructurado el sistema eléctrico mexicano y de los programas de aplicación que se están desarrollando para el Centro Nacional de Control de Energía. Estos programas se están elaborando en conjunto para formar un sistema. Se explica cuáles son los programas y en que consisten. Se menciona la ubicación del subsistema Coordinación Hidro Térmica y de la tarea Ajuste Interactivo de Generación dentro de la arquitectura total del sistema de programas. De esta forma queda ubicada la tarea (Ajuste Interactivo de Generación) que será desarrollada como programa

ejemplo.

2.1 Prefacio.

En México la energía eléctrica es proporcionada como un servicio público y para manejar todo lo relacionado con ella existe un organismo denominado Comisión Federal de Electricidad (CFE). Esta institución tiene como funciones: la programación, el diseño, la construcción y la instalación del sistema eléctrico de potencia y sus componentes. Estas funciones están asociadas a las actividades de generación, transmisión, distribución y venta de energía eléctrica.

(SAD81)

2.2 Características Relevantes Del Sistema Eléctrico Mexicano.

2.2.1 Organización. -

En la actualidad el Sistema Eléctrico Mexicano (SEM) se encuentra dividido en ocho áreas. Seis de estas áreas están interconectadas. Las otras dos operan aisladas en las penínsulas de Yucatán y de Baja California. Cada una de estas áreas cuenta con un centro de control, desde este se coordinan las actividades de generación, transmisión, distribución y venta de la energía eléctrica para el área correspondiente. Además se cuenta con un Centro Nacional de Control de Energía (CENACE), desde el cual se controlan los centros de las áreas.

La CFE creó al CENACE para dirigir la operación de los sistemas eléctricos del país. Entre sus funciones están las de operación de instalaciones, manejo de otros recursos como combustibles, escurrimientos hidráulicos y personal de operación. Además es el responsable de la seguridad y economía del SEM. [SAUB1]

Los criterios de operación que manejan todos los centros de control de área son continuidad, calidad y economía. Continuidad quiere decir que se proporcione el servicio de una manera ininterrumpida. Calidad, que el servicio proporcionado este dentro de los márgenes seguros de voltaje y frecuencia. Economía, que se genere energía eléctrica y que se distribuya en forma económica.

El desarrollo de los centros de control a dado lugar a inversiones y desarrollos metodológicos, para implantar en línea funciones de seguridad y economía como núcleo vital del centro de control. [SAB81]

2.2.2 Características Eléctricas -

Existen dos niveles de tensión mayor: uno de 400 KV y otro de 230 KV. La potencia total instalada en 1981 era de 14 400 MW, en cuatro tipos de instalaciones, con un total de 150 unidades distribuidas de la siguiente forma:

- a) 77 hidro eléctricas.
- b) 33 de vapor.
- c) 2 de ciclo combinado.
- d) 8 de turbo gas.

En las áreas de operación existe una gran diversidad en sus características de: recursos de generación, comportamiento de la carga, efectos del clima, ubicación de los centros de generación y ubicación de los centros de consumo. Todo esto repercute sobre la estructura de la red eléctrica. Por ejemplo, la Ciudad de México absorbe en promedio el 35% de la demanda máxima diaria, mientras que las hidroeléctricas mayores en el sureste y el suroeste están a 750 y 350 KM de la Cd. de México, respectivamente. [SAD81]

2.2.3 Factores Que Afectan La Red. -

Hay tres variaciones de carga importantes: en la madrugada, al anochecer y cercana a la media noche.

Otro factor importante es la condición aleatoria del agua. El 85% del volumen escurrido suele ocurrir durante el período de lluvias.

El tamaño, el diseño y la complejidad del SEM, así como las grandes concentraciones de energía y la diversidad de recursos, dan origen a problemas complejos, tanto en la operación de la red, como en la concepción del esquema de control de la misma. Se deben hacer una serie de consideraciones especiales para manejar al mismo tiempo las restricciones que imponen los problemas de estabilidad y la optimización que se debe hacer de los recursos hidráulicos. [SAD81]

El Centro Nacional de Control de Energía (CENACE), en unión con los centros de área debe afrontar estos problemas. El CENACE tiene la tarea de dirigir la operación de los sistemas eléctricos del país.

El CENACE a su vez esta dividido para ejecutar sus funciones en tres niveles. Cada nivel con sus funciones y sus responsabilidades. La división con mayor jerarquía tiene la autoridad y responsabilidad completa de la operación del sistema eléctrico. La división de segunda jerarquía opera a nivel de las áreas interconectadas y la división de tercera jerarquía trabaja a nivel de operación de ciudad o subárea. [CAL81]

2.3 Desarrollo De Programas.

Se equipará al CENACE y a los demás centros de control con un sistema de programas de cómputo que faciliten la operación del SEM en forma automática. Hay cuatro programas de ese sistema que son de particular importancia; estos son: Pronóstico de Carga (PC), Programa de Cálculo Automático de Intercambios entre áreas (PCAI), Despacho Económico Restringido (DER) y Coordinación Hidro Térmica (CHT). [CAL81]

A continuación se dará una breve descripción de cada uno de estos subsistemas.

2.3.1 Pronóstico De Cargas A Corto Plazo (PC). -

La meta que persigue el subsistema PC es calcular el pronóstico de demanda diaria en intervalos de una a seis horas para:

-las seis áreas que operan el sistema nacional interconectado.

-resiones climatológicas que CFE puede definir dentro de cada área.

Los resultados de Pronóstico de Carga son utilizados por los subsistemas Despacho Económico y Coordinación Hidro Térmica. Se planea que PC sea ejecutado por lo menos una vez al día, ya que al predecir la demanda, se podrá programar la generación de cada una de las unidades del sistema.

2.3.2 Programa De Cálculo Automático De Intercambios (PCAI). -

Bajo este nombre se agrupan dos funciones, la primera es el Cálculo Automático de Intercambios y la segunda es el Tabulador de Intercambios (Los intercambios de que se habla son intercambios de energía eléctrica). [SAN82], [MAR83].

El objetivo de la tarea PCAI es calcular, cada T segundos, bajo condiciones normales de operación del sistema, las Potencias Netas de Intercambio de Cada Área (PNIA) del sistema eléctrico interconectado de CFE.

Los valores calculados de las PNIA's se deberán tener disponibles para que sean transmitidas cada T segundos a los Centros de Control de Área (CCA) correspondientes.

El periodo de ejecución del PCAI, T segundos, será especificado por el operador como un parámetro del subsistema.

La estructura del sistema interconectado de CFE es tal que puede ser operado como dos o más subsistemas separados eléctricamente, por periodos de tiempo relativamente largos. PCAI considerará que un sistema eléctrico estará formado por al menos dos áreas conectadas eléctricamente.

El subsistema PCAI realiza básicamente tres funciones: el manejo de información del subsistema, el cálculo de las potencias netas de intercambio de las áreas y la evaluación de estadísticas.

El Tabulador de Intercambios (TI) deberá facilitar el manejo y actualización de la información de los intercambios programados, intercambios netos programados e intercambios programados pasados.

La función principal del subsistema TI es la de obtener los segmentos de intercambios netos programados. Se habla de segmentos porque después de que se ha obtenido el intercambio neto, este se puede representar gráficamente por medio de una curva, en la que dos puntos de inflexión consecutivos definen un segmento.

2.3.3 Despacho Económico Restringido (DER). -

El objetivo del DER es minimizar el costo total de la generación de la potencia activa, tomando en consideración: [ROM80]

- La demanda total del sistema.

- Pérdidas de transmisión.

- Restricciones de seguridad y economía como:

- flujos en las líneas/transformadores.

- flujos en los grupos de líneas/transformadores.

- reserva rodante del sistema.

El DER produce el punto base óptimo de generación de potencia activa, para cada unidad o planta generadora en el modo de operación despachable y también produce los siguientes factores:

- Factor de participación de cada unidad o planta reguladora de frecuencia.

- Factor de sensibilidad de las áreas por el cambio de demanda total del sistema.

- Factor de sensibilidad de las áreas por el cambio en el intercambio con sistemas externos.

Para calcular la reserva rodante se hace la diferencia entre la capacidad máxima de generación de una unidad y su generación asignada. La suma de las diferencias de cada unidad dentro del sistema es lo que constituye la reserva rodante del mismo. Esto sirve para saber cuánta energía se puede proporcionar en caso de cualquier eventualidad dentro del sistema, por ejemplo, que se tenga que sacar de generación una

unidad por fallas mecánicas.

El DER se inicia partiendo del modelo eléctrico de la red, en el que se incluyen sus restricciones (de la red). Tales restricciones son: los flujos que existen en las líneas, los intercambios que se producen entre áreas, las reservas que existen por áreas y la capacidad disponible. A partir de estas restricciones se construye un modelo matemático que se deberá optimizar.

Este problema quedará planteado en términos de programación lineal y será resuelto con el método de Dantzig-Wolfe. [ROMBO]

2.3.4 Coordinación Hidro Térmica (CHT). -

Es necesario que la operación del Sistema Eléctrico de Potencia (SEP) sea efectuada teniendo en cuenta políticas óptimas de generación a corto plazo. Estas servirán para proporcionar planes de operación preliminares, que serán la base para efectuar un despacho económico del sistema.

El objetivo es establecer un plan horario de generación (o redespacho) para todas las unidades que hay en el sistema, a lo largo de un horizonte de 168 horas, que constituyen una semana. Se deben minimizar los costos de operación de las unidades generadoras.

La optimización se llevará a cabo mediante la asignación programada de unidades ("unit commitment") y estableciendo un plan de generación hidráulica y térmica.

Descripción General.

Este subsistema tiene cuatro actividades primarias, estas son:

1. **Actividad escenario de planeación.** Aquí se definen los diferentes casos de planeación con las características dadas a cada horizonte de planeación.
2. **Actividad preparación de información para ejecución de tareas de CHT.** Se detecta posible incompatibilidad en los datos y se preparan estos para que sean accedados más eficientemente por las tareas de CHT.
3. **Actividad cálculo del Plan de operación.** Controla la ejecución de aquellas tareas de CHT que conducen a la determinación del Plan de operación, a la consulta de resultados y manejo de mensajes.
4. **Actividad resultados.** Se puede actualizar el Plan de operación si los resultados de la actividad 3 son satisfactorios, se pueden obtener reportes y resultados gráficos.

En la figura 2-1 se puede ver el modelo conceptual del subsistema CHT como quedará implementado en la computadora VAX 11/780 [NAVB3]. La explicación de todos los elementos de la figura 2-1 escapa a los objetivos de este trabajo, por lo tanto, solo se explicarán las tareas importantes para obtener el Plan de operación, como se pueden ver en la actividad 3.

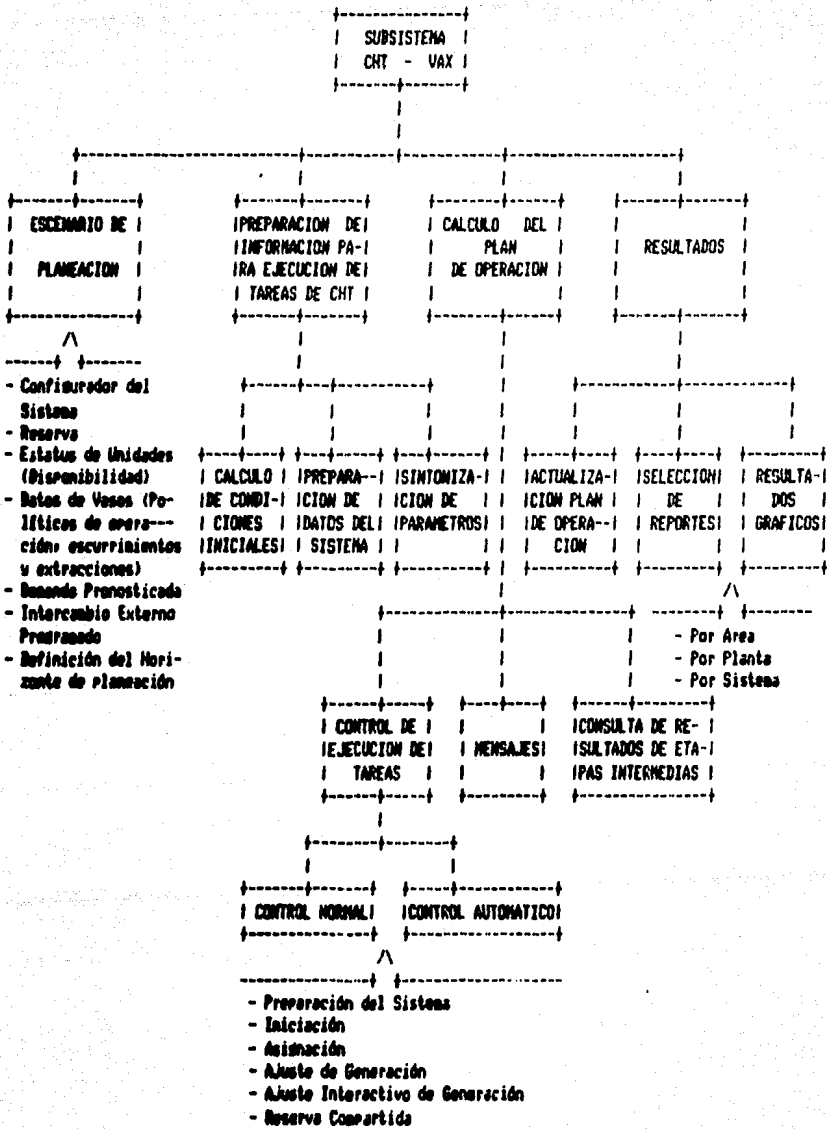


FIGURA 2-1 MODELO CONCEPTUAL CHT

Tareas:

1. Interfaz Hombre-Máquina.
2. Preparador del Sistema.
3. Inicialización Semanal.
4. Asignación de Unidades.
5. Ajuste Automático de Generación.
6. Ajuste Interactivo de Generación.
7. Reserva Compartida.

2.3.4.1 Interfaz Hombre-Máquina (IHM). -

El objetivo es proporcionar al usuario formas de interacción con la máquina (computadora) para definir datos, ejecutar o detener programas y solicitar resultados.

2.3.4.2 Preparador Del Sistema (PS). -

Prepara información de entrada a los diversos programas de CHT, proporciona resultados generales relacionados con la planificación del sistema, tales como cortes de carga y reserva; además identifica violaciones a las restricciones de tiempos mínimos de operación y de paro de las unidades termoeléctricas y verifica incompatibilidad entre

casos de configuración y grupos de generación, para el horizonte de planeación definido por el usuario.

Los tiempos mínimos de operación y de paro se refieren al intervalo de tiempo que como mínimo una unidad generadora debe estar parada u operando.

Casos de configuración. Esto se refiere a las formas en que esta dividida la red eléctrica para su estudio. Es decir las diferentes configuraciones de topología que se pueden elaborar para estudiar y planear sobre el sistema eléctrico.

Grupos de generación son, como su nombre lo indica, generadores que se asocian en un solo conjunto con la finalidad de cumplir con ciertas restricciones que impone la red eléctrica, como son: reserva rodante mínima, y flujos de carga en líneas.

Horizonte de Planeación es el periodo de tiempo para el que se planea la actividad.

2.3.4.3 Inicialización Semanal (IS). -

Deberá programar un plan preliminar de operación factible para todas las plantas hidroeléctricas programables. Este plan será para darle a CHT una aproximación a la operación más económica, para que posteriormente la optimice, o para facilitar al operador la decisión sobre la cantidad de agua que sería conveniente utilizar durante el horizonte de planeación de una semana. IS realiza lo siguiente:

- Conociendo la cota inicial de cada vaso, calcula la cota final en

cada intervalo.

- Calcula la energía a generar por intervalo en las plantas programables.
- Proporciona el número de unidades en línea y el gasto para cada una en cada intervalo y da las descargas de agua para cada vaso en cada intervalo.

2.3.4.4 Asignación De Unidades (AU). -

Esta constituido por Asignación de Unidades Termoeléctricas (AUT) y Asignación de Unidades Hidroeléctricas (AUH).

AUT: tiene como objetivo obtener una asignación económicamente conveniente de unidades termoeléctricas, respetando restricciones de operación de las unidades y de reserva rodante necesaria para el sistema. La asignación es realizada intervalo por intervalo para todo el horizonte de planeación. Se busca primero una solución cercana al óptimo y después se mejora dicha solución con el manejo de optimizadores o programación dinámica.

AUH: a partir de una asignación de unidades termoeléctricas, provee una asignación y despacho de unidades hidroeléctricas procurando que el costo incremental del sistema sea igual en todo el horizonte de estudio.

Costo incremental de la potencia entregada desde la planta 'n' es la derivada de la curva de costo contra potencia entregada al centro de carga y representa cuanto aumenta el costo de producción total si

se entrega 1 KW más en el centro de carga, generándolo junto con el incremento de pérdidas de transmisión que esto provoca, desde la planta 'n'. [MAR]

2.3.4.5 Ajuste Automático De Generación (CHAG). -

Objetivo: Determinar si los esquemas de generación a nivel horario son factibles desde el punto de vista de restricciones de la red eléctrica. [IND83]

Método: Realizar desplazamiento de generación en el punto inicial de operación de unidades termoeléctricas e hidroeléctricas proporcionado por asignación de unidades diario, bajo el marco de un despacho económico para aliviar violaciones a las restricciones impuestas por la red. Se contempla una lógica para efectuar cortes de carga en caso necesario.

2.3.4.6 Ajuste Interactivo De Generación (AIG). -

El programa anterior (CHAG) y el programa Movilidad de Unidades (CHMO) actuando interactivamente con el usuario componen la tarea Ajuste Interactivo de Generación (AIG).

La forma en que funciona la tarea es la siguiente: Después de que se ha obtenido un primer plan preliminar de operación, es posible que ese plan contenga cortes de carga. Entonces el programa Movilidad de Unidades (CHMO) se puede activar para aliviar a través de él algunas restricciones con las que se tuvo que cumplir al obtener el

plan preliminar. Al hacer más flexibles las restricciones se puede volver a ejecutar el Programa Ajuste Automático de Generación (CHAG), con lo cual se obtiene otro plan de operación. Lo más probable es que esta nueva corrida arroje un plan de operación con menos o ningún corte de carga, en cuyo caso el usuario decidirá si ese plan de operación le conviene o si debe volver a ejecutar CHMO para aliviar otras restricciones y obtener un nuevo plan de operación.[FRAB2] Este proceso de interactuar con el usuario y con el programa CHAG se repite tantas veces como el usuario desee.

La implementación de la tarea AIG es la que se mostrará como ejemplo de desarrollo, usando la metodología del capítulo uno. Como se vio, la tarea está compuesta por dos programas y una parte de interacción con el operador. El programa CHAG fue desarrollado previamente como prototipo por el IIE, por lo tanto, solo se mostrará como ejemplo de desarrollo de la tarea AIG, el programa CHMO y la parte interactiva.

2.3.4.7 Reserva Compartida (RC). -

Calcula a partir de información sobre las unidades generadoras, sobre la topología de la red eléctrica y sobre los resultados del plan de operación diario, el valor de la reserva compartida de cada una de las regiones definidas por el usuario para un máximo de 48 intervalos horarios.

Reserva compartida es la reserva de potencia activa máxima con la que puede contar una región, proveniente de ella misma y de las demás regiones, tomando en cuenta las limitaciones a través de los enlaces.
[RPCB1]

Entonces en el inciso 2.3.4.6 se puede encontrar definida la parte del subsistema CHT que es desarrollada en los capítulos 3 y 4 para mostrar como se aplica la metodología que se explicó en el capítulo 1.

C A P I T U L O I I I .

Requerimientos y Diseño de la Tarea

Ajuste Interactivo de Generación.

3.0 FASES DE REQUERIMIENTOS Y DE DISEÑO.

A lo largo de este capítulo se mostrará como ejemplo la forma en que se fue desarrollando la tarea Ajuste Interactivo de Generación (AIG).

Para mostrar el desarrollo de la tarea AIG se usará la metodología que se describió en el capítulo 1.

En este capítulo se abarcará la fase de definición de requerimientos y la fase de diseño.

En el siguiente capítulo se mostrará la fase de desarrollo.

Para no hacer el documento muy voluminoso solo se incluyen partes representativas de la documentación generada. Primero se presentan los requerimientos a nivel general, de ellos se extraen los que son necesarios para el subsistema Coordinación Hidro Térmica (CHT). De los requerimientos del subsistema se obtienen los necesarios para desarrollar la tarea AIG. Una vez identificados los requerimientos de la tarea, se hace el Diseño del Programa de Computadora (DPC). El Diseño de Programa de Computadora es el documento en el que se detalla como se pasan los requerimientos a diseños de programas de computadora. Más adelante, en este capítulo, se presentan detalles del DPC del programa Movilidad de Unidades (CHMO).

3.1 Fase De Definición De Requerimientos.

3.1.1 Nivel General. -

La especificación de requerimientos para todo el sistema quedó plasmada en un solo documento. A lo largo de este documento el grado de abstracción es diferente, pasando de un grado de abstracción alto a uno más bajo. En la tabla 3-1 se muestra la hoja inicial del documento [RPC81].

1. Introducción.

1.1 Identificación.

Este documento establece los requerimientos de los programas de aplicación (PA):

Pronóstico de carga (PC).
Coordinación hidrotérmica a corto plazo (CHT).
Despacho económico (DER).
Cálculo automático de intercambios (PCAI).
Tabulación de intercambios externos (TI).

que el Instituto de Investigaciones Eléctricas desarrolla para el sistema de Información y Control en Tiempo Real del Centro Nacional de Control de Energía.

Tabla 3-1

Primero se especifican con un alto grado de abstracción los requerimientos que deben cumplir los cuatro subsistemas (PCAI, PC, DER y CHT) que integran el sistema completo. En la tabla 3-2 se muestra la forma en que se han especificado estos requerimientos (a nivel general) tratándose en este caso del subsistema CHT.

2. DEFINICION DEL SUBSISTEMA 2: COORDINACION HIDROTERMICA. (CHT)

2.1 ESPECIFICACION FUNCIONAL.

2.1.1 OBJETIVO.

Determinar el plan de operación de las unidades hidro y termoeléctricas que satisfacen la demanda pronosticada, administrando el agua disponible para lograr un costo mínimo de generación termoeléctrica.

2.1.2 GENERALIDADES.

El horizonte de planeación es de siete días. El plan de operación para los dos primeros días se determinará a nivel horario. Para los días restantes se determinará en intervalos de seis horas de amplitud.

El plan de generación será revisado diariamente para tomar en cuenta cambios de:

- a) Pronóstico de demanda.
- b) Niveles de agua en vasos de regulación.
- c) Disponibilidad de componentes del sistema eléctrico.
- d) Intercambios programados con sistemas eléctricos externos.

Tabla 3-2

En la tabla 3-3 se muestra la especificación, a nivel general, de lo que se desea que la tarea AIG ejecute dentro de CHT.

3.1.5 AJUSTE INERACTIVO DE GENERACION. (AIG)

Durante los intervalos criticos en que la tarea Ajuste Automático de Generación (AAG) no haya podido satisfacer la totalidad de las restricciones sin tener que realizar cortes de carga, esta tarea debe permitir la interacción con el usuario para que este modifique a voluntad la asignación de las unidades, para entonces determinar automáticamente la generación de las unidades hidroeléctricas y termoeléctricas asignadas que, a costo mínimo, satisfice la totalidad de las restricciones. El algoritmo debe incluir una lógica para realizar cortes de carga en caso de que la solución así lo requiera.

Tabla 3-3

3.1.2 Nivel Subsistema. -

Para elaborar una especificación de requerimientos a nivel subsistema es necesario tomar todos sus componentes e ir detallándolos. Para esto se cuenta con un modelo conceptual (ver inciso 2.3.4 en el capítulo 2 de esta tesis). Es conveniente mencionar la importancia que tiene el modelo conceptual, entre sus ventajas se tienen: ayuda a dividir el subsistema para facilitar su elaboración; hace que sea más fácil entender las funciones que realiza el subsistema; las revisiones son más eficaces ya que se detecta más claramente tanto a las funciones como a su interacción con el medio. A través del trabajo sobre el modelo conceptual (análisis y refinamiento) se llegan a establecer los requerimientos para el subsistema. Ya que se tienen los requerimientos, se define la forma

en que estos pasarán a convertirse en programas (arquitectura) y esto queda detallado gráficamente en el diagrama de estructura del subsistema (ver figura 3-1).

En el diagrama de estructura, figura 3-1, se pueden observar los programas que llevan a cabo las tareas que se describen en el capítulo dos para el subsistema CHT, además de muchos otros que quedan fuera del alcance de este trabajo. Para cada uno de estos programas se hizo una definición de requerimientos desde el nivel subsistema.

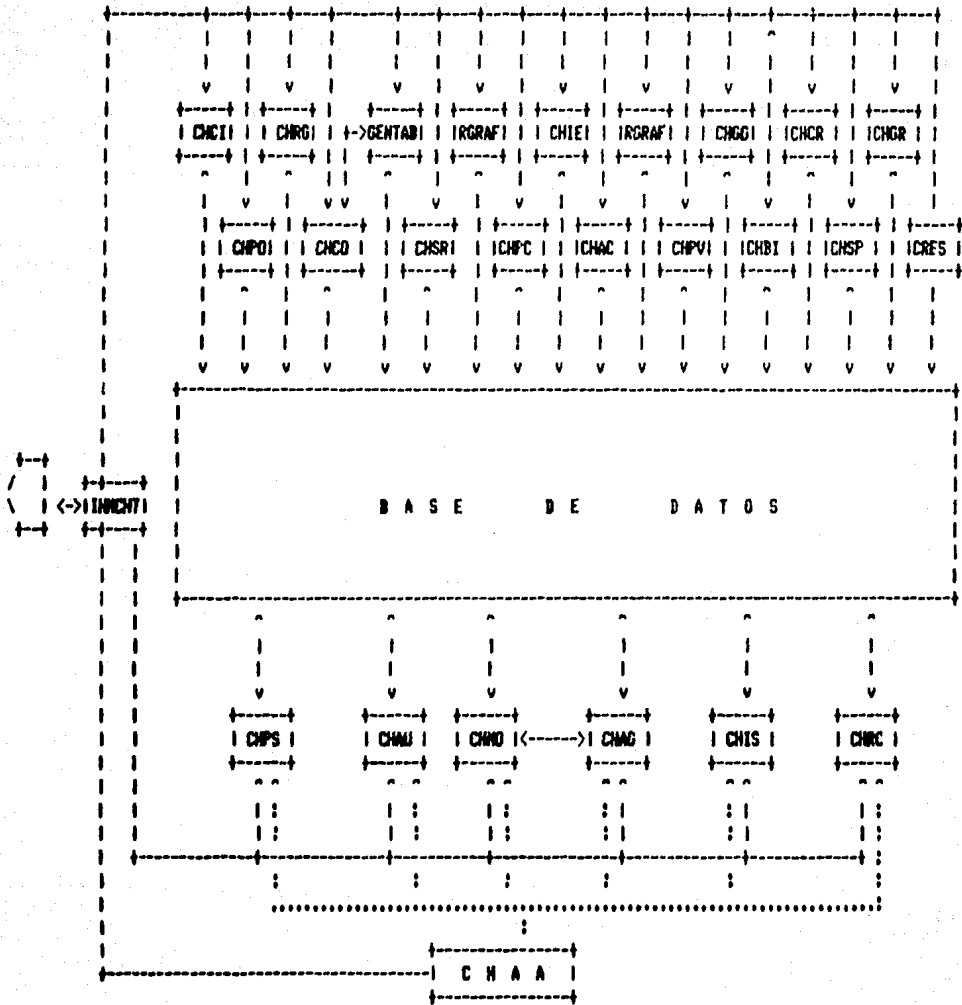


FIGURA 3-1 DIAGRAMA DE ESTRUCTURA DE CNT.
 (RELACION PROGRAMAS - INTERFAZ HOMBRE MAQUINA - BASE DE DATOS)

En la tabla 3-4 se muestran los requerimientos que a la tarea AIG se le asignan desde la perspectiva de observación de CHT. Esta tarea se desarrollará como programa ejemplo en este y los siguientes capítulos.

Se debe hacer un análisis de factibilidad de estos requerimientos para adecuarlos al desarrollo de la tarea en el subsistema.

Los requerimientos de la tarea AIG son, en forma breve:

1. Cambios al esquema de generación.
 - 1.1 Mover el punto base de operación de las unidades que están a generación fija.
 - 1.2 Fijar la generación de las unidades que originalmente se encuentren en modo libre de operación.
2. Asignación o desasignación de unidades.
 - 2.1 Asignar o desasignar unidades termoeléctricas por medio de un despliegado.
3. Alteración de límites de restricciones impuestos a:
 - 3.1 Líneas.
 - 3.2 Transformadores.
 - 3.3 Grupos de líneas y transformadores.
 - 3.4 Límites de reserva en grupos de generadores.

3.5 Requerimientos de la Tarea Ajuste Interactivo de Generación (AIG).

3.5.1 Especificación funcional de AIG.

3.5.1.1 Objetivo.

La tarea AIG tiene como objetivo principal el siguiente:
Dar al operador la opción de tratar de eliminar cortes de carga, en aquellos intervalos críticos para los cuales la tarea AAG arrojó una solución con cortes.

3.5.1.2 Modo de uso.

3.5.1.2.1 AIG es una tarea del subsistema CHT, que es activada por el usuario, a través del monitor general de CHT.

Se puede ejecutar para los intervalos críticos que resultaron con cortes de carga en la solución de AAG.

AIG es ejecutada para intervalos-isla individuales.

Los datos que el usuario podrá modificar a través de los despliegues de entrada serán los siguientes.

3.5.1.2.2 Cambios al esquema de generación. El usuario podrá mover el punto base de operación de unidades que se encuentran a generación fija. Podrá también fijar la generación de unidades que originalmente se encontraban en modo libre de operación (despachables).

3.5.1.2.3 Asignación o desasignación de unidades. El usuario podrá asignar o desasignar unidades tarea, de acuerdo a una lista de unidades posibles de asignar o desasignar en el intervalo en cuestión, que le será proporcionada a través de un despliegue.

3.5.1.2.4 Alteración de límites de restricciones: El usuario podrá alterar (modificar, anular) los límites de las restricciones impuestas en líneas, transformadores, grupos de líneas y transformadores, así como los límites de reserva (total y mínima tarea) en los grupos de generadores.

Tabla 3-4

3.1.3 Nivel Programa. -

El programa encargado de ejecutar los requerimientos de AIG se denominó Movilidad de Unidades (CHMO). Para localizar su ubicación dentro del diagrama de estructura del subsistema ver la figura 3-1, abajo del bloque Base de Datos, tercer bloque de izquierda a derecha.

En la tabla 3-5 se muestra el objetivo del programa CHMO, como se podrá observar, se satisface básicamente al inciso 2 de los requerimientos. Para incorporar los incisos 1 y 3, se le incluyó un desplazado que sirve para hacer la asignación o desasignación de unidades y se hizo interactuar el programa CHMO con el programa CHAG.

MOVILIDAD DE UNIDADES

a) Objetivo.

Para un intervalo J determinar dos conjuntos de unidades: unidades promovibles en el intervalo J y unidades renovables en el intervalo J . Cada uno de estos conjuntos se presenta al operador ordenado ascendente según la magnitud de la violación a las restricciones de tiempos mínimos de operación o de paro.

Tabla 3-5

En la figura 3-2 se muestra el diagrama de estructura del programa CHMO. En la tabla 3-6 se describe con prosa como llevará a cabo dicho programa su proceso. Resumiendo, lo que el programa CHMO hará es: Buscar en el intervalo que el usuario le indique, las unidades que estén operando o paradas. Reunir en un grupo a las que estén operando y en otro a las que estén paradas. Para cada unidad dentro de cada grupo, evaluar cuanto le falta para cumplir con el tiempo mínimo requerido de operación (o de paro), en los intervalos adyacentes al solicitado, adelante del intervalo y atrás del intervalo señalado. Finalmente ejecutar el programa CHAG para buscar un nuevo punto de operación.

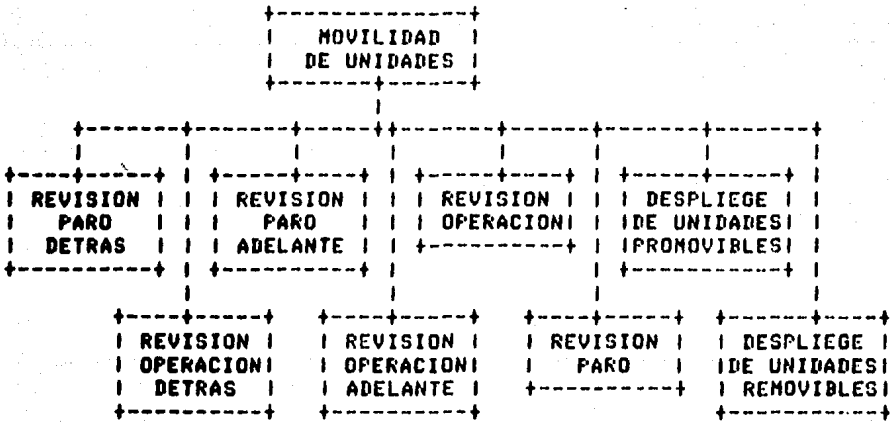


Figura 3-2.

MOVILIDAD DE UNIDADES

f) Proceso.

El conjunto de unidades en el intervalo J esta formado por todas las unidades que son asignables y no estan en operaci3n en dicho intervalo. Este conjunto se denota K en el LDP.

El conjunto de unidades renovibles en el intervalo J esta formado por todas las unidades que son asignables y estan en operaci3n en dicho intervalo. Este conjunto se denota L en el LDP. Evidentemente $K \cap L = 0$.

La instrucci3n 3 del LDP procesa las unidades de K, una por una, y con ellas forma el conjunto K', mismo que ser3 procesado por la instrucci3n 4.

La funci3n 'Revisi3n Paro Detr3s' calcula la magnitud de la violaci3n al tiempo m3nimo de paro en el lapso de paro inmediato anterior al intervalo J, si la unidad u se asignara en el intervalo J. Esta magnitud se mide en n3mero de intervalos y es igual a cero cuando no hay violaci3n; cuando hay violaci3n es igual a la diferencia entre el tiempo m3nimo de paro y el lapso de paro.

La funci3n 'Revisi3n Paro Adelante', realiza lo mismo que la funci3n anterior, con la diferencia que el lapso considerado es el lapso de paro inmediato posterior al intervalo J.

La funci3n 'Revisi3n Operaci3n', calcula la magnitud de la violaci3n al tiempo m3nimo de operaci3n, en el lapso de operaci3n que incluye al intervalo J, si la unidad se asignara en el intervalo J. Evidentemente, si la unidad esta en operaci3n en cualquiera de los intervalos adyacentes al intervalo J, no se incurre en violaci3n al tiempo m3nimo de operaci3n. La magnitud de la violaci3n se mide en n3mero de intervalos y es igual a cero cuando no hay violaci3n; cuando hay violaci3n es igual a la diferencia entre el tiempo m3nimo de operaci3n y el lapso de operaci3n.

La instrucci3n 3.6 calcula la violaci3n total, esto es, la suma de las cantidades calculadas por las tres funciones descritas. La violaci3n total, VTOTAL, es el criterio 3nico para ordenar las unidades. La instrucci3n 4 Primero

Tabla 3-6

ordena las unidades del conjunto L' según el valor de VTOTAL y luego las desfilas al operador.

La instrucción 7 del LHP procesa las unidades de L' una por una, y con ellas forma el conjunto L'' mismo que será procesado por la instrucción 8.

La función 'Revisión Operación Detrás', calcula la magnitud de la violación al tiempo mínimo de operación en el lapso de operación inmediato anterior al intervalo J , si la unidad u se desasignara en el intervalo J . Esta magnitud se mide en número de intervalos y es igual a cero cuando no hay violación; cuando hay violación es igual a la diferencia entre el tiempo mínimo de operación y el lapso de operación.

La función 'Revisión Operación Adelante' realiza lo mismo que la función anterior, con la diferencia que el lapso considerado es el lapso de operación inmediato posterior al intervalo J .

La función 'Revisión Faro' calcula la magnitud de la violación al tiempo mínimo de fero, en el lapso de fero que incluye al intervalo J ; si la unidad se desasignara en el intervalo J . Evidentemente, si la unidad está en fero en cualquiera de los intervalos adyacentes al intervalo J , no se incurre en violación al tiempo mínimo de fero. La magnitud de la violación se mide en número de intervalos y es igual a cero cuando no hay violación; cuando hay violación es igual a la diferencia entre el tiempo mínimo de fero y el lapso de fero.

La instrucción 7.6 calcula la violación total, esto es, la suma de las cantidades calculadas por las 3 funciones descritas. La violación total, VTOTAL, es el criterio único para ordenar las unidades. La instrucción 8 ordena primero las unidades del conjunto L' según el valor de VTOTAL y luego las desfilas al operador.

Tabla 3-6 (Continuación)

A continuación se explican algunos conceptos que se mencionan en las tablas.

Intervalo: cuando se habla de intervalos, se está refiriendo a las

particiones de tiempo en que se divide la semana para propósitos de planeación, en este caso cada intervalo representa una hora.

Intervalo crítico: es un intervalo en el que se ha programado, para alguna región del Sistema Eléctrico Mexicano (SEM), un corte de carga.

Unidades: siempre que se hable de unidades se estará refiriendo a las unidades generadoras termoeléctricas.

Asignable: Una unidad es assignable cuando depende del subsistema CHT si se asigna o no, para que con su generación contribuya a la satisfacción de la demanda.

Una unidad es assignable si en un intervalo dado, forzosamente debe ser asignada y contribuir a la satisfacción de la demanda.

Coordinable: Es una unidad que de ser asignada en el intervalo, su nivel de generación será determinado por el subsistema CHT.

No coordinable es una unidad que de ser asignada en el intervalo, su nivel de generación será preestablecido por el operador.

Operación: Se dice que una unidad está en operación cuando se encuentra generando en el intervalo referido.

Una unidad que no está en operación en un intervalo, está en paro y es una unidad que no está generando en dicho intervalo.

El tiempo mínimo (de operación o de paro) es un número mínimo de horas (intervalos horarios) que una unidad debe estar en ese estatus (operación o paro).

La magnitud de violación se refiere al lapso de intervalos horarios que a una unidad le faltaron para cumplir con el tiempo mínimo (de operación o de paro).

En la tabla 3-7 usando un Lenguaje de Diseño de Programas (LDP)[CRO81] se detalla lo que ha sido explicado por medio de prosa en la tabla 3-6.

NOVILIDAD DE UNIDADES	
d)	LDP
1.	$K = A(J) - O(J)$
2.	$K' = \emptyset$
3.	EJECUTA MIENTRAS ($K \neq \emptyset$)
	A. Sea u E K
	B. $K = K - \{u\}$
	C. $K' = K' \cup \{u\}$
	D. LLAMA Revisión Faro Detrás
	E. LLAMA Revisión Faro Adelante
	F. LLAMA Revisión Operación
	G. $VTOTAL(u) = VPD(u) + VFA(u) + VO(u)$
4.	LLAMA Despliege de Unidades Promovibles
5.	$L = O(J) \cap A(J)$
6.	$L' = \emptyset$
7.	EJECUTA MIENTRAS ($L \neq \emptyset$)
	A. Sea u E L
	B. $L = L - \{u\}$
	C. $L' = L' \cup \{u\}$
	D. LLAMA Revisión Operación Detrás
	E. LLAMA Revisión Operación Adelante
	F. LLAMA Revisión Faro
	G. $VTOTAL(u) = VOD(u) + VOA(u) + VP(u)$
8.	LLAMA Despliege Unidades Removibles
9.	REGRESA

Tabla 3-7

Del uso y la utilidad del LDP se habló en el capítulo 1.

Cada uno de los módulos componentes del Programa CHMD, ver figura 3-2, se especificó por separado. En la tabla 3-8 se muestra el objetivo de la función 'Revisión Paro Adelante'. En la tabla 3-9 se muestra la especificación del modo de uso de esta función. En la figura 3-3 se detalla la ubicación del módulo dentro del conjunto que compone Movilidad de Unidades. En la tabla 3-10 se mencionan las variables de entrada. En la tabla 3-11 se mencionan las variables de salida. En la tabla 3-12 se explica su proceso usando prosa y en la tabla 3-13 usando LDF.

REVISION PARO ADELANTE

a) Objetivo.

Dada una unidad u y un intervalo J , calcular la magnitud de la violación al tiempo mínimo de paro de la unidad, en el lapso de paro inmediato posterior al intervalo J , si la unidad u se asignara en el intervalo J .

Tabla 3-8

b) Modo de Uso.

Esta función es activada por la función Movilidad de Unidades.

Tabla 3-9

c) Diagrama de Estructura.

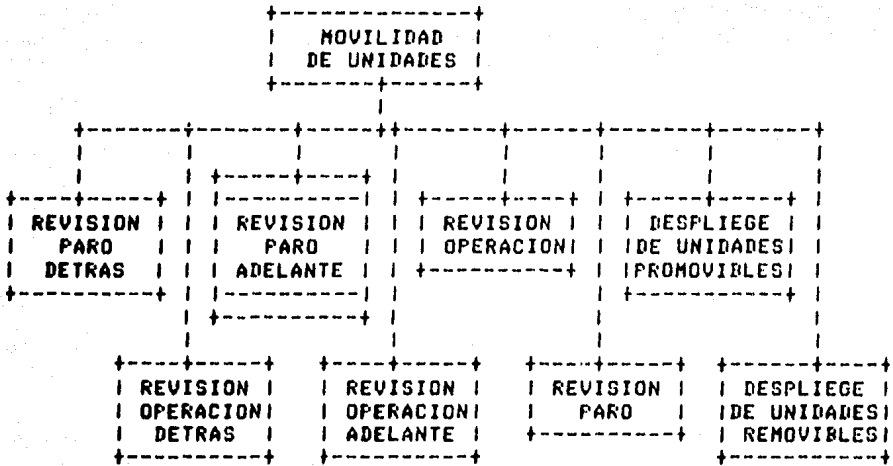


Figura 3-3.

d) Entradas.

- Clave de la unidad u
- Intervalo de base J
- Número de intervalos en el periodo de estudio
- Tiempo mínimo de paro

Tabla 3-10

e) Salidas.

- Magnitud de la violación al tiempo mínimo de paro en el lapso de paro inmediato posterior al intervalo J, si la unidad se asignara en el intervalo J.

Tabla 3-11

REVISION PARO ADELANTE

f) Proceso.

Primersamente (instrucciones 1 y 2 del LDP) se detecta el primer intervalo posterior a J en el que la unidad u no este en paro frio. Si este intervalo es igual a $J+1$ o si la unidad permanece en paro frio durante todos los intervalos $J+1, J+2, \dots, n$ entonces no hay violación al tiempo mínimo de paro de la unidad u , por lo que la magnitud de la violación, VPA, es igual a cero (instrucción 4.A).

En caso contrario, es decir, si el primer intervalo posterior a J en el que la unidad u no esta en paro frio cae entre los valores $J+2$ y n (incluidos límites), entonces la magnitud de la violación (si hay) es igual a la diferencia entre el tiempo mínimo de paro de la unidad, $n(u)$, y el tiempo que la unidad u estaria en paro, es decir $t-J-1$. Si esta diferencia es menor o igual que cero, entonces no hay violación y $VPA=0$.

Tabla 3-12

REVISION PARO ADELANTE

d) LDP

1. $t = J + 1$
2. EJECUTA MIENTRAS (($t \leq n$) y ($u \in F(t)$))
 - A. $t = t + 1$
3. SI (($t \leq n$) y ($t > J+1$)) ENTONCES
 - A. $VPA(u) = n(u) - (t-J-1)$
 - B. SI ($VPA(u) < 0$)
 1. $VPA(u) = 0$
4. SINO
 - A. $VPA(u) = 0$
5. REGRESA

Tabla 3-13

El índice completo de la especificación de requerimientos se muestra a continuación y debe ser llenado completamente para cada uno de los módulos mostrados en la figura 3-2.

- a) Objetivo. El objetivo del módulo.
- b) Modo de uso. Como se usará el módulo dentro del programa.
- c) Diagrama de estructura. Donde se encuentra ubicada la función dentro del diagrama de estructura que define al programa.
- d) Entradas. Las variables de entrada necesarias.
- e) Salidas. Las variables de salida que proporciona el módulo.
- f) Proceso. Descripción por medio de prosa de la función que ejecuta el módulo.
- g) LFD. Descripción por medio de LFD del proceso (del inciso f).

3.2 Fase De Diseño.

Siguiendo la metodología presentada en el capítulo uno se ha elaborado una guía denominada: "Guía Para el Diseño de Programas de Computadora" [RODB3]. La guía tiene como objetivo proporcionar una serie de normas y lineamientos que permitan producir un documento llamado "Diseño de Programa de Computadora" (DPC), en donde se reportan los resultados de la etapa de diseño de un programa de computadora. [RODB3]

El índice que esta guía señala que debe contener cualquier DPC de los que se elaboran para el sistema de programas de cómputo, se muestra en la tabla 1-4, del capítulo 1.

A continuación se explicará brevemente como quedó el DFC de CHMO. No se incluye todo el DFC porque son alrededor de 70 páginas.

En el punto 1 del índice, panorama general, se da un resumen de cual es el alcance del documento y se incluyen las siguientes secciones:

1. Introducción. Se define brevemente el programa, se menciona cual es su ubicación dentro del sistema al que pertenece, su alcance, etc.
2. Objetivo. Se establece el objetivo del documento DFC.
3. Resumen Funcional. Se explica brevemente como funciona el programa. Se incluye la descripción, en un párrafo breve, de como funciona cada uno de los módulos que usa el programa.

El punto 2 del índice, 'Referencias', es para señalar toda la bibliografía que se ha usado para desarrollar el DFC, como capítulos del RFC, manuales usados, etc.

3.2.1 Diagrama De Estructura. -

El diagrama de Estructura, punto 3 del índice, sirve para mostrar los módulos que componen el programa y sus interrelaciones. En la figura 3-4 se muestra la mayor parte del diagrama de estructura del programa CHMO. El módulo principal lleva el mismo nombre del programa (CHMO). En la figura 3-4 se puede apreciar al módulo principal y a los bloques que lo integran. Si se compara este diagrama con el de la figura 3-2 se notará que en esencia son lo mismo. La diferencia es que en la figura 3-2 aparecen los módulos 'Revisión Paso Detrás' y

'Revisión Operación Detrás', mientras que en la figura 3-4 estos se han reemplazado por un módulo único llamado 'Revisión Detrás' ('RVDTRS'). Esto se debe a que en el diseño (basado en los requerimientos), se observó que la función era exactamente la misma, solo que en un caso se analizaba operación y en el otro paro. Se decidió entonces unir las funciones en un solo módulo y pasar el estado a analizar (operación o paro) como parámetro. De esta forma los dos módulos mencionados de la figura 3-2 quedaron unidos en un solo módulo llamado 'Revisión Detrás'. El mismo proceso de reducción se siguió para los módulos 'Revisión Paro Adelante' y 'Revisión Operación Adelante' de la figura 3-2, que quedaron en un módulo denominado 'Revisión Adelante' (REVAD) en la figura 3-4. De la misma manera los módulos 'Revisión Operación' y 'Revisión Paro' de la figura 3-2 quedaron como 'Revisión en el Intervalo' (REVINT) en la figura 3-4. Finalmente los módulos que en la figura 3-2 se mencionan como despliegue de unidades, quedaron unidos en la figura 3-4 como 'DSPLY'. En este último módulo también se incluye la parte interactiva del programa. En la figura 3-4 aparece también un módulo denominado 'LEDATO' y otro llamado 'UTILER'. 'LEDATO' es el que se encarga de hacer la lectura de datos y se desdosa en la parte inferior de la misma figura. El módulo 'UTILER' está constituido por pequeños programas que se desarrollaron para darle facilidades al usuario y hacer la parte interactiva más clara.

En la figura 3-4 (continuación) se puede observar la composición del módulo 'DSPLY', como se verá no solo es la parte de los desplegados de unidades Promovibles (DESPUP) y de unidades removibles (DESPUR), sino son también una serie de utilerías de validación de datos y de interacción con el usuario y los archivos de la Base de Datos (BD). Esto se puede observar en el desplazo de los módulos 'DESPUP' y 'DESPUR' que se muestra en la parte inferior de la figura 3-4 (continuación).

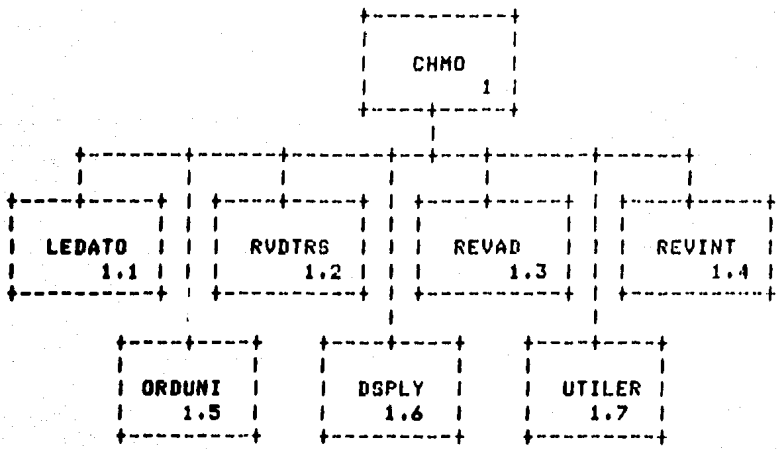


Figura 3-4

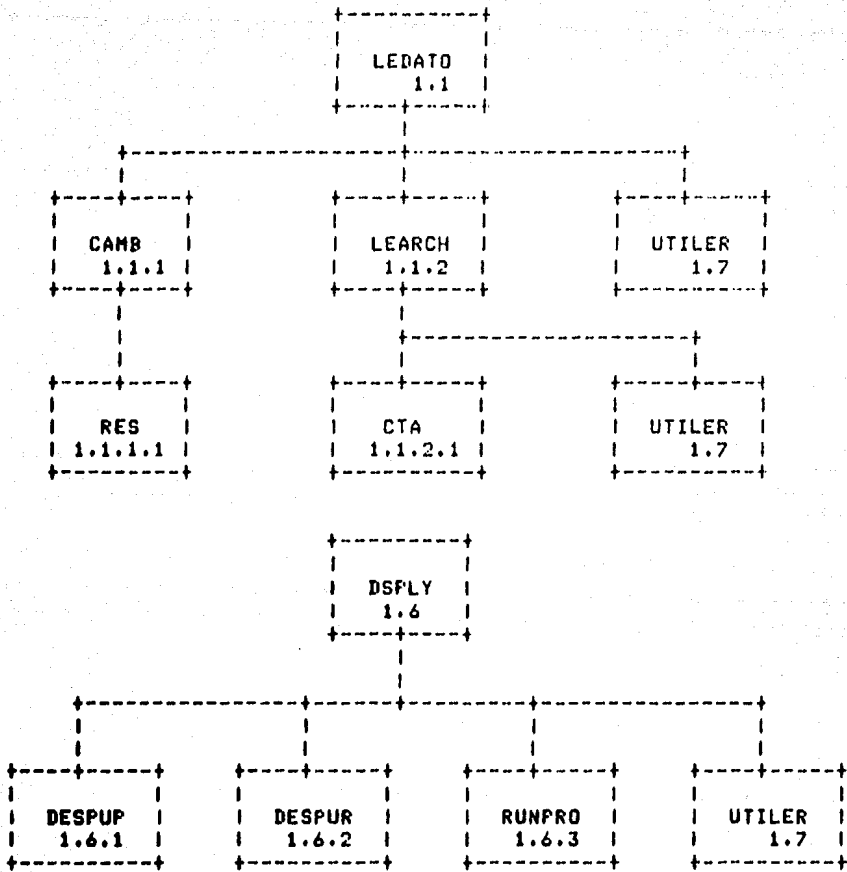


Figura 3-4 (Continuación)

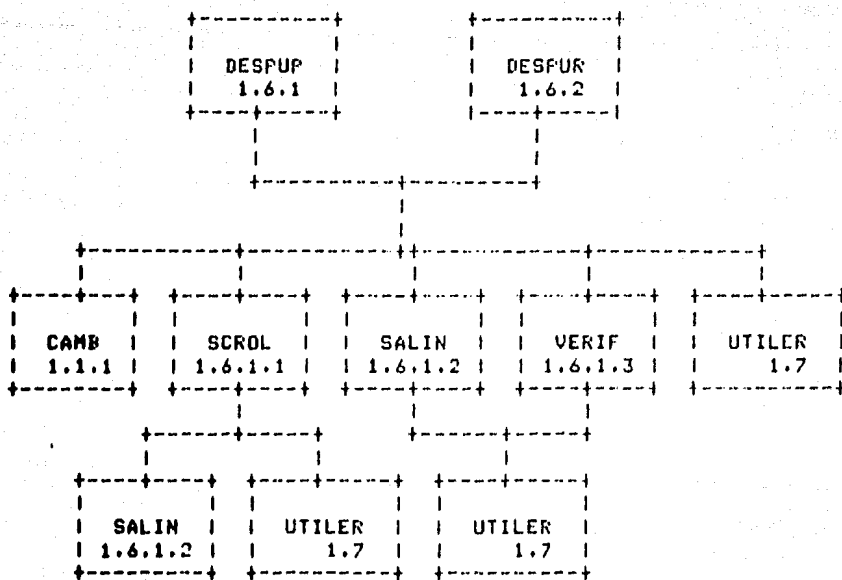


Figura 3-4 (Continuación)

En la figura 3-5 se puede observar el diagrama de flujo de información; punto 4 en el índice del DPC, donde se aprecia claramente como fluyen los datos entre módulos, base de datos y operador.

IHMCHT es la Interfaz Hombre Máquina de CHT. La interfaz es un canal de comunicación entre el subsistema CHT y el operador. Por ejemplo, el operador a través de un desplegado de la IHMCHT puede pedir que se ejecute la tarea AIG.

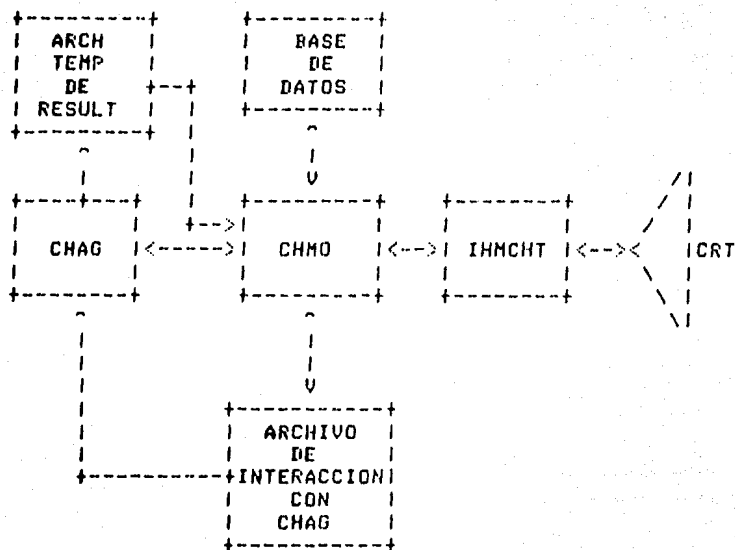


Figura 3-5

3.2.2 Detallado De Módulos. -

El punto 5 del índice del DPC se debe repetir para cada módulo de los detallados en el diagrama de estructura. Por ejemplo, en la tabla 3-14, se detalla el módulo principal CHMO que ya fue ilustrado en la figura 3-4.

A continuación se explican los aspectos que se deben incluir en el detallado de módulos, punto 5 del índice del DPC.

La reseña es una descripción en forma breve de lo que realiza el módulo, Junto con las utilerías que usa, etc.

El detalle del proceso es la especificación de la función reseñada usando pseudocódigo estructurado. Se puede comparar al detallado de la tabla 3-14 con el LDF de requerimientos de la tabla 3-7 para apreciar su parecido.

En cuanto a las variables, estas se dividen en tres tipos: externas, globales y locales. Estos tipos están descompuestos, cada uno, en: de entrada, de salida y de entrada/salida.

Las variables externas son aquellas que están en el medio ambiente que rodea a la tarea AIG y que existen antes y después de que se active la tarea. Por lo tanto AIG las puede usar en cualquier momento, el ejemplo más común son las variables de los archivos de la BD.

Las variables globales son aquellas que se usan en varios módulos del programa, es decir, son comunes a varios de ellos. El sentido de global se refiere a que pueden abarcar a todos los módulos que componen el programa.

Las variables locales son aquellas que se usan solamente en un módulo del programa y que por lo tanto son exclusivas de él. Estas se dividen en argumentos e internas.

El que sean de entrada, de salida o de entrada/salida, para cada módulo, se explica por sí mismo. Como ejemplo se puede ver la descripción de variables del módulo CHMO en la tabla 3-14.

5.1 Componente Movilidad de Unidades (CHMO).

A) Reseña del Proceso.

El componente CHMO es el programa principal, primero obtiene los datos que necesita el programa para correr (con la rutina LEDATO), calcula la magnitud de la violación en que incurre cada unidad (con las rutinas: RVDTRS, REVAD Y REVINT). Llama a una rutina (ORDUNI) que ordena las unidades, las promovibles y las removibles, en orden ascendente según la magnitud de la violación en que haya incurrido cada unidad. Llama a DSFLY para que el operador elija una forma de continuar con el programa. Finalmente le presunta al operador si desea repetir el proceso.

B) Detalle del proceso.

INICIA CHMO

(*EJECUTA TANTAS VECES COMO EL OPERADOR QUIERA*)

COND=CIERTO

EJECUTA MIENTRAS (COND)

HAYUNI=FLASO

(*PARA LEER DE LA BASE DE DATOS*)

LLAMA LEDATO

(*SI HAY UNIDADES SE EJECUTA EL PROCESO*)

SI (HAYUNI) ENTONCES

(*CICLO PARA PENALIZAR A LAS UNIDADES PROMOVIABLES*)

REPITE KONT=1, KNTUPR

LLAMA RVDTRS

LLAMA REVAD

LLAMA REVINT

VIOLTP=K1*VIOLPD+K1*VIOLPA+K2*VIOLOP

FIN(REPITE)

Tabla 3-14

```

(*CICLO PARA PENALIZAR A LAS UNIDADES REMOVIBLES*)
REPITE KONT=1,KNTURM
  LLAMA RVDTRS
  LLAMA REVAD
  LLAMA REVINT
  VIOLTR=K2*VIOLDD+K2*VIOLDA+K1*VIOLFR
FIN(REPITE)

(*PARA ORDENAR LAS UNIDADES PROMOVIBLES*)
LLAMA ORDUNI

(*PARA ORDENAR LAS UNIDADES REMOVIBLES*)
LLAMA ORDUNI

(*SE INICIALIZAN CONTADORES*)
ICOR=0
ILIM=0
ISTA=0
ESNRID=FALSO
(*ENTRADA A LOS DESPLEGADOS*)
LLAMA DSPLY
FIN(SI)

(*PREGUNTAR AL OPERADOR SI DESEA SEGUIR EJECUTANDO CHMO*)
RESP=' '
EJECUTA MIENTRAS (RESP<>'S' Y RESP<>'N')
  ESCRIBE EN PANTALLA:'QUIERE ANALIZAR OTRA ISLA O INTERVALO?'
  LEE DE PANTALLA:RESP
  SI (RESP<>'S' Y RESP<>'N') ENTONCES
    ESCRIBE EN PANTALLA:'RESPONDA SOLO SI O NO'
  FIN(SI)
FIN(MIENTRAS)
SI (RESP='N' O ESNRID) COND=FALSO
FIN(MIENTRAS)

TERMINA CHMO

C)Variables.

-Variables Globales.
de entrada:
KNTUPR
KNTURM.
K1

```

Tabla 3-14 (Continuación)

K2
OPERA
PAROF
TMINOF
TMINPF
TSTAIN

de entrada/salida:

ESNRIO
HAYUNI
ICOR
ILIM
ISTA
VIOLOA
VIOLOD
VIOLOP
VIOLPA
VIOLPD
VIOLPR
VIOLTP
VIOLTR

-Variables Locales.

Internas:

COND:Condición para ejecutar el proceso mientras el
operador quiera.

tipo:lógica.

rango:verdadero o falso.

KONT:Contador en los ciclos de penalización.

tipo:entero.

RESP:Contiene la respuesta del operador.

tipo:carácter.

SUBI:Subíndice.

tipo:entera.

rango:1 a KNTURM.

Tabla 3-14 (Continuación)

Finalmente, para mostrar la interacción de los programas CHMO y CHAG que componen la tarea AIG, se puede ver en la tabla 3-15 una parte del diseño del módulo 'RUNPRO' que se encarga de ejecutar desde CHMO, a solicitud del operador, al programa CHAG.

El punto 6 del índice del DPC se refiere a las interfaces externas; esto es para mencionar todas las variables y programas que son externos; en el sentido que a externo se le dio anteriormente. Para ejemplificar como se ha especificado este punto, en la tabla 3-16 se muestra como se definen algunas variables externas y en la tabla 3-17 se puede ver como se forma la matriz de variables externas.

El punto 7 del DPC es igual que el anterior, con la diferencia de que ahora las variables especificadas son las globales.

El punto 8 es en cuanto al equipo de cómputo que se usará y al lenguaje de alto nivel en que se codificarán los diseños.

El punto 9 es para especificar bajo que condiciones de fallos se puede recuperar el programa.

El punto 10 es para que se especifiquen las certificaciones de calidad con respecto a los requerimientos cumplidos y para incluir una matriz de requerimientos. En esta última se muestra que módulos cubren los requerimientos del programa de computadora. Ver tabla 3-18 y tabla 3-4.

Finalmente el punto 11 es un glosario de términos.

Con esto se da por terminada la fase de diseño, en la fase de desarrollo se verá como estos diseños se convirtieron a código.

La fase de diseño esta completamente basada en los requerimientos. Para la parte interactiva, que no esta especificada en los requerimientos, lo que se hizo fue un diseño basado en desplazados de las utilerias del sistema de servicios. Sin embargo, el uso de las rutinas no se describe en el pseudocódigo del diseño de los módulos involucrados para no hacerlos perder generalidad. Finalmente hay que mencionar que esta fase de diseño será retroalimentada por la fase de desarrollo.

COMPONENTE QUE EJECUTA OTRO PROGRAMA

A) Reseña del Proceso.

Esta rutina hace uso de las utilerías del 'SYSTEM SERVICE', para ejecutar al programa CHAG como un proceso externo y tomar el control nuevamente cuando este se ha terminado de ejecutar. El programa primero obtiene el nombre de la terminal sobre la que se está corriendo el programa; después crea un 'Mail Box' para comunicarse con el proceso que se va a ejecutar. A continuación obtiene la información necesaria del 'Mail Box' para pasársela a la utilería que crea el programa. En seguida crea el proceso que ejecuta al otro programa. Finalmente espera a que termine el proceso creado para tomar el control nuevamente.

B) Detalle del Proceso.

INICIA RUNPRO

OBTIENE NOMBRE DE LA TERMINAL

SI (ERROR) ENTONCES

ESCRIBE EN PANTALLA: 'ERROR EN EL NOMBRE DE LA TERMINAL.'

TERMINA RUNPRO

FINSI

CREA UN 'MAIL BOX'

SI (ERROR) ENTONCES

ESCRIBE EN PANTALLA: 'ERROR EN EL MAIL BOX.'

TERMINA RUNPRO

FINSI

OBTIENE LA INFORMACION DEL MAIL BOX

SI (ERROR) ENTONCES

ESCRIBE EN PANTALLA: 'ERROR EN CARACTERISTICAS DEL MAILBOX'

TERMINA RUNPRO

FINSI

CREA EL OTRO PROCESO

SI (ERROR) ENTONCES

ESCRIBE EN PANTALLA: 'ERROR EN LA CREACION DEL PROCESO.'

TERMINA RUNPRO

FINSI

ESPERA A QUE TERMINE EL PROCESO CREADO

SI (ERROR) ENTONCES

ESCRIBE EN PANTALLA: 'ERROR AL SALIR DEL PROCESO.'

FINSI

TERMINA RUNPRO

Tabla 3-15

Interfases Externas.

Variables Externas.

CLVEUN:Clave de la Unidad.
tipo:caracter.
dimensión:1 a NMTOTU.

CPCDD:Capacidad de generación por unidad.
tipo:real.
dimensión:1 a NMTOTU.

CORTE:Corte en los nodos que pertenecen a la isla que se analiza.
tipo:real.

EDOCOR:Arreglo que contiene el estado de coordinabilidad por unidad.
tipo:caracter.
dimensión:1 a NMNPE, 1 a NMTOTU.

FECHA:Fecha inicial con la que se cargaron los datos en la BD.
tipo:caracter.

LIMAX:Límite máximo de generación por unidad.
tipo:real.
dimensión:1 a NMTOTU.

LIMIN:Límite mínimo de generación por unidad.
tipo:real.
dimensión:1 a NMTOTU.

PLANTA:Clave de la Planta a la que pertenece la Unidad.
tipo:caracter.
dimensión:1 a NMTOTU.

STAIN:Status por Unidad por Intervalo.
tipo:entero.
dimensión:1 a NMNPE, 1 a NMTOTU.
rango:OPERA, PAROF o Paro Caliente.

Tabla 3-16

MATRIZ DE VARIABLES EXTERNAS

VARIABLE	CLASIFICACION	COMPONENTE	USO
CLUEUN	CONSTANTE	LEARCH	E
CORTE	.	LEDATO	E
EDDCOR	VARIABLE	LEARCH	E
FECHA	CONSTANTE	LEDATO	E
LIMAX	VARIABLE	LEARCH	E
LIMIN	CONSTANTE	LEARCH	E
PLANTA	.	LEARCH	E
STAIN	.	LEARCH	E
THINOP	.	LEARCH	E
THINPF	.	LEARCH	E
TSTAIN	.	LEARCH	E

E=entrada

S=salida

E/S=entrada/salida

Tabla 3-17

REQUERIMIENTO NUMERO	MODULO QUE LO SATISFACE
3.5.1.2.1	CHMO
3.5.1.2.2	DSFLY
3.5.1.2.3	DSPLY
3.5.1.2.4	DSFLY

Tabla 3-18

C A P I T U L O I V .

Desarrollo y Pruebas de la Tarea AIG.

4.0. FASE DE DESARROLLO.

En el presente capítulo se abordará el desarrollo de la tarea Ajuste Interactivo de Generación (AIG) del subsistema Coordinación Hidro Térmica (CHT).

No hay que perder de vista que el objetivo es mostrar como se ha desarrollado la tarea AIG al seguir la metodología que se planteó en el capítulo uno.

Con base en la documentación de los diseños se hace el código. Esta etapa puede generar modificaciones al diseño, sobre todo en la parte de declaración de variables. En este capítulo se presenta

documentación que sirve para ejemplificar como quedó la codificación de los módulos.

Después de la codificación se hizo el plan de pruebas, se presenta aquí documentación sobre las pruebas y sobre la guía que se usó para desarrollarlas.

La última parte fue de procedimientos para ejecutar las pruebas, se presenta documentación sobre los procedimientos y sobre la guía que sirvió para elaborarlos.

La documentación, como se verá a lo largo del capítulo, no es muy extensa, esto es con el fin de no hacer la tesis muy voluminosa.

4.1 Codificación.

En esta etapa se mostrará como se codificaron los módulos del programa Movilidad de Unidades (CHMO).

Para llevar a cabo la codificación se tomó como base el documento Diseño de Programa de Computadora (DPC), ya que en este se encuentran documentados los diseños de los módulos. El proceso de codificación hizo necesario en algunos casos actualizar el diseño, en cuanto a forma, no en contenido.

En la tabla 4-1 se muestra la codificación del módulo principal CHMO.

PROGRAM CHMO

```
C
C*****
C*****          CHMO          *****
C*****
C
C  PROGRAMAS DE APLICACION AVANZADA
C
C  INSTITUTO DE INVESTIGACIONES
C
C  ELECTRICAS
C
C  DIVISION DE SISTEMAS DE POTENCIA
C
C  DEPARTAMENTO DE ANALISIS DE REDES
C*****
C
C  PROGRAMA FUENTE:
C  CHMO.FOR
C  PROPOSITO:
C  ESTE ES EL PROGRAMA PRINCIPAL, CONTROLA LAS LLAMADAS A LOS MODULOS
C  DE LECTURA DE DATOS, DE PENALIZACION DE UNIDADES, PARA ORDENAR LAS
C  UNIDADES Y PARA HACER EL DESPLEGADO DE OPCIONES, ADEMÁS CONTROLA
C  LAS REPETICIONES DEL PROGRAMA CHMO.
C  REFERENCIAS:
C  RPC DE CNT, SECCIONES CORRESPONDIENTES A LAS TAREAS DE
C  AIG Y AAG Y EL DPC DE CHMO.
C  LIMITACIONES:
C  LAS QUE MARCA EL DPC.
C  ACLARACIONES:
C  NINGUNA.
C  NOMBRE Y FECHA DE IMPLEMENTACION:
C  JCPC 03 06 07
C  DESCRIPCION DE IDENTIFICADORES.
C  VARIABLES GLOBALES:
C  DE ENTRADA:
C  KNTUPR:NUMERO DE UNIDADES PROMOVIBLES.
C  KNTURN:NUMERO DE UNIDADES REMOVIBLES.
C  K1:CONSTANTE DE PONDERACION.
C  K2:CONSTANTE DE PONDERACION.
C  OPERA:REPRESENTACION DEL ESTATUS DE OPERACION.
C  PAROF:REPRESENTACION DEL ESTATUS DE PARO FRIO.
C  TMINOP:ARREGLO CON EL TIEMPO MINIMO DE OPERACION POR UNIDAD.
C  TMINPF:ARREGLO CON EL TIEMPO MINIMO DE PARO FRIO POR UNIDAD.
C  TSTAIN:ARREGLO CON EL TIEMPO EN EL ESTATUS INICIAL POR UNIDAD.
C
```

Tabla 4-1

```

C      DE ENTRADA/SALIDA: *
C      VILOA:ARREGLO QUE CONTIENE LA VIOLACION DE OPERACION ADELANTE. *
C      VILOD:ARREGLO QUE CONTIENE LA VIOLACION DE OPERACION DETRAS. *
C      VIOLOF:ARREGLO QUE CONTIENE LA VIOLACION DE OPERACION. *
C      VIOLPA:ARREGLO QUE CONTIENE LA VIOLACION AL PARO ADELANTE. *
C      VIOLPD:ARREGLO QUE CONTIENE LA VIOLACION AL PARO DETRAS. *
C      VIOLPR:ARREGLO QUE CONTIENE LA VIOLACION AL PARO. *
C      VIOLTP:VIOLACION TOTAL PROMOVIBLE POR UNIDAD. *
C      VIOLTR:VIOLACION TOTAL REMOVIBLE POR UNIDAD. *
C      VARIABLES LOCALES: *
C      INTERNAS: *
C      COND:CONDICION DE EJECUTA MIENTRAS. *
C      KONT:INDICE. *
C      SUBI:SUBINDICE. *
C      UNIORD: BANDERA PARA UNIDADES PROMOVIBLES O REMOVIBLES. *
C      DESCRIPCION DE LOS MODULOS USADOS. *
C      DSPLY:NUESTRA EL DESPLEGADO DE OPCIONES AL OPERADOR. *
C      LEDATO:LEE E INICIALIZA LOS DATOS NECESARIOS PARA TODOS LOS MODULOS *
C      ORDUNI:EFECTUA EL ORDENAMIENTO DE LOS INDICES DE LAS UNIDADES. *
C      REVAD:REVISLA LOS INTERVALOS POSTERIORES AL INTERVALO BASE. *
C      REVINT:REVISLA LOS INTERVALOS ADYACENTES AL INTERVALO BASE. *
C      RVDTRS:REVISLA LOS INTERVALOS ANTERIORES AL INTERVALO BASE. *
C*****
C      INTEGER      KNTUPR , KNTURN , KONT ,
>      OPERA ,
>      PAROF ,
>      SUBI ,
>      THINOP(150) , THINPF(150) , TSTAIN(150) ,
>      VIOLOA(150) , VIOLOD(150) , VIOLOP(150) ,
>      VIOLPA(150) , VIOLPD(150) , VIOLPR(150)
C      LOGICAL      HAYUNI ,
>      UNIORD
C      REAL      K1 , K2 ,
>      VIOLTP(150) , VIOLTR(150)

COMMON / HAYUNI / HAYUNI
COMMON / K1 / K1
COMMON / K2 / K2
COMMON / KNTUPR / KNTUPR
COMMON / KNTURN / KNTURN
COMMON / OPERA / OPERA
COMMON / PAROF / PAROF
COMMON / THINOP / THINOP
COMMON / THINPF / THINPF
COMMON / TSTAIN / TSTAIN
COMMON / VIOLOA / VIOLOA
COMMON / VIOLOD / VIOLOD
COMMON / VIOLOP / VIOLOP

```

Tabla 4-1 (Continuación)

```

COMMON / VIOLPA / VIOLPA
COMMON / VIOLPD / VIOLPD
COMMON / VIOLPR / VIOLPR
COMMON / VIOLTP / VIOLTP
COMMON / VIOLTR / VIOLTR

```

```

COND=.TRUE.
DO WHILE (COND)

```

```

-----
*PARA TRAER VALORES DE LA BASE DE DATOS*
-----

```

```

CALL LEDATO

```

```

IF (NAYUNI) THEN

```

```

-----
*CICLO PARA PENALIZAR A LAS UNIDADES PROMOVIBLES*
-----

```

```

IF (KNTUPR.NE.0) THEN

```

```

DO KONT=1,KNTUPR

```

```

CALL RVDTRS(PAROF,TSTAIN(KONT),THINPF(KONT),
VIOLPD(KONT),KONT)

```

```

CALL REVAD(PAROF,THINPF(KONT),VIOLPA(KONT),KONT)

```

```

CALL REVINT(PAROF,TSTAIN(KONT),THINOP(KONT),

```

```

VIOLOP(KONT),KONT)

```

```

VIOLTP(KONT)=K1*VIOLPD(KONT)+K2*VIOLPA(KONT)
+K3*VIOLOP(KONT)

```

```

ENDDO

```

```

ENDIF

```

```

-----
*CICLO PARA PENALIZAR A LAS UNIDADES PROMOVIBLES*
-----

```

```

IF (KNTURN-KNTUPR.NE.0) THEN

```

```

SUBI=1

```

```

DO KONT=KNTUPR+1,KNTURN

```

```

CALL RVDTRS(OPERA,TSTAIN(KONT),THINOP(KONT),
VIOLOP(SUBI),KONT)

```

```

CALL REVAD(OPERA,THINOP(KONT),VIOLOA(SUBI),KONT)

```

```

CALL REVINT(OPERA,TSTAIN(KONT),THINPF(KONT),

```

```

VIOLPR(SUBI),KONT)

```

```

VIOLTR(SUBI)=K1*VIOLOA(SUBI)+K2*VIOLOP(SUBI)
+K3*VIOLPR(SUBI)

```

```

SUBI=SUBI+1

```

```

ENDDO

```

```

ENDIF

```

```

-----
*PARA ORDENAR LAS UNIDADES PROMOVIBLES*
-----

```

```

IF (KNTUPR.NE.0) CALL ORDUNI(VIOLTP)

```

Table 4-1 (Continuación)

```

C      -----
C      #PARA ORDENAR LA UNIDADES REMOVIBLES#
C      -----
C      IF (KNTURN-KNTUPR.NE.C) CALL ORDUNI(VIOLTR)
C
C      -----
C      #ENTRADA A LOS DESPLEGADOS#
C      -----
C      CALL DSPLY
      ENDIF
      ENDDO
      ENDICMDO

```

Tabla 4-1 (Continuación)

4.1.1 Normas. -

Es necesario mencionar que existe un documento de normas de codificación, mismo que sirve para homogeneizar la forma en que se codifican los módulos de todos los programas. En este documento se encuentran redactadas las normas que se deben usar para codificar los módulos de los programas (ver apéndice B). Estas normas tienen la finalidad de homogeneizar la forma de presentación de los programas codificados. Además establecen una serie de reglas sobre el uso de las instrucciones disponibles en el lenguaje de alto nivel usado. Lo que se pretende es la claridad en los programas, es decir, que sean fácilmente entendibles y mantenibles.

Estas normas de ninguna manera intentan restringir la libertad del programador, simplemente se trata de que el sistema en general presente en cada uno de sus módulos un desarrollo similar.

4.2 Planes De Prueba (PLP).

Para llevar a cabo la integración de la tarea AIG con el subsistema CHT, es necesario estar seguros de que la tarea funciona correctamente y de que cumple con los requerimientos que le fueron especificados.

Para hacer la verificación de que el programa realiza su función y cumple con los requerimientos es necesario elaborar una serie de pruebas.

Estas pruebas deben abarcar todas las características de la función y quedarán especificadas, para su uso posterior, en un documento denominado: Planes de Prueba (PLP). Ver apéndice C.

En el documento de PLP para la tarea AIG se han descrito dos pruebas. La primera muestra los casos de prueba que cotejan la parte interactiva del programa Movilidad de Unidades (CHMO). La segunda muestra los casos de prueba que verifican la función que realiza este programa.

Los "casos de prueba" son pruebas que se realizan parcialmente. Por ejemplo, para la parte interactiva, un caso de prueba es que el programa verifique el valor que introduce el operador y que responda adecuadamente según haya o no un error.

Los casos de prueba de toda la parte interactiva del programa constituyen la prueba de interacción.

A través de estos casos de prueba se cuida que el usuario no cometa errores que puedan dañar al sistema o al programa en lo que este ejecutará posteriormente.

En la segunda parte del PFP están descritos los casos que constituyen la prueba funcional del programa CHMO.

Esta prueba servirá para verificar que se realicen dos cosas:

- i) la función que debe cumplir la tarea AIG con el programa CHMO.
- ii) los requerimientos planteados al inicio para la tarea.

Se prueban los requerimientos que se mencionaron en el capítulo de Requerimientos y Diseño, sección 3.1.2 de esta tesis.

La función que se prueba es la misma que se especificó en el capítulo 3 para el programa CHMO (ver sección 3.1.3). Esta consiste brevemente en:

Seleccionar las unidades que están en paro y las que están en operación en el intervalo escogido.

Ordenarlas en conjuntos de unidades promovibles y unidades removibles respectivamente.

Para cada unidad promotiva, evaluar la violación al paro hacia adelante y hacia atrás del intervalo escogido y evaluar la violación a la operación en el intervalo escogido.

Para cada unidad removible, evaluar la violación a la operación hacia adelante y hacia atrás del intervalo escogido y evaluar la violación al paro en el intervalo escogido.

Presentarle al operador en desplegados separados las unidades

promovibles y las unidades removibles, ordenadas según la magnitud total de violación en que incurrió cada unidad.

La elaboración del PLP se hizo siguiendo las normas para Planes de Prueba. En la tabla 4-2 se muestra la Presentación del documento suya para elaborar los Planes de Prueba [NIEB3].

NORMAS PARA LA DOCUMENTACION DE PLANES DE PRUEBA.

PROYECTO DE PROGRAMAS DE APLICACION PARA EL SISTEMA DE INFORMACION Y CONTROL EN TIEMPO REAL DEL CENACE.

Las normas que se presentan a continuación deberán ser aplicadas por los responsables de documentar el plan de pruebas para los programas de aplicación. El propósito de las normas es el de facilitar la elaboración de una documentación homogénea y completa.

CONSIDERACIONES GENERALES.

Cuando en casos particulares sea impráctico el seguimiento de una norma dada, podrá esta ser modificada o ignorada con la autorización del responsable del área de Análisis de Sistemas.

El plan de pruebas para los programas de aplicación estará comprendido en un solo documento. El documento será elaborado utilizando el procesador de textos disponible en la computadora VAX del IIE.

Tabla 4-2

En la tabla 4-3 se muestra, en "La Documentación de una Prueba", el índice del contenido que debe llevar cada prueba de las que se especifique en el PLP.

LA DOCUMENTACION DE UNA PRUEBA

La documentación de una prueba deberá constar de una portada y de 6 subsecciones:

- 1) Identificación.
- 2) Análisis.
- 3) Caracterización de salidas.
- 4) Caracterización de entradas.
- 5) Reseña de la prueba.
- 6) Casos de prueba.

Las subsecciones 1, 5 y 6 serán necesarias para elaborar los procedimientos de prueba que a su vez serán necesarios para poder ejecutar la prueba.

Se requieren las 6 subsecciones para documentar el diseño de la prueba. La documentación deberá permitir a terceras personas revisar y juzgar dicho diseño.

Tabla 4-3

La identificación sirve para ubicar la prueba dentro del subsistema, es decir, a que tarea pertenece la prueba y que tipo de prueba se va a realizar.

El análisis, como su nombre lo indica, es un análisis conceptual sobre la parte del programa que se va a probar, considerando las características particulares de las variables que se van a probar y tratando de que estas características sean completamente abarcadas para luego plasmarlas en los casos de prueba.

La caracterización de la salida es parte del análisis y consiste en revisar las variables de salida y especificar sus particularidades para que sean probadas. Al hacer la caracterización, se pueden agrupar en clases las variables que tengan rasgos similares para ser probados.

La caracterización de la entrada es también parte del análisis y es igual que el caso anterior, solo que ahora con las variables de entrada. También se pueden formar clases.

En la reseña de la prueba se describe cuales han sido los casos de prueba seleccionados y que se pretende abarcar con ellos.

Finalmente los casos de prueba propuestos, que también resultan del análisis, son presentados en una lista en donde están agrupados por clase y se especifica para cada dato de entrada mostrado cual es la salida válida que se deberá recibir.

Si siguiendo la secuencia que marca el índice de la tabla 4-3, se muestran en las tablas 4-4 a 4-7 ejemplos de como quedó documentada en el PLP la prueba de validación de datos de la parte interactiva de la tarea AIG.

<p>Identificación.</p> <p>El programa CHMO necesita que el operador le proporcione datos específicos para ejecutar su función. Estos datos pueden contener errores, por lo que se hace necesario validarlos después de que han sido introducidos. Este documento muestra unos casos de prueba que se han desarrollado para validar los datos de entrada.</p> <p>PRUEBA 3.A</p> <p>PRUEBA FUNCIONAL DE: La Validación que Efectúa CHMO de sus Datos de Entrada.</p> <p>REFERENCIA RPC: Sección 4.2.3.5</p> <p style="text-align: center;">Tabla 4-4</p>

Análisis.

Esta prueba servirá para cotejar que la validación de los datos que introduce el usuario trabaja correctamente.

Se validarán los datos de acuerdo a sus características particulares, es decir, hay datos de dos tipos: numéricos y caracteres.

Para los datos numéricos existen dos casos: los enteros y los reales, hay que hacer notar que estos datos deben ser siempre positivos. También es necesario cotejar el rango de los datos de tipo numérico.

Para los datos de tipo carácter se deberá cotejar si son datos válidos y si concuerdan con la información que se debe introducir en cada caso específico.

Tabla 4-5

Caracterización de la salida.

Las salidas que proporciona el programa para la validación de datos se agrupan en tres clases. La primera abarcará a las salidas que provengan de introducir datos numéricos. La segunda clase contendrá a las salidas que corresponden a la introducción de datos de caracteres. La clase tres corresponde a salidas que emite la utilería 'Form Driver' (FDV).

Cuando se crea la forma que se despliega en la pantalla, a cada uno de los campos de datos que se le van a solicitar al operador se le asigna un tipo de dato, por ejemplo, solo números, o solo letras y números, etc. Cuando el operador se equivoca, la utilería FDV le manda un mensaje informándole cual es el tipo de dato que esa entrada requiere.

El conjunto de salidas válidas se presenta clasificado en la Tabla 1.

TABLA 1: Salidas de Validación de Datos.

S(1.1)	V	NUMERO FUERA DE RANGO. (RANGO [1,168])
S(1.2)	V	NUMERO FUERA DE RANGO. (RANGO [1,4])
S(1.3)	V	ERROR: EN EL CAMPO 'STATUS' SOLO SE ACEPTA UN '1'.
S(1.4)	V	ERROR: EN EL CAMPO 'STATUS' SOLO SE ACEPTA UN '0'.
S(1.5)	V	ERROR: EN EL CAMPO 'EDO. COOR.' SOLO SE ACEPTA UN '1' EN LAS UNIDADES QUE TIENEN '0'.
S(1.6)	V	ERROR: EL NUEVO LIMITE MAXIMO DEBE ESTAR DENTRO DEL RANGO CORRECTO.
S(2.1)	V	ERROR: TECLEE SOLAMENTE UNA OPCION.
S(2.2)	V	ERROR: TECLEE SOLO D: PARA DESPLEGAR. E: PARA EJECUTAR CHAG. PF2: AYUDA GENERAL. 2: PARA REGRESAR AL DESPLEGADO ESENAIO DE PLANEACION.

Tabla 4-6

Reseña de la Prueba.

Los casos de prueba que se proponen pretenden cubrir las combinaciones de entradas que se pueden producir para cada dato. Cabe hacer notar el hecho de que son casos de pruebas para validar datos, por esta razón es que solo se indica el tipo de dato que se debe introducir, pudiendo el operador escoger la combinación de caracteres que desee dentro del tipo indicado.

Se proponen 23 casos de prueba. Algunos casos son parecidos entre sí, sin embargo son necesarios porque se prueban variables diferentes en el programa.

Para la clase de datos de tipo numérico, los casos de prueba que se proponen son: i) introducir cualquier cadena de caracteres o caracteres mezclados con números, ii) introducir un número diferente del esperado, o fuera del rango esperado.

Para la clase de datos de tipo carácter, los casos de prueba que se proponen son: i) una secuencia de números o números y caracteres diferente de la esperada, ii) una secuencia de números.

La verificación de esta prueba, es decir de la validación que efectúa el programa, es mediante la observación de los resultados que se obtienen en pantalla cuando se tecléa un número, o una secuencia de caracteres diferentes de lo esperado.

Casos de Prueba.

N° de caso	Clase	Entrada	Salida
1	E(1.1)	NMINBS caracteres o símbolos de puntuación.	NUMERIC REQUIRED
2		otros números	NUMERO FUERA DE RANGO (RANGO[1:168])
3	E(1.2)	ISLA caracteres o símbolos de puntuación.	NUMERIC REQUIRED
4		otros números	NUMERO FUERA DE RANGO (RANGO[1:4])

Tabla 4-7

4.3 Procedimientos De Prueba (PRP).

Una vez que se han terminado los planes de prueba, se deben elaborar unos Procedimientos de Prueba.

Estos procedimientos sirven para que se pueda mostrar al usuario la forma en que se ejecutan las pruebas descritas en el PLP.

Los procedimientos son una guía paso a paso de la ejecución de las pruebas. En estos se explica que datos pide el programa, que responde si se dan datos correctos, que responde si son datos incorrectos, cual es el siguiente dato que pide el programa en su secuencia, etc. Con base en los resultados de las pruebas el usuario debe decidir si el programa cumple o no con las especificaciones impuestas para la tarea.

Es decir, los PRP's constituyen un procedimiento formal de aceptación del programa.

Para elaborar el documento PRP se cuenta con una guía. Esta guía persigue el fin de que todos los PRP's que se elaboren, para los programas del sistema, sean homogéneos.

En la tabla 4-8 se muestra el índice de lo que debe contener un PRP, según lo establece la guía para la elaboración de PRP's [R0083].

EL INDICE DE UN DOCUMENTO DE PROCEDIMIENTOS DE PRUEBAS

Un PRP deberá contener los siguientes encabezados de sección:

1. Panorama general.
2. Referencias.
3. Requerimientos de operación.
 - 3.1 Equipo.
 - 3.2 Programas.
 - 3.3 Bases de datos.
 - 3.4 Sistema operativo.
 - 3.5 Entradas.
 - 3.6 Salidas.
 - 3.7 Métodos.
 - 3.8 Personal.
4. Procedimientos de Pruebas de Aceptación.

Esta sección se divide en subsecciones, una para cada prueba y cada prueba contendrá las siguientes subsecciones:

 - Nombre de la prueba.
 - Reseña.
 - Condiciones.
 - Requerimientos satisfechos.
 - Procedimientos de prueba.

Tabla 4-8

En el panorama general se da una breve referencia del alcance del documento y se ubica al programa dentro del subsistema al que pertenece.

En las referencias se establecen todos los documentos que han servido para elaborar los procedimientos que se describen. Por ejemplo, que secciones del RPC, cual DFC, etc.

Los requerimientos de operación se refieren a las necesidades que en cuanto a equipo, programas, archivos de la BD y datos de entrada se deben proporcionar para que el programa pueda trabajar correctamente. Se debe especificar también: bajo que sistema

operativo corre el programa, que métodos se han usado para diseñar las pruebas que se presentan en el PLP, que personal por parte de los fabricantes y por parte de los clientes estará presente cuando se realicen las pruebas y que clase de salidas proporcionará el programa cuando se ejecute.

En cuanto a los procedimientos de pruebas de aceptación se incluyen:

- Un nombre con el que se identifica la prueba.
- Una reseña en la que se describe cual es el objetivo de la prueba.
- Las condiciones que cada prueba necesita (de las condiciones que fueron mencionadas en los Requerimientos de Operación).
- Requerimientos que quedan satisfechos con cada prueba y finalmente
- Los procedimientos o pasos necesarios para llevar a cabo cada prueba.

Para el programa CHMO los casos de prueba propuestos se han reunido todos en una sola prueba general. Se ha hecho así porque el programa tiene una primera parte interactiva en la cual el usuario debe introducir datos. A continuación, con los datos que ha introducido el usuario, el programa realiza su función y presenta sus resultados en un par de desplegados. Finalmente, sobre los desplegados que presenta el programa, el usuario puede interactivamente, introducir modificaciones.

Como se puede ver, para que el programa muestre los resultados de la función que ejecuta, es necesario pasar primero la secuencia de introducción de datos. Es por eso que se decidió probar la validación de la secuencia de datos interactivos, al mismo tiempo que se prueba la función del programa.

En las tablas 4-9 a 4-11 se mostrará como ha quedado el documento PRP de la tarea AIG.

Panorama General.

En este documento se describe el procedimiento que se debe seguir para llevar a cabo las pruebas que se han elaborado para el programa Movilidad de Unidades (CHMO) el cual forma parte del subsistema CHT, y trabajará interactivamente con el programa Ajuste Automático de Generación (CHAG). Para más referencias sobre los métodos y las interacciones de CHAG y de CHMO consultar los RPCs y DFCs correspondientes.

Referencias.

- RPC de CHT secciones correspondientes al Ajuste Interactivo de Generación.
- Dr. Rolando Nieva G.
Guía para el desarrollo de Planes de Prueba, 1983
- Dr. Guillermo Rodríguez D.
Guía para el desarrollo de Procedimientos de Prueba, 1983
- JC Pérez C.
DFC de Movilidad de Unidades (CHMO). Junio de 1983.
- JC Pérez C.
PLP del programa CHMO. Julio de 1983

Requerimientos de Operación.

Equipo.

- a) Computadora VAX-11/780.
- b) Terminal VT100 conectada al nodo A de la computadora VAX.
Ambos dispositivos de la marca DIGITAL.

3.2 Programas.

Archivos que se encuentran en la cuenta [PROY1688.1603.CHMO]:

CHMO.FDR	Código Fuente.
CHMO.EXE	Código Ejecutable.
CHAG.EXE	Código Ejecutable.

Tabla 4-9

3.8 Personal.

3.8.1 Cliente.

El cliente estará representado por:
Ins. José L. Calderón.

3.8.2 IIE.

El IIE estará representado por:
Dr. Juan Frausto.

4 Procedimientos de Pruebas de Aceptación.

4.1 Prueba General del Programa CHMO.

4.1.1 Reseña.

A través de esta prueba general se verifican los casos de pruebas que se describen en los Planes de Prueba[2]. Los casos se han considerado en una sola prueba por el hecho de que la introducción de datos sigue una secuencia bien definida y los desplegados de la función del programa aparecen a la mitad de la secuencia.

4.1.2 Condiciones.

El equipo y el personal son los mismos que se mencionan en la sección 3.

4.1.3 Requerimientos Satisfechos.

Requerimientos de CHMO en la sección correspondiente de los requerimientos del subsistema CHT.

4.1.4 Procedimientos de Prueba.

Para llevar a cabo estos procedimientos de prueba, se deben seguir previamente los siguientes pasos:

ACCION 1:

- Entrar al sistema VAX a través de una terminal VT100 conectada al nodo 'A'.
- Colocarse en la cuenta [PROY1688.1603.CHMO].
- Verificar la existencia en la BD de los archivos mencionados en los incisos 3.2 y 3.3.
- Teclar 'CHT'. De esta manera el usuario se posiciona en el directorio en que se encuentra el programa IHMCHT.

Tabla 4-10

- Teclar 'RUN CHT'. Aparecerá un desplegado denominado 'SISTEMA DE PLANEACION A CORTO PLAZO'.
- Posicionarse con la tecla 'TAB' en el renglón que dice: AJUSTE INTERACTIVO DE GENERACION y teclar 'E'.

RESULTADO: Aparece un desplegado informándole al usuario sobre el día inicial en el horizonte de planeación y pidiéndole introduzca el intervalo crítico que desea analizar. A continuación se muestra dicho desplegado.

Desplegado 1.

PROGRAMA AJUSTE INTERACTIVO DE GENERACIÓN.

EL DIA INICIAL EN EL HORIZONTE DE PLANEACION ES: MI:25:05:83
 POR LO TANTO EL INTERVALO NUMERO UNO CORRESPONDE A LA HORA
 NUMERO UNO DE ESE DIA.

ESCOJA UN NUMERO DE INTERVALO:---

Este desplegado corresponde a los datos preparados para los Planes de Prueba.

ACCION 2: Tratar de introducir datos no numéricos, caracteres o símbolos de puntuación.

RESULTADO: Aparece el mensaje:

NUMERIC REQUIRED

ACCION 3: Tratar de introducir números mayores a 168, que es el intervalo máximo aceptado.

RESULTADO: Aparece el mensaje:

NUMERO FUERA DE RANGO. (RANGO [1,168])

ACCION 4: Teclar 'PF2'.

RESULTADO: Aparecerá el siguiente mensaje de ayuda:

EL INTERVALO ES UN NUMERO ENTRE 1 Y 168.

Tabla 4-10 (Continuación)

ACCION 5: Introducir un número dentro del rango marcado. Cabe mencionar que el caso que esta preparado en los Planes de Prueba corresponde al intervalo número 20, independientemente de esto; el usuario puede introducir otros números para probar lo que se explica a continuación.

RESULTADO: Se tienen los siguientes posibles resultados:

- a) Si el número corresponde a un intervalo que no tiene unidades promovibles aparecerá el mensaje:

NO HAY UNIDADES PROMOVIBLES EN EL INTERVALO XXX

- b) Si el número corresponde a un intervalo que no tiene unidades removibles aparecerá el mensaje:

NO HAY UNIDADES REMOVIBLES EN EL INTERVALO XXX

- c) Si el número corresponde a un intervalo que no tiene unidades promovibles ni removibles se mandará el siguiente mensaje:

NO HAY UNIDADES PROMOVIBLES NI REMOVIBLES EN EL INTERVALO XXX
Después de este último despliado aparecerá otro en el que se le presentará al usuario si desea volver a ejecutar el Programa con otro intervalo.

QUIERE ANALIZAR OTRA ISLA O INTERVALO?

- d) Si el intervalo que introdujo el usuario tiene unidades promovibles o unidades removibles o ambas, el programa le desplegará: (este despliado corresponde al intervalo número 20, que es el que se ha preparado para los Planes de Prueba)

Despliado 2.

```

PROGRAMA AJUSTE INTERACTIVO DE GENERACION.
LAS ISLAS QUE EXISTEN EN EL INTERVALO 20 SON:

ISLA      CASO DE      CORTE
          CONFIGURACION  EN MW

      1          5      XXX XXX.X

QUE ISLA DESEA ANALIZAR? _
  
```

Tabla 4-10 (Continuación)

ACCION 15: Cuando el usuario teclee una D en el campo de las unidades promovibles.

RESULTADO: Aparecerá el siguiente desplegado: (que es el que producen los datos almacenados para los Planes de Prueba)

Desplegado 5.

PROGRAMA AJUSTE INTERACTIVO DE GENERACION

UNIDADES PROMOVIBLES

FECHA INICIAL: MI:25:05:83

INTERVALO: 20

ISLA: X

CORTE: XXX XXX.X (MW)

CLAVE DE UNIDAD PLANTA	ESTATUS	EDO. COOR.	LIMITE MAX.	VIOLACIONES			
				PARD DETRAS	PARD ADELANTE	OPERA EN INT.	TOTAL
TUCU2	TULP2	0	105.3	0	0	5	5.0
CABU1	CABP1	0	8.6	0	0	5	5.0
NZBU3	NZBP3	0	135.0	11	0	0	11.0
SUZU1	SUZP1	0	112.5	3	4	5	12.0
JOLU1	JOLP1	0	28.8	0	18	0	18.0
FVLU2	FVLP1	0	14.8	0	0	26	26.0
TPOU1	TPOP1	0	18.0	5	9	26	40.0
NTYU1	TTYP1	0	33.8	13	0	29	42.0
TULU1	TULP1	0	100.0	0	19	35	54.0
FVLU1	FVLP1	0	14.8	20	14	26	60.0
JOLU4	JOLP1	0	36.9	29	5	29	63.0
VBHU2	VBHP1	0	67.5	31	11	32	74.0

ACCION 16: Teclar <RETURN> estando el desplegado 5 en la pantalla.

RESULTADO: Se sale de este desplegado y se vuelve al desplegado 3.

Tabla 4-11

C O N C L U S I O N E S

En el presente trabajo se ha mostrado una metodología que sirve para el desarrollo de sistemas de programas. Para ejemplificar el uso de esta metodología se muestra el desarrollo de uno de los programas del sistema.

La metodología tiene varias finalidades a lo largo del desarrollo del sistema:

- i) dividir el desarrollo a través de diferentes etapas;
- ii) emitir documentos que puedan servir para controlar el desarrollo del sistema;
- iii) utilizar los documentos generados para hacer correcciones o modificaciones durante el desarrollo y durante el mantenimiento (esto último después que el sistema se ha liberado).

Los problemas generales que esta metodología trata de evitar son: sobre costos durante el desarrollo, retraso en la entrega del sistema terminado, insatisfacción del usuario por las soluciones brindadas, abatir los costos de mantenimiento, etc.

Los beneficios generales que se obtienen al usar esta metodología son:

Se tiene una visión panorámica del sistema total a través de la construcción de modelos. Esto sirve para que se definan los objetivos y los requerimientos del sistema.

Al construir modelos se puede subdividir el problema, esto reducirá a su vez la complejidad y facilitará la elaboración del sistema.

Si se subdivide el problema se pueden detectar las acciones que se pueden desarrollar en paralelo y las acciones que tienen relaciones de precedencia y que se deben ejecutar en orden.

La realimentación en el sistema a través de la revisión y aprobación o corrección de los documentos generados, con lo que se ejercen acciones correctivas, adaptivas y perfectivas en el sistema aun antes de terminarlo.

En general se lleva el control de todo el proyecto, desde su definición de objetivos y requerimientos hasta la fase de pruebas al sistema generado.

Se pueden distinguir en cada una de las fases beneficios particulares, como son:

Fase de Definición de Requerimientos. Se construyen los modelos conceptuales: con enfoque de usuario para definir los objetivos del sistema; con enfoque de analista para detallar las funciones que ejecutará el sistema. Además sirven como medio de comunicación entre: i) personal no especializado en computación y el líder del proyecto; ii) diseñadores, analistas y programadores. También pueden servir para generar en forma más organizada el documento de requerimientos, mismo que servirá de guía para el desarrollo a lo largo de todo el sistema. La estructura del documento de requerimientos es tal que permite la localización rápida y en forma clara de los requerimientos y además permite introducir modificaciones al costo mínimo.

Fase de Diseño. A partir de los requerimientos y con ayuda del modelo conceptual se construye un diagrama de estructura. Este diagrama permite diseñar el sistema en forma que se cumpla con sus funciones y se respeten los requerimientos. Además permite identificar los programas que constituyen el sistema y como se relacionarán entre ellos, bases de datos y otros sistemas, manteniendo la consistencia entre estos tres elementos. Es aquí donde se detectan las relaciones de precedencia que pudieran existir y el paralelismo para el desarrollo de los programas que forman el sistema.

Cada programa se desarrolla en pseudocódigo, esto da la ventaja de tener un 'lenguaje' universal de comunicación entre analistas, diseñadores y programadores. Además el pseudocódigo queda como documentación del código puro (lenguaje de programación); la utilidad de esto se verá mejor durante la fase de mantenimiento.

Fase de Desarrollo y Pruebas. La codificación de los módulos se convierte en una tarea sencilla por el apoyo que da el pseudocódigo ya generado en la fase anterior. Las etapas de integración y pruebas dependen de las condiciones de cada programa particular y de la capacidad del analista y/o diseñador para detectar situaciones que pudieran generar errores.

Fase de Operación y Mantenimiento. Para llevar a cabo las acciones de corrección de errores latentes, de adaptaciones al medio y de perfeccionamiento, se cuenta con el apoyo de toda la documentación generada a lo largo de las fases anteriores, lo que hace disminuir el costo del mantenimiento.

Como comentario final, hay que recalcar que al control de calidad le corresponde un papel muy importante durante el desarrollo del sistema y que su actividad se verá reflejada en toda la documentación que se deje para el sistema.

A P E N D I C E A

Estructuras de Control.

A continuación se muestran las estructuras básicas de control con que cuenta la programación estructurada, se muestran también algunos ejemplos. La especificación formal de estos ejemplos se puede ver en [RODS1].

La secuencia, consiste en enunciar una tras otra un grupo de instrucciones. Por ejemplo:

BLOQUE : SECUENCIA DE ENUNCIADOS Y COMENTARIOS

COMENTARIO : (* FRASE EN ESPAÑOL *)

**ENUNCIADO : ASIGNACION o DECISION o REPETICION o
INVOCACION A MODULOS o INSTRUCCION DE LECTURA Y ESCRITURA**

ASIGNACION : IDENTIFICADOR=EXPRESION

EXPRESION : EXPRESION ARITMETICA o EXPRESION LOGICA

CONDICION : EXPRESION LOGICA

La decisión, esta estructura da alternativas para ejecutar una secuencia de instrucciones dependiendo de una condición. La estructura tiene tres variantes: sencilla, doble y múltiple.

La decisión sencilla ejecuta una o un conjunto de instrucciones si se cumple la condición. Si la condición no se cumple salta el grupo de instrucciones.

La decisión doble cuenta con dos secuencias, separadas, de instrucciones; la condición se prueba para determinar que secuencia de instrucciones se ejecuta; entonces, si la condición es cierta se ejecuta una secuencia de instrucciones y si es falsa se ejecuta la otra.

La decisión múltiple es el caso general de la decisión; se tienen varias secuencias separadas de instrucciones y una condición general; la secuencia que se va a ejecutar depende del valor que tome la condición. Cada vez que se ejecuta esta estructura solo se ejecuta una secuencia de instrucciones; las demás se saltan.

Ejemplo:

```
DECISION:  SI (CONDICION) ENTONCES
           SECUENCIA DE ENUNCIADOS Y COMENTARIOS
           SINO
           SECUENCIA DE ENUNCIADOS Y COMENTARIOS
           FIN (SI)
```

La repetición. Esta estructura permite ejecutar una misma secuencia de instrucciones un número determinado de veces. La estructura cuenta con una condición para determinar el número de veces que se ejecutará la secuencia. Hay varios esquemas para esta estructura de control, la diferencia entre ellos esta en la forma de probar la condición, se tienen como esquemas: mientras, ejecuta-hasta, repite y ciclo.

En el esquema mientras la condición se prueba antes de ejecutar la secuencia, si la condición es cierta se ejecuta la secuencia si no es cierta se termina la ejecución de esta estructura.

El esquema ejecuta-hasta permite la ejecución de la secuencia de instrucciones, después de la ejecución prueba la condición, si esta es verdadera vuelve a ejecutar la secuencia, si es falsa se termina la ejecución de la estructura.

El esquema repite cuenta con tres variables auxiliares, una para establecer el valor inicial, la segunda para establecer el valor final y la tercera para indicar el incremento que tomará la variable con el valor inicial. Esta asignación se hace al inicio de la ejecución de la estructura. En este esquema la condición consiste en ir comparando el valor de la variable inicial con el valor de la variable final, si el valor inicial es menor o igual al valor de la variable final se incrementa la variable inicial en lo que indique la tercera variable y se ejecuta la secuencia de instrucciones, en otro caso se termina la ejecución de la estructura.

El ciclo cuenta con una expresión y una secuencia de instrucciones; al inicio de la ejecución se evalúa la expresión; si el resultado da mayor que cero se ejecutará la secuencia ese número de veces. Si el resultado da menor o igual que cero se termina la ejecución de la estructura.

Ejemplos:

REPETICION : MIENTRAS o EJECUTA o REPITE o CICLO

MIENTRAS : MIENTRAS (CONDICION)
 SECUENCIA DE ENUNCIADOS Y COMENTARIOS
 FIN (MIENTRAS)

EJECUTA : EJECUTA
 SECUENCIA DE ENUNCIADOS Y COMENTARIOS
 HASTA (CONDICION)

REPITE : REPITE IDENTIFICADOR=INICIO,FIN,INCREMENTO
 SECUENCIA DE ENUNCIADOS Y COMENTARIOS
 FIN (REPITE)

CICLO : CICLO (EXPRESION ARITMETICA)
 SECUENCIA DE ENUNCIADOS Y COMENTARIOS
 SALIDA CICLO SI (CONDICION)
 SECUENCIA DE ENUNCIADOS Y COMENTARIOS
 FIN (CICLO)

A P E N D I C E B

Normas de Programación.

A continuación se describen algunas de las normas que se emitieron para gobernar la generación de código ejecutable en FORTRAN 77. [MIE82]

Las normas están divididas en dos partes:

- A) Normas de encabezados.
- B) Normas de estilo de codificación para FORTRAN 77.

A) Normas de encabezado (para el programa principal).

- El primer registro del encabezado es el nombre del programa.
- Podrán utilizarse tantos registros como sea necesario para documentar el propósito del programa principal, la bibliografía, las limitaciones y aclaraciones. En caso de que alguna(s) de

esta(s) información(es) no exista(n), se debe aclarar anotando 'Ninguna' o 'Ninguno'.

-Deberán existir necesariamente referencias al RPC y al DPC en el formato señalado.

Para ver la forma de un encabezado referirse al capítulo 4, tabla 4-1. Las normas para encabezado de subrutinas y de bloque de datos son similares.

B) Normas de estilo de codificación para FORTRAN 77.

Estas normas se dividen de la siguiente manera:

1- Indentación. Todo anidamiento de código deberá ser desplazado tres columnas a la derecha del anidamiento de referencia, incluyendo comentarios.

2- Líneas de Continuación. Estarán definidas por el símbolo ">" en la columna seis.

3- Comentarios:

a) deberán estar indentados.

b) deberán tener el siguiente estilo:

```
-----  
*?????????????????????????????????:????????*  
-----
```

c) ? es el lugar reservado para los comentarios y es de longitud variable.

d) los comentarios llevan una 'C' en la columna uno.

4- Areas comunes.

- a) no se permiten 'COMMONs' sin etiquetar.
- b) se declara una variable por etiqueta.
- c) los nombres de las etiquetas en 'COMMON' deberán aparecer en orden alfabético.

Para ver el estilo referirse a la tabla 4-1 del capítulo 4.

5- Instrucciones prohibidas:

- INPLICIT
- ENTRY
- IF aritmético
- STOP
- COMMON no etiquetado
- DO con línea CONTINUE

6- Instrucciones limitadas.

- RETURN, uno por cada subrutina y exactamente antes del END.
- GO TO, solo si está justificado y lo aprueba el responsable de esa parte del proyecto.
- DATA, solo en el BLOCK DATA o para variables locales en subrutinas.
- EQUIVALENCE, igual que el GO TO.
- CALL EXIT, igual que el GO TO o en el programa principal exactamente antes del END.
- END, aparecerá en las instrucciones READ o WRITE solo cuando lo autorice el responsable de esa parte del proyecto.
- FUNCTION, se autoriza por el responsable de la parte del proyecto

involucrada.

- **COMPLEX**, se autoriza por el responsable de la parte del proyecto involucrada.

7- **Etiquetas**. Se definirán de 100 en 100 en orden ascendente a partir del número 5000 y se ajustarán a la izquierda en los primeros 5 campos de la línea.

8- **Declaración de variables**.

-todas las variables deberán ser explícitamente declaradas.

-todas las declaraciones (excepto **LOGICAL**) deberán indicar explícitamente el número de palabras reservadas.

Para ver el estilo de declarar variables en el capítulo 4, tabla 4-1 se muestran varios ejemplos.

9- **Números mágicos**.

Las instrucciones de **PARAMETER** deben seguir el siguiente estilo:

```
C
C   FECHA INICIAL
    PARAMETER ( AAAAAA = N1N1N1 )
```

Las instrucciones **DATA** deberán seguir el siguiente estilo:

```
C
C   ARCHIVO DE LECTURA
    DATA AAAAAA / N1N1N1 /
```

En PARAMETER y en DATA se comentarán las variables a partir de la columna 7 en el formato señalado. Entre instrucción e instrucción se incluye una línea de comentario vacío.

A P É N D I C E C

Normas de Planes de Pruebas.

A continuación se detallan algunas de las normas que se emitieron para gobernar la elaboración del documento planes de prueba.[NIE82]

1.- Consideraciones generales.

El plan de pruebas para los programas de aplicación estará comprendido en un solo documento.

El documento constará de un capítulo de introducción y los restantes uno para cada subsistema.

2.- La portada de un capítulo. Deberá presentar la siguiente información:

-El número y nombre del capítulo.

- El nombre y firma del responsable del capítulo.
- Los nombres de los investigadores colaboradores.
- Los nombres y firmas del personal que autoriza el capítulo por parte del IIE y de CFE.
- La fecha de la última revisión.
- Número de página consecutivo inicial.

3.- La lista del contenido del capítulo.

Se presentara después de la portada y contendrá la lista de secciones que integran el capítulo. No se incluirán las subsecciones en esta lista.

4.- La introducción al capítulo. Deberá constar del encabezado, un párrafo inicial y dos subsecciones que son:

- Un panorama general del capítulo. Este incluye el número total de secciones y el número de secciones dedicadas a cada uno de los distintos aspectos del subsistema como son: Interfaz Hombre Máquina y tareas.
- Un resumen del plan de pruebas. Este resumen incluye por cada prueba la clave de esta, una descripción breve, el número de casos que la constituyen, la prueba y las secciones del RPC cuyos requerimientos son cubiertos por la prueba.

5.- Lista de referencias. Referencias que:

- Describen los requerimientos y especificaciones del programa a probar.
- Describen los fundamentos del análisis sobre los cuales se basan

las pruebas propuestas.

- Describen las normas a seguir para los planes de prueba.
- Describen las técnicas para diseñar casos de prueba.

6.- Documentación de una prueba. Constará de una portada y seis subsecciones:

- Identificación.
- Análisis.
- Caracterización de las salidas.
- Caracterización de las entradas.
- Reseña de la prueba.
- Casos de prueba.

La identificación deberá incluir la referencia a la sección o secciones principales del RFC donde se reportan los requerimientos a probar y una lista de elementos de la matriz de requerimientos que serán considerados en la prueba.

El análisis debe contener los siguientes dos puntos:

- a) Conceptos claves sobre los cuales se fundamentará el diseño de los casos de prueba.
- b) Lista de consideraciones que pueden, a criterio del analista, incrementar la susceptibilidad del código y diseño a errores de computación.

La caracterización de las salidas contiene los siguientes dos puntos:

- a) Descripción de la salida, haciendo referencia a las secciones

correspondientes del RPC.

- b) Agrupamiento de las salidas posibles en clases.

La caracterización de las entradas contiene los dos puntos siguientes:

- a) Descripción de la entrada, haciendo referencia a las secciones correspondientes del RPC.

- b) Agrupamiento de las entradas posibles en clases.

El resumen de la prueba debe contener:

- a) Número total de casos de prueba.

- b) Descripción de casos o grupos de casos de prueba con características similares.

- c) Método o métodos que deberán ser utilizados para verificar los resultados de la prueba.

- d) Criterios de aceptación de los resultados de la prueba.

Finalmente se presentan todos y cada uno de los casos de prueba propuestos.

A P E N D I C E D

Manual de Usuario.

A continuación se presenta una propuesta para documentar el manual que usará el operador cuando trabaje, a través de la terminal, con el Programa Movilidad de Unidades (CHMO), que forma parte de la tarea Ajuste Interactivo de Generación (AIG) del subsistema Coordinación Hidro Térmica (CHT) y que se ha presentado en esta tesis como programa ejemplo.

Esta propuesta será evaluada junto con otras propuestas que se presentarán para documentar este tipo de manual. Finalmente, de esta evaluación surgirá un modelo único para documentar este tipo de manual.

Introducción.

El programa Movilidad de Unidades (CHMO), Junto con el Programa Ajuste Automático de Generación (CHAG) componen la tarea Ajuste Interactivo de Generación (AIG).

El objetivo de la tarea AIG es eliminar cortes de carga en los intervalos críticos.

El procedimiento que sigue la tarea es como sigue. Se ejecuta CHMO para los intervalos críticos del Plan de operación, a través del programa el usuario aliviará las restricciones que considere más convenientes para tratar de eliminar cortes. A continuación ejecutará CHAG para que le proporcione un nuevo Plan de operación.

El programa CHMO tiene la facilidad de aliviar ciertas restricciones. Se alivian las restricciones para que al generar un nuevo Plan de operación esto resulte sin cortes de carga.

El tipo de restricciones que se puede aliviar son:

Para las unidades que en el intervalo crítico estén operando o estén en paro frío.

- cambiar estado: de no-coordinables a coordinables.
- cambiar límite máximo de generación.
- cambiar estatus: las unidades que estén en operación se pueden cambiar a paro frío. Las unidades que estén en paro frío se pueden poner en operación.

El usuario podrá consultar a través de un despliegado de CHMO si realmente se ha eliminado el corte de carga. Si se ha eliminado el corte y al usuario le parece que el plan de operación así como sus modificaciones son buenas, puede actualizar los archivos de la Base de Datos (BD) con sus modificaciones.

Si el nuevo plan de operación no satisface al usuario puede volver a ejecutar el ciclo completo de la tarea AIG hasta que encuentre un plan de operación que le satisfaga o hasta que considere que ha ejecutado el proceso un número conveniente de veces.

Modo de ejecutar AIG.

Para ejecutar la tarea AIG se debe revisar que existan en el número de cuenta en que se está trabajando los archivos: CHMO.EXE, CHAG.EXE y CHT.EXE. Si los archivos no existen se debe solicitar información con la persona encargada del subsistema CHT.

Si los tres archivos existen, la primera instrucción que se teclea es :

```
$ RUN CHT
```

Al correr este programa le presenta al usuario un despliegado. En este despliegado se encuentra localizada la tarea AIG como sigue:

```
.  
. .  
AJUSTE INTERACTIVO DE GENERACION .....  
. .  
.
```

Para que se ejecute la tarea AIG se debe teclear en el campo asociado una 'E', como se muestra a continuación:

```
.
.
.
AJUSTE INTERACTIVO DE GENERACION .....E<RETURN>
.
.
```

Cuando se empieza a ejecutar la tarea AIG el programa CHMO manda un desplazado como el que se muestra a continuación:

Desplazado 1.

```
+-----+
|          PROGRAMA AJUSTE INTERACTIVO DE GENERACION.          |
| EL DIA INICIAL EN EL HORIZONTE DE PALNEACION ES: MI:25:05:83 |
| POR LO TANTO EL INTERVALO NUMERO UNO CORRESPONDE A LA HORA  |
| NUMERO UNO DE ESE DIA.                                       |
| ESCOJA UN NUMERO DE INTERVALO:___                          |
+-----+
```

La fecha mostrada se especifica como día:día:mes:año y en este caso el intervalo número uno corresponde a la hora uno del miércoles 25. Por ejemplo, el intervalo 35 corresponde a la hora 10 del Jueves 26 y el intervalo 68 corresponde a la hora 20 del viernes 27.

El usuario debe escribir sobre las líneas del desplazado 1 el intervalo crítico para el que desea ejecutar CHMO. En una semana hay

RETURN: de esta forma se indicará que se debe oprimir esa tecla de la terminal.

168 intervalos horarios, por lo tanto, el programa solo aceptará números entre 1 y 168.

Cuando se oprime la tecla 'PF2' de la terminal se obtiene un mensaje de ayuda como el que se muestra a continuación:

'LOS NUMEROS DE INTERVALO VALIDOS SON DEL 1 AL 168.'

Después de que el usuario ha escosido el intervalo critico se pueden presentar cuatro situaciones:

1) Que no haya unidades promovibles ni removibles en el intervalo escosido. En este caso el programa manda el mensaje que sigue y termina su ejecución.

NO HAY UNIDADES PROMOVIbles NI REMOVIBLES EN EL INTERVALO XXX.

2) Que no haya unidades promovibles en el intervalo escosido. En este caso aparece el siguiente mensaje y el programa continúa su ejecución con el desplizado 2.

NO HAY UNIDADES PROMOVIbles EN EL INTERVALO XXX.

3) Que no haya unidades removibles en el intervalo escosido. En este caso el programa manda el siguiente mensaje y continúa su ejecución con el desplizado 2.

NO HAY UNIDADES REMOVIBLES EN EL INTERVALO XXX.

d) Que haya tanto unidades promovibles como removibles en el

intervalo escogido, en este caso el programa continúa con el despliegado 2.

El despliegado dos muestra las islas que hay en el intervalo escogido, el corte que hay en cada isla (si lo hay) y el caso de configuración asociado.

Despliegado 2.

```
PROGRAMA AJUSTE INTERACTIVO DE GENERACION,  
LAS ISLAS QUE EXISTEN EN EL INTERVALO 20 SON:  
  
ISLA      CASO DE      CORTE  
          CONFIGURACION  EN MW  
  
  1         X      XXX XXX.X  
  2         X      XXX XXX.X  
  3         X      XXX XXX.X  
  4         X      XXX XXX.X  
  
QUE ISLA DESEA ANALIZAR? _
```

El usuario debe escoger una de las islas que se le muestran. En caso de que se mande ejecutar el programa CHAG, los resultados que arroje para esta isla se van a comparar con los resultados que para la misma isla muestra el despliegado 2.

Se obtiene un mensaje de ayuda para el despliegado 2 cuando se oprime la tecla 'PF2' de la terminal. El mensaje dice:

'LOS NUMEROS DE ISLA VALIDOS SON DEL 1 AL 4.'

El programa continúa su ejecución con el desplegado 3. Este desplegado da opciones sobre la forma de continuar la ejecución de la tarea AIG.

Desplegado 3.

```
┌-----┐
│ DESPLEGADO DE UNIDADES PROMOVIBLES      - │
│ DESPLEGADO DE UNIDADES REMOVIBLES      - │
└-----┘
```

Las opciones consisten en:

Desplegado 4.

```
┌-----┐
│          PROGRAMA AJUSTE INTERACTIVO DE GENERACION          │
│          OPCIONES VALIDAS                                    │
│ DESPLEGADO DE UNIDADES PROMOVIBLES      D, E, 'PF2', 2    │
│ DESPLEGADO DE UNIDADES REMOVIBLES      D, E, 'PF2', 2    │
│          D:  PARA DESPLEGAR                                │
│          E:  PARA EJECUTAR CHAG                           │
│          PF2: AYUDA GENERAL                                │
│          2:  REGRESAR AL DESPLEGADO ESENAARIO DE PLANEACION │
└-----┘
```

El desplegado 4 se obtiene si se oprime dos veces la tecla 'PF2' de la terminal cuando se esta en el desplegado 3. A continuación se explican las opciones que contiene este desplegado.

D: Si se pone sobre el campo correspondiente a las unidades promovibles aparecerá el desplegado 5. Si se pone sobre el campo

correspondiente a las unidades removibles, aparecerá el desplegado

7.

E: Se manda ejecutar el Programa CHAG.

PF2: Aparece el desplegado 4 en la pantalla de la terminal.

2: El programa finaliza y le devuelve el control a la interfaz hombre máquina de CHT.

Si el usuario teclea en el campo correspondiente del desplegado 3 cualquier otro caracter diferente de los mostrados, aparecerá el siguiente mensaje:

ERROR: TECLEE SOLO.

D: PARA DESPLEGAR.

E: PARA EJECUTAR CHAG.

PF2: AYUDA GENERAL.

2: PARA REGRESAR AL DESPLEGADO ESCENARIO DE PLANEACION.

Si en el desplegado 3, en el campo correspondiente a las unidades removibles, se teclea una D, aparecerá el desplegado 5.

Desplegado 5.

PROGRAMA AJUSTE INTERACTIVO DE GENERACION								
UNIDADES PROMOVIBLES								
FECHA INICIAL: MI:25:05:83								
INTERVALO: 20		ISLA: X		CORTE: XXX (M. A. (M))				
CLAVE DE UNIDAD PLANTA	ESTATUS	EDO. COOR.	LIMITE MAX.	VISUALIZACIONES				
				PARO DETRAS	FARO ADELANTE EN INT.	OPERA	TOTAL	
TUCU2	TULP2	0	0	105.3	0	0	5	5.0
CADU1	CADP1	0	0	8.6	0	0	5	5.0
MZDU3	MZDP3	0	1	135.0	11	0	0	11.0
SUZU1	SUZP1	0	1	112.5	3	4	5	12.0
JOLU1	JOLP1	0	0	28.8	0	18	0	18.0
FVLU2	FVLP1	0	0	14.8	0	0	26	26.0
TPOU1	TPOP1	0	1	18.0	5	9	26	40.0
MTYU1	TTYU1	0	1	33.8	13	0	29	42.0
TULU1	TULP1	0	1	100.0	0	19	35	54.0
FVLU1	FVLP1	0	1	14.8	20	14	26	60.0
JOLU4	JOLP1	0	0	36.9	29	5	29	63.0
VDNU2	VDNP1	0	0	67.5	31	11	32	74.0

Sobre el desplegado 5 el usuario puede modificar los datos que explica el desplegado 6.

Desplegado 6.

PROGRAMA AJUSTE INTERACTIVO DE GENERACION

AYUDA PARA EL DESPLIEGADO DE UNIDADES PROMOVIBLES
LAS UNIDADES QUE SE DESPLIEGARON ESTAN PARADAS Y SE PUEDEN
PONER EN OPERACION. (PROMOVER)

LOS DATOS QUE SE PUEDEN MODIFICAR EN ESTE DESPLIEGADO SON:

ESTATUS: A CUALQUIER UNIDAD PUEDE CAMBIARSE SU ESTATUS, PERO
SOLAMENTE A 'OPERACION', PARA ESTO SE TECLEA UN '1' EN EL
CAMPO ASOCIADO.

EDO. COOR.: SOLO LAS UNIDADES QUE ESTAN COMO NO COORDINABLES (0)
SE PUEDEN PONER COMO COORDINABLES TECLEANDO UN '1' EN EL CAMPO
ASOCIADO.

LIMITE MAX.: SE PUEDE MODIFICAR EL LIMITE MAXIMO DE GENERACION
DE CUALQUIER UNIDAD. EL NUMERO QUE SE TECLEE EN EL CAMPO ASOCIADO
ES COTEJADO CONTRA LA POTENCIA MAXIMA PERMITIDA.

El despliegado 6 se puede obtener si cuando se esta en el
despliegado 5 se teclaea dos veces 'PF2'. Además, sobre cada uno de los
campos modificables del despliegado 5 al oprimir una vez 'PF2' se
obtiene un mensaje de ayuda. Estos mensajes de ayuda son:

Sobre el campo 'ESTATUS'

'A CUALQUIER UNIDAD PUEDE CAMBIARSE SU ESTATUS TECLEANDO UN 1.'

Sobre el campo 'EDO. COOR.'

'SOLO LAS UNIDADES QUE TIENEN UN 0 PUEDEN CAMBIAR A 1.'

Sobre el campo 'LIMITE MAX.'

SE PUEDE CAMBIAR EL LIMITE MAXIMO PERO DENTRO DEL RANGO ADECUADO.

Tanto en el desplegado 5 como en el 7, para moverse entre los campos que se pueden modificar, las teclas de la terminal que se usan son:

<TAB> para avanzar hacia adelante sobre los campos desplegados en la forma y que se pueden modificar.

<BACK SPACE> para avanzar hacia atrás sobre los campos desplegados en la forma y que se pueden modificar.

<^> para moverse hacia arriba en forma vertical sobre un campo de datos desplegado.

<V> para moverse hacia abajo en forma vertical sobre un campo de datos del desplegado.

Si en el desplegado 3, en el campo correspondiente a las unidades renovables se tecléa una D aparecerá el desplegado 7.

Desplegado 7.

PROGRAMA AJUSTE INTERACTIVO DE GENERACION								
UNIDADES REMOVIBLES								
INTERVALO: 20		ISLA: X		CORTE: XXX XXX.X (MW)				
CLAVE DE	ESTATUS	EDO.	LIMITE	VIOLACIONES				
UNIDAD PLANTA		COORD.	MAX.	OPERA	OPERA	PARO	TOTAL	
				DETRAS	ADELANTE	EN INT.		
JLEU2	JLEP1	1	0	13.5	0	0	5	5.0
CUAU4	CUAP1	1	0	12.6	1	1	5	7.0
NOBU1	NOGP1	1	0	8.1	3	0	5	8.0
CRUU1	CRUP1	1	0	16.6	6	1	17	24.0
SLMU1	SLMP1	1	0	67.5	0	25	0	25.0
PGUU1	PGUP1	1	0	11.3	5	11	17	33.0
CUAU4	CUAP1	1	0	18.9	16	20	23	59.0

En el desplegado 7 se pueden modificar los mismos campos que en el desplegado 5, pero ahora con las condiciones que marca el desplegado 8.

Desplegado 8.

PROGRAMA AJUSTE INTERACTIVO DE GENERACION

AYUDA PARA EL DESPLEGADO DE UNIDADES REMOVIBLES

LAS UNIDADES QUE SE DESPLEGARON ESTAN EN OPERACION Y SE PUEDEN PARAR (REMOVED)

LOS DATOS QUE SE PUEDEN MODIFICAR EN ESTE DESPLEGADO SON:

ESTATUS: A CUALQUIER UNIDAD PUEDE CAMBIARSE SU ESTATUS, PERO SOLAMENTE A 'PARO', PARA ESTO SE TECLEA UN '0' EN EL CAMPO ASOCIADO.

EDO. COOR.: SOLO LAS UNIDADES QUE ESTAN COMO NO COORDINABLES (0) SE PUEDEN PONER COMO COORDINABLES TECLEANDO UN '1' EN EL CAMPO ASOCIADO.

LIMITE MAX.: SE PUEDE MODIFICAR EL LIMITE MAXIMO DE GENERACION DE CUALQUIER UNIDAD. EL NUMERO QUE SE TECLEE EN EL CAMPO ASOCIADO ES COTEJADO CONTRA LA POTENCIA MAXIMA PERMITIDA.

El desplegado 8 también se puede obtener desde el desplegado 7 tecleando dos veces 'PF2'. Es posible obtener mensajes de ayuda en los campos modificables del desplegado 7 tecleando una vez 'PF2'. Estos mensajes son:

Sobre el campo 'STATUS'

'A CUALQUIER UNIDAD PUEDE CAMBIARSE SU ESTATUS TECLEANDO UN 0.'

Sobre el campo 'EDO. COOR.'

'SOLO LAS UNIDADES QUE TIENEN UN 0 PUEDEN CAMBIAR A 1.'

Sobre el campo 'LIMITE MAX.'

'SE PUEDE CAMBIAR EL LIMITE MAXIMO PERO DENTRO DEL RANGO ADECUADO.'

Desde el desplegado 3 el usuario puede mandar ejecutar CHAG, tecleando una E en cualquiera de sus dos campos. Si el programa CHAG se ejecuta se le mostrará al usuario el desplegado 9, de comparación entre los resultados de esta corrida y la anterior, en la isla que seleccionó el usuario.

Desplegado 9.

```

+-----+
|          PROGRAMA AJUSTE INTERACTIVO DE GENERACION          |
|          DESPLEGADO DE COMPARACION DE RESULTADOS DE CHAG    |
|          ISLA: X          INTERVALO: 20          FECHA: MI:25:05:83 |
|          COSTO ANTERIOR: $ XXX XXX.X          COSTO NUEVO: $ XXX XXX.X |
|          DIFERENCIA: $ XXX XXX.X              |
|          CORTE ANTERIOR: XXX XXX.X MW          CORTE NUEVO: XXX XXX.X MW |
|          DIFERENCIA: XXX XXX.X MW              |
|          UNIDAD          CAMBIO:              |
|          XXXXX          X                      |
|          XXXXX          X                      |
+-----+

```

Después del despliegado ? se le manda al usuario el siguiente mensaje:

DESEA ACTUALIZAR LA BASE DE DATOS?(S/N)_

Si el usuario responde afirmativamente, las modificaciones que hizo a los despliegados 5 y 7 se actualizarán en la RD y cuando esto se haya hecho se le avisará con el mensaje que sigue:

BASE DE DATOS ACTUALIZADA.

Si el usuario responde que no desea que se actualice la RD el programa continúa. El mensaje que aparece a continuación es:

DESEA ANALIZAR OTRA ISLA O INTERVALO?(S/N)_

Si el usuario responde que no desea seguir analizando datos, el programa termina y le devuelve el control a al interfaz hombre máquina del CHT.

Si el usuario responde que desea continuar analizando datos, el programa vuelve a empezar desde el principio, mostrando el despliegado 1.

B I B L I O G R A F I A .

[ARR]

Arroyo Asuilera, Gonzalo.

Aplicaciones de Control en Sistemas Eléctricos de Potencia.

CFE, Gerencia General de Operación y ESIME, IPN Sección de Graduados.

[CAL81]

Ing. José L. Calderón, CFE, México 1981.

Desarrollo en México de Programas de Aplicación para el Sistema de Información y Control en Tiempo Real de CFE.

[FAG81]

Fadenbaum, Joel.

A New Breed: The Software Engineer.

IEEE Spectrum, September 1981, pp. 62-66.

[FR82]

Dr. Juan Frausto, IIE, México 1982.

Análisis de Requerimientos para el Diseño de Programas de Computadora de CHT a Corto Plazo.

[GER78]

Dr. Victor Gerez, M. en C. Manuel Grijalva.

El Enfoque de Sistemas. Ed. LIMUSA, México 1978.

[GER81]

Dr. Victor Gerez, M. en C. Mauricio Mier, Ing. Motoki Inoue.

Ingeniería de Programación en la División Sistemas de Potencia del Instituto de Investigaciones Eléctricas.

IEEE, MEXICON 1981.

[GER83]

Dr. Victor Gerez, IIE, México 1983.

Apuntes para un libro inédito.

[IEEE76]

IEEE, Power Engineering Society, Publication Guide for Power Engineers.

New York, USA, 1976.

[IND83]

M. C. Adrián Inda, Ing. David Burciaga, IIE, México 1983.

Optimum Active Power Constrained Re-scheduling and Load Shedding to Alleviate Power Systems Operating Constraints Violations.

[LEH80]

Lehman, Meir M.

Programs, Life Cycles and Laws of Software Evolution.

Proceedings of the IEEE, Vol. 68, Num. 9, September 1980.

[LIA]

Thomas E. Di Liacco, IEEE.

An Overview of Power System Control Centers.

[MAR82]

M. C. Salvador Martínez, IIE, México 1982.

Análisis del Subsistema Pronóstico de Carga a Corto Plazo.

[HAR83]

Ins. Javier Martínez, IIE, México 1983.
Manual de Mantenimiento del Programa Cálculo de Intercambio
Sección correspondiente al DPC de CIIH.

[HAR]

Ins. Ferdinando Martone, Honeywell, S. A., México.
El Despacho Económico de la Energía Eléctrica por Medio de
Computadoras.

[MIE82]

M. C. Mauricio Mier, IIE, México 1982.
Normas de Codificación FORTRAN 77.

[MIE82]

Dr. Rolando Nieva, IIE, México 1982.
Guía para el Desarrollo de Planes de Prueba.

[PRE82]

Roger S. Pressman, McGraw-Hill 1982.
Software Engineering: A Practitioner's Approach.

[ROD81]

Dr. Guillermo Rodríguez, IIE, México 1981.
Lenguaje para el Diseño de Procesos (LDP).

[ROD82]

Dr. Guillermo Rodríguez, IIE, México 1982.
Guía para la Documentación de Procedimientos de Pruebas de
Aceptación de Programas de Computadora.

[ROM90]

M. C. Rikica Romano, IIE, México 1980.

Despacho Económico Restringido.

[RPC81]

Documento RPC de CHT, sección 4.

[SAD81]

Ing. Jesús Sada, Ing. Gonzalo Arroyo, CFE, México 1981.

El Sistema Eléctrico Mexicano, Descripción y Filosofía de Operación, Problemática Actual.

[SAN82]

M. C. Manuel Santos, Ing. Javier Martínez, IIE, México 1982.

Manual de Mantenimiento del Programa Cálculo de Intercambio Neto entre las Areas (CINA). Subsistema PCAI-II

Sección correspondiente a los requerimientos de la tarea PCAI (CINA).

Ing. Javier Martínez, IIE, México 1982.

Sección correspondiente al Diseño de Programa de Computadora (DPC) del CINA.

[YEH80]

Yeh, R. T. and Zave, F.

Specifying Software Requirements.

Proceedings of the IEEE, Vol. 68, Num. 9, pp. 1077-1084, 1980.

[YOU75]

Yourdon, E. and L. Constantine.

Structured Design.

Yourdon Press, New York, 1975.