

24
14



UNIVERSIDAD NACIONAL
AUTÓNOMA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO E IMPLEMENTACION DE UN SISTEMA
DE BANCO DE HORAS
VIA MICROCOMPUTADORA

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
PRESENTA:
JUAN CARLOS REZA GARCIA

DIRECTOR DE TESIS
ING. SEBASTIAN FOBLANO ORDÓÑEZ

MEXICO, D.F.

1986.



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

INTRODUCCION	1
.....	
DESARROLLO DE LA INFORMATICA	7
.....	
ANALISIS	15
.....	
Planteamientos de los Objetivos.....	15
Análisis de Factibilidad	17
Especificación de los Requerimientos	22
FASE DE DISEÑO	32
.....	
Diseño de Entradas	33
Diseño de Salidas	42

Diseño de Archivos	50
Diagrama de Estructura	57
Diseño Detallado	75
Selección del Leng. de Programación	110
Documentación	112

FASE DE DESARROLLO
..... 114

Codificación	114
Integración	116
Pruebas del Sistema	119

FASE DE OPERACION Y MANTENIMIENTO
..... 124

Manual del Usuario	124
Mantenimiento	149

EVALUACION DEL SISTEMA
..... 152

APENDICES

A) Descripción del Equipo Utilizado	155
B) Transportabilidad del Sistema	166

BIBLIOGRAFIA
..... 171

I N T R O D U C C I O N

En toda institución, el correcto desarrollo de las funciones administrativas hace necesario buscar herramientas que permitan controlar ágilmente todos los procesos y trámites, a la vez, que contar con información oportuna y veráz. De aquí que, debido al gran volumen de información que se maneja en los procesos administrativos de la Universidad Nacional Autónoma de México, se ha hecho de la computadora un instrumento de gran importancia, desarrollandose una serie de sistemas automatizados, entre ellos el elaborado por esta tésis.

Este trabajo surge como una necesidad por parte de algunas escuelas y facultades, que han solicitado a la Dirección General de Servicios de Cómputo para la Administración, la creación de un sistema que permita llevar, un control riguroso de la información del

personal docente por medio de una microcomputadora, considerando asignaturas, grupos, horarios, días, categorías y códigos programáticos.

Este Sistema permite crear un Banco de Datos que refleja la situación de cada uno de los profesores. Para el personal que no es profesor de asignatura, se registra lo correspondiente a sus nombramientos.

Entre las funciones que se contemplan para el Sistema, están la emisión de diversos reportes así como la posibilidad de poder consultar el Banco de Datos en el momento en que se desee, de tal manera que el sistema sea un apoyo sustancial en las actividades administrativas, para llevar más ágil y eficaz un control, tanto del personal como del presupuesto, y contar con información completa y actualizada para la toma de decisiones y la planeación adecuada.

Además el sistema contará con cada uno de los elementos necesarios que permitan satisfacer los requerimientos generales que implica llevar a cabo un control de Banco de Horas en escuelas y facultades, así como en dependencias de docencia e investigación.

La presente obra esta dividida en fases de desarrollo, que permiten llevar un control efectivo del Sistema durante todo el proceso de diseño e implementación, cada una de estas fases abarca un capítulo y estan divididas como sigue:

En el Capítulo 1 se da un panorama general sobre el desarrollo y la problemática dentro de la informática, mostrando el impacto que ha provocado esta tecnología, la forma en que ha ido evolucionando desde sus orígenes hasta nuestros días, se muestran las características de las computadoras, mismas que se van haciendo cada vez más potentes y más pequeñas. Aparte se muestra como, no obstante todos estos avances, existen problemas que enturbian el futuro de la computación como es el costo de la programación, y que debido a esto, se hace necesario desarrollar programas en forma sistemática.

En el Capítulo 2, se tiene ya lo que es la fase inicial del Sistema de Banco de Horas, aquí se estudian los primeros pasos que son necesarios para el diseño del sistema, la identificación del problema, el planteamiento de los objetivos, así como la propuesta y evaluación de las diversas alternativas que se presenten y que mediante herramientas ya existentes como el análisis de factibilidad, se toma la que se considera que cumple, de mejor manera con los objetivos propuestos, tomando en consideración factores económicos, técnicos y legales.

Finalmente se plantearán los requerimientos, en base a los cuales el sistema deberá ser diseñado. Una vez terminado esto, se dará paso a la siguiente fase del desarrollo del sistema que se muestra en el siguiente capítulo.

En el Capitulo 3, se realiza el Diseño del Sistema, y este inicia con la identificación de cada uno de los procesos que se van a necesitar para poder cumplir con los requerimientos del mismo. Esta fase permite el diseño de las entradas y de las salidas, la forma con que el sistema va a interaccionar con el medio ambiente, así como la manera en que se relacionan los archivos que se utilizarán.

Las técnicas utilizadas se basan en el Diseño Estructurado y la Programación Esctructurada, así como en la elaboración de Pseudocódigo, que permite manejar los procesos como módulos, antes de ser codificados en algún lenguaje de alto nivel.

Al término del capítulo, se realiza una guía para la Documentación general del Sistema, de tal forma que nos ayude posteriormente para los planes de Mantenimiento, además de permitir verificar constantemente el Desarrollo del Sistema.

En el Capitulo 4, la fase de Desarrollo tiene la finalidad de codificar los módulos, ya que se traducen las especificaciones de proceso para cada módulo descritas en la fase anterior, en instrucciones ejecutables de un lenguaje de alto nivel. También se verifica el correcto funcionamiento de cada uno de los módulos, así como la integración funcional de cada uno de

ellos.

Ya al último, quedaría la etapa de las pruebas del Sistema, las cuales se harían para encontrar errores de análisis, de especificación y diseño.

El Capítulo 5, presenta la última fase del proceso de programación, y es en donde se reflejan los aciertos o los errores de las fases previas. Esta etapa se caracteriza por la creación del Manual del Usuario, que es el documento que sirve para instalar y operar el sistema. Otra parte importante es el concepto de Mantenimiento, que consiste en proponer los cambios necesarios en la programación, para que el sistema pueda ser implementado con los requerimientos de otro usuario, mismos que pueden ser diferentes en cuanto a las definiciones de los mismos.

En el Capítulo 6, se describe una evaluación en cuanto al funcionamiento del sistema, tratando de establecer si se cumplieron con los objetivos que se plantearon desde un inicio, con los requerimientos que se propusieron, y si el sistema es lo bastante versátil para poder ser implementado en otros equipos de microcomputación.

Un vez terminado con los capítulos referentes a la metodología para la elaboración del sistema, se presentan dos apéndices. Uno que describe el equipo de cómputo con el que fué implementado el Sistema de Banco de Horas,

haciendo referencia a las características de Hardware y Software. El otro, se dedica a describir las capacidades de Estandarización del lenguaje utilizado, poniendo en relieve, la posibilidad de ser transportado a otro equipo.

Finalmente, para concluir este trabajo, se muestra la bibliografía utilizada en la elaboración de esta tesis.

DESARROLLO DE LA INFORMATICA

De todos los inventos desarrollados por el hombre, la computadora electrónica destaca muy por encima de los demás. Y es que en solo unos años ha logrado avances gigantescos que comparados con las mejoras realizadas a los automoviles desde su creación a inicios de este siglo, estos resultan insignificantes.

Las primeras computadoras eran máquinas que servían para realizar calculos a gran velocidad y que con su aspecto imponente, estaban reservadas solamente a los centros industriales y de investigación más importantes.

Hoy en día las computadoras forman parte de nuestra sociedad, pues se les puede encontrar en universidades, empresas y oficinas, desempeñan un importantísimo papel

en las telecomunicaciones y la medicina, así como en la robótica que se emplea para procesos de gran importancia dentro de la industria y en general en todas las ramas que forman parte de las actividades del hombre.

Es así como nuestra sociedad está cambiando, debido a los avances en la tecnología particularmente en la microelectrónica, misma que ha hecho posible un incremento fenomenal en los números y usos de computadoras de todos los tamaños, así como en su capacidad de procesamiento de datos.

Varios eventos claves contribuyeron al surgimiento y dominio de la computación. Esto se muestra en la figura 1 que se encuentra en la página 13.

Esta figura enfatiza no solo la velocidad con que ha estado cambiando la tecnología, sino también el avance ocurrido en el desempeño del procesamiento de información por unidad de costo.

Se menciona previamente el modelo matemático para la computadora de programa interno de Von Newman, misma que consiste básicamente de un procesador central, una memoria y conexiones entre ambos, que transmiten unidades de información. Esta máquina así como otras de su época fueron posibles gracias a la invención del bulbo. Estas mismas pudieron ejecutar cálculos en milisegundos, en vez de los segundos requeridos por dispositivos mecánicos. A estas computadoras se les hace referencia

como computadoras de la primera generación y el periodo aproximado de su uso fue de 1950 a 1954.

Pero no solo fueron voluminosas, también eran poco confiables; los miles de valvulas de vacío emitían grandes cantidades de calor y frecuentemente fallaban. El transistor un pequeño y más confiable sucesor del tubo de vacío fue inventado en 1947. Sin embargo, computadoras que usaron transistores no fueron producidas en cantidad sino una década después.

A estas computadoras se les llamó también como de la segunda generación, las cuales usaron grandes cantidades de transistores, dominando de 1959 a 1964 el mercado de procesamiento de datos. Estas máquinas estaban habilitadas para reducir calculos de milisegundos a microsegundos.

Un evento clave fue la introducción de computadoras de propósito general en 1964. Estas máquinas fueron diseñadas para satisfacer las necesidades de procesamiento de datos para la ciencia y la industria.

El desarrollo de la microelectronica hizo posible reducir el tamaño de los transistores e integrar gran cantidad de elementos de circuitos en pequeños "chips" de silicón. Las computadoras fueron diseñadas para usar la tecnología de los circuitos integrados (CI) siendo llamadas computadoras de la tercera generación y el

tiempo aproximado asociado con esas maquinas fue de 1956 a 1979. Ellas podían ejecutar muchas operaciones de procesamiento de datos en nanosegundos.

Se ha progresado de circuitos de pequeña escala (SSI) los cuales contenían unos pocos transistores, por medio de la integración a gran escala (LSI) en integración a muy larga escala (VLSI) mismos que tienen arriba de 10,000 transistores.

En la actualidad se observa que los avances de la microminiaturización podrán hacer en un futuro próximo alcanzar en avance para que el concepto de generaciones de computadoras vuelva a aplicarse para medir el progreso en la potencialidad de la computadora durante todo lo que resta del siglo.

La Tabla 1 de la página 14 muestra la reducción en el costo que están teniendo los componentes de las computadoras. Pero no solamente ha disminuido su costo, sino que ha aumentado simultáneamente su capacidad reduciéndose al mismo tiempo su tamaño, de ahí que la industria de la computación tenga un crecimiento tan espectacular.

Aunque todo esto hable maravillas de la computación existe un hecho que ensombrece su futuro, ya que a inicio de la década pasada se detectó serios problemas que no permitían que se aprovecharan las computadoras debidamente: la carencia de recursos humanos, los altos

costos y constantes retrasos en el desarrollo de sistemas y, finalmente, la construcción de sistemas poco confiables. Todos estos conceptos se pueden resumir en la figura 2 de la página 14.

Así en lo que respecta al desarrollo de sistemas los costos se incrementaban considerablemente sin producir mayores logros, pues la mayor parte de los recursos era absorbido por las labores de mantenimiento de sistemas. Es así que la comunidad informática se esforzó por dar a la labor de desarrollo de sistemas de procesamiento de datos, el carácter de una disciplina ingenieril, en contraste con el estado artesanal en que esta actividad se encontraba.

Esta disciplina tiene la finalidad de investigar el proceso lógico por el cual se construye un sistema para satisfacer cierta necesidad, llegando con ello a la creación de métodos que permitan desarrollar en forma práctica sistemas. Los principales objetivos que persiguen estos métodos son :

- la producción de sistemas confiables con bajos presupuestos y tiempos de desarrollo planeados y controlados.

- la reducción de los costos de mantenimiento de sistemas.

- la portabilidad de los sistemas desarrollados con un mínimo de cambios.

-la eficiencia del sistema.

La metodología que se seguirá en esta obra es el llamado "ENFOQUE DE SISTEMAS", la cual prevé la división del proceso de desarrollo de programación en varias fases que siguen una secuencia temporal como sigue:

- 1.- Analisis.
- 2.- Diseño.
- 3.- Desarrollo.
- 4.- Operación y Mantenimiento.
- 5.- Evaluación del Sistema.

incremento en la
potencialidad

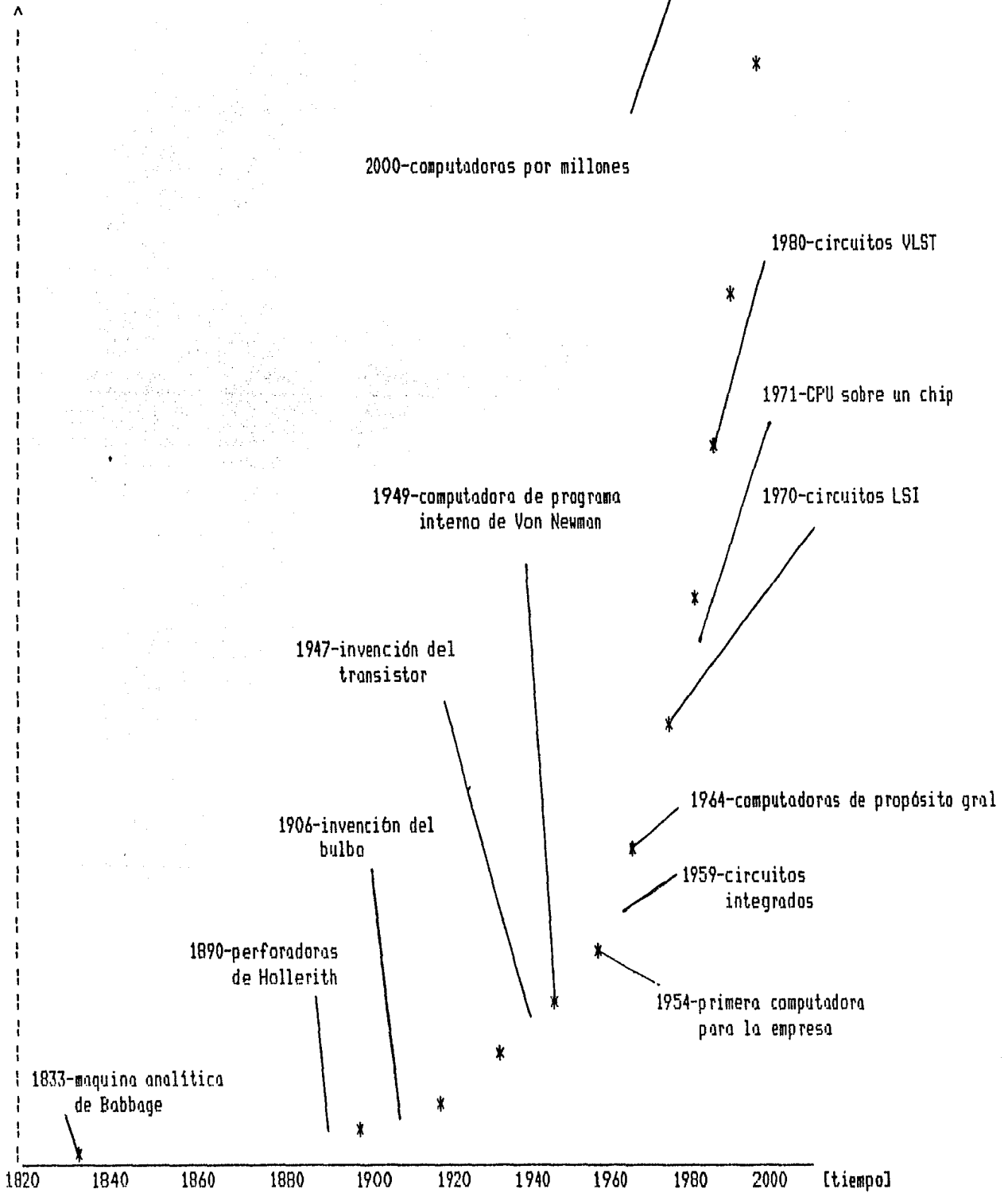


FIGURA 1

AÑO	TECNOLOGIA	\$/ELEMENTO DE CIRCUITO	\$/CIRCUITO LOGICO	TIEMPO DE OPERACION
1958	Tubo de Vacio	\$6.00	\$160.00	16 mseg
1965	Transistor	\$0.25	\$ 12.00	4 useg
1972	Circuitos Integrados	\$0.02	\$200.00	40 nseg
1980	Circuitos Integrados a Larga Escala	\$0.001	\$ 0.05	0,2 nseg

TABLA 1

EVOLUCION DEL PRECIO DEL EQUIPO Y LA PROGRAMACION

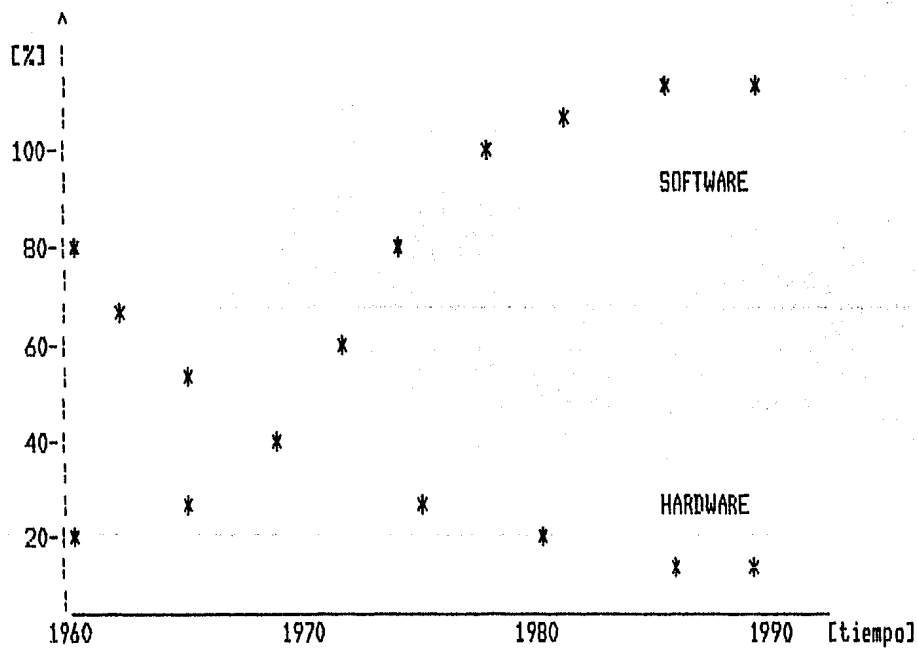


FIGURA 2

A N A L I S I S

La primera fase del desarrollo de un sistema, es la de Análisis, y ésta consiste en un planteamiento inicial de los objetivos con los cuales deberá cumplir el sistema que se desarrolle, para después hacer un análisis de las posibles soluciones y finalmente, se procederá a la definición de los requerimientos para el Sistema.

PLANTEAMIENTO DE LOS OBJETIVOS .- Con esta parte, se inicia la elaboración de un sistema, esta formada por el planteamiento adecuado de los objetivos que deberá cumplir el producto final. Efectivamente, los sistemas no pueden ser creados sin una documentación que describa cabalmente las funciones que deberá tener el sistema a desarrollar.

Dentro del Sistema de Banco de Horas que se va a

realizar tenemos que los objetivos identificados son los que siguen:

a) Diseñar un Sistema que permita llevar el control de personal docente, que labore en escuelas y facultades dentro de la universidad.

b) El sistema deberá tener la habilidad de dar información vía terminal así como la emisión de reportes que reflejen el estado de la información dentro del sistema.

c) El sistema deberá tener la facilidad para poder ser adaptado a las necesidades de diferentes usuarios dentro de la universidad.

d) Este sistema deberá tener la característica de poder ser fácilmente transportado a otro equipo de cómputo que sea considerado Estandard (de preferencia Microcomputadora).

A N A L I S I S .- El propósito de ésta etapa es el de la elaboración de alternativas que conlleven a la realización de los objetivos planteados en la etapa anterior y que, en función de las mismas se decidirá el rumbo que se seguirá en la elaboración del sistema (si se estima que es conveniente).

Entre las alternativas propuestas se pueden observar

las siguientes:

- a) Elaborar el Sistema Banco de Horas en un Lenguaje convencional estandarizado.
- b) Utilizar un manejador de base de datos para microcomputadora.
- c) Cancelar la elaboración del sistema.

Para la evaluación de las alternativas se utilizaron herramientas como el análisis de factibilidad así como de un diagrama de flujo de datos, que nos sirve para visualizar la forma en que el sistema deberá controlar toda la información que entra y sale dentro del mismo.

Analizando los objetivos de los sistemas se puede dar un bosquejo general de como debe estar configurado el sistema, en base a un diagrama de flujo de datos y que se muestra en la figura 3 de la página 18.

ANALISIS DE FACTIBILIDAD .- El propósito del análisis de factibilidad, es seleccionar la mejor alternativa que cumpla con los objetivos planteados en la fase anterior.

Este estudio de factibilidad se concentra en cuatro áreas generales descritas a continuación:

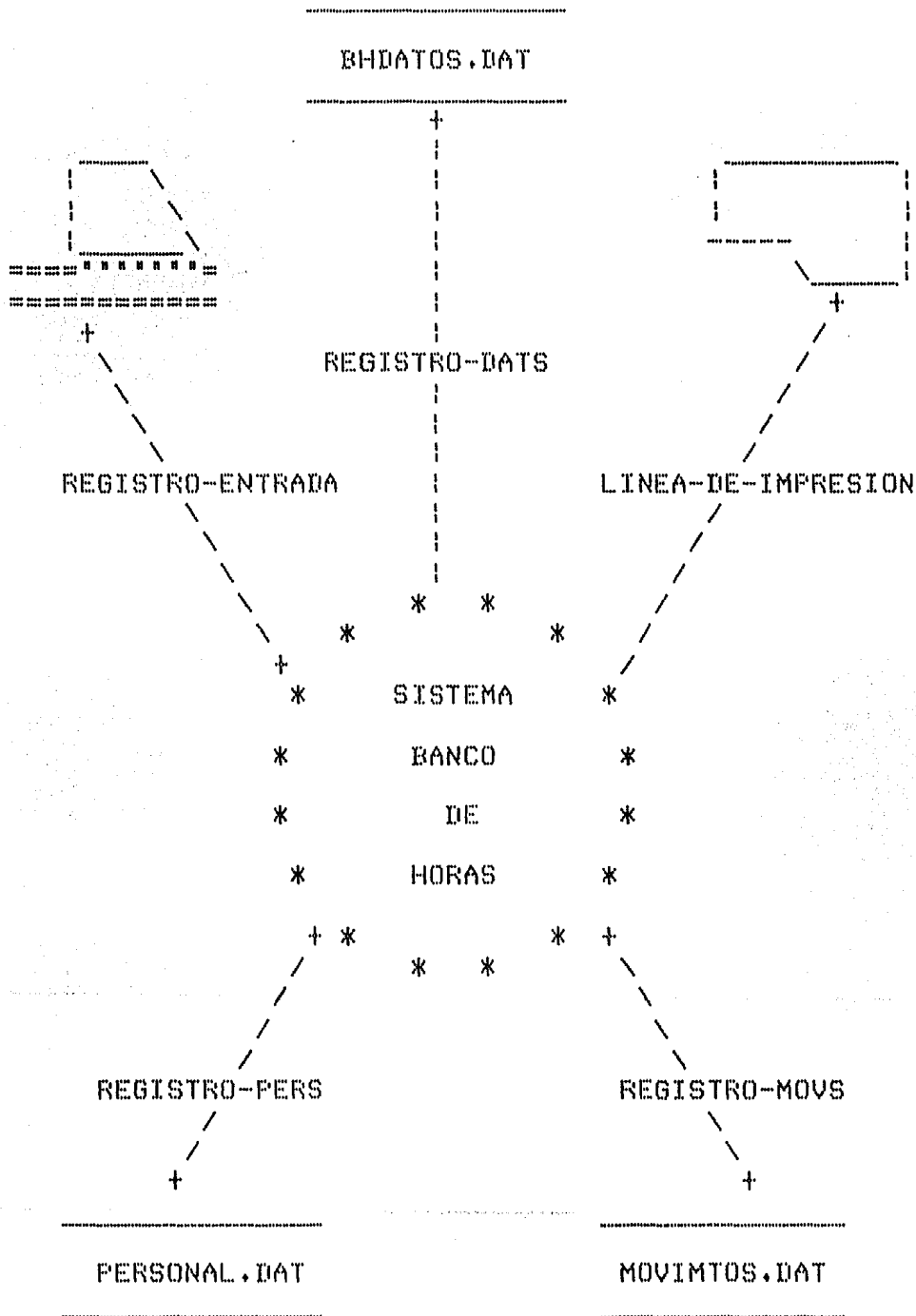


FIGURA 3

-Factibilidad Económica o Análisis de Costo-Beneficio .- Este análisis es complejo y depende mucho de las características del sistema, además cuando el proyecto tiene una obvia justificación económica, no es recomendable aplicársele, como es el caso del Sistema de Banco de Horas, pues se está considerando que no existen otros paquetes que cumplan los con objetivos planteados, en cambio, este sistema debe ser implementado a nivel de dependencia, la cual deberá contar con el equipo necesario para poderse implementar.

-Factibilidad Técnica .- En cuanto a esta parte, no hay ninguna restricción en cuanto al equipo de cómputo, pues existen tanto un equipo Cromemco System Three (256K de memoria y disco duro) como equipos PC (con memoria de 512K y disco duro), con su respectivo Software, con los cuales deberá ejecutarse el Sistema. Los recursos humanos están limitados a un Analista-Programador de Sistemas que tendrá a su cargo todo el proceso desde el análisis hasta la liberación del sistema .

-Factibilidad Legal .- No existe ninguna restricción en el manejo del Sistema Banco de Horas, debido a que su utilización será dentro de la universidad.

-Alternativas .- En esta parte se hará una evaluación de cada una de las alternativas para escoger la que mejor se adapte a los objetivos del sistema.

Tomando la primera alternativa, observamos que el sistema a desarrollarse tendría que ser complejo, dado que la implementación en un lenguaje de alto nivel llevaría determinado tiempo de desarrollo.

En cuanto a la segunda alternativa, se tendría la posibilidad de desarrollar el sistema en menor tiempo que la primera alternativa, pero involucraría estudiar algunos de los paquetes ya existentes en el mercado para su evaluación, y una vez considerado un paquete el adecuado, tomar un curso para la utilización correcta del manejador de datos escogido.

Estudiando la alternativa uno respecto a la alternativa dos, tenemos:

a) La alternativa 1 es inmediata, mientras que la alternativa 2, se necesita realizar un estudio para la correcta evaluación de los administradores de bases de datos, para que esta sea compatible con el Sistema Operativo Cromix y el MS-DOS.

b) La alternativa 1 tiene la posibilidad de modularizar el Sistema, por lo tanto, no sería impedimento la limitación de memoria, pues en el sistema tres de Cromemco con un procesador central Z80, cuenta con solamente 64K de memoria para usuario, mientras que en MS-DOS permite expandirla hasta 8 veces la misma cantidad de bytes, dependiendo de la configuración. En

cuanto la alternativa 2, esta restringida, pues muchos administradores de bases de datos para microcomputadora, necesitan cuando menos 128K de memoria para poder utilizarlas.

Existe DBASE II, la cual se puede correr tanto en una Cromemco como en una PC, únicamente se tendrían que transferir los programas de una máquina a otra, pero existe una limitante en cuanto a utilizar éste paquete, y es que es un intérprete, provocando con ello una lentitud en el tiempo de respuesta del sistema al manejar grandes volúmenes de información; aparte de tener que manejar siempre los programas fuentes y que para correr una aplicación, se tiene que entrar primero a DBASE II, provocando con ello confusión a los usuarios.

Además DBASE II no permite manejar más de dos archivos a la vez durante la ejecución de un proceso. Si se toma en cuenta que el tercer objetivo del sistema que indica que el sistema debe ser fácilmente adaptado a las necesidades de cualquier usuario, no se cumpliría cabalmente con esta opción. Todo esto provocaría que el mantenimiento a los programas se tornará tedioso, pues se tendría que corregir cada uno de los módulos del sistema cada vez que se tuviera que adaptar a una dependencia.

En cuanto a la alternativa 3, esta es infactible, pues las necesidades en dependencias universitarias de llevar un paquete de Banco de Datos es tal, que les

permite llevar un mejor control de los gastos efectuados por el personal académico, permitiendo con ello, un mejor manejo y planeación de los recursos asignados al área académica.

De todo lo dicho anteriormente, se puede concluir que la alternativa 1, es la mejor alternativa que cumple con los objetivos planteados para el sistema, además de poder ser aplicada de inmediato.

ESPECIFICACION DE LOS REQUERIMIENTOS .- La función de esta sección es la de especificar de una manera formal y objetiva las cualidades deseadas en el producto final, así como el medio ambiente en el cual el usuario lo operará.

Dado que ya se conocen las funciones y los métodos de solución que deberán ser empleados para satisfacer las necesidades del usuario, se generarán los requerimientos con los que debe cumplir el Sistema que se Implementará.

Una manera eficiente de caracterizar esta parte, es por medio de un modelo conceptual, el cual es una representación de la especificación de requerimientos del usuario, así tenemos que el sistema estaría definido como se observa en la figura 4.

Así, teniendo identificado cada proceso, podemos definir los siguientes requerimientos:

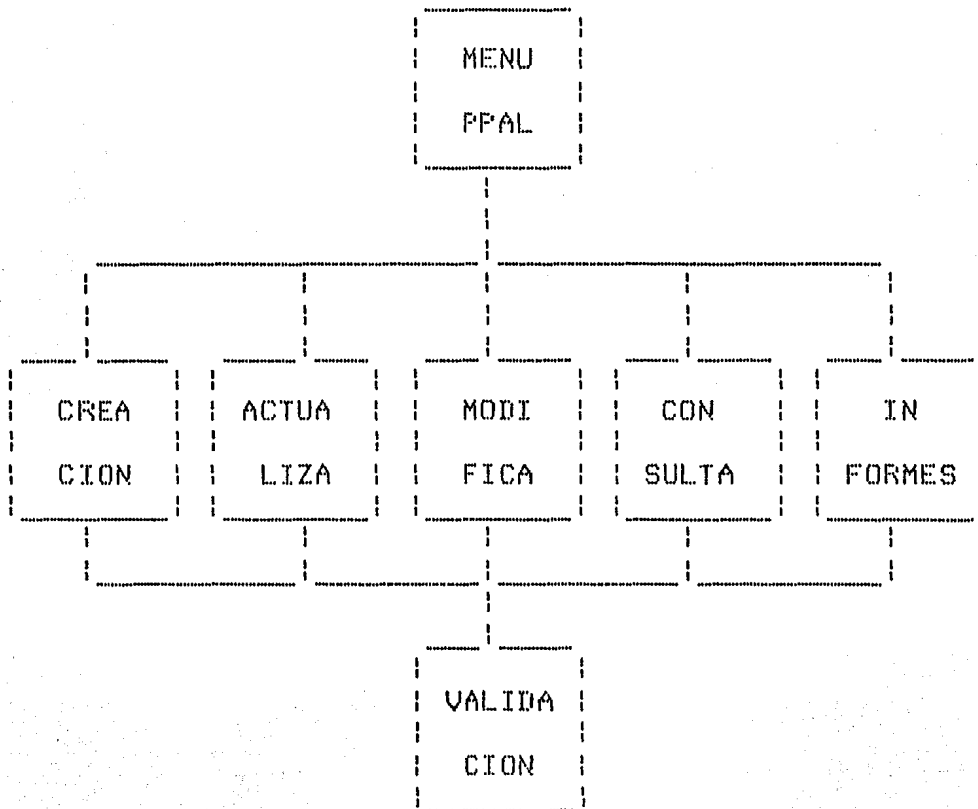


FIGURA 4

-EJECUCION-

El programa se podrá ejecutar por medio del comando
SISTBH

Este comando producirá el desplegado del Menú

Principal, el cual tendrá las tareas generales del sistema.

Dentro de las tareas que se mostrarán en el Menú, se tendrá la posibilidad de crear el banco de datos, la actualización del mismo, el que se pueda realizar la consulta de los datos en forma interactiva y la contemplación de la generación de reportes del banco de datos.

-CREACION-

En este proceso, se deberá tener la habilidad de crear el Banco de Datos en dos formas. La primera por medio de un archivo conteniendo todos los datos personales y de movimientos y la otra de modo conversacional.

El primer modo tendrá la forma de creación en lote de datos, que serán leídos por el sistema realizando la validación de cada uno de los registros, y en caso de que esté correcto, será dado de alta en el banco de datos. También se deberá generar un archivo con el resultado de la validación de cada uno de los registros de datos que entren al sistema. Una vez que se termine de procesar toda la información, se habrán creado dos archivos, uno conteniendo los registros que estuvieron correctos llamado "BMBUENAS.DAT" y otro, con los registros que

tuvieron algun error llamado "BHMALAS.DAT". Finalmente, se deberá regresar al Menú Principal una vez que haya terminado todo este proceso.

La segunda forma de creación, consistirá en la captura de la información por medio de la pantalla, este permitirá ir capturando cada uno de los datos que el sistema vaya pidiendo. Esta captura se realizará en dos partes. En la primera, se desplegará una forma que permitirá la captura de los datos personales del profesor; junto a esta captura se irá validando cada uno de los campos que se intenten meter, y en caso de algún error, el sistema desplegará de inmediato un mensaje indicando la causa de la anomalía.

Una vez que se terminan de capturar los datos personales correctamente, se desplegará la segunda forma de captura para los movimientos que tendrán cada uno de los profesores, éstos se capturarán de igual manera que la descrita en el parrafo anterior. Para finalizar la entrada de datos bastará con teclear FIN, en vez de capturar el lugar de adscripción.

Ya que se terminó la captura de movimientos, se volverá a desplegar la primera forma, para así volver a teclear otro empleado, repitiendose nuevamente el proceso descrito anteriormente. En caso de que no se quiera teclear más empleados, se podrá salir por medio de FIN, capturandolo en lugar del RFC, regresando al Menú

Principal.

Otra característica que deberán tener cada uno de los tipos de creación, será el de poder dar de alta en cualquier momento un profesor con sus respectivos movimientos, sin necesidad de tener que volver a ejecutar todo el proceso de creación del banco de datos.

-ACTUALIZACION DEL BANCO DE HORAS-

Para ésta tarea, se deberá dar el RFC del empleado a actualizar con nuevos movimientos, en caso de no existir el RFC en el banco de datos, el sistema desplegará un error indicándolo e iniciando nuevamente la captura del RFC.

Una vez que el RFC es válido, se procede a la captura de los nuevos movimientos que serán añadidos al empleado. Cuando se termine de capturar, deberá salirse de esta tarea por medio del comando FIN, tecleado en el campo del lugar de adscripción, regresando al menú principal para realizar algún otro proceso.

-MODIFICACIONES AL BANCO DE HORAS-

El Sistema deberá tener la habilidad de hacer cambios en el Banco de Datos. Estos deberán realizarse cada vez que se requiera. La forma de entrada será por

medio del RFC, en caso de no existir en el Banco, se emitirá un mensaje de error, y se deberá de intentar de nueva cuenta.

Cuando el sistema haya podido encontrar el RFC requerido, se procederá a desplegar un menú, donde se identificará, a qué se le va a hacer cambios. (Datos Personales o Movimientos). Después se escogerá la forma en que se modificará el Banco por medio de otro menú, el cual mostrará el tipo de modificación: Cancelación o Corrección.

Todas las correcciones se realizarán en forma interactiva, mostrando cada uno de los campos y validando los nuevos datos que se intenten introducir, y en caso de existir algún error, el sistema lo hará saber por medio de un mensaje.

Si se realiza una cancelación de un RFC, el sistema deberá cancelar todos los movimientos que tuviera el profesor que se le esta dando de baja.

-CONSULTA-

El sistema deberá permitir el acceso a la información del banco de horas por medio de un menú. Este deberá mostrar los tipos de consulta que podrá realizar el sistema.

Deberá tener una consulta por datos personales, que muestre toda la información concerniente a los datos personales del profesor. Además deberá permitir la consulta de cada uno de los movimientos del mismo.

Para iniciar la consulta, se deberá validar que exista el RFC, en caso contrario se regresará al menú de la consulta.

-GENERACION DE INFORMES-

El sistema deberá generar reportes, que muestren el estado de la información contenida en el banco de datos. Estos reportes se podrán obtener a partir de un Menú, mismo que indicará el tipo de informe que se requiera obtener en ese momento.

Cada uno de los reportes deberá tener la posibilidad de poderse generar en forma global (que no tome en cuenta el nivel de adscripción) o por niveles (que considere los niveles de adscripción).

Para cada reporte se pedirá la fecha con la cual se emitirá el informe, esta será validada para que no ocurra algún problema durante la ejecución del reporte.

El primer informe que se obtendrá, será un Directorio que nos permita conocer los datos personales de cada uno de los profesores; este reporte nos mostrará

además el número total de horas que imparte por todas sus asignaturas. Este informe estará sorteado por el RFC del personal, además, si es por niveles, se añadirán los datos referentes a los niveles de adscripción, en caso contrario se omitirán.

Si el informe que se pide es por niveles, se permitirá que se repitan RFC's, pero deberá ser único para cada uno de los niveles que se obtengan.

El segundo informe nos permitirá conocer cada uno de los movimientos vigentes por el profesor en forma detallada. El listado obtenido estará ordenado por registro federal de causantes en caso de que sea solicitado globalmente, o en base a los tres niveles de la estructura de la dependencia usuaria y RFC's respectivamente.

Este informe nos deberá de dar adicionalmente información del total de horas que imparte cada uno de los distintos empleados académicos.

-VALIDACION-

La validación de la información se dará toda vez que se quiera meter información al Banco de Datos y estará presente en cada uno de los módulos que componen el Sistema, ejecutándose cada vez que se requiera. Esta

validación será por medio de tablas, las cuales estarán formadas por los catálogos que se deberán definir previamente. Esto será definido por el usuario y su utilización será de uso exclusivo.

-SALIDA DEL SISTEMA-

Para salirse del sistema, se deberá estar en el Menú Principal y habrá una opción que permite la salida del sistema, retornando al ambiente del sistema operativo del equipo en que se encuentra el Sistema.

Finalmente, en base a las especificaciones que se describieron en ésta sección, podemos resumirlas en un diagrama de flujo de datos, que se muestra en la siguiente hoja.

DIAGRAMA DE FLUJO DE DATOS DEL SISTEMA BANCO DE HORAS

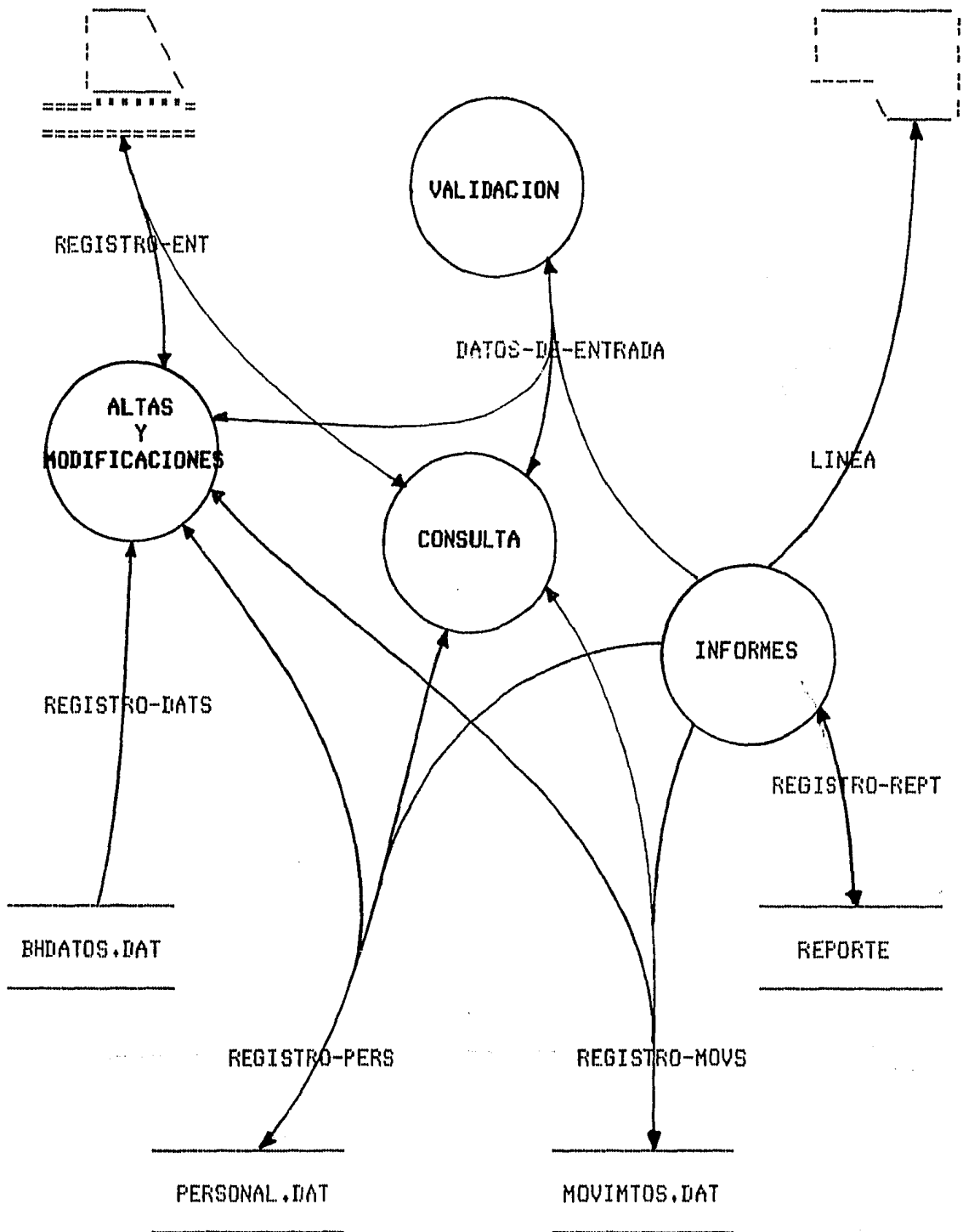


FIG. 5

D I S E Ñ O

La fase central en el desarrollo de un sistema, es la etapa de diseño. Esta parte es un proceso a través de la cual los requerimientos son trasladados de una representación esquemática, a una serie de instrucciones de algún lenguaje que pudiera ser fácilmente traducido por la computadora.

Una vez que se haya terminado de especificar los requerimientos para el sistema, se procede a la elaboración de las actividades del proyecto en forma extensa y detallada.

Para el modelo del Sistema Banco de Horas, se utilizó la convención del Análisis Estructurado, este modelo se logra con el uso de un "diagrama de flujo de datos" y de un "diccionario de datos".

Un diagrama de flujo de datos es una representación en forma de red de un sistema. Muestra al mismo en términos de los componentes de los procesos y declara todas las interfaces entre las partes que lo integran.

Por diccionario de datos, se entiende como un conjunto de definiciones de las interfaces declaradas en el diagrama de flujo de datos. Se define cada una de esas interfaces en términos de sus componentes.

Con estas herramientas, se procedió a la elaboración del diseño del Sistema Banco de Horas, mismo que se explica en las siguientes secciones de este trabajo.

DISEÑO DE LAS ENTRADAS .- En esta parte, se definen los procesos que permiten la comunicación entre el sistema y el exterior, (generalmente dispositivos manejados por seres humanos). Se debe tomar en cuenta el elemento humano en orden, para asegurar una rápida y exacta entrada de datos de un documento fuente, o bien de la misma pantalla de la microcomputadora. Todo esto permitirá la prevención de errores y una rápida entrada de datos.

Esta comunicación, para nuestro sistema se realiza de dos modos, el primero por medio de la pantalla, la cual muestra una serie de formas de entrada, estas

dependen de la utilización que en ese momento se le este dando al sistema, nos permitirán introducir información al mismo. En esta entrada se darán los datos que serán utilizados para la creación del banco de datos, así como de los datos que actualizarán al mismo, ya sea mediante modificaciones a los registros o mediante nuevos movimientos.

La segunda entrada que será reconocida por el sistema será por medio de un archivo, donde estarán contenidos los datos para la creación o actualización del banco de datos, este archivo será identificado como:

"BH DATOS.DAT".

Resumiendo las entradas del sistema en un diagrama de flujo de datos, obtendremos el esquema que se muestra en la figura 7 de la siguiente página.

Definiendo las entradas por pantalla mediante un diccionario de datos, tendremos lo siguiente:

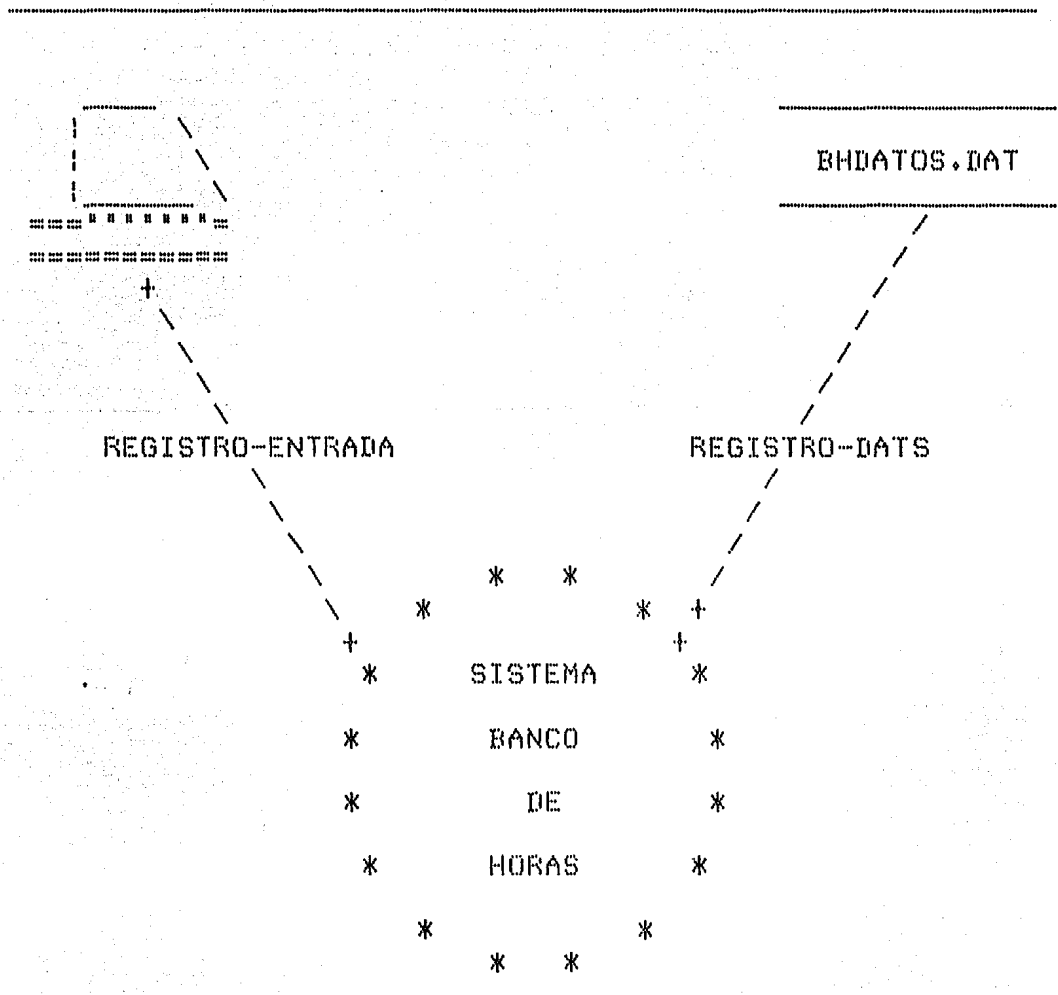


FIGURA 6

REGISTRO-ENTRADA = * Formas para introducir información al sistema *

= [REGISTRO-CREACION ;
REGISTRO-ACTUALIZACION ;
REGISTRO-MODIFICACION],

REGISTRO-CREACION = * Formas que permiten la alta de personal académico *

= FORMA-1-C +
FORMA-2-C,

FORMA-1-C = * Datos personales a darse de alta *

= ENT-RFC +
[DATOS-PARTE-1 ;
DATOS-PARTE-2],

ENT-RFC = * Registro Fed. de Causantes del profesor *

= RFC-ANIO +
RFC-MES +
RFC-DIA +
RFC-LETRAS,

RFC-ANIO = * Año válido *

= ['10'..'68'],

RFC-MES = * Mes válido *

= ['01'..'12'],

RFC-DIA = * Día válido *

= ['01'..'31'],

RFC-LETRAS = ['A'..'Z'],

DATOS-PARTE-1 = ENT-FOLIO +
ENT-NOMBRE +

ENT-TELEFONO.

DATOS-PARTE-2 = ENT-DOMICILIO +
 ENT-COLONIA +
 ENT-CODIGO-POSTAL +
 ENT-FE-INGRESO +
 ENT-ANTIG-UNAM +
 ENT-TP-PERSONAL +
 ENT-NACIONALIDAD +
 ENT-SEXO +
 ENT-ESTADO-CIVIL +
 ENT-FE-ACTUALIZA.

ENT-TP-PERSONAL = * Actividad desarrollada en la
 UNAM *
 = ['1' | '2' | '3']

ENT-NACIONALIDAD = * Nacionalidad del empleado *
 = ['0' * mexicano *
 | '1' * extranjero *].

ENT-SEXO = * sexo del empleado *
 = ['F' * femenino *
 | 'M' * masculino *].

ENT-ESTADO-CIVIL = * edo. civil del empleado *
 = ['S' * soltero *
 | 'C' * casado *
 | 'V' * viudo *
 | 'D' * divorciado *
 | 'X' * otro *].

FORMA-2-C = * Captura de los movimientos que
 tenga el profesor *
 = DATOS-PARTE-3.

DATOS-PARTE-3 = ENT-ADSCRIPCION
 ENT-CARRERA
 ENT-TP-MOVIMTO
 ENT-CAUSA
 ENT-CATEGORIA
 ENT-PLAZA
 ENT-FE-INICIO

ENT-FE-TERMINO
 ENT-ASIGNATURA
 ENT-GRUPO
 ENT-SALON
 ENT-DIAS
 ENT-HR-ENTRADA
 ENT-HR-SALIDA
 ENT-HRS-TEORCAS
 ENT-HRS-PRACTCAS
 ENT-HRS-APOYO
 ENT-COD-PROGMTCO.

ENT-TP-MOVIMTO = * Tipo de movimiento *
 = ['1' * Alta *
 | '2' * Baja *
 | '3' * Licencia con goce
 de sueldo *
 | '4' * Licencia sin goce
 de sueldo * 1.

ENT-CAUSA = * Causas que maneja el
 sistema *
 = ['A' * Nuevo ingreso *
 | 'B' * Reingreso *
 | 'C' * Prorroga *
 | 'D' * Otro Nombram. *
 | 'E' * Reenumerac. Adic. *
 | 'F' * Aumento de Horas *
 | 'G' * Promoción *
 | 'H' * Reanudac. Labors *
 | 'I' * Interinato *
 | 'J' * Reclasificación *
 | 'K' * Transferencia *
 | 'L' * Disminución Hrs *
 | 'M' * Enfermedad *
 | 'N' * Gravidéz *
 | 'O' * Año Sabatico *
 | 'P' * Comisión Oficial *
 | 'Q' * Revocación *
 | 'R' * Suspensión *
 | 'S' * Renuncia *
 | 'T' * Defunción *
 | 'U' * Est. en el Pais *
 | 'V' * Est. en el Ext. *
 | 'W' * Particular *
 | 'X' * Rescisión *
 | 'Y' * Prorroga por
 comb. de Materia *

1.

ENT-CATEGORIA = * Categoria del profesor *
 = ['01' * Prof. Asig. A *
 | '02' * Prof. Asig. B *
 | '03' * Ayud. Prof. A *
 | '04' * Ayud. Prof. B *
].

ENT-PLAZA = * Tipo de plaza *
 = ['1' * Interina *
 | '2' * Definitiva *].

ENT-DIAS = ['1' | '0' | ' '] * lun *
 + ['2' | '0' | ' '] * mar *
 + ['3' | '0' | ' '] * mie *
 + ['4' | '0' | ' '] * jue *
 + ['5' | '0' | ' '] * vie *
 + ['6' | '0' | ' ']. * sab *

ENT-COD-PROGMTICO = * Niveles del codigo progmtico *
 = PROGRAMA +
 SUBPROGRAMA +
 DEPENDENCIA +
 SUBDEPENDENCIA +
 PARTIDA +
 DIGITO DE CONTROL.

REGISTRO-ACTUALIZACION = * Alta de nuevos movimientos por RFC *
 = FORMA-2-A.

FORMA-2-A = * Campos que de la FORMA-2-C *
 = FORMA-2-C.

REGISTRO-MODIFICACION = * Tiene la función de cancelar o corregir información *
 = FORMA-1-M +
 FORMA-2-M.

FORMA-1-M = * Esta forma contiene los mismos campos que la forma FORMA-1-C *

= FORMA-1-C,
 FORMA-2-M = * Esta forma contiene los
 mismos campos que la forma
 FORMA-2-C *
 = FORMA-2-C.

El segundo tipo de entrada, es por medio de un
 archivo de datos, este archivo contiene la información
 necesaria para dar de alta a un profesor con sus
 respectivos movimientos. Estos datos se definen como
 sigue:

REGISTRO-DATS = * Alta de profesores *
 = DATS-RFC +
 DATS-NUMERO-REGISTRO +
 DATS-REGISTRO.

DATS-RFC = ENT-RFC

DATS-REGISTRO = * Registro de datos concernientes
 a datos personales y movimientos
 para cada profesor *
 = [REGISTRO-1 ;
 REGISTRO-2 ;
 REGISTRO-N].

REGISTRO-1 = * Datos personales del empleado *
 = DATOS-FOLIO +
 DATOS-NOMBRE +
 DATOS-TELEFONO.

REGISTRO-2 = * Continuación de los datos
 personales *

= DATOS-DOMICILIO +
DATOS-COLONIA +
DATOS-CODIGO-POSTAL +
DATOS-FE-INGRESO +
DATOS-ANTIG-UNAM +
DATOS-TP-PERSONAL +
DATOS-NACIONALIDAD +
DATOS-SEXO +
DATOS-ESTADO-CIVIL +
DATOS-FE-ACTUALIZA.

REGISTRO-N

= * Datos de movimientos *

= DATOS-ADSCRIPCION +
DATOS-CARRERA +
DATOS-TP-MOVIMTO +
DATOS-CAUSA +
DATOS-CATEGORIA +
DATOS-PLAZA +
DATOS-FE-INICIO +
DATOS-FE-TERMINO +
DATOS-ASIGNATURA +
DATOS-GRUPO +
DATOS-SALON +
DATOS-DIAS +
DATOS-HR-ENTRADA +
DATOS-HR-SALIDA +
DATOS-HRS-TEORCAS +
DATOS-HRS-PRACTCAS
DATOS-HRS-APOYO +
DATOS-COD-PROGMTCO.

DISEÑO DE SALIDAS .- Esta etapa se inicia cuando se están especificando los requerimientos del sistema, y es cuando se elaboran las salidas que tendrá el sistema en detalle. En este diseño, se tiene que especificar las diferentes presentaciones de la información; la exacta localización de todas las líneas de caracteres, el espaciado entre líneas, etc. Para el Sistema Banco de Horas, se tienen dos formas de salida, al igual que en las entradas del sistema.

El primer modo de salida estará dado por pantalla y esta se utiliza durante el proceso de consulta y el de modificaciones, ya que en el primero se muestra la información que se desea consultar, mientras que para el proceso de modificación se utilizan para verificar que los datos desplegados sean los que se tienen que cancelar o corregir.

El segundo tipo se conforma por la impresión de los informes que se desean obtener, estos reportes están definidos primero por Datos Personales y en segundo lugar, el Detallado de los movimientos que tenga cada profesor.

Haciendo un Diagrama de Flujo de Datos que nos muestre gráficamente las salidas que deberá tener el sistema, obtendríamos el esquema que se muestra en la

figura 8.

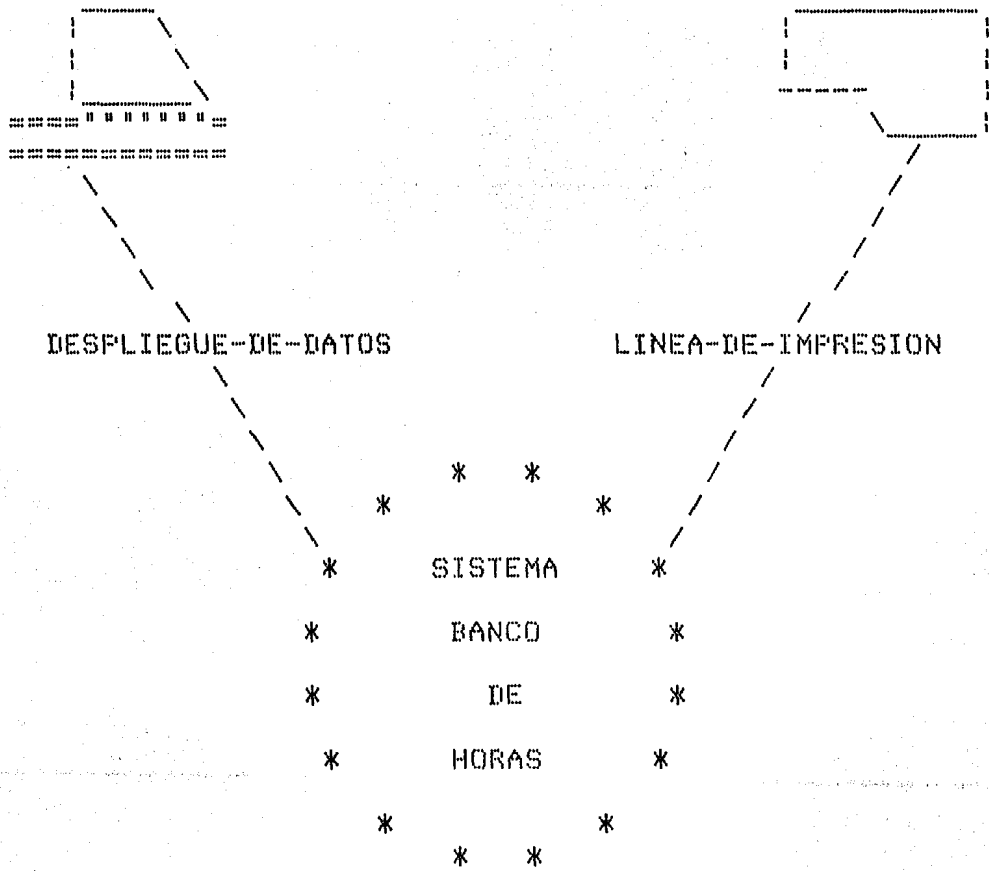


FIGURA 8

DESPLIEGUE-DE-DATOS = * Información del sistema a la terminal *

= [DESPL-CONSULTA ;
DESPL-MODIFICACIONES],

DESPL-CONSULTA = * Despliega tipo de información desenda *

= [DESPL-DATOS-PERSONALES-C ;
DESPL-MOVIMIENTOS-C],

DESPL-DATOS-PERSONALES-C = * Información referente a los datos personales *

= DESPL-RFC +
DESPL-FOLIO +
DESPL-NOMBRE +
DESPL-TELEFONO +
DESPL-DOMICILIO +
DESPL-COLONIA +
DESPL-CODIGO-POSTAL +
DESPL-FE-INGRESO +
DESPL-ANTIG-UNAM +
DESPL-TP-PERSONAL +
DESPL-NACIONALIDAD +
DESPL-SEXO +
DESPL-ESTADO-CIVIL +
DESPL-FE-ACTUALIZA,

DESPL-MOVIMIENTOS-C = * Despliegue de movimientos para de los profesores *

= DESPL-RFC +
DESPL-ADSCRIPCION +
DESPL-CARRERA +
DESPL-TP-MOVIMTO +
DESPL-CAUSA +
DESPL-CATEGORIA +
DESPL-PLAZA +
DESPL-FE-INICIO +
DESPL-FE-TERMINO +
DESPL-ASIGNATURA +
DESPL-GRUPO +
DESPL-SALON +
DESPL-DIAS +

DESPL-HR-ENTRADA +
 DESPL-HR-SALIDA +
 DESPL-HRS-TEORCAS +
 DESPL-HRS-PRACTCAS +
 DESPL-HRS-APOYO +
 DESPL-COD-PROGMTCO.

DESPL-MODIFICACIONES = * Despliegue de información que se va a modificar *
 = [DESPL-DATOS-PERSONALES-M ;
 DESPL-MOVIMIENTOS-M],

DESPL-DATOS-PERSONALES-M = * En virtud de que este campo tiene la misma estructura que el despliegue de datos personales de la consulta se toman los mismos campos *.
 = DESPL-DATOS-PERSONALES-C.

DESPL-MOVIMIENTOS-M = * Aquí se aplica el mismo criterio que en el caso de arriba, solo que los campos son de movimientos *
 = DESPL-MOVIMIENTOS-C.

Para la información obtenida por medio de los reportes, se define como sigue:

LINEA-DE-IMPRESION = * Salida de los informes generados por el sistema *
 = ENCA-UNAM
 ENCA-FECHA-PAG
 ENCA-BH
 ENCA-DEPENDENCIA +
 ENCA-NIVELES +
 [ENCA-INFORME-1 ;
 ENCA-INFORME-2],

ENCA-NIVELES = * Niveles que componen el lugar de la adscripción *
 = ENCA-NIVEL1 +

[ENCA-NIVEL2 :
ENCA-NIVEL3 :
ESPACIOS].

ENCA-INFORME-1 = * Letreros para el informe del directorio de personal *

= ENCA-SUB-TITULO +
ENCA-REGISTRO.

ENCA-SUBTITULO = 'R. F. C.' +
18C' ']' +
'NOMBRE' +
16C' ']' +
'HORAS' +
17C' ']' +
'DOMICILIO' +
24C' ']' +
'COLONIA' +
7C' ']' +
'C. P.' +
2C' ']' +
'TELEFONO'.

ENCA-REGISTRO = * Datos personales a imprimirse *

= SAL-RFC +
' ' +
SAL-NOMBRE +
' ' +
SAL-TOTAL-HORAS +
' ' +
SAL-DOMICILIO +
' ' +
SAL-COLONIA +
' ' +
SAL-CODIGO-POSTAL +
' ' +
SAL-TELEFONO.

ENCA-INFORME-2 = * Detallado de movimientos a imprimirse *

= ENCA-SUB-TITULO1 +
ENCA-SUB-TITULO2 +
ENCA-SUB-TITULO3 +
ENCA-SUB-TITULO4 +
ENCA-SUB-TITULO5 +
ENCA-REGISTRO +
ENCA-HRS-TEORCAS +
ENCA-HRS-PRACTCAS +

ENCA-HRS-APOYO +
ENCA-TOTAL-HORAS.

ENCA-SUB-TITULO1 = * Datos personales de RFC *

= 12C' ']' +
'NOMBRE' +
SAL-NOMBRE +
15C' ']' +
'DIRECCION' +
SAL-DOMICILIO +
11C' '],

ENCA-SUB-TITULO2 = * Datos personales restantes *

= 6C' ']' +
'R , F, C.:' +
SAL-RFC +
10C' ']' +
'TELEFONO:' +
SAL-TELEFONO +
16C' ']' +
'COLONIA:' +
SAL-COLONIA +
10C' ']' +
'CODIGO-POSTAL:' +
SAL-CODIGO-POSTAL +
6C' '],

ENCA-SUB-TITULO = * Impresión de movimientos *

= SAL-CARRERA-TEXTO +
3C' ']' +
'MATERIA:' +
SAL-ASIGNATURA +
3C' ']' +
'CATEGORIA:' +
SAL-CATEGORIA +
3C' ']' +
'PLAZA:' +
SAL-PLAZA +
3C' ']' +
'CARRERA:' +
SAL-CARRERA.

ENCA-SUB-TITULO4 = * Letrero para cada grupo de movimientos (primero) *

= 12C' ']' +
'DIAS DE CLASE ' +
'HORA HORA ' +

'HORAS HORAS HORAS' +
 11C' '] +
 'TIPO' +
 15C' '] +
 'CODIGO ' +
 'FECHA FECHA' +
 ' ESTADO'.

ENCA-SUB-TITULO5 = * Letrero para cada grupo de movimientos (segundo) *

= 'GPO SALON ' +
 'L M M J V S ' +
 'ENTRA SALDA ' +
 'TEORS PRATS APOYO' +
 ' MOVIMIENTO -' +
 'CAUSA ' +
 'PROGRAMATICO ' +
 'INICIO TERMINO' +
 ' ACTUAL'.

ENCA-REGISTRO = * Campos impresos en el informe *

= SAL-GRUPO +
 2C' '] +
 SAL-SALON +
 2C' '] +
 SAL-DIAS +
 3C' '] +
 SAL-HR-ENTRADA +
 2C' '] +
 SAL-HR-SALIDA +
 2C' '] +
 SAL-HRS-TEORCAS +
 2C' '] +
 SAL-HRS-PRACTCAS +
 2C' '] +
 SAL-HRS-APOYO +
 3C' '] +
 SAL-CAUSA +
 SAL-TP-MOVIMTO +
 3C' '] +
 SAL-COD-FROGMTCO +
 3C' '] +
 SAL-FE-INICIO +
 2C' '] +
 SAL-FE-TERMINO +
 3C' '] +
 SAL-EDO-ACTUAL.

ENCA-HORAS-TEORCAS = 53C' '] +
 'HORAS TEORICAS ' +

SAL-TOT-HRS-TEORCAS +
55C' '3.

ENCA-HORAS-PRACTCAS = 53C' '3 +
'HORAS PRACTICAS ' +
SAL-TOT-HRS-PRACTCAS +
55C' '3.

ENCA-HORAS-APOYO = 53C' '3 +
'HORAS DE APOYO ' +
SAL-TOTLHRS-APOYO +
55C' '3.

ENCA-TOTAL-HORAS = 53C' '3 +
'TOTAL DE HORAS ' +
SAL-TOTAL-HORAS +
55C' '3.

DISEÑO DE LOS ARCHIVOS .- Los archivos de datos son los elementos críticos dentro de los sistemas de información, ya que se requieren muchos pasos de proceso para que puedan ser almacenados dentro de un fichero.

Los pasos de proceso que se requieren para el uso de los archivos, se establecen como parte del diseño general del sistema. Los archivos pueden ser en nuestro caso de Datos Personales y de Movimientos, que es como se ha identificado en la especificación de los requerimientos.

Tomando esta información, se procede a diseñar el archivo de Personal, se tiene que tomar en cuenta que debe estar relacionado este archivo con el de movimientos, de tal forma que al momento de acceder el RFC que se desee, inmediatamente después se pueda tener acceso a cada uno de los movimientos que tenga el profesor.

Para el acceso del archivo de Personal, se podrá obtener información inmediatamente por medio del RFC, que servirá de llave de acceso, por lo que el archivo será de tipo indexado por RFC, permitiendo controlar los RFC's, de tal forma que este tiene que ser único, como a continuación se muestra:

REGISTRO-PERS

= * Datos personales de los
profesores *

```

= PERS-RFC +
  PERS-PARTE-1 +
  PERS-PARTE-2 +
  APUNTADOR-A-MOVS.

PERS-PARTE-1 = * Primeros datos personales *

= PERS-FOLIO +
  PERS-NOMBRE +
  PERS-TELEFONO.

PERS-PARTE-2 = * Demas datos personales *

= PERS-DOMICILIO +
  PERS-COLONIA +
  PERS-CODIGO-POSTAL +
  PERS-FE-INGRESO +
  PERS-ANTIG-UNAM +
  PERS-TP-PERSONAL +
  PERS-NACIONALIDAD +
  PERS-SEXO +
  PERS-ESTADO-CIVIL +
  PERS-FE-ACTUALIZA.

```

En cuanto al archivo de movimientos, se procede a su diseño, este deberá ser de acceso aleatorio, ya que con el archivo de datos se podrá tener disponible el registro deseado por medio de un apuntador.

Para llevar el correcto control del total de los registros se tiene que tomar en cuenta el índice que controla cada movimiento, para ello se deberá guardar el apuntador y esto se hace en el primer registro.

Otra característica que se tiene que tomar en cuenta, es llevar el control de cada uno de los movimientos para determinado RFC, para ello se tendrá otro apuntador que nos indicará el lugar del archivo en que se encuentra el siguiente movimiento del profesor.

Ya para finalizar se debe tener un control de cada uno de los niveles de adscripción, a fin de que cuando se deseen obtener informes divididos por nivel de adscripción, estos se puedan obtener de manera rápida.

De lo anterior podemos definir la estructura del archivo de movimientos como sigue:

```
REGISTRO-MOVS      = * Movimientos de cada profesor *
                  =
                  MOVS-RFC +
                  MOVS-PARTE-3 +
                  MOVS-LIGA-A-MOV +
                  MOVS-LIGA-A-NIVEL.

MOVS-PARTE-3      = * Tercera parte de los movimientos *
                  =
                  MOVS-ADSCRIPCION +
                  MOVS-CARRERA +
                  MOVS-TP-MOVIMTO +
                  MOVS-CAUSA +
                  MOVS-CATEGORIA +
                  MOVS-PLAZA +
                  MOVS-FE-INICIO +
                  MOVS-FE-TERMINO +
                  MOVS-ASIGNATURA +
                  MOVS-GRUPO +
                  MOVS-SALON +
                  MOVS-DIAS +
                  MOVS-HR-ENTRADA +
                  MOVS-HR-SALIDA +
                  MOVS-HRS-TEORCAS +
                  MOVS-HRS-PRACTICAS +
                  MOVS-HRS-APOYO +
                  MOVS-COD-FROGMTICO.
```

Resumiendo la estructura de los archivos que utiliza el sistema, el almacenamiento de los datos introducidos al mismo, se representa a continuación en la figura 9.

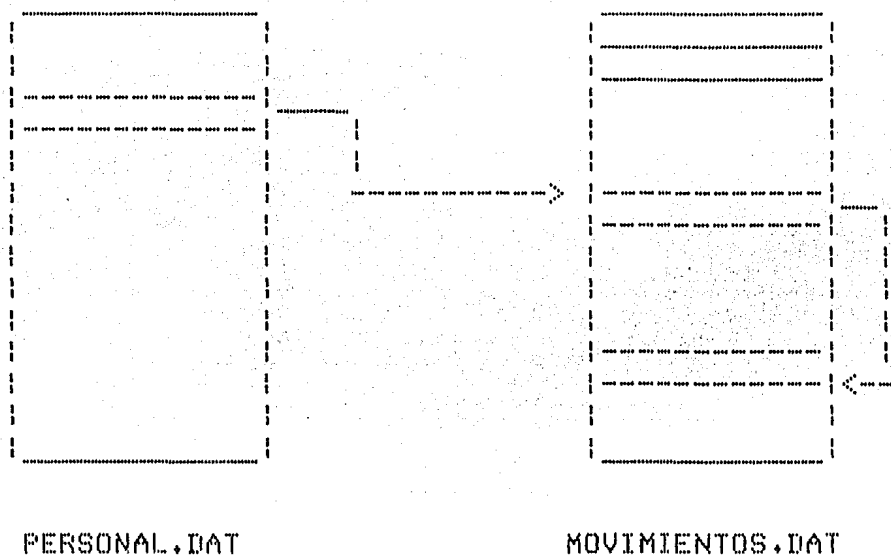


FIGURA 9

Otro archivo que utiliza el sistema de Banco de Horas será el del generador del reporte, este es utilizado por los informes que se quieran obtener, y tendrá diferente estructura para cada uno de los reportes a obtenerse. Tendrá la característica de que en un informe se pueda obtener dos formas uno global y otro subdividido por niveles de adscripción.

Ya que el archivo deberá ser dinámico, para la obtención de la variedad de informes a generarse, la estructura que este archivo tiene es indexada, para así tener la posibilidad de manejar distintas llaves

necesarias para la obtención de cada uno de los informes a la vez que se va sorteando el archivo.

La estructura de este archivo se muestra a continuación:

```
REGISTRO-REFORTE      = *  Generador de informes  *
                      = [  REGISTRO-REPT-DIR  |
                          REGISTRO-REPT-MOVS ].
```

```
REGISTRO-REPT-DIR    = *  Datos del directorio
                          de personal  *
                      =  REPT-LLAVE-D +
                          REPT-HRS-TEORCAS +
                          REPT-HRS-PRACTCAS +
                          REPT-HRS-APOYO,
```

```
REPT-LLAVE-D         = *  Llave del archivo  *
                      =  REPT-ADSCRIPCION +
                          REPT-RFC +
                          REPT-NUMEROS,
```

```
REGISTRO-REPT-MOVS   = *  Datos del reporte detallado de
                          movimientos  *
                      =  REPT-LLAVE-M +
                          REPT-CARRERA +
                          REPT-TP-MOVIMTO +
                          REPT-CAUSA +
                          REPT-CATEGORIA +
                          REPT-PLAZA +
                          REPT-FE-INICIO +
                          REPT-FE-TERMINO +
                          REPT-SALON +
                          REPT-DIAS +
                          REPT-HR-ENTRADA +
                          REPT-HR-SALIDA +
                          REPT-HRS-TEORCAS +
                          REPT-HRS-PRACTCAS +
                          REPT-HRS-APOYO +
                          REPT-COD-PROGTCO,
```

```

REPT-LLAVE-M           = * Llave del archivo *
                        = REPT-ADSCRIPCION +
                          REPT-RFC +
                          REPT-ASIGNATURA +
                          REPT-GRUPO +
                          REPT-NUMEROS.

```

Un registro de gran importancia, es el que se refiere a la comunicación de los módulos del sistema con el proceso que define la validación, este registro transfiere los parámetros a validar, así como el resultado de dicha evaluación.

```

DATOS-DE-ENTRADA      = * Comunicación del sistema con
                        el proceso de validación *
                        = PROCESO +
                          NUMERO-REGISTRO +
                          NUMERO-CAMPO +
                          INFORMACION +
                          ENT-RFC +
                          [ DATOS-PARTE-3  |
                            DATOS-PARTE-2  |
                            DATOS-PARTE-1  ].

```

Existen una serie de pequeños archivos que se utilizan en forma auxiliar por parte del sistema, estos tienen la característica de tener los letreros que se necesitan durante la obtención de un reporte, siendo únicamente este su uso.

Los archivos descritos están formados como se muestra a continuación:

```

ADNIVEL1              = * Primer nivel de adscripción *
                        = NEL1-CLAVE +
                          NEL1-DESCRIPCION.

```

ADNIVEL2 = * Segundo nivel de adscripción *

 = NEL2-CLAVE +
 NEL2-DESCRIPCION,

ADNIVEL3 = * Tercer nivel de adscripción *

 = NEL3-CLAVE +
 NEL3-DESCRIPCION,

CARRERAS = * Carreras que maneja el
 sistema *

 = CARR-CLAVE +
 CARR-DESCRIPCION,

MATERIAS = * Materias que acepta el
 sistema *

 = MATS-CLAVE +
 MATS-DESCRIPCION,

DIAGRAMA DE ESTRUCTURA .- Esta fase tiene la función de mostrar los programas definidos, describiendo sus funciones y sus interfaces, pero sin mostrar su estructura interna. Para cada programa identificado en la arquitectura del sistema, se procede a elaborar un diagrama de estructura, mismo que representa los módulos jerarquizados de cada uno de los procesos.

Así, podemos deducir que el diagrama de estructura de un programa es una representación gráfica de la relación entre sus subrutinas, ya que muestra para cada subrutina o módulo, cuales son las subrutinas que lo activan y cuales subrutinas son activadas por el módulo.

Para la elaboración de los diagramas de estructuras del sistema, primero se tiene que tener bien definido cada uno de los módulos que componen el sistema, para ello en primera instancia, se debe tomar cada diagrama de flujo de datos para cada proceso de que esta compuesto el sistema, a fin de adecuar el diagrama jerárquico al mismo sistema.

Así tenemos que los procesos o programas a desarrollar estan conformados como se muestra a través de las siguientes páginas.

SISTEMA BANCO DE HORAS

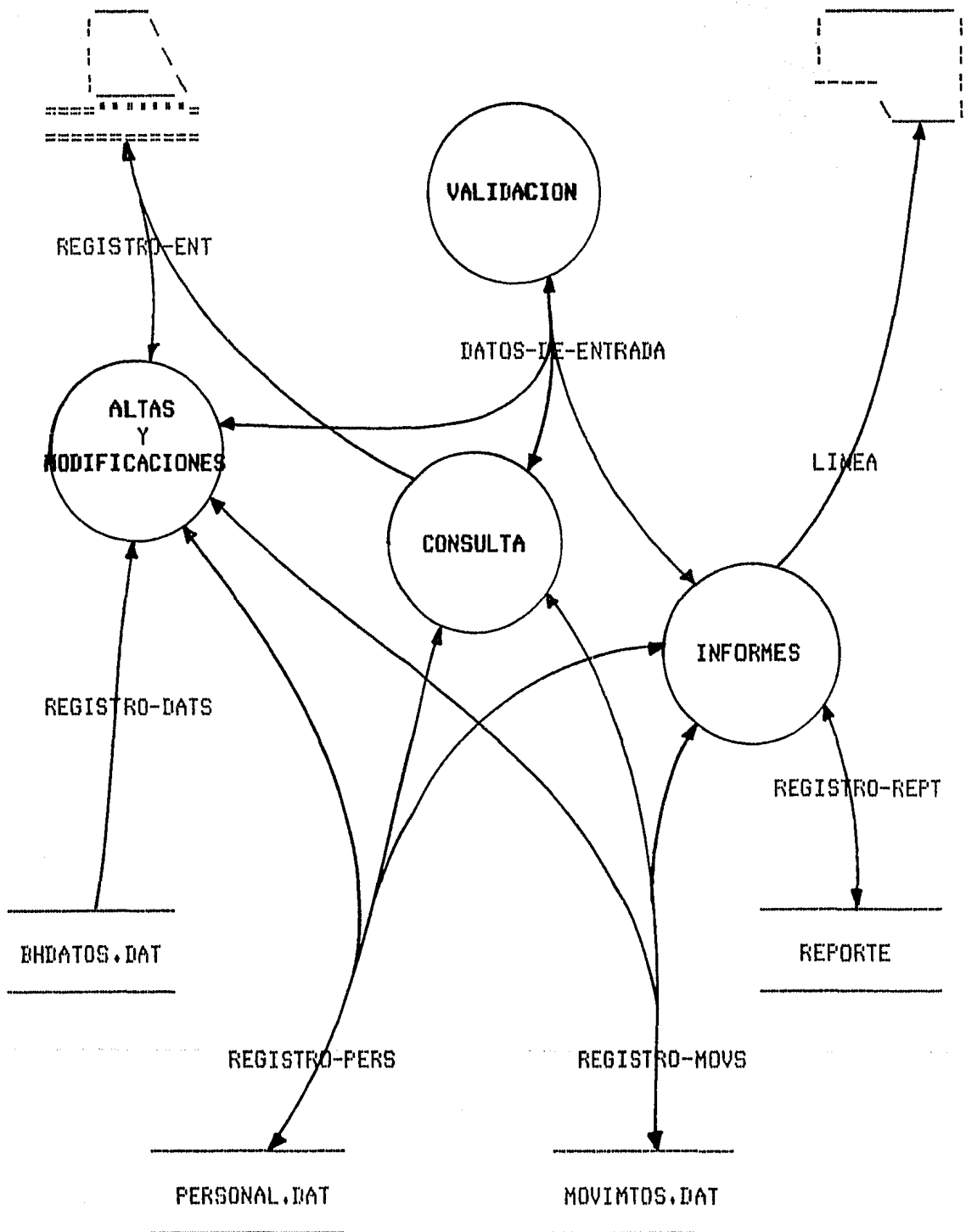


FIG. 10

PROCESO DE ALTAS Y MODIFICACIONES

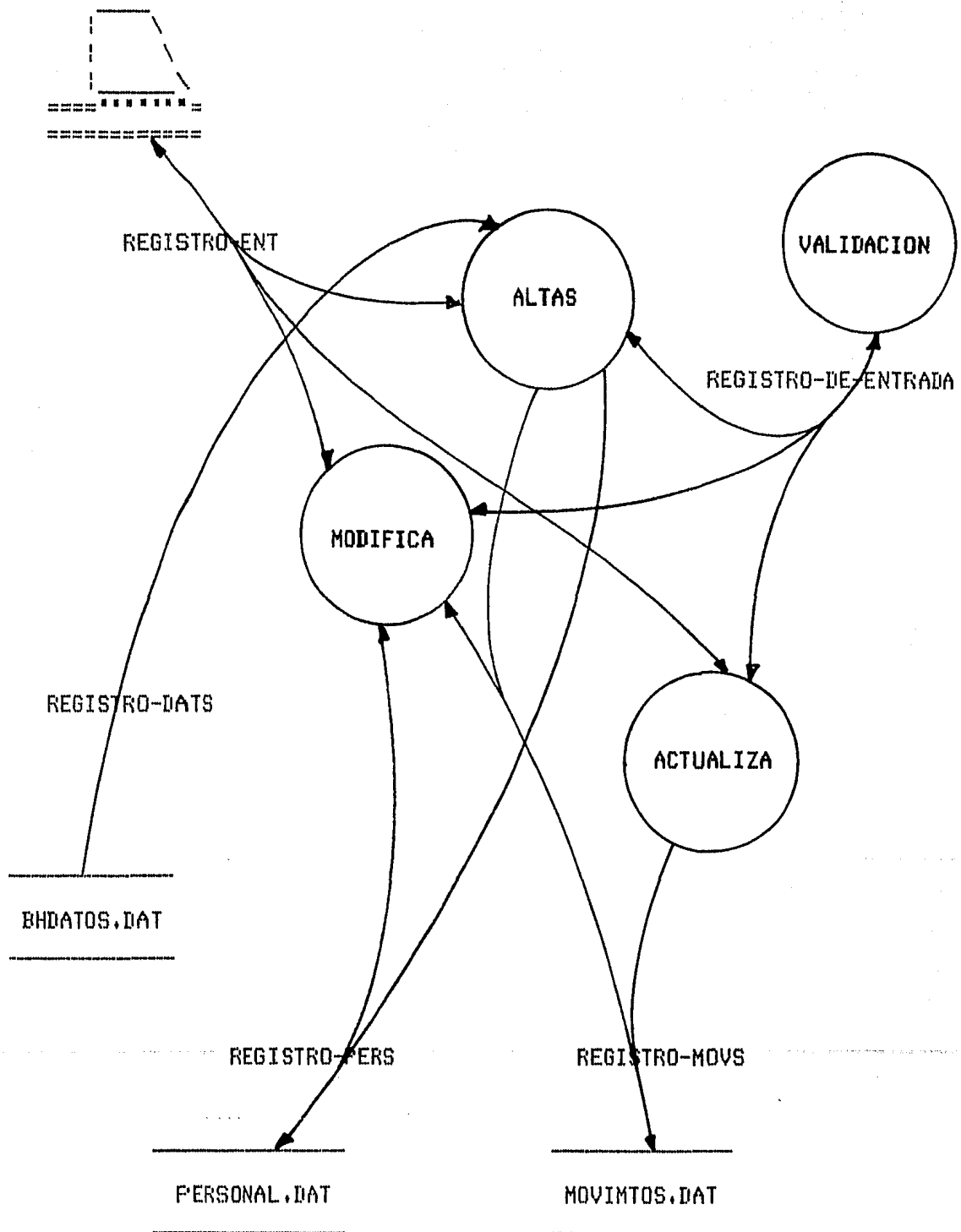


FIG. 11

PROCESO DE ALTAS

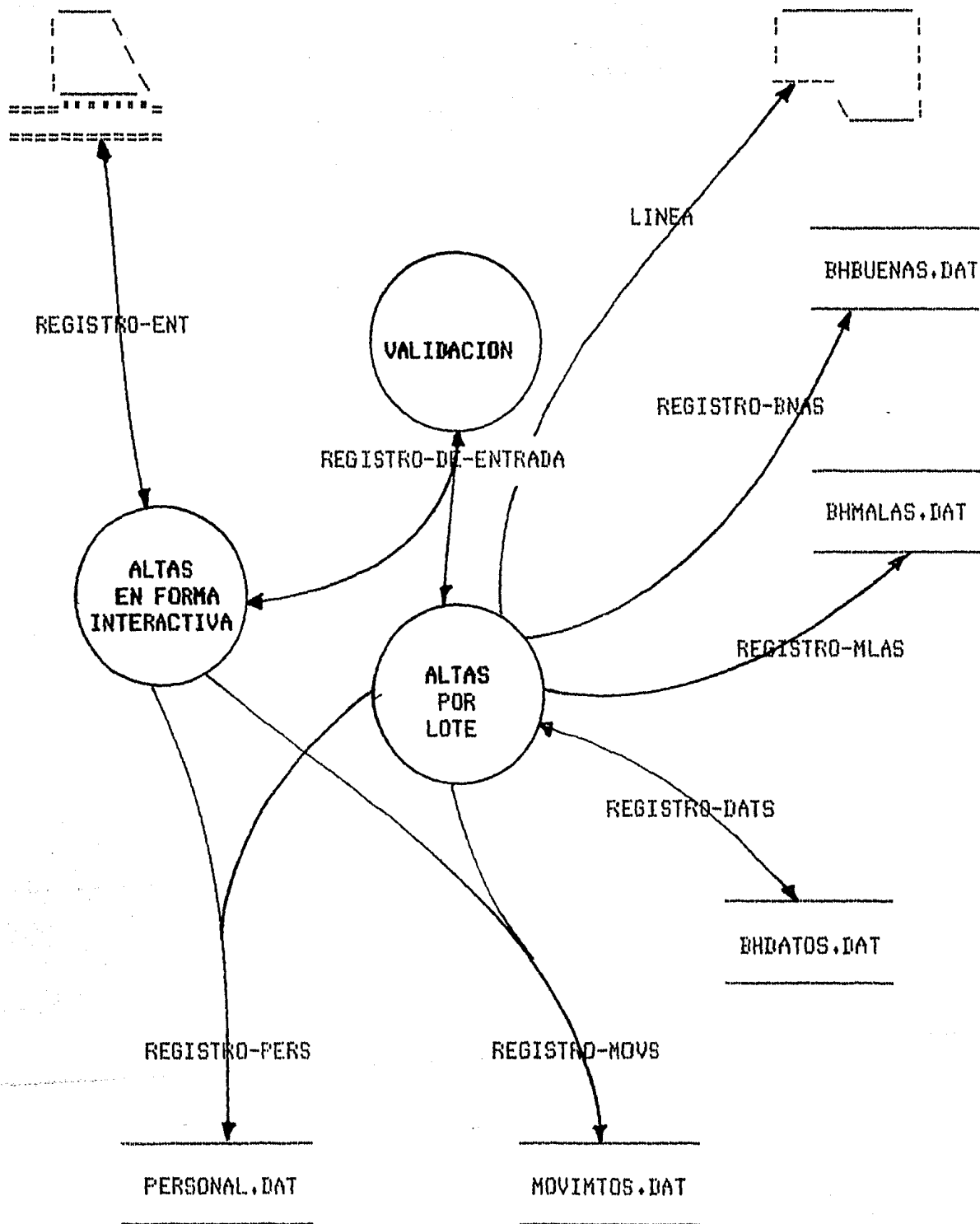


FIG. 12

PROCESO DE ACTUALIZACION

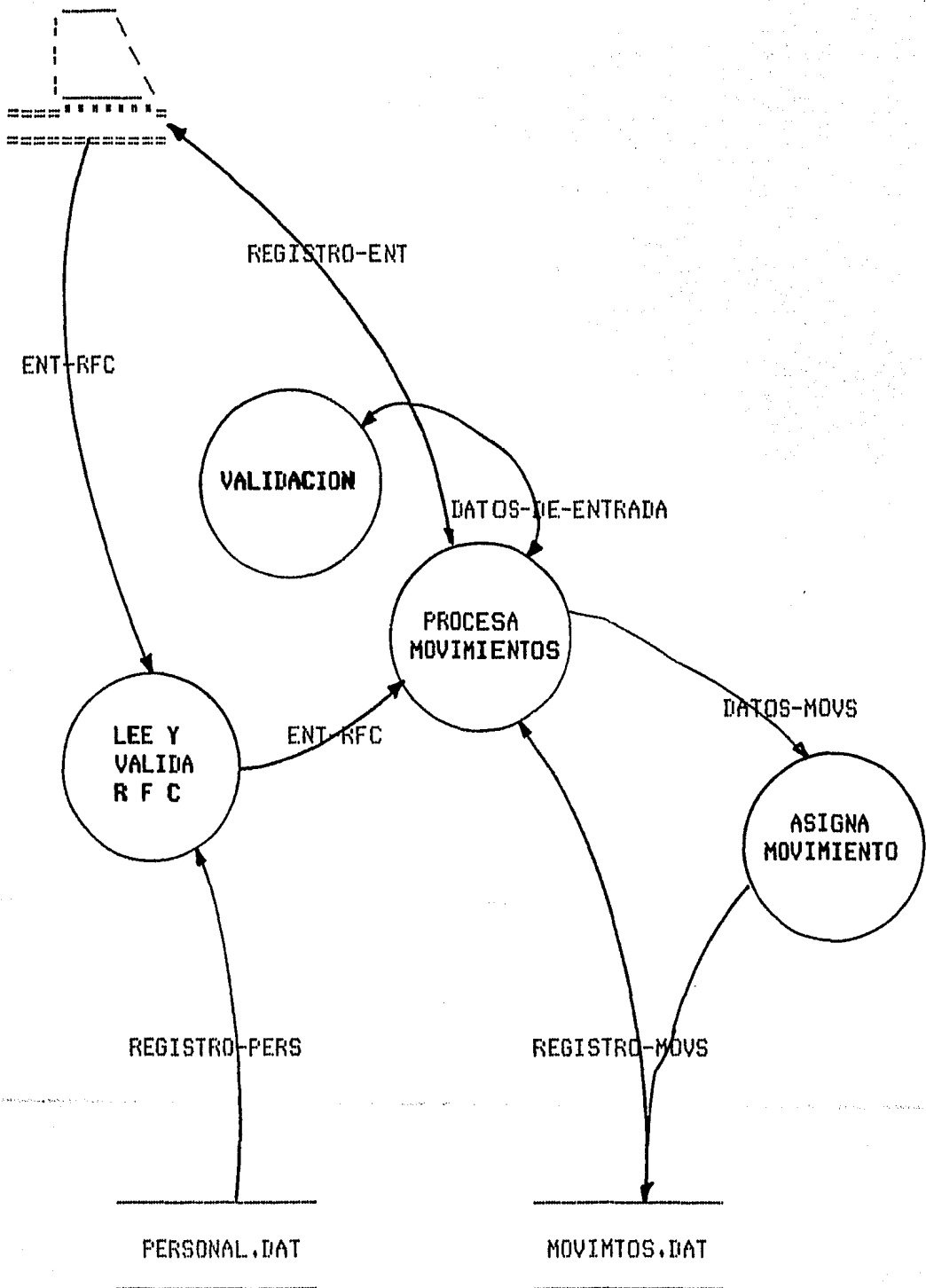


FIG. 13

PROCESO DE MODIFICACIONES

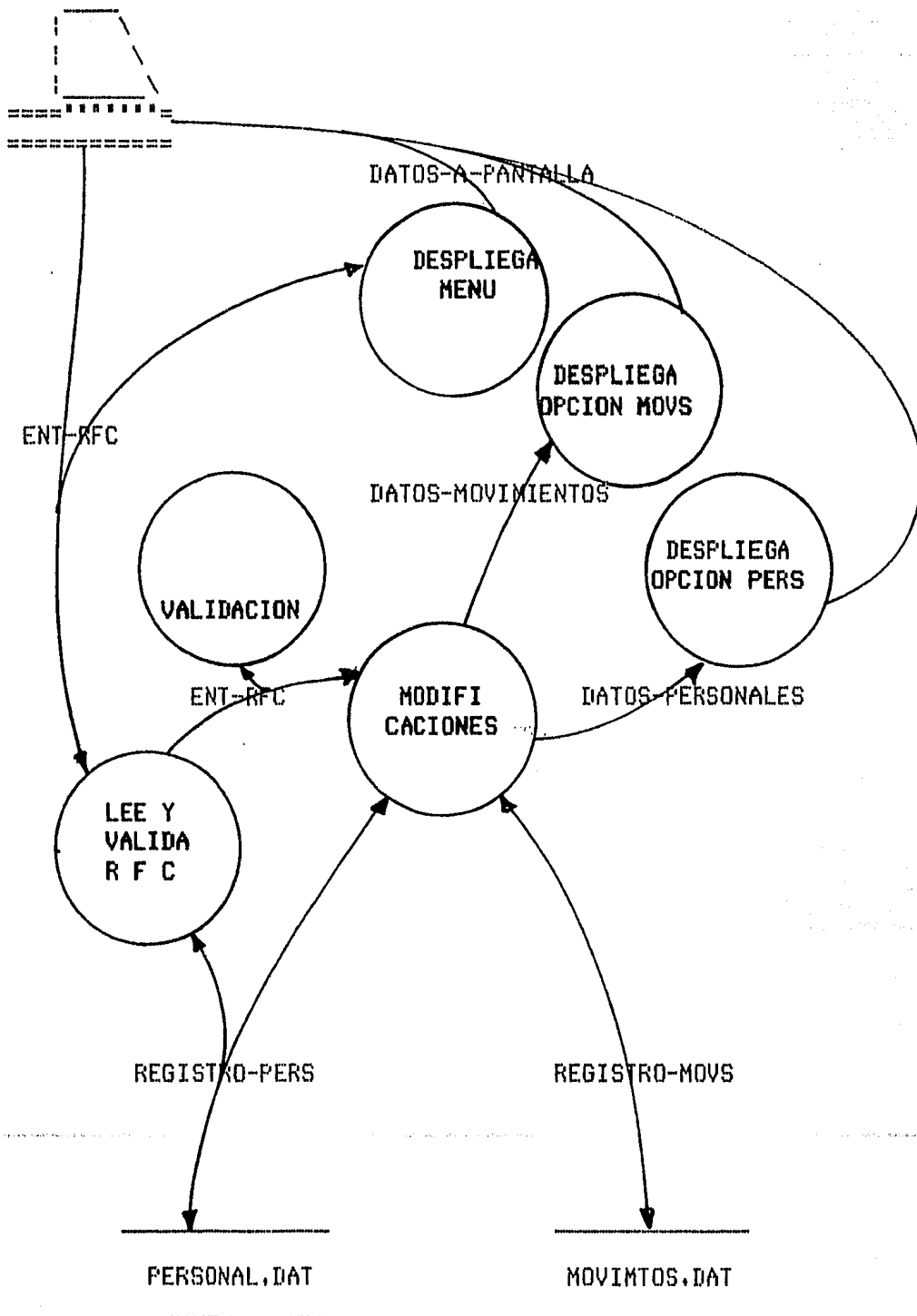


FIG. 14

PROCESO DE CONSULTA

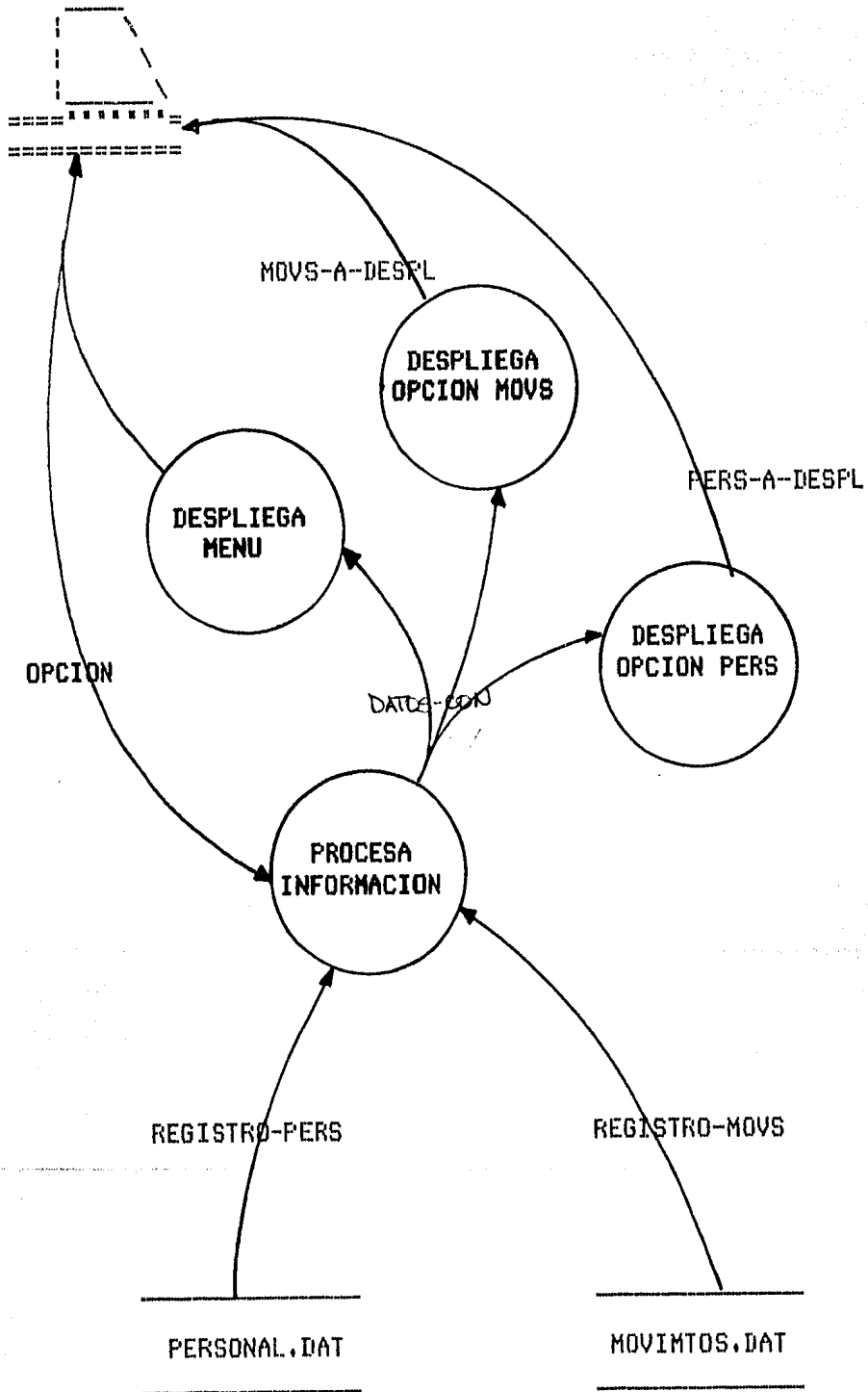


FIG. 15

PROCESO DE INFORMES

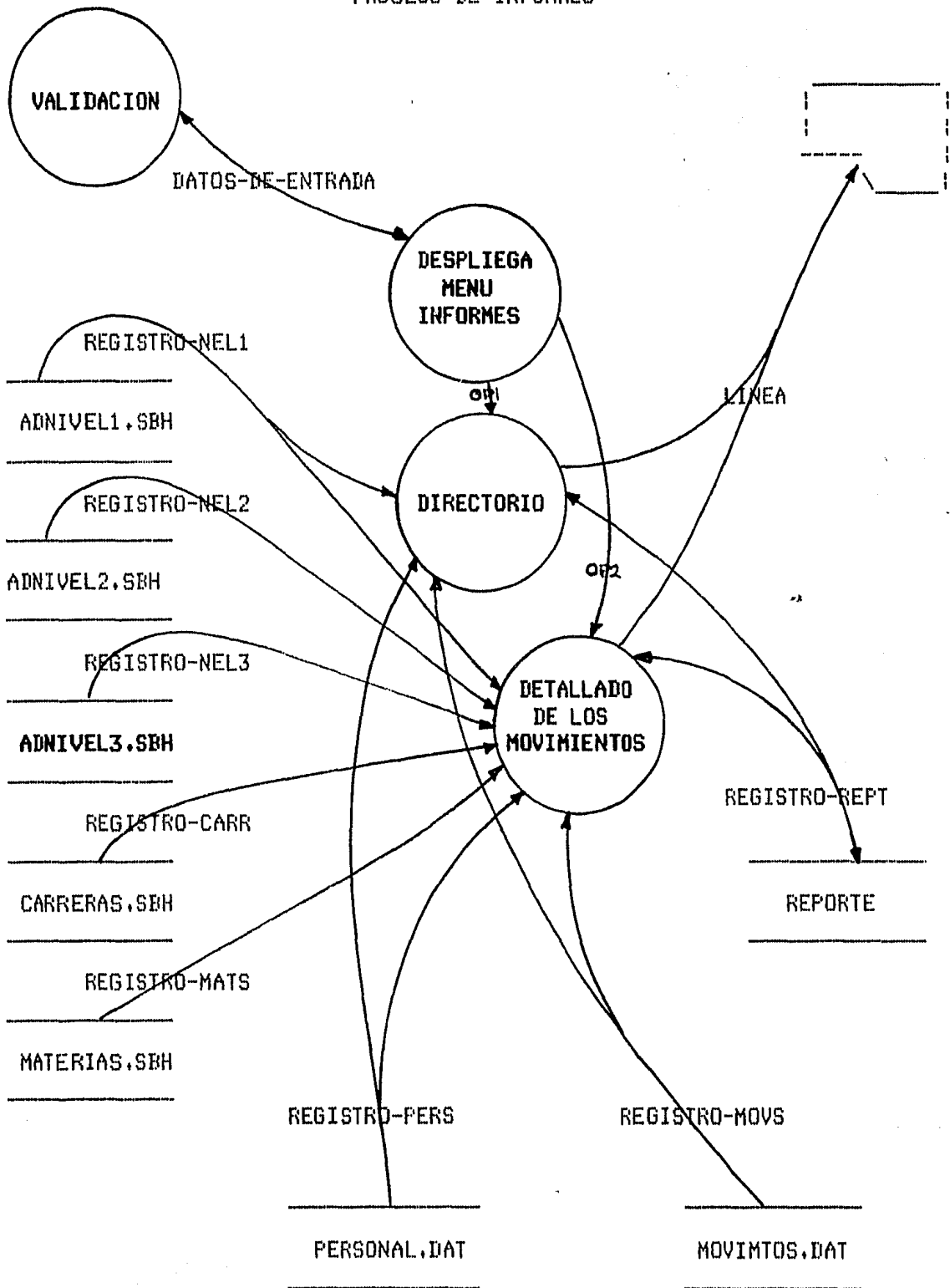


FIG. 16

PROCESO DE VALIDACION

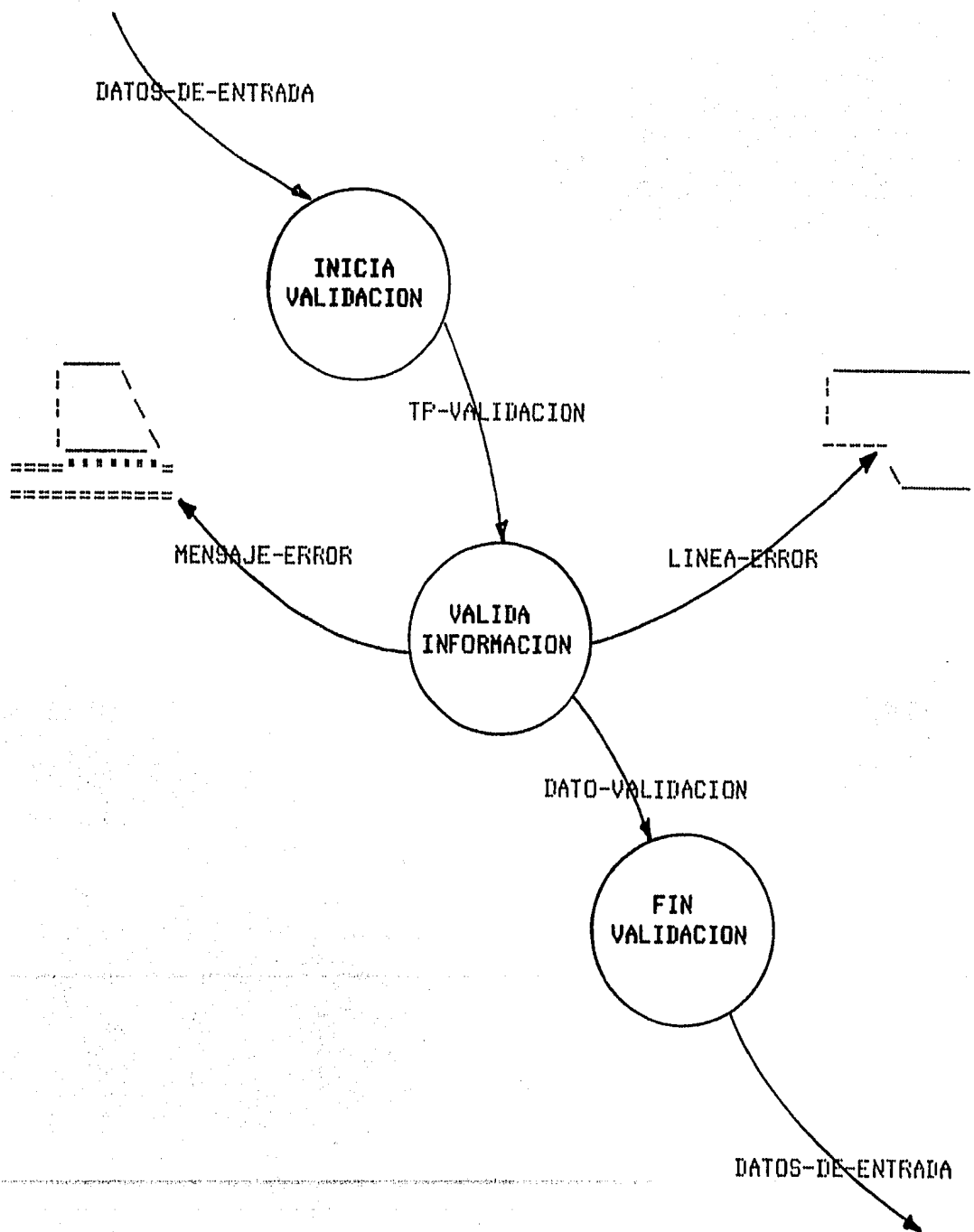


FIG. 17

DIAGRAMA DE ESTRUCTURA DEL SISTEMA BANCO DE HORAS

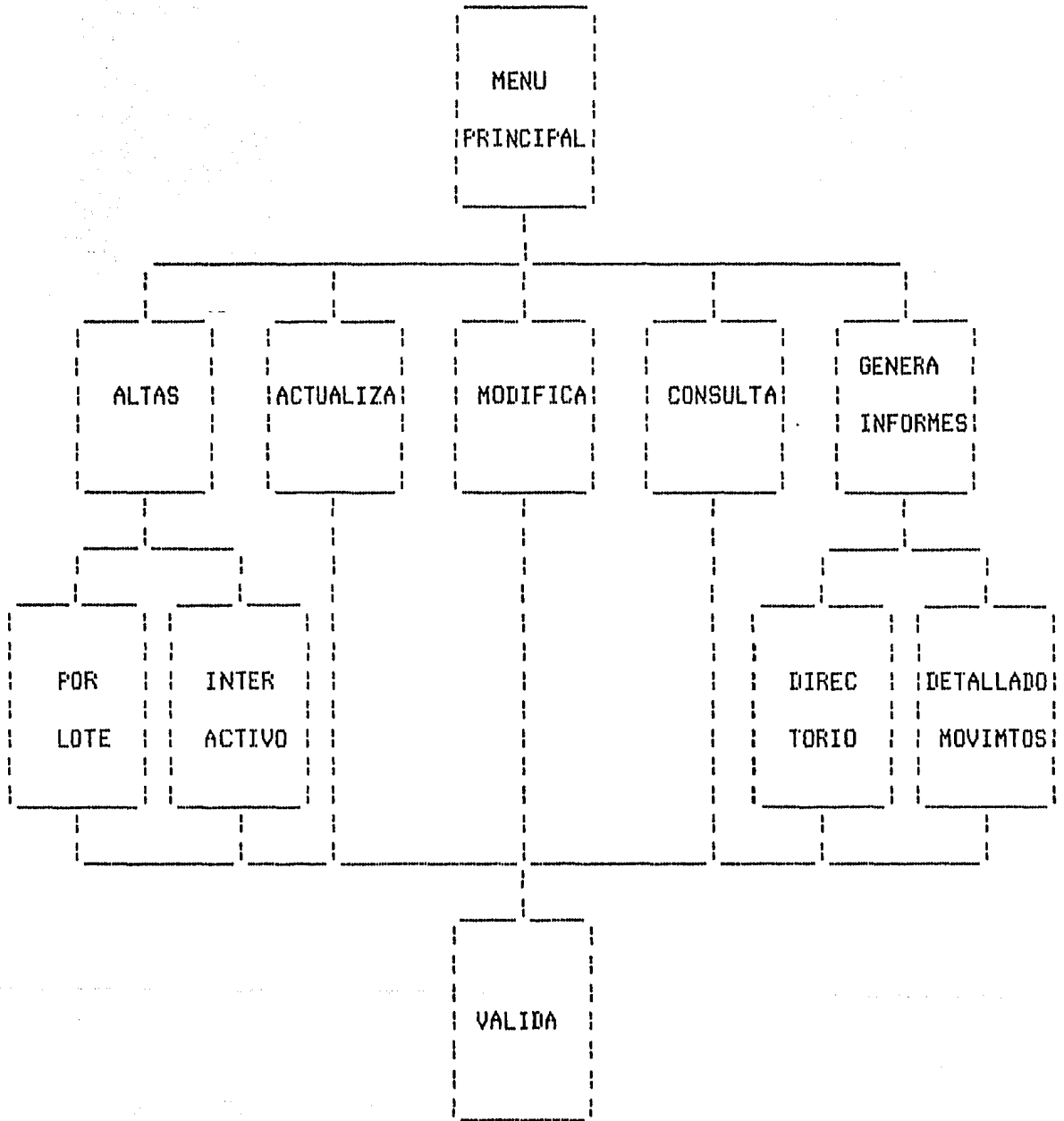


FIG. 18

DIAGRAMA DE ESTRUCTURA DEL PROCESO ALTASINT

(ALTAS INTERACTIVO)

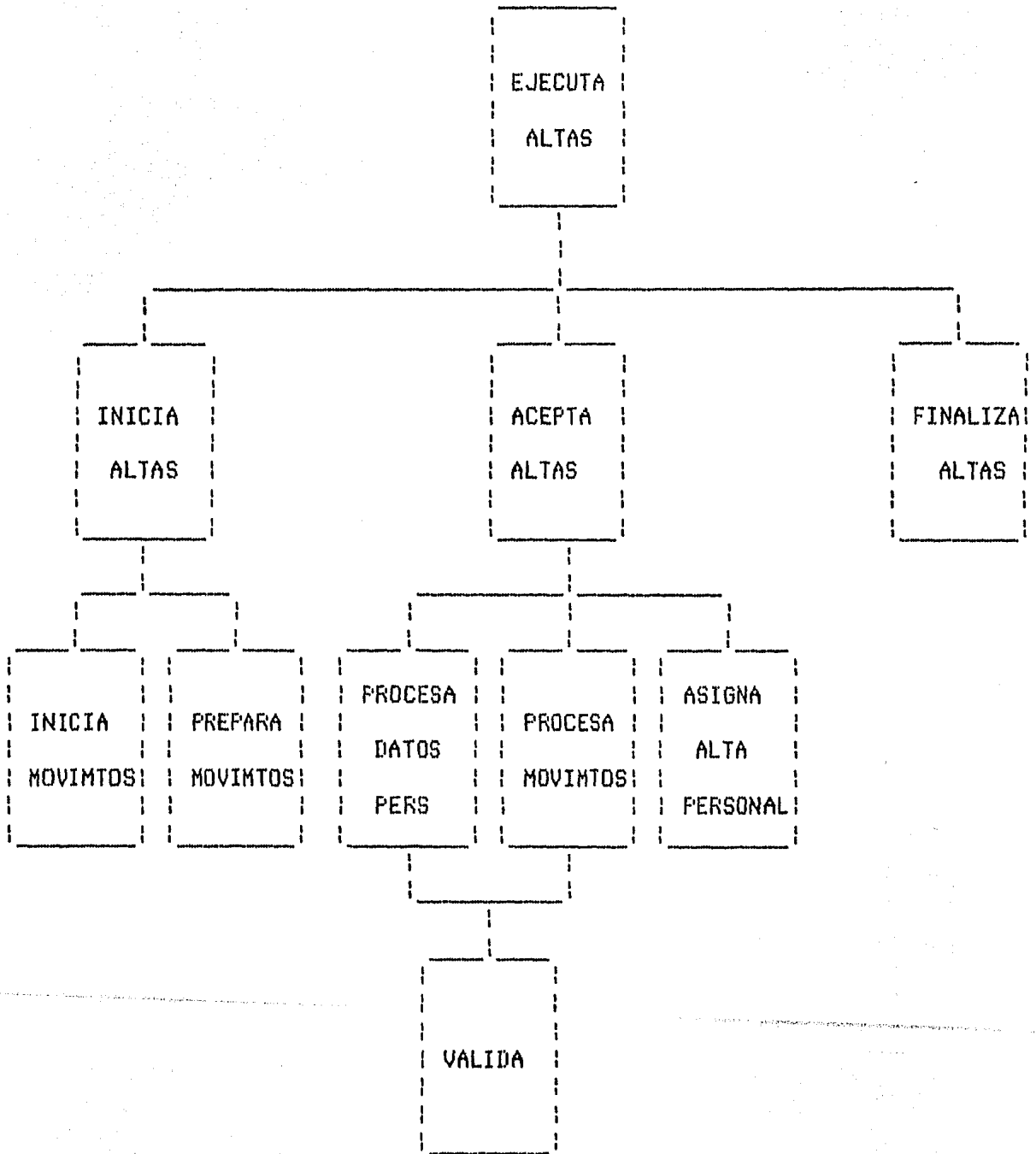


FIG. 19

DIAGRAMA DE ESTRUCTURA DEL PROCESO ALTASBAT

(ALTAS POR LOTE)

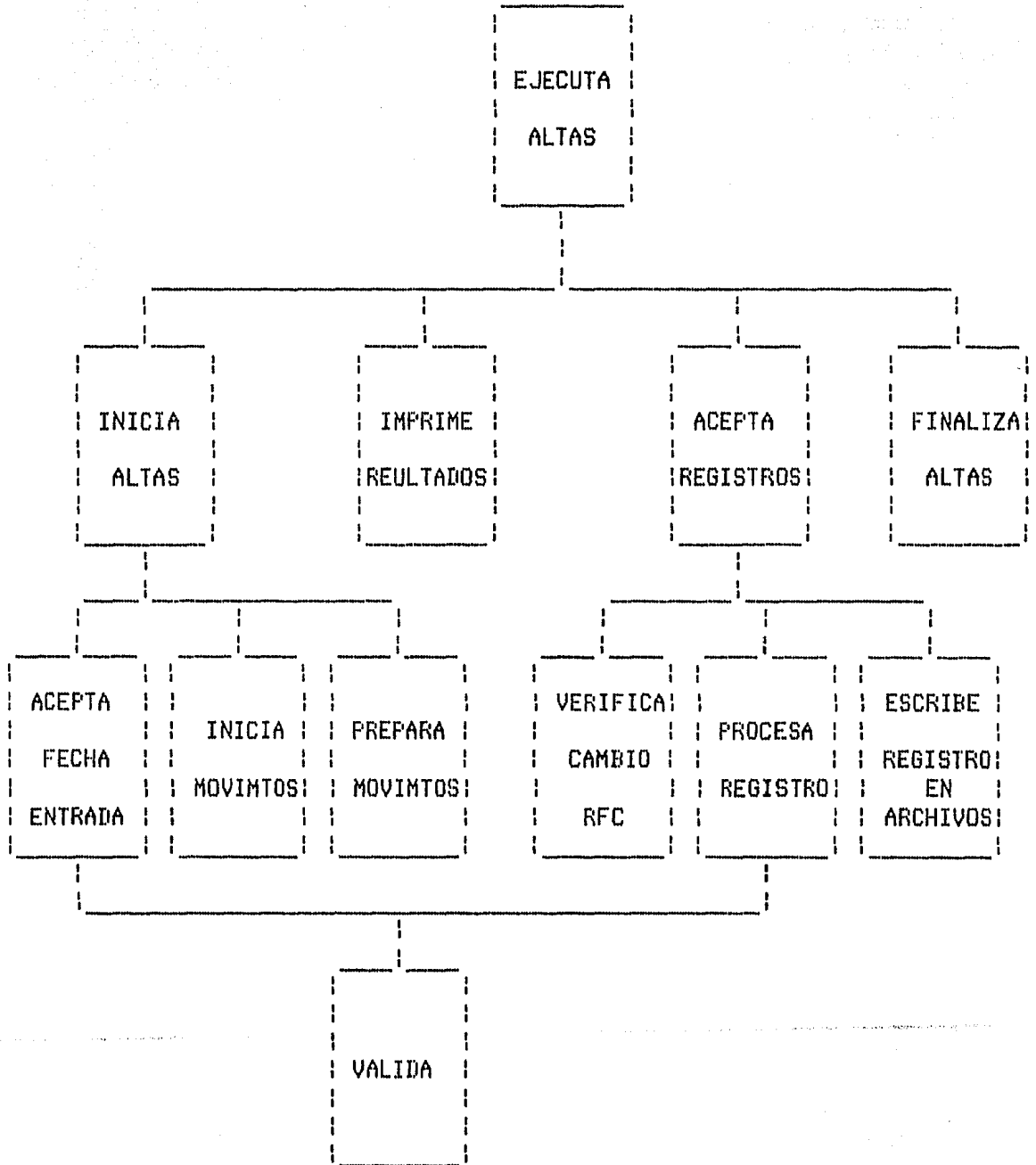


FIG. 20

DIAGRAMA DE ESTRUCTURA DEL PROCESO ACTUALZA

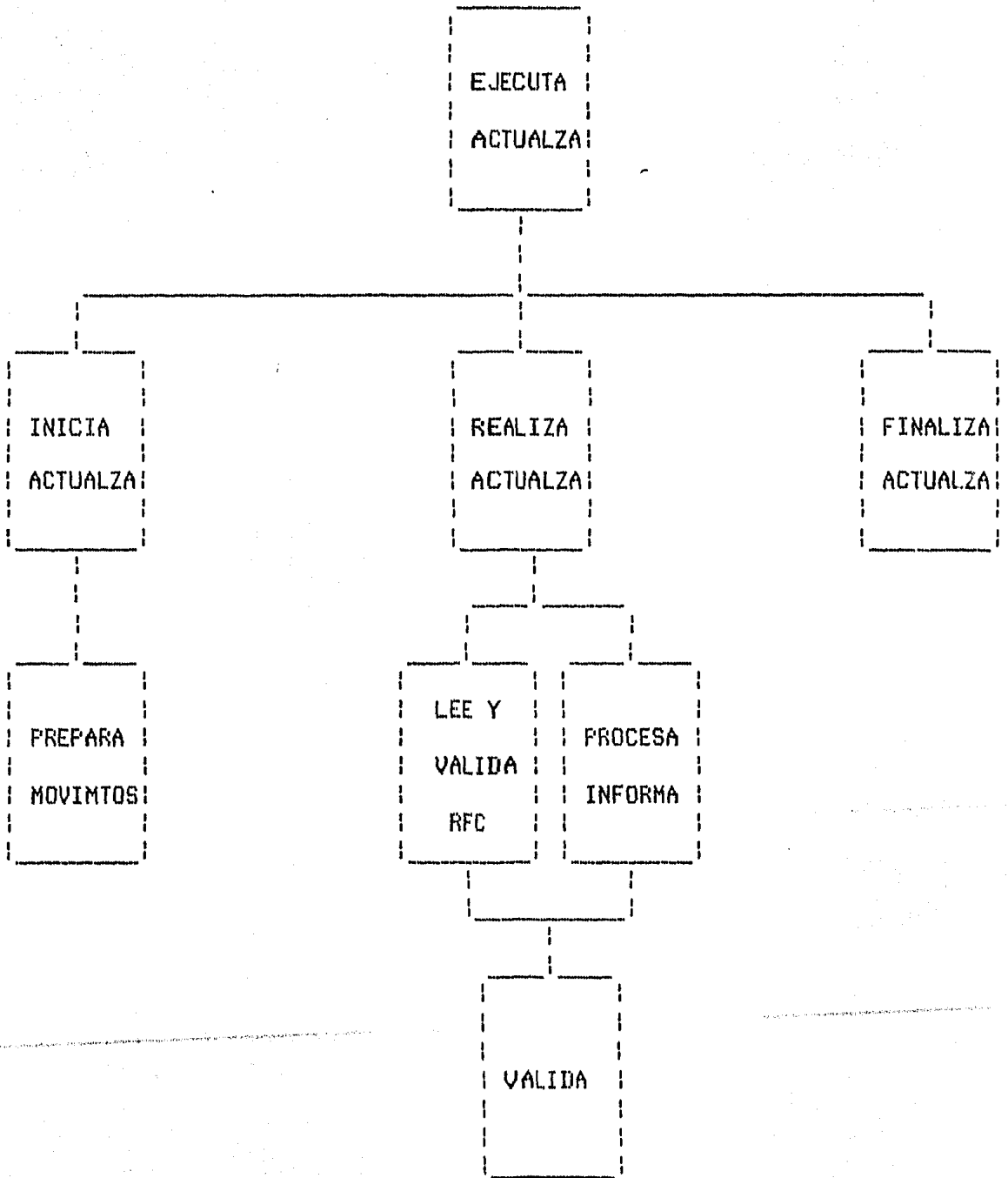


FIG. 21

DIAGRAMA DE ESTRUCTURA DEL PROCESO MODIFICA

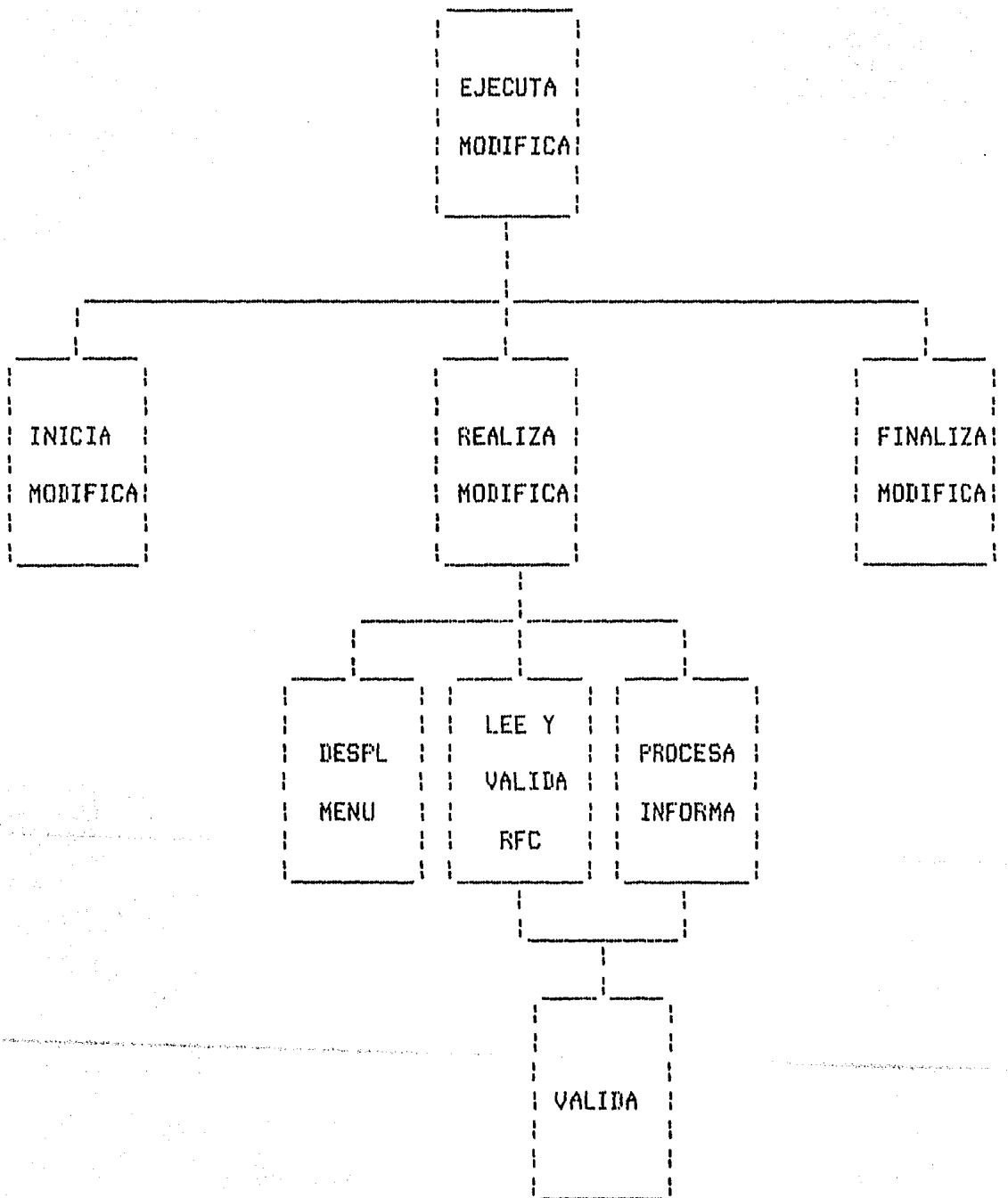


FIG. 22

DIAGRAMA DE ESTRUCTURA DEL PROCESO CONSULTA

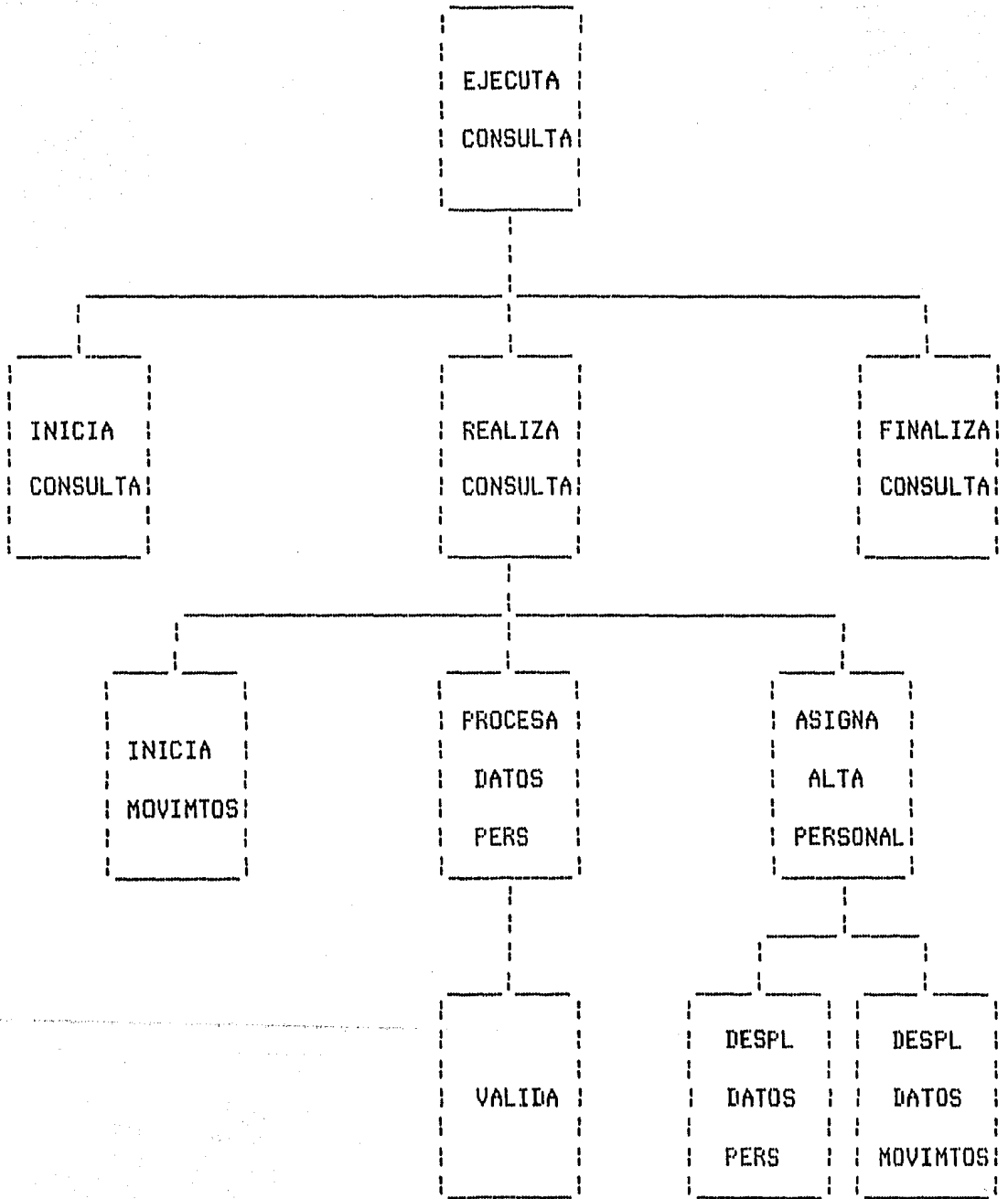


FIG. 23

DIAGRAMA DE ESTRUCTURA DEL DETALLADO DE MOVIMIENTOS

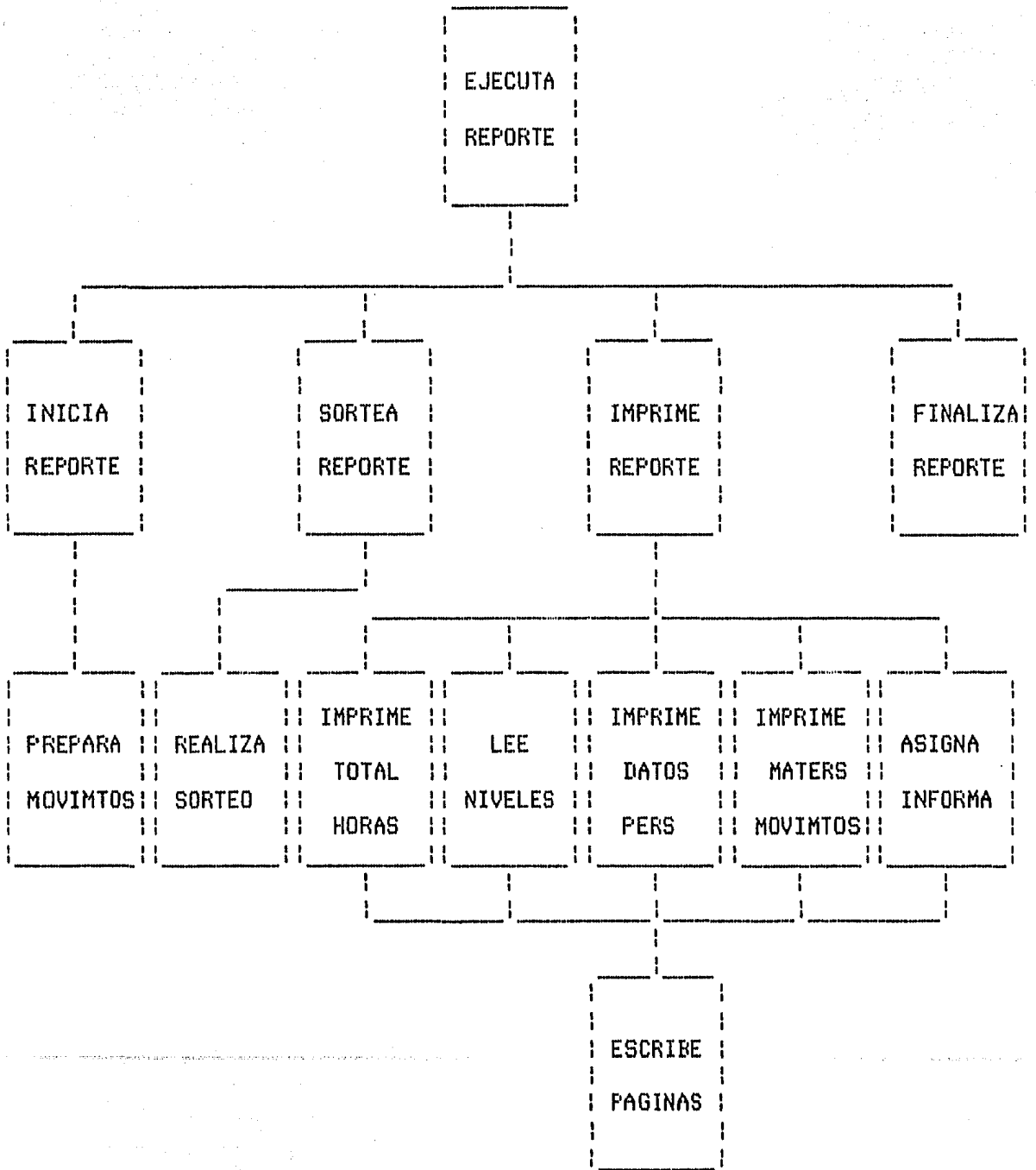


FIG. 24

DIAGRAMA DE ESTRUCTURA DEL DIRECTORIO

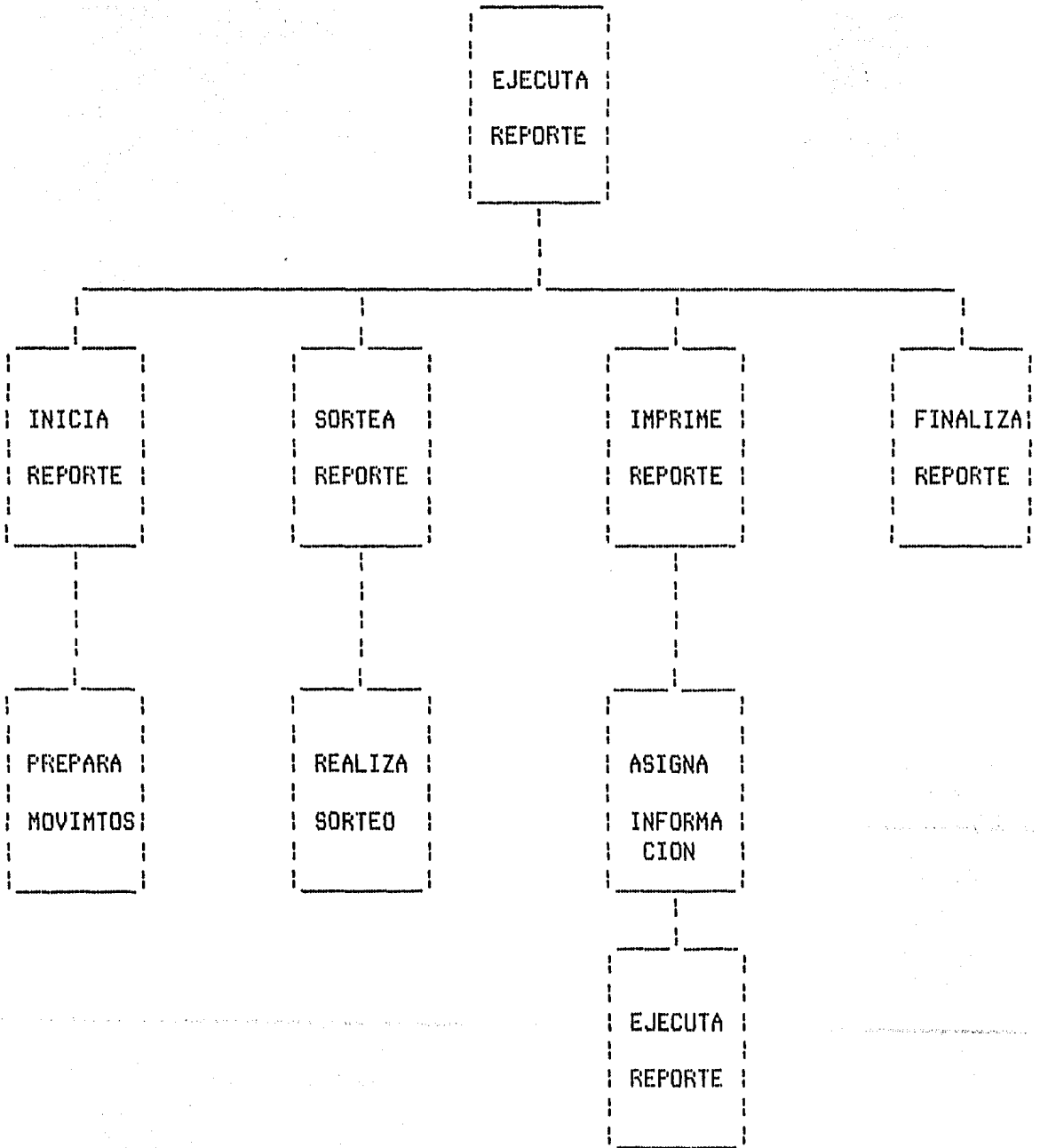


FIG. 25

DIAGRAMA DE ESTRUCTURA DEL PROCESO DE VALIDACION

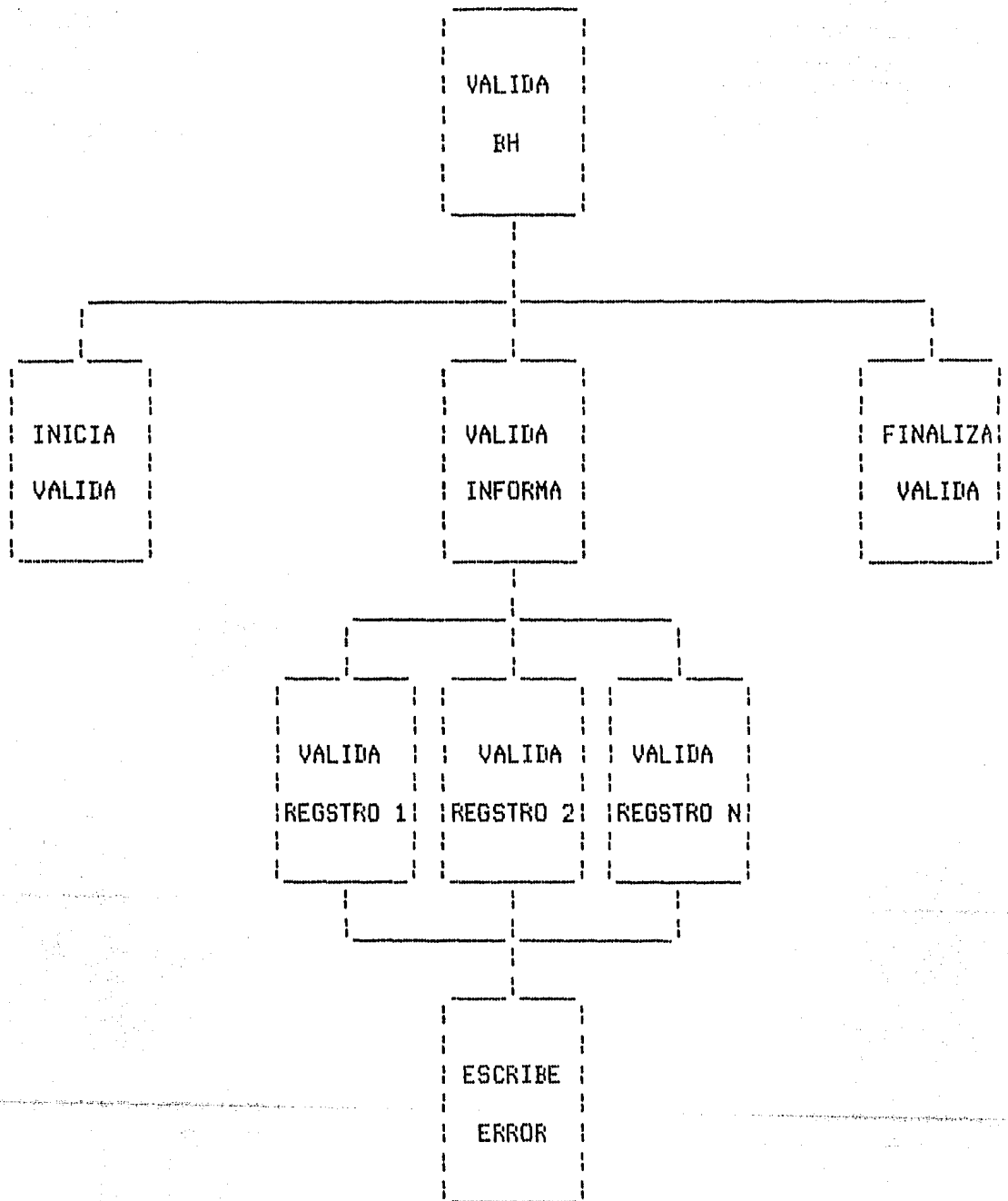


FIG. 26

DISEÑO DETALLADO .- En esta etapa del diseño, se procede a la especificación del proceso o función que un módulo realiza. El uso de lenguaje natural para este diseño podría ser suficiente, pero no cuenta con formas estructurales efectivas para la representación de procesos.

Por otro lado, aunque algunos lenguajes de programación tienen las formas estructuradas para la representación de procesos, requieren de un nivel muy bajo de expresión y los conceptos de diseño se perderían en un mar de detalles. Además, los lenguajes de programación tienen una sintaxis muy estricta y convenciones que no son necesarias para este nivel de diseño con que se está trabajando el Sistema Banco de Horas.

El lenguaje para diseño de programas y que se utilizó para la elaboración de este trabajo, consiste en usar el pseudo-código de estructuras de proceso. La especificación de las tareas de cada módulo se describe utilizando técnicas de programación estructurada.

Existen tres estructuras de proceso por medio de las cuales se puede realizar el sistema y que se muestran a continuación:

La Secuencia .- Esta permite enunciar una serie de instrucciones una después de la otra.

La Decisión .- Deja especificar alternativas en la ejecución de instrucciones dependiendo de una condición.

La decisión se divide en tres tipos:

Decisión Simple :

SI condición ENTONCES instruccion(es)

Decisión Doble :

SI condición ENTONCES instruccion(es)

SI NO instruccion(es)

Decisión Múltiple :

CASO condición-1 instruccion(es)

 condición-2 instruccion(es)

 |

SI NO condición-N instruccion(es)

La Repetición .- Da lugar a la especificación de la ejecución iterativa de una serie de instrucciones. Tiene dos tipos de estructuras:

Mientras :

MIENTRAS condición(es)
 instrucción(es)

FIN (MIENTRAS)

Ejecuta-hasta :

EJECUTA
 instrucción(es)
HASTA condición(es)

Repita :

REPITE var = inicio, final, incremento
 instrucción(es)

FIN (REPITE)

El anidamiento de estructuras de proceso, se representa usando el sangrado de las estructuras y frases en lenguaje natural.

A continuación se presenta el detallado de cada uno de los módulos utilizados por el sistema de banco de horas.

PROCESO DE OPCIONES

(* EJECUTA OPCIONES *)

INICIA-SISTEMA.
PRESENTA-OPCIONES.
EJECUTA-PROCESO.
FINALIZA-SISTEMA.

(* INICIA-SISTEMA *)

INICIALIZA VARIABLES
Y ABRE ARCHIVOS.

(* PRESENTA-OPCIONES *)

MIENTRAS NO OCURRA OPCION-BIEN
MUESTRA MENU-PRINCIPAL
LEE OPCION
FIN (MIENTRAS).

SI OPCION = 1
MIENTRAS NO OCURRA OPCION-AL-BIEN
MUESTRA MENU-ALTAS
LEE OPCION-AL
FIN (MIENTRAS)

SINO
SI OPCION = 5
MIENTRAS NO OCURRA OPCION-IN-BIEN
MUESTRA MENU-ALTAS
LEE OPCION-IN
MIENTRAS NO OCURRA DATOS-REPORTE-BIEN
MUESTRA "FECHA (ddmmaa) >> "
LEE ENT-FECHA
VALIDA ENT-FECHA
SI INFORMACION-CORRECTA

```
MUESTRA "TIPO DE INFORME >> "  
COLOCA-PARAMENTROS  
FIN (MIENTRAS).
```

```
(* EJECUTA-PROCESO *)
```

```
SI OPCION = 1  
  SI OPCION-AL = 1  
    EJECUTA "ALTASINT"  
  SINO  
    EJECUTA "ALTASBAT"  
SINO  
  SI OPCION = 2  
    EJECUTA "ACTUALZA"  
  SINO  
    SI OPCION = 3  
      EJECUTA "MODIFICA"  
    SINO  
      SI OPCION = 4  
        EJECUTA "CONSULTA"  
      SINO  
        SI OPCION =5  
          SI OPCION-IN = 1  
            EJECUTA "DIRPERS"  
          SINO  
            SI OPCION-IN = 2  
              EJECUTA "MOVSPERS".
```

```
(* FINALIZA-OPCIONES *)
```

```
BORRA PANTALLA  
MUESTRA "FIN SISTEMA BANCO DE HORAS"  
MUESTRA "=== ===== ===== == ====="  
FIN.
```

PROCESO DE ALTAS INTERACTIVA

(* EJECUTA-ALTAS *)

INICIA-ALTAS.

MIENTRAS NO OCURRA FIN-ALTAS
ACEPTA-ALTAS
FIN (MIENTRAS).

FINALIZA-ALTAS.

(* INICIA-ALTAS *)

INICIALIZA VARIABLES
REPITE VAR = 1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)

SI EXISTEN-ARCHIVOS
(* INICIA-MOVIMTOS *)
REPITE VAR =1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)

SINO

(* PREPARA-MOVIMTOS *)
LEE PRIMER-REGISTRO-MOV
REPITE VAR =1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
LEE SEGUNDO-REGISTRO-MOV
REPITE VAR =1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
APUNTA-DOR-MOVS --> MOVS-IND.

(* ACEPTA-ALTAS *)

(* PROCESA-DATOS-PERSONALES *)
BORRA PANTALLA
MUESTRA CAMPOS-FORMA-1
INICIALIZA-VARIABLES
MIENTRAS INFORMACION-INCORRECTA O FIN-ALTAS
LEE ENT-RFC
SI NO OCURRE FIN-ALTAS
SI NO ES PRIMER REGISTRO EN PERSONAL

PROCESO DE ALTAS INTERACTIVA

(* EJECUTA-ALTAS *)

INICIA-ALTAS.

MIENTRAS NO OCURRA FIN-ALTAS
ACEPTA-ALTAS
FIN (MIENTRAS).

FINALIZA-ALTAS.

(* INICIA-ALTAS *)

INICIALIZA VARIABLES
REPITE VAR = 1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)

SI EXISTEN-ARCHIVOS

(* INICIA-MOVIMTOS *)
REPITE VAR =1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)

SINO

(* PREPARA-MOVIMTOS *)
LEE PRIMER-REGISTRO-MOV
REPITE VAR =1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
LEE SEGUNDO-REGISTRO-MOV
REPITE VAR =1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
APUNTA-DOR-MOVS --> MOVS-IND.

(* ACEPTA-ALTAS *)

(* PROCESA-DATOS-PERSONALES *)

BORRA PANTALLA
MUESTRA CAMPOS-FORMA-1
INICIALIZA-VARIABLES
MIENTRAS INFORMACION-INCORRECTA O FIN-ALTAS
LEE ENT-RFC
SI NO OCURRE FIN-ALTAS
SI NO ES PRIMER REGISTRO EN PERSONAL

ENT-RFC --> PERS-RFC
LEE PERSONAL
SI EXISTE RFC
MUESTRA "** ESTE RFC YA FUE DADO DE ALTA **"
SINO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)

SI NO OCURRE FIN-ALTAS
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-FOLIO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-NOMBRE
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-TELEFONO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-DOMICILIO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-COLONIA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-CODIGO-POSTAL
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-FE-INGRESO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-ANTIG-UNAM
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-TP-PERSONAL
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-NACIONALIDAD
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-SEXO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-ESTADO-CIVIL

VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-FE-ACTUALIZA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)

(* PROCESA MOVIMIENTOS *)

MIENTRAS NO OCURRA FIN-MOVIMTOS O FIN-ALTAS
BORRA PANTALLA
MUESTRA CAMPOS-FORMA-2
DESPLIEGA-RFC
INCIALIZA-VARIABLES
MIENTRAS INFORMACION-INCORRECTA O OCURRA FIN-MOVIMTOS
LEE ENT-ADSCRIPCION
SI NO OCURRE FIN-MOVIMTOS
SI NO ES PRIMER REGISTRO EN PERSONAL
ENT-RFC --> PERS-RFC
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)

SI NO OCURRE FIN-MOVIMTOS
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-CARRERA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-TP-MOVIMTO
VALIDA DATOS-DE-ENTRADA
SI INFORMACION-INCORRECTA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-CAUSA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-CATEGORIA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-PLAZA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-FE-INICIO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-FE-TERMINO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA

```

        LEE ENT-ASIGNATURA
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-GRUPO
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-SALON
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-DIAS
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-HR-ENTRADA
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-HR-SALIDA
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-HRS-TEORCAS
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-PRACTCAS
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-APOYO
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)
    MIENTRAS INFORMACION-INCORRECTA
        LEE ENT-COD-PROGMTCO
        VALIDA DATOS-DE-ENTRADA
    FIN (MIENTRAS)

```

(* ASIGNA MOVIMIENTO *)

SI NO OCURRIO FIN-MOVIMTOS

(* PROCESA-NIVEL *)

```

    ACTUALIZA APUNTAORES DE MOVS
    (* VERIFICA MOVIMIENTOS *)
        LEE MOVIMTOS
        ACTUALIZA APUNTAORES DE MOVS
        REESCRIBE MOVIMTOS
    (* ESCRIBE-MOVIMIENTO *)
        SI ES PRIMER MOVIMIENTO
            MOVS-IND --> PERS-APUNTAOR-MOVS
            ENT-PARTE-3 --> MOVS-PARTE-3
        ESCRIBE EN MOVIMTOS

```

```
SINO
  SI NO ES EL PRIMER MOVIMIENTO
    INICIALIZA APUNTAORES
  SINO
    ACTUALIZA APUNTAORES DE MOVS
```

```
(* ESCRIBE ALTA PERSONAL *)
```

```
ENT-RFC --> PERS-RFC
ENT-PARTE-1 --> PERS-PARTE-1
ENT-PARTE-2 --> PERS-PARTE-2
ESCRIBE EN PERSONAL
```

```
(* FINALIZA ALTAS *)
```

```
MOVS-IND --> ULTIMO-MOV
1 --> MOVS-IND
REPITE VAR = 1 ,10 ,1
  INTERCAMBIA-NIVELES
FIN (REPITE)
REESCRIBE EN MOVIMTOS
2 --> MOVS-IND
REPITE VAR =1 ,10 ,1
  INTERCAMBIA-NIVELES
FIN (REPITE)
REESCRIBE EN MOVIMTOS
CIERRA ARCHIVOS
EJECUTA "OPCIONES".
```

```
(* INTERCAMBIA-NIVELES *)
```

```
SI VA A LEER APUNTAORES
  SI MOVS-IND = 1
    REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-I (VAR)
  SINO
    REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-F (VAR)
SINO
  SI MOVS-IND = 1
    TABLA-NIVELES-I (VAR) --> REG-TABLA-NIVELES (VAR)
  SINO
    TABLA-NIVELES-F (VAR) --> REG-TABLA-NIVELES (VAR)
```


PROCESO DE ALTAS POR LOTE

(* EJECUTA ALTAS *)

INICIA-ALTASEAT.

MIENTRAS NO OCURRA FIN-DATOS
 ACEPTA-REGISTROS
FIN (MIENTRAS).

IMPRIME-RESULTADOS.

FINALIZA-ALTASINT.

(* INICIA-ALTASINT *)

INICIALIZA VARIABLES

MIENTRAS INFORMACION-INCORRECTA

 (* ACEPTA-FECHA-ENTRADA *)

 BORRA PANTALLA

 MUESTRA "TECLEE FECHA (ddmmaa) >> "

 LEE ENT-FECHA

 VALIDA DATOS-DE-ENTRADA

 SI INFORMACION CORRECTA

 ENT-DIA --> DIA

 ENT-MES --> MES

 ENT-ANIO --> ANIO

 INICIALIZA VARIABLES

 SI EXISTEN ARCHIVOS

 (* INICIA MOVIMTOS *)

 REPITE VAR=1 ,10 ,1

 INTERCAMBIA-NIVELES

 FIN (REPITE)

 SINO

 (* PREPARA MOVIMTOS *)

 LEE PRIMER REGISTRO

 REPITE VAR=1 ,10 ,1

 INTERCAMBIA-NIVELES

 FIN (REPITE)

 LEE SEGUNDO REGISTRO

 REPITE VAR=1 ,10 ,1

 INTERCAMBIA-NIVELES

 FIN (REPITE)

 APUNTADOR-MOVS --> MOVS-IND.

(* ACEPTA-REGISTROS *)

```

MIENTRAS DATS-RFC <> BH-RFC
  (* VERIFICA CAMBIO DE EFC *)
  SI BH-RFC LLEGA POR PRIMERA VEZ
    (* ASIGNA PERSONAL *)
    0 --> MOVS-LIGA
    REESCRIBE MOVIMTOS
    ENT-RFC --> PERS-RFC
    DATOS-PERS-1 --> PERS-PARTE-1
    DATOS-PERS-2 --> PERS-PARTE-2
    ESCRIBE EN PERSONAL
FIN (MIENTRAS)

```

```

(* PROCESA REGISTROS *)

```

```

DATS-NUM-REGISTRO --> REGISTRO
SI REGISTRO = 1
  ESCRIBE-REGISTRO
  LEE-Y-VAL-RFC
  SI NO EXISTE RFC
    VALIDA DATOS-DE-ENTRADA
    SI INFORMACION INCORRECTA
      ERR-PERS + 1 --> ERR-PERS
    SINO
      DATOS-PARTE-1 --> DATOS-PERS-1
    SINO
      ERR-PERS + 1 --> ERR-PERS
  SINO
    SI REGISTRO = 2
      ESCRIBE-REGISTRO
      VALIDA DATOS-DE-ENTRADA
      SI INFORMACION INCORRECTA
        ERR-PERS + 1 --> ERR-PERS
      SINO
        DATOS-PARTE-2 --> DATOS-PERS-2
    SINO
      REGISTRO-DATS --> DATOS-PARTE-3
      ESCRIBE-REGISTRO
      VALIDA DATOS-DE-ENTRADA
      SI INFORMACION INCORRECTA
        ERR-MOVS + 1 --> ERR-MOVS
      SINO
        SI ERR-PERS = 0
          (* ASIGNA MOVIMIENTO *)
          ACTUALIZA APUNTAORES-MOV
          SI TABLA-NIVELES-I (NIVEL) > 0
            TABLA-NIVELES-F (NIVEL) --> MOVS-IND
            (* VERIFICA MOVIMIENTOS *)
            LEE MOVIMTOS
            MOVS-IND + 1 --> MOVS-LIGA
            REESCRIBE EN MOVIMTOS
            MOVS-IND-AUX --> MOVS-NIVEL
          SINO
            MOVS-IND --> TABLA-NIVELES-F (NIVEL)
            (* ESCRIBE MOVIMIENTO *)

```

SI ES PRIMER MOVIMIENTO
MOVS-IND --> PERS-APUNTADOR-MOVS
DATOS-PARTE-3 --> MOVS-PARTE-3
ESCRIBE EN MOVIMTOS

(* ESCRIBE-REGISTRO *)

SI REGISTRO = 1
 DATOS-RFC --> LIN-RFC
 IMPRIME LINEA
 DATOS-PERS-PARTE-1 --> LIN-PERS-PARTE-1
 IMPRIME LINEA
SINO
 SI REGISTRO = 2
 DATOS-PERS-PARTE-21 --> LIN-PERS-PARTE-21
 IMPRIME LINEA
 DATOS-PERS-PARTE-22 --> LIN-PERS-PARTE-22
 IMPRIME LINEA
 DATOS-FE-ACTUALIZA --> LIN-FE-ACTUALIZA
 IMPRIME LINEA
 SINO
 DATOS-MOVS-PARTE-1 --> LIN-MOVS-PARTE-1
 IMPRIME LINEA
 DATOS-MOVS-PARTE-2 --> LIN-MOVS-PARTE-2
 IMPRIME LINEA
 DATOS-MOVS-PARTE-3 --> LIN-MOVS-PARTE-3
 IMPRIME LINEA
 DATOS-MOVS-PARTE-4 --> LIN-MOVS-PARTE-4
 IMPRIME LINEA,

(* ESCRIBE EN ARCHIVOS *)

SI NUMERO REGISTRO > 2
 SI ERR-PERS = 0 Y ERR-MOVS = 0
 CONT-BUENAS + 1 --> CONT-BUENAS
 ESCRIBE EN BUENAS
 SINO
 CONT-MALAS + 1 --> CONT-MALAS
 ESCRIBE EN MALAS
SINO
 SI ERR-PERS = 0
 CONT-BUENS + 1 --> CONT-BUENAS
 ESCRIBE EN BUENAS
 SINO
 CONT-MALAS + 1 --> CONT-MALAS
 ESCRIBE EN MALAS

(* IMPRIME RESULTADOS *)

MUESTRA "---- IMPRIMIENDO RESULTADOS ----"

```

SI CONTADOR-BUENAS > 0
  LEE BUENAS
  1 --> ESC-ARCH
  MIENTRAS NO OCURRA FIN-BUENAS
    IMPRIME-ARCHIVOS
  FIN (MIENTRAS)
SI CONTADOR-MALAS > 0
  LEE MALAS
  2 --> ESC-ARCH
  MIENTRAS NO OCURRA FIN-MALAS
    IMPRIME-ARCHIVOS
  FIN (MIENTRAS)
  LEE DIAGNOSTICO
  3 --> ESC-ARCH
  MIENTRAS NO OCURRA FIN-DIAGSTCO
    IMPRIME-ARCHIVOS
  FIN (MIENTRAS)

```

(* IMPRIME ARCHIVOS *)

```

SI ESC-ARCH = 1
  BNAS-RFC --> ARCH-RFC
  REGISTRO-BNAS --> REG-AUXILIAR
SINO
  SI ESC-ARCH = 2
    MLAS-RFC --> ARCH-RFC
    REGISTRO-MLAS --> REG-AUXILIAR
SI ARCH-RFC <> BH-RFC
  ESCRIBE REG-AUXILIAR
SI ESC-ARCH = 1
  LEE BUENAS
SINO
  SI ESC-ARCH = 2
    LEE MALAS
  SINO
    LEE DIAGNOSTICO

```

(* FINALIZA-ALTASBAT *)

```

MOVS-IND --> ULTIMO-MOV
1 --> MOVS-IND
REPITE VAR = 1 ,10 ,1
  INTERCAMBIA-NIVELES
FIN (REPITE)
REESCRIBE EN MOVIMTOS
2 --> MOVS-IND
REPITE VAR =1 ,10 ,1
  INTERCAMBIA-NIVELES
FIN (REPITE)
REESCRIBE EN MOVIMTOS
CIERRA ARCHIVOS
EJECUTA "OPCIONES".

```

(* INTERCAMBIA-NIVELES *)

SI SE VA A LEER APUNTA^oDORES

SI MOVS-IND = 1

REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-I (VAR)

SINO

REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-F (VAR)

SINO

SI MOVS-IND = 1

TABLA-NIVELES-I (VAR) --> REG-TABLA-NIVELES (VAR)

SINO

TABLA-NIVELES-F (VAR) --> REG-TABLA-NIVELES (VAR)

PROCESO DE ACTUALIZACION

(* EJECUTA ACTUALIZACION *)

INICIA-ACTUALIZA.

MIENTRAS NO OCURRA FIN-ACTUALIZA
REALIZA-ACTUALIZA
FIN (MIENTRAS),

FINALIZA-ACTUALIZA.

(* INICIA-ACTUALIZA *)

INICIALIZA VARIABLES
ABRE ARCHIVOS
SI NO EXISTEN ARCHIVOS
BORRA PANTALLA
MUESTRA "NO ENCUENTR LOS ARHIVOS"
MUESTRA "PARA REALIZAR LA ACTUALIZACION"
SINO

(* PREPARA-MOVIMTOS *)
LEE PRIMER REGISTRO
REPITE VAR=1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
LEE SEGUNDO REGISTRO
REPITE VAR=1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
AFUNTADOR-MOVS --> MOVS-IND,

(* REALIZA-ACTUALIZA *)

MIENTRAS NO HAYA INFORMACION CORRECTA
LEE-Y-VALIDA-RFC
FIN (MIENTRAS)

PROCESA INFORMACION

(* LEE-Y-VALIDA-RFC *)

BORRA PANTALLA
MUESTRA MENSAJE

```
LEE ENT-RFC
VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-RFC --> PERS-RFC
  LEE PERSONAL
SI NO EXISTE RFC
  MUESTRA "*** NO EXISTE ESTE RFC ***"
  MUESTRA "*** EN PERSONAL ***"
```

(* PROCESA INFORMACION *)

```
SI PERS-APUNTADOR-MOVS > 0
  MIENTRAS NO OCURRA FIN-MOVIMTOS
    RECORRE MOVIMTOS
  FIN (MIENTRAS)
  MIENTRAS NO OCURRA FIN-MOVIMTOS
    PROCESA-MOVIMIENTOS
  FIN (MIENTRAS)
```

(* RECORRE MOVIMTOS *)

```
LEE MOVIMTOS
SI MOVS-LIGA > 0
  MOVS-LIGA --> MOVS-IND
SINO
  ACTUALIZA MOVS-LIGA
  REESCRIBE MOVIMTOS
```

(* PROCESA-MOVIMTOS *)

```
BORRA PANTALLA
MUESTRA CAMPOS-FORMA-2
DESPLIEGA-RFC
INICIALIZA-VARIABLES
MIENTRAS INFORMACION-INCORRECTA O FIN-MOVIMTOS
  LEE ENT-ADSCRIPCION
  SI NO OCURRE FIN-MOVIMTOS
    SI NO ES PRIMER REGISTRO EN PERSONAL
      ENT-RFC --> PERS-RFC
      VALIDA DATOS-DE-ENTRADA
  FIN (MIENTRAS)
SI NO OCURRE FIN-MOVIMTOS
  MIENTRAS INFORMACION-INCORRECTA
    LEE ENT-CARRERA
    VALIDA DATOS-DE-ENTRADA
  FIN (MIENTRAS)
  MIENTRAS INFORMACION-INCORRECTA
    LEE ENT-TP-MOVIMTO
    VALIDA DATOS-DE-ENTRADA
    SI INFORMACION-INCORRECTA
      FIN (MIENTRAS)
  MIENTRAS INFORMACION-INCORRECTA
```

LEE ENT-CAUSA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-CATEGORIA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-PLAZA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-FE-INICIO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-FE-TERMINO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-ASIGNATURA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-GRUPO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-SALON
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-DIAS
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-HR-ENTRADA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-HR-SALIDA
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-HRS-TEORCAS
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-PRACTCAS
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
LEE ENT-APOYO
VALIDA DATOS-DE-ENTRADA
FIN (MIENTRAS)


```

MIENTRAS INFORMACION-INCORRECTA
  LEE ENT-COD-FROGMTCO
  VALIDA DATOS-DE-ENTRADA
  FIN (MIENTRAS)
(* ASIGNA MOVIMIENTO *)
SI NO OCURRIO FIN-MOVIMTOS
  (* PROCESA-NIVEL *)
  ACTUALIZA APUNTAORES DE MOVS
  (* VERIFICA MOVIMIENTOS *)
  LEE MOVIMTOS
  ACTUALIZA APUNTAORES DE MOV
  REESCRIBE MOVIMTOS
  (* ESCRIBE-MOVIMIENTO *)
  SI ES PRIMER MOVIMIENTO
    MOVS-IND --> PERS-APUNTADOR-MOVS
    ENT-PARTE-3 --> MOVS-PARTE-3
    ESCRIBE EN MOVIMTOS
SINO
  SI NO ES EL PRIMER MOVIMIENTO
    INICIALIZA APUNTAORES
  SINO
    ACTUALIZA APUNTAORES DE MOVS

```

```
(* FINALIZA ACTUALIZA *)
```

```

MOVS-IND --> ULTIMO-MOV
1 --> MOVS-IND
REPITE VAR = 1 ,10 ,1
  INTERCAMBIA-NIVELES
FIN (REPITE)
REESCRIBE EN MOVIMTOS
2 --> MOVS-IND
REPITE VAR =1 ,10 ,1
  INTERCAMBIA-NIVELES
FIN (REPITE)
REESCRIBE EN MOVIMTOS
CIERRA ARCHIVOS
EJECUTA "OPCIONES".

```

```
(* INTERCAMBIA-NIVELES *)
```

```

SI SE VA A LEER APUNTAORES
  SI MOVS-IND = 1
    REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-I (VAR)
  SINO
    REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-F (VAR)
SINO
  SI MOVS-IND = 1
    TABLA-NIVELES-I (VAR) --> REG-TABLA-NIVELES (VAR)
  SINO
    TABLA-NIVELES-F (VAR) --> REG-TABLA-NIVELES (VAR)

```

PROCESO DE MODIFICACION

(* EJECUTA MODIFICACION *)

INICIA-MODIFICA.

MIENTRAS NO OCURRA FIN-MODIFICA
REALIZA-MODIFICA
FIN (MODIFICA).

FINALIZA-MODIFICA.

(* INICIA-MODIFICA *)

INICIALIZA VARIABLES
SI NO EXISTEN ARCHIVOS
BORRA PANTALLA
MUESTRA "NO ESTAN LOS ARCHIVOS"
MUESTRA "PARA REALIZAR LA MODIFICACION"

(* REALIZA-MODIFICA *)

MIENTRAS NO HAYA OPCION CORRECTA
DESPLIEGA-MENU
FIN (MIENTRAS)
MIENTRAS NO SE LEA CORRECTAMENTE RFC
LEE-Y-VALIDA-RFC
FIN (MIENTRAS)

(* DESPLIEGA-MENU *)

BORRA PANTALLA
MUESTRA DESPL-MENU-OPCIONES
LEE OPCION-M
SI OPCION-M = 1 o 2
MIENTRAS OPCION-TP SEA INCORRECTA
(* TOMA-TP-MODIFICACION *)
BORRA PANTALLA
MUESTRA MENU-CN-CR
LEE OPCION-TP
SI OPCION-TP = "CR"
"CORREGIR" --> TP-MENSAJE
SINO

```

"CANCELAR" --> TP-MENSAJE
FIN (MIENTRAS)
SINO
SI OPCION = 3
"SI" --> MODIFICA --> INFORMACION

```

```
(* LEE-Y-VALIDA-RFC *)
```

```

BORRA PANTALLA
MUESTRA MENSAJE
LEE ENT-RFC
VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
ENT-RFC --> PERS-RFC
LEE PERSONAL
SI NO EXISTE RFC
MUESTRA "*** NO EXISTE ESTE RFC ***"
MUESTRA "*** EN PERSONAL ***"

```

```
(* PROCESA-INFORMACION *)
```

```

SI PERS-APUNTA-DOR-MOVS > 0
"NO" --> CNCR-MOVS
SINO
"SI" --> CNCR-MOVS
(* MODIFICACIONES *)
BORRA PANTALLA
SI OPCION-M = 1
(* DESPLIEGA-OPCION-PERS *)
PREPARA DATOS PERSONALES PARA SER DESPLEGADOS
BORRA PANTALLA
MUESTRA DESPL-DATOS-PERSONALES
SI OPCION-TP = "CN"
SI CNCR-MOVS = "NO"
MIENTRAS NO OCURRA FIN-CNCR-MOVS
CN-CR-MOVIMIENTOS
FIN (MIENTRAS)
BORRA REGISTRO DE PERSONAL
SINO
SINO
MIENTRAS INFORMACION-INCORRECTA
MUESTRA "TECLEE NVO. NUM. FOLIO [no]"
LEE ENT-FOLIO
VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
ENT-FOLIO --> PERS-FOLIO
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
MUESTRA "TECLEE NVO. NOMBRE [no]"
LEE ENT-NOMBRE
VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA

```

ENT-NOMBRE --> PERS-NOMBRE
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVO, TELEFONO [no]"
 LEE ENT-TELEFONO
 VALIDA DATOS-DE-ENTRADA
 SI INFORMACION CORRECTA
 ENT-TELEFONO --> PERS-TELEFONO
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVO, DOMICILIO [no]"
 LEE ENT-DOMICILIO
 VALIDA DATOS-DE-ENTRADA
 SI INFORMACION CORRECTA
 ENT-DOMICILIO --> PERS-DOMICILIO
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVA, COLONIA [no]"
 LEE ENT-COLONIA
 VALIDA DATOS-DE-ENTRADA
 SI INFORMACION CORRECTA
 ENT-COLONIA --> PERS-COLONIA
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVO, COD POSTAL [no]"
 LEE ENT-CODIGO-POSTAL
 VALIDA DATOS-DE-ENTRADA
 SI INFORMACION CORRECTA
 ENT-CODIGO-POSTAL --> PERS-CODIGO-POSTAL
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVA, FE INGRESO [no]"
 LEE ENT-FE-INGRESO
 VALIDA DATOS-DE-ENTRADA
 SI INFORMACION CORRECTA
 ENT-FE-INGRESO --> PERS-FE-INGRESO
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVA, ANTIG UNAM [no]"
 LEE ENT-ANTIG-UNAM
 VALIDA DATOS-DE-ENTRADA
 SI INFORMACION CORRECTA
 ENT-ANTIG-UNAM --> PERS-ANTIG-UNAM
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVO, TP PERSONAL [no]"
 LEE ENT-TP-PERSONAL
 VALIDA DATOS-DE-ENTRADA
 SI INFORMACION CORRECTA
 ENT-TP-PERSONAL --> PERS-TP-PERSONAL
 FIN (MIENTRAS)
 MIENTRAS INFORMACION-INCORRECTA
 MUESTRA "TECLEE NVA, NACIONALIDAD [no]"
 LEE ENT-NACIONALIDAD
 VALIDA DATOS-DE-ENTRADA

```

    SI INFORMACION CORRECTA
      ENT-NACIONALIDAD --> PERS-NACIONALIDAD
    FIN (MIENTRAS)
  MIENTRAS INFORMACION-INCORRECTA
    MUESTRA "TECLEE NVO. SEXO [no]"
    LEE ENT-SEXO
    VALIDA DATOS-DE-ENTRADA
    SI INFORMACION CORRECTA
      ENT-SEXO --> PERS-SEXO
    FIN (MIENTRAS)
  MIENTRAS INFORMACION-INCORRECTA
    MUESTRA "TECLEE NVO. ESTADO CIVIL [no]"
    LEE ENT-ESTADO-CIVIL
    VALIDA DATOS-DE-ENTRADA
    SI INFORMACION CORRECTA
      ENT-ESTADO-CIVIL --> PERS-ESTADO-CIVIL
    FIN (MIENTRAS)
  MIENTRAS INFORMACION-INCORRECTA
    MUESTRA "TECLEE NVA. FECHA ACTUALIZA [no]"
    LEE ENT-FE-ACTUALIZA
    VALIDA DATOS-DE-ENTRADA
    SI INFORMACION CORRECTA
      ENT-FE-ACTUALIZA --> PERS-FE-ACTUALIZA
    FIN (MIENTRAS)
  SINO
    SI CNCR-MOVS = "NO"
      MIENTRAS NO OCURRA FIN-CNCR-MOVS
      CNCR-MOVIMIENTOS
    FIN (MIENTRAS)
  SINO
    BORRA PANTALLA
    MUESTRA "NO EXISTEN MOVIMIENTOS"

```

(* CN-CR-MOVIMIENTOS *)

```

  LEE MOVIMTOS
  SI OPCION-M = 2
    (* DESPLIEGA-OPCION-MOVS *)
    LEE MOVIMTOS
    PREPARA MOVIMIENTOS PARA SER DESPLEGADOS
    BORRA PANTALLA
    MUESTRA DESPL-MOVIMIENTOS
    MIENTRAS NO HAYA RESPUESTA-CORRECTA
      ACEPTA-RESPUESTA
    FIN (MIENTRAS)
  SINO
    SI OPCION-TP = "CN"
      ETIQUETA REGISTRO PARA CANCELAR
    SINO
      MIENTRAS INFORMACION-INCORRECTA
        MUESTRA "TECLEE NVA CARRERA [no]"
        LEE ENT-CARRERA
        VALIDA DATOS-DE-ENTRADA
        SI INFORMACION CORRECTA

```

```

      ENT-CARRERA --> MOVS-CARRERA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVO TP-MOVIMTO [no]"
  LEE ENT-TP-MOVIMTO
  VALIDA DATOS-DE-ENTRADA
  SI INFORMACION CORRECTA
    ENT-TP-MOVIMTO --> MOVS-TP-MOVIMTO
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA CAUSA [no]"
  LEE ENT-CAUSA
  VALIDA DATOS-DE-ENTRADA
  SI INFORMACION CORRECTA
    ENT-CAUSA --> MOVS-CAUSA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA CATEGORIA [no]"
  LEE ENT-CATEGORIA
  VALIDA DATOS-DE-ENTRADA
  SI INFORMACION CORRECTA
    ENT-CATEGORIA --> MOVS-CATEGORIA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA PLAZA [no]"
  LEE ENT-PLAZA
  VALIDA DATOS-DE-ENTRADA
  SI INFORMACION CORRECTA
    ENT-PLAZA --> MOVS-PLAZA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA FE-INICIO [no]"
  LEE ENT-FE-INICIO
  VALIDA DATOS-DE-ENTRADA
  SI INFORMACION CORRECTA
    ENT-FE-INICIO --> MOVS-FE-INICIO
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA FE TERMINO [no]"
  LEE ENT-FE-TERMINO
  VALIDA DATOS-DE-ENTRADA
  SI INFORMACION CORRECTA
    ENT-FE-TERMINO --> MOVS-FE-TERMINO
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA ASIGNATURA [no]"
  LEE ENT-ASIGNATURA
  VALIDA DATOS-DE-ENTRADA
  SI INFORMACION CORRECTA
    ENT-ASIGNATURA --> MOVS-ASIGNATURA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVO GRUPO [no]"
  LEE ENT-GRUPO
  VALIDA DATOS-DE-ENTRADA

```

```

SI INFORMACION CORRECTA
  ENT-GRUPO --> MOVS-GRUPO
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVO SALON [no]"
  LEE ENT-SALON
  VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-SALON --> MOVS-SALON
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVOS DIAS [no]"
  LEE ENT-DIAS
  VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-DIAS --> MOVS-DIAS
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA HR-ENTRADA [no]"
  LEE ENT-HR-ENTRADA
  VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-HR-ENTRADA --> MOVS-HR-ENTRADA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVA HR SALIDA [no]"
  LEE ENT-HR-SALIDA
  VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-HR-SALIDA --> MOVS-HR-SALIDA
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVAS HRS TEORCAS [no]"
  LEE ENT-HRS-TEORCAS
  VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-HRS-TEORCAS --> MOVS-HRS-TEORCAS
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVAS HRS PRACTCAS [no]"
  LEE ENT-PRACTCAS
  VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-HRS-PRACTCAS --> MOVS-HRS-PRACTCAS
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVAS HRS APOYO [no]"
  LEE ENT-HRS-APOYO
  VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-HRS-APOYO --> MOVS-HRS-APOYO
FIN (MIENTRAS)
MIENTRAS INFORMACION-INCORRECTA
  MUESTRA "TECLEE NVO COD PROGMTCO [no]"
  LEE ENT-COD-PROGMTCO

```

```
VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
  ENT-COD-PROGMTCO --> MOVS-COD-PROGMTCO
  FIN (MIENTRAS)
MIENTRAS NO HAYA RESPUESTA-CORRECTA
  ACEPTA-RESPUESTA
  FIN (MIENTRAS)
SI RESPUESTA = "SI"
  REESCRIBE MOVIMTOS
```

(* ACEPTA RESPUESTA *)

```
SI OPCION-TP = "CR"
  MUESTRA "ES ESTE EL REGISTRO A CORREGIR?"
SINO
  MUESTRA "ES ESTE EL REGISTRO A CANCELAR?"
SI RESPUESTA = "SI"
  REESCRIBE MOVIMTOS
```

(* FINALIZA MODIFICA *)

```
SI EXISTEN ARCHIVOS
  CIERRALOS
BORRA PANTALLA
MUESTRA "FIN DE MODIFICACIONES"
MUESTRA "=====",
```


PROCESO DE CONSULTA

(* EJECUTA CONSULTA *)

INICIA-CONSULTA.

MIENTRAS NO OCURRA FIN-CONSULTA
REALIZA CONSULTA
FIN (MIENTRAS).

FINALIZA-CONSULTA.

(* INICIA-CONSULTA *)

INICIALIZA VARIABLES
SI NO EXISTEN ARCHIVOS
BORRA PANTALLA
MUESTRA "NO ESTAN ARCHIVOS PARA"
MUESTRA "REALIZAR LA CONSULTA ".

(* REALIZA-CONSULTA *)

MIENTRAS NO HAYA OPCION-CORRECTA
(* DESPLIEGA-MENU *)
BORRA PANTALLA
DESPLIEGA OPCIONES
FIN (MIENTRAS)
MIENTRAS NO HAYA INFORMACION-CORRECTA
(* LEE-Y-VALIDA-RFC *)
BORRA PANTALLA
MUESTRA MENSAJE
LEE ENT-RFC
VALIDA DATOS-DE-ENTRADA
SI INFORMACION CORRECTA
ENT-RFC --> PERS-RFC
LEE PERSONAL
SI NO EXISTE RFC
MUESTRA "*** NO EXISTE ESTE RFC ***"
MUESTRA "*** EN PERSONAL ***"
FIN (MIENTRAS)
PROCESA INFORMACION

(* PROCESA INFORMACION *)

SI OPCION = 1
DESPLIEGA-OPCION-PERS

SINO
SI PERS-APUNTADOR-MOVS > 0
MIENTRAS NO OCURRA FIN-MOVIMTOS
DESPLIEGA-OPCION-MOVS
FIN (MIENTRAS)
SI NO SE ENCONTRARON MOVIMIENTOS
MUESTRA " NO SE ENCONTRARON MOVIMIENTOS"

(* DESPLIEGA-OPCION-PERS *)

PREPARA DATOS PERSONALES PARA SER DESPLEGADOS
BORRA PANTALLA
MUESTRA DESPL-DATOS-PERSONALES

(* DESPLIEGA-OPCION-MOVS *)

PREPARA DATOS-MOVIMIENTOS PARA SER DESPLEGADOS
BORRA PANTALLA
MUESTRA DESPL-DATOS-MOVIMTOS

(* FINALIZA CONSULTA *)

SI EXISTEN ARCHIVOS
CIERRALOS
BORRA PANTALLA
MUESTRA "FIN DE CONSULTA"
MUESTRA "=====
EJECUTA "OPCIONES"

PROCESO PARA INFORME DE DIRECTORIO

(* EJECUTA-REPORTE *)

INICIA-REPORTE.

MIENTRAS NO OCURRA FIN-SORTEO
SORTEA-REPORTE
FIN (MIENTRAS).

MIENTRAS NO OCURRA FIN-PROCESO
IMPRIME-REPORTE
FIN (MIENTRAS).

FINALIZA-REPORTE.

(* INICIA REPORTE *)

INICIALIZA VARIABLES
ABRE ARCHIVOS
SI NO EXISTE ARCHIVOS
BORRA PANTALLA
MUESTRA "NO ENCUENTRA ARCHIVOS"
MUESTRA "PARA REALIZAR ESTE REPORTE"
SINO
MUESTRA "PROCESANDO DATOS PARA REPORTE"
TOMA PARAMETROS DEL INFORME

(* PREPARA-MOVIMTOS *)

LEE PRIMER REGISTRO
REPITE VAR=1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
LEE SEGUNDO REGISTRO
REPITE VAR=1 ,10 ,1
INTERCAMBIA-NIVELES
FIN (REPITE)
APUNTAOR-MOVS --> MOVS-IND.

(* SORTEA REPORTE *)

MUESTRA "SORTEANDO REPORTE"
SI TIPO-INFORME = "NIVELS"
PREPARA APUNTAORES PARA NIVEL
MIENTRAS NO OCURRA FIN-SORTEO
REALIZA-SORTEO

```
FIN (MIENTRAS)
SINO
  ESPACIOS --> REP-ADSCRIPCION
  REPITE VAR=3 ,ULTIMO-MOV ,1
  REALIZA-SORTEO
  FIN (REPITE)
MUESTRA "IMPRIMIENDO REPORTE"
```

```
(* REALIZA-SORTEO *)
```

```
VAR --> MOVS-IND
LEE MOVIMTOS
SI TIPO-INFORME = "GLOBAL"
  ESPACIOS --> REP-ADSCRIPCION
SINO
  MOVS-ADSCRIPCION --> REP-ADSCRIPCION
ASIGNA CAMPOS AL REPORTE
ESCRIBE REPORTE
```

```
(* IMPRIME-REPORTE *)
```

```
LEE REPORTE
SI HAY CAMBIO EN RFC
  ESCRIBE PAGINAS
  ASIGNA INFORMACION
SUMA HORAS-CLASE
```

```
(* ESCRIBE-PAGINAS *)
```

```
SI FIN-PAGINA
  ESCRIBE ENCABEZADOS
ESCRIBE LINEA
```

```
(* ASIGNA INFORMACION *)
```

```
REP-RFC --> PERS-RFC
LEE PERSONAL
ASIGNA CAMPOS DE PERSONAL A CAMPOS DE IMPRESION
IMPRIME LINEA
```

```
(* FINALIZA REPORTE *)
```

```
CIERRA ARCHIVOS
BORRA PANTALLA
MUESTRA "FIN DE REPORTE"
MUESTRA "======"
EJECUTA "OPCIONES"
```

(* INTERCAMBIA-NIVELES *)

SI MOVS-IND = 1

REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-I (VAR)

SINO

REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-F (VAR)

PROCESO DE INFORME DETALLADO DE MOVIMIENTOS

(* EJECUTA REPORTE *)

INICIA REPORTE.

MIENTRAS NO OCURRA FIN-SORTEO
SORTEA REPORTE
FIN (MIENTRAS).

MIENTRAS NO OCURRA FIN-INFORME
IMPRIME REPORTE
FIN (MIENTRAS).

FINALIZA REPORTE.

(* INICIA REPORTE *)

INICIALIZA VARIABLES

ABRE ARCHIVOS

SI NO EXISTE ARCHIVOS

BORRA PANTALLA

MUESTRA "NO ENCUENTRA ARCHIVOS"

MUESTRA "PARA REALIZAR ESTE REPORTE"

SINO

MUESTRA "PROCESANDO DATOS PARA REPORTE"

TOMA PARAMETROS DEL INFORME

(* PREPARA-MOVIMIENTOS *)

LEE PRIMER REGISTRO

REPITE VAR=1 ,10 ,1

INTERCAMBIA-NIVELES

FIN (REPITE)

LEE SEGUNDO REGISTRO

REPITE VAR=1 ,10 ,1

INTERCAMBIA-NIVELES

FIN (REPITE)

AFUNTADOR-MOVS --> MOVS-IND.

(* SORTEA REPORTE *)

MUESTRA "SORTEANDO REPORTE"

SI TIPO-INFORME = "NIVELS"

PREPARA AFUNTADORES PARA NIVEL

MIENTRAS NO OCURRA FIN-SORTEO

REALIZA-SORTEO

```
FIN (MIENTRAS)
SINO
  ESPACIOS --> REP-ADSCRIPCION
  REPITE VAR=3 ,ULTIMO-MOV ,1
  REALIZA-SORTEO
  FIN (REPITE)
MUESTRA "IMPRIMIENDO REPORTE"
```

```
(* REALIZA-SORTEO *)
```

```
VAR --> MOVS-IND
LEE MOVIMTOS
SI TIPO-INFORME = "GLOBAL"
  ESPACIOS --> REP-ADSCRIPCION
SINO
  MOVS-ADSCRIPCION --> REP-ADSCRIPCION
ASIGNA CAMPOS AL REPORTE
ESCRIBE REPORTE
```

```
(* IMPRIME-REPORTE *)
```

```
LEE REPORTE
SI HAY CAMBIO EN ADSCRIPCION
  IMPRIME-TOTAL-HORAS
  LEE-NIVELES
  IMPRIME-DATOS-PERS
  IMPRIME-MATER-MOVS
SINO
  SI HAY CAMBIO EN RFC
    IMPRIME-TOTAL-HORAS
    IMPRIME-DATOS-PERS
    IMPRIME-MATER-MOVS
  SINO
    SI HAY CAMBIO EN ASIGNATURA
      IMPRIME-MATER-MOVS
    ASIGNA INFORMACION
```

```
(* TOTAL-HORAS *)
SI NO ES LA PRIMERA VEZ
  IMPRIME TOTAL-HORAS TEORCAS
  IMPRIME TOTAL-HORAS PRACTCAS
  IMPRIME TOTAL-HORAS APOYO
  IMPRIME TOTAL-HORAS
```

```
(* LEE-NIVELES *)
```

```
SI NIVEL1 <> REP-NIVEL
  LEE NIVEL1
SI NIVEL2 <> REP-NIVEL2
  LEE NIVEL2
SI NIVEL3 <> REP-NIVEL3
```

LEE NIVEL3.

(* IMPRIME-DATOS-PERS *)

REPT-RFC --> PERS-RFC
LEE PERSONAL
ASIGNA CAMPOS DE PERSONAL A LINEA
IMPRIME LINEA
ESCRIBE PAGINAS

(* IMPRIME-MATER-MOVS *)

SI HAY CAMBIO EN CARRERA
LEE CARRERAS
SI HAY CAMBIO EN ASIGNATURA
LEE MATERIAS

(* ESCRIBE-PAGINAS *)

SI FIN-PAGINA
ESCRIBE ENCABEZADOS
IMPRIME LINEA

(* ASIGNA INFORMACION *)

ASIGNA CAMPOS DE MOVIMIENTOS A CAMPOS DE IMPRESION
SI FE-INICIO > FE-TERMINO
ESPACIOS --> ESTADO-ACTUAL
SINO
"VENCIDO" --> ESTADO-ACTUAL
IMPRIME LINEA
ESCRIBE PAGINAS

(* FINALIZA REPORTE *)

CIERRA ARCHIVOS
BORRA PANTALLA
MUESTRA "FIN DE REPORTE"
MUESTRA "=====
EJECUTA "OPCIONES"

(* INTERCAMBIA-NIVELES *)

SI MOVS-IND = 1
REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-I (VAR)
SINO
REG-TABLA-NIVELES (VAR) --> TABLA-NIVELES-F (VAR)

PROCESO DE VALIDACION

(* VALIDACION *)

INICIA-VALIDA.

REALIZA VALIDA-INFORMACION HASTA FIN-VALIDA-INFORMACION.

FIN-VALIDA.

(* INICIA VALIDACION *)

INICIALIZA VARIABLES

SI PROCESO = "I"

NUMERO-CAMPO --> OPCION

(* VALIDA-INFORMACION *)

SI PROCESO = "B"

VALIDA RFC

SI REGISTRO = 1

VALIDA DESDE FOLIO HASTA NOMBRE

SINO

SI REGISTRO = 2

VALIDA DESDE TELEFONO HASTA ESTADO-CIVIL

SINO

VALIDA DESDE ADSCRIPCION HASTA COD-PROGMTCO

SINO

SI REGISTRO = 1

CASO NUMERO-CAMPO=1 VALIDA RFC

NUMERO-CAMPO=2 VALIDA FOLIO

NUMERO-CAMPO=3 VALIDA NOMBRE

NUMERO-CAMPO=4 VALIDA TELEFONO

SINO

SI REGISTRO = 2

CASO NUMERO-CAMPO=5 VALIDA DOMICILIO

NUMERO-CAMPO=6 VALIDA COLONIA

NUMERO-CAMPO=7 VALIDA CODIGO POSTAL

NUMERO-CAMPO=8 VALIDA FE-INGRESO

NUMERO-CAMPO=9 VALIDA ANTIG-UNAM

NUMERO-CAMPO=10 VALIDA TP-PERSONAL

NUMERO-CAMPO=11 VALIDA NACIONALIDAD

NUMERO-CAMPO=12 VALIDA SEXO

NUMERO-CAMPO=13 VALIDA ESTADO CIVIL

NUMERO-CAMPO=14 VALIDA FE-ACTUALIZA

SINO

CASO NUMERO-CAMPO=15 VALIDA ADSCRIPCION

NUMERO-CAMPO=16 VALIDA CARRERA
NUMERO-CAMPO=17 VALIDA TP-MOVIMTO
NUMERO-CAMPO=18 VALIDA CAUSA
NUMERO-CAMPO=19 VALIDA CATEGORIA
NUMERO-CAMPO=20 VALIDA PLAZA
NUMERO-CAMPO=21 VALIDA FE-INICIO
NUMERO-CAMPO=22 VALIDA FE-TERMINO
NUMERO-CAMPO=23 VALIDA ASIGNATURA
NUMERO-CAMPO=24 VALIDA GRUPO
NUMERO-CAMPO=25 VALIDA SALON
NUMERO-CAMPO=26 VALIDA DIAS
NUMERO-CAMPO=27 VALIDA HR-ENTRADA
NUMERO-CAMPO=28 VALIDA HR-SALIDA
NUMERO-CAMPO=29 VALIDA HRS-TEORICAS
NUMERO-CAMPO=30 VALIDA HRS-PRACTICAS
NUMERO-CAMPO=31 VALIDA HRS-APOYO
NUMERO-CAMPO=32 VALIDA COD-PROGMTCO
NUMERO-CAMPO=33 VALIDA FECHA.

(* FIN-VALIDACION *).

(* FIN-VALIDA *)

PREPARA PARAMETROS
REGRESA.

SELECCION DEL LENGUAJE DE PROGRAMACION .- Debido a que uno de los objetivos en la elaboración del sistema era que el programa pudiera fácilmente implementarse en otro equipo, con un mínimo de cambios, (si es que ninguno), la selección del lenguaje se realizó desde la fase de Análisis.

Se tiene que el factor más importante a considerar es la minimización del costo de vida de la programación, incluyendo tanto el costo de desarrollo, como el de mantenimiento y de acuerdo con la aplicación de la metodología de sistemas desarrollada durante este trabajo.

Para la selección del lenguaje de programación se tomaron varios pasos: primero el de hacer una lista de lenguajes que pudieran ser factibles como: Pascal / Fortran / Basic / Cobol.

Eliminándose algunos lenguajes de esta lista, debido a que no cumplen con las necesidades de información previstas por el sistema, como son el manejo avanzado de archivos, estandarización de los lenguajes, y fácil transportabilidad.

Así tenemos, que la posibilidad de usar el lenguaje para el desarrollo del sistema esta entre Basic y Cobol.

Estudiando algunos otros factores para la evaluación de los lenguajes, tenemos:

Transportabilidad .- Aquí se observa, que el compilador de Cobol traduce programas que son altamente compatibles, el cambio de los programas fuentes no es necesario (o al menos en forma mínima), mientras que en el Basic, de Cromemco, para que sea compatible con el Basic de una PC, es necesario realizar cambios principalmente al momento de imprimir y manejar cadenas de caracteres.

Otro factor tomado en cuenta, es la eficiencia en espacio y que es muy importante para la implementación del sistema, ya que se debe poder adaptar para una capacidad mínima de 64 K de memoria. En este caso se tiene que el Basic está restringido, pues mientras que en una PC ocupa aproximadamente 60 K de memoria, en una Cromemco con todas sus rutinas avanzadas ocupa 32 K, y si tomamos en cuenta el espacio del sistema operativo, el espacio en memoria es raquítico. Otro parámetro es que Basic es un intérprete en los dos equipos, mientras que Cobol es un compilador lo que posibilita el crear programas que ocupen un menor espacio y con una velocidad mayor para la ejecución de programas.

Como conclusión, se tiene que el lenguaje a utilizar para este sistema es Cobol, pues permite el manejo de archivos indexados, random y secuenciales, además de

poder manipular la pantalla lo que nos ayuda en gran manera a alcanzar los fines del sistema.

DOCUMENTACION .- La documentación del sistema es de suma importancia, ya que nos permite un gran control de cada una de las fases dentro de la vida del sistema. Esta documentación esta dividida en varios reportes mismos que se van obteniendo desde el momento mismo que se solicita el diseño de un sistema, hasta que este se pone en operación, ya que se tienen que realizar manuales para el usuario, para el mantenimiento de tal forma que permitan adecuar fácilmente el sistema a nuevas especificaciones.

Así tenemos que durante la fase de Análisis, el documento que se obtiene es el llamado Documento de Objetivos, mismo que trae una descripción detallada de cada uno de los objetivos con que debe cumplir el sistema que se termine. Otro documento dentro del Análisis es el llamado reporte de Análisis que sirve para negociar y justificar los compromisos y las prioridades adoptadas. Finalmente dentro de esta etapa esta el documento de la Especificación de Requerimientos, que es el más importante, pues nos indicará claramente lo que el sistema debe ejecutar, obtener, procesar en base a los objetivos planteados.

Durante la fase de diseño, se genera otro reporte

que es el de arquitectura el cual indica como esta dividido el sistema, o sea el diagrama de estructura y el detallado de los módulos del sistema.

Ya en la etapa de Operación se realiza un reporte y que es de los mas importantes, dado que indica como se puede utilizar el sistema desarrollado y es el Manual del Usuario, este documento enfatiza aspectos que permiten al usuario obtener el mejor provecho de las funciones que el sistema realiza.

DESARROLLO

Esta etapa tiene como objetivos la Codificación, de los módulos, el verificar el correcto funcionamiento de ellos, la integración, a fin de darle forma a los programas, y finalmente, la integración de los módulos para formar los sistemas de programación.

La fase de desarrollo queda terminada cuando los programas generados han sido codificados y verificados para que finalmente sean implementados, una vez que se están obteniendo los resultados esperados y planteados durante la fase de análisis.

CODIFICACION .- La parte de Codificación de programas tiene como objetivo traducir las especificaciones de proceso de cada módulo, descritas anteriormente y manteniendo la estructura jerárquica de la

fase de Diseño en instrucciones ejecutables por un lenguaje de programación específico (en el caso de este trabajo es COBOL).

Existen dos tipos de enfoque que permiten trabajar para la codificación, uno el radical y otro el conservador. El primero especifica que se pueden programar y probar cada uno de los módulos inmediatamente después de diseñar al mismo, y este se realiza generalmente cuando hay premura de tiempo. El segundo enfoque se utiliza cuando el diseño del sistema se termina completamente y acto seguido, se procede a la programación de cada uno de los módulos.

Durante la codificación se procuró que los nombres de las variables fueran explícitamente declaradas en concordancia con los datos empleados durante la fase de Diseño.

También se siguió la técnica de codificación TOP DOWN, o sea desde el primer módulo (el módulo de opciones del sistema) hacia cada uno de las tareas realizadas por el Programa, aunque también se utilizó la técnica BOTTOM-UP con uno de los módulos, todo esto se observa a lo largo de la siguiente sección.

Durante la elaboración de este sistema en esta etapa, se encontraron problemas para la correcta codificación de los módulos, pues mientras el diseño detallado estaba estructurado, algunas instrucciones de

COBOL al utilizarse con una estructura IF-THEN-ELSE provocaban que se tuviera que utilizar un "perform" extra, debiéndose hacer algunos cambios en el diseño original.

Otra característica que no se percató, es que Cobol no presenta la estructura REPEAT-UNTIL, y que al principio provocó que se tuviera que añadir algunas instrucciones al codificar los módulos.

Una vez que se codificaron cada uno de los módulos, se inicio con la Integración de los mismos. Esto se explica en detalle en la siguiente sección.

Para la demostración de los programas fuentes ya codificados y compilados, se sugiere ir a los listados de cada uno de los módulos que se encuentran en un anexo de este trabajo.

INTEGRACION .- El objetivo principal de esta etapa es la integración funcional de los módulos de un programa de computadora, ajustándolos a las particularidades del Sistema de Banco de Horas.

Existen dos tipos de integración y validación de los módulos, uno el de Integración y Pruebas No Incremental y el otro el de Integración y Pruebas Incremental.

El proceso de integración y pruebas no incremental

es donde la validación de computadora se hace a partir de pruebas modulares independientes, mientras en el otro caso, se realiza la validación de nuevos módulos (no probados) agregando los módulos ya probados e integrados.

El proceso de integración incremental requiere un menor esfuerzo, ya que cuando se integra y prueba un módulo de un nivel jerárquico bajo dentro del diagrama de estructura, este no necesariamente requiere llamar a otros. En cambio con la integración no incremental es mayor el esfuerzo, ya que el probar el módulo que llama a otros a ejecución, se tiene que diseñar módulos que simulen las funciones de los subordinados, tarea que resulta más complicada para la generación de este tipo de módulos.

Para la integración, se tomó como base un diagrama jerárquico donde se observan las tareas del sistema y su importancia, pues están divididos en niveles como se ilustra en la fig. 27 de la siguiente página.

Así, en este trabajo se tuvo que la integración del Sistema de Banco de Horas se realizó bajo los dos tipos de integración, durante la primera parte de la integración se realizó en base a la No Incremental, pues se generaron los módulos de OPCIONES, que es el que se encarga de llamar a todas las tareas generales del Sistema.

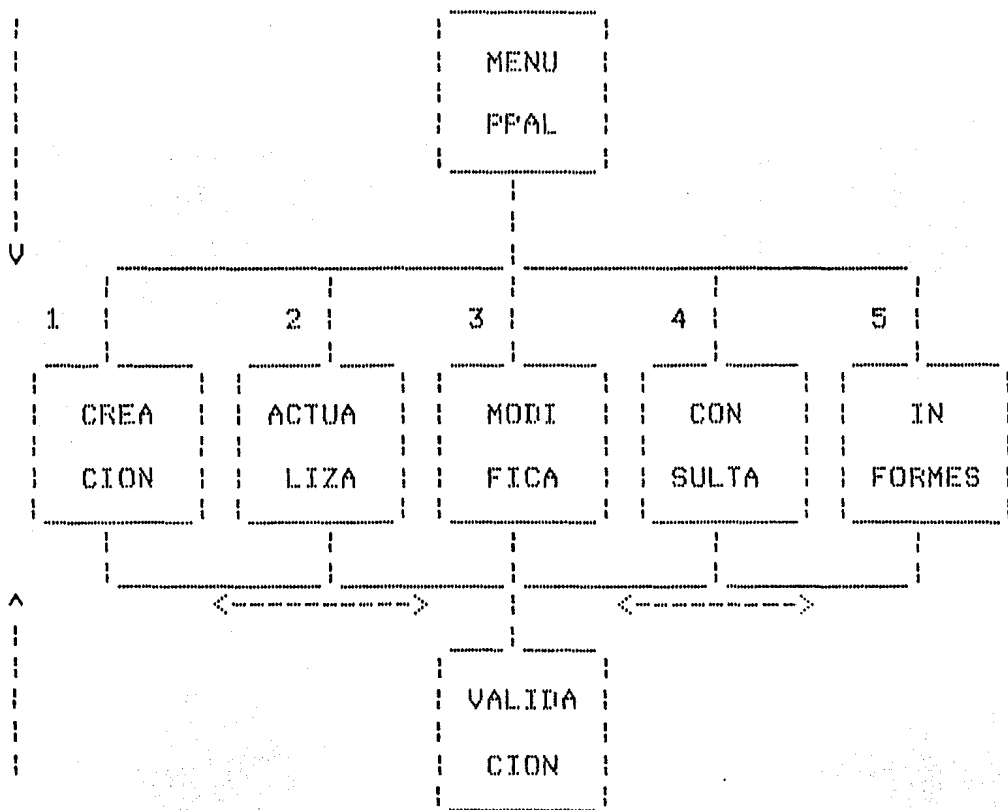


FIGURA 27

En la Integración de los siguientes módulos, se usó tanto la técnica Incremental como la No Incremental, pues por ejemplo en la elaboración del primer módulo dentro del mismo nivel jerárquico, se tuvo que realizar la codificación e integración del módulo de VALIDACION, que es el módulo con mas bajo nivel jerárquico, y que es de los más importantes, pues tiene comunicación con todos los módulos restantes. Esto permitió que se tuviera seguridad sobre la correcta

comunicación de este módulo y los demás.

PRUEBAS DEL SISTEMA .- Esta parte se caracteriza por encontrar los errores cometidos a lo largo de todos los procesos del Desarrollo del Sistema. Esto involucra desde el planteamiento de requerimientos, el Análisis, y Diseño, a fin de poder corregir cualquier posible conflicto entre lo diseñado y lo que se intenta producir.

Se tiene que el concepto de prueba es el proceso de ejecutar un programa con el fin de encontrar errores. Dentro de la etapa de pruebas del software se consideran básicamente 4 pasos:

1) Inspecciones y recorridos de los productos generados.

2) Diseño de casos de prueba.

3) Ejecución de las pruebas.

4) Evaluación de las pruebas.

El objetivo de las inspecciones y recorridos es el de detectar errores de omisión, de lógica o de consistencia.

El diseño de casos de prueba esta dividido es dos técnicas que permitan detectar el mayor número de posibles errores. estas técnicas son: la de caja negra y la de caja blanca.

Pruebas de Caja Blanca .- Las pruebas de caja

blanca o lógicas, permiten diseñar los casos de prueba de acuerdo a la estructura interna del programa. Los criterios para el diseño de casos de prueba se describen como sigue:

Cobertura de Instrucciones .- La idea de este criterio es que todas las instrucciones del programa sean ejecutadas al menos una vez. Este criterio es muy deficiente por lo que se considera inútil.

Cobertura de Decisiones .- Este criterio establece que cada secuencia lógica de instrucciones (lazo) debe ejecutarse una vez si y otra no; esto puede traducirse a que cada decisión debe tener un falso y un verdadero.

Cobertura de Condiciones .- Este criterio establece que cada condición en cada decisión tenga todos los resultados posibles. Esto es, que si una decisión contiene varias condiciones, estas deberán tener todos los resultados posibles.

Cobertura de Condición Múltiple .- Este criterio establece que, todas las combinaciones posibles de resultados de condición en cada decisión y todos los puntos de entrada se invoquen por lo menos una vez.

Técnica de Caja Negra .- Las pruebas producidas por entrada/salida permiten diseñar los casos de prueba en función de las especificaciones del programa sin tener

la necesidad de consultar su lógica interna. Se tiene la ventaja, por lo tanto, de que cuando se diseñan los casos de prueba no es necesario conocer el código, pudiendo además desarrollar el código en forma paralela. Por lo tanto al momento de revisar los requerimientos se pueden detectar errores.

Los criterios para el diseño de casos de prueba con esta técnica se presentan a continuación:

Particiones de equivalencia .- Aquí se realiza una partición del conjunto de casos de pruebas que pueden ser cubiertas con el mínimo número de casos de prueba.

Este criterio está compuesto de dos pasos: el primero se ejecuta haciendo divisiones en las condiciones de entrada (usualmente una especificación) en subconjuntos de válidas e inválidas. El segundo consiste en asignar un número único a cada clase de equivalencia, escribir un caso de prueba que cubra el máximo de equivalencias válidas que no han sido cubiertas, repitiendo el proceso hasta que no existan equivalencias válidas, y escribir un caso de prueba para cada una de las equivalencias inválidas.

Ejecución de los Pruebas .- Una vez que se han diseñado los casos de prueba, se procede a la ejecución para de esta manera detectar los errores. Existen básicamente cinco tipos de prueba que están divididos en los siguientes niveles:

Pruebas Modulares .- En esta parte se procede en primera instancia a probar cada uno de los módulos del sistema. El objetivo de este tipo de prueba es el de detectar errores en la codificación.

Pruebas de Integración .- Las pruebas de Integración se realizan sobre los módulos ya probados individualmente. Estas tienen el objetivo de localizar los errores generados en la codificación y en el diseño.

Pruebas Funcionales .- Estas pruebas se realizan a programas que tienen cierto grado de complejidad (sistema). Este tipo de programas contiene requerimientos y objetivos formales. El objetivo principal es encontrar errores de especificación de requerimientos, y son creados por las discrepancias existentes entre el análisis y la especificación.

Pruebas al Sistema .- Esta es la más difícil, ya que el objetivo es el identificar las discrepancias entre el producto original y las funciones generales del sistema que debe de realizar.

Pruebas de Aceptación .- Estas consisten en comparar el producto final, con el planteamiento de los objetivos.

Una vez que se realizaron las pruebas, se procedió a corregir de manera sistemática y organizada al Sistema. Así, se elaboraron pruebas modulares que se basaron en la

experiencia para encontrar situaciones, que se pudieran presentar en determinado momento, durante la ejecución de cada uno de los módulos, hasta que no ocurrieron errores, para después integrar cada una de las entes que conforman el sistema general.

OPERACION Y MANTENIMIENTO

Esta fase tiene su inicio cuando el sistema se termina y se hace una revisión de los requerimientos, observando que el mismo cumple cabalmente con ellos. Una característica de esta fase es la creación del Manual del Usuario.

Manual del Usuario .- El Manual del Usuario es el instructivo que se realiza para poder implementar, instalar y operar el Sistema Banco de Horas y esta formado como sigue:

Introducción.- El Sistema Banco de Horas tiene el fin de facilitar el control y manejo de la información del personal docente que labora en las diversas dependencias universitarias.

El propósito de este Manual es proporcionar los

elementos básicos para la Implementación y Operación del Sistema Banco de Horas en cada una de las microcomputadoras que utilicen los siguientes sistemas operativos: Cromix y MS-DOS (esto es en primera instancia).

Para poder tener alguna de las versiones se deberá consultar la parte de implementación presentada a continuación.

Implementación.- Para que el Sistema Banco de Horas pueda ser implementado en cualquier dependencia universitaria, se requiere que se elaboren una serie de catálogos auxiliares en los cuales se basa el sistema para su implementación.

Estos catálogos corresponden a los que la dependencia va a utilizar, los cuales están relacionados con los niveles de adscripción, carreras y asignaturas que se imparten.

Catálogo de Adscripción.- Aquí se determina la manera en que se controlará internamente el banco de datos, estableciendo tres niveles de adscripción de acuerdo a su estructura organizacional y sus necesidades.

Por ejemplo:

- 1) Nivel 1 : División
Nivel 2 : Programa
Nivel 3 : Unidad Responsable
- 2) Nivel 1 : Plantel
Nivel 2 : Turno

Nivel 3 : Colegio Académico

- 3) Nivel 1 : División
Nivel 2 : Secretaría
Nivel 3 : Departamento

Elaborar una lista de los nombres correspondientes a cada nivel, asociándoles un número con las siguientes posiciones:

Nivel 1 1 Posición
Nivel 2 2 Posiciones
Nivel 3 3 Posiciones

Considerando los niveles del primer ejemplo:

Nivel 1 División
 1 Estudios Profesionales
 2 Posgrado

Nivel 2 Programa
 11 Docencia
 22 Investigación

Nivel 3 Unidad Responsable
 216 Depto. de Matematicas
 240 Depto. de Sociologia
 321 Depto. de Ciencias Politicas
 715 Depto. de Ciencias Basicas

Catálogo de Carreras.- Elaborar una lista de todas las carreras que existan en la dependencia, asociándoles un número de dos dígitos o bien en el otro caso de que así se quiera asignar dos letras o una combinación de ambas.

Ejemplo:

11 LICENCIADO EN INFORMATICA
15 LICENCIADO EN ADMINISTRACION
23 LICENCIADO EN DERECHO

Catálogo de Asignaturas.- Elaborar una lista de todas las asignaturas que se imparten en la dependencia,

asociándoles un número de 4 posiciones a cada una de ellas y anteponiéndole el número de la carrera a la cual corresponden.

Ejemplo:

```
11 0046 MATEMATICAS APLICADAS
11 0072 SOCIOLOGIA DE MEXICO
15 0032 ADMON. DE PERSONAL PUBLICO
15 0034 ADMON. DE PROYECTOS
23 0020 DERECHO CONSTITUCIONAL
23 0021 DERECHO ADMINISTRATIVO
```

Instalación del Sistema.- Para la instalación del sistema en cualquiera de los equipos en los que se va a implementar, se deben traspasar todos los archivos que conforman el sistema por medio de las utilerías que forman parte del equipo a utilizar.

Por ejemplo si se desea instalar el sistema en un equipo Sistema Tres de Cormemco, con Disco Duro, se tiene primero que crear un directorio llamado BH; luego poner el disco del sistema en una unidad de discos flexibles auxiliar, Ahora estando en el directorio desde BH se colocarán los módulos del Sistema Banco de Horas por medio del comando CDOSCOPY -V SFDA *.* (suponiendo que la unidad auxiliar sea A:).

Ahora bien, si la máquina es una microcomputadora de las llamadas PC con configuración de Disco Duro se seguirán los pasos similares a los del ejemplo anterior, solo que para el paso de archivos se teleará COPY A:*. * C: (suponiendo que el sistema se encuentra en la unidad

A:)), y el Sistema será copiado al directorio de Banco de Horas.

Una vez hecho lo anterior, el Sistema estará listo para poder ser ejecutado. En secciones posteriores se explicará cuales son los comandos requeridos para poder trabajar correctamente con el Sistema Banco de Horas.

Codificación de la Información.- En esta parte se explican los criterios de codificación tanto para los datos personales como de los movimientos que tengan cada uno de los RFC's. Para ello se muestran las formas que se deberán llenar para poder darle información al sistema.

Para la forma de captura del sistema de banco de horas, (ver anexo "1") se tiene lo siguiente:

Registro Número 1

COLUMNA	AMPLITUD DE CAMPO	TIPO DE CAMPO	DESCRIPCION
1 a 10	10	Alfanumérico	Anotar el Registro Federal de Causantes del empleado.
11 a 12	2	Numérico	Marcar el número de Registro, en este caso siempre será 01.
13 a 16	4	Numérico	Registrar el número de folio contenido en las solicitudes de personal.
17 a 51	35	Alfanumérico	Apuntar el nombre completo de empleado empezando por el apellido paterno.
52 a 58	7	Numérico	Facilitar el número

telefónico del mismo.

Registro Número 2

COLUMNA	AMPLITUD DE CAMPO	TIPO DE CAMPO	DESCRIPCION
1 a 10	10	Alfanumérico	Anotar el Registro Federal de Causantes del empleado.
11 a 12	2	Numérico	Marcar el número de Registro, en este caso siempre será 02.
13 a 52	40	Alfanumérico	Facilitar el domicilio del empleado anotando el nombre de la calle y el número.
53 a 72	20	Alfanumérica	Señalar la colonia.
73 a 77	5	Numérico	Proporcionar el código postal.
78 a 83	6	Numérico	Registrar la fecha de ingreso a la dependencia siguiendo el formato (ddmmaa).
84 a 85	2	Numérico	Indicar para el empleado la antigüedad en la U.N.A.M. en años cumplidos.
86	1	Numérico	Anotar el tipo de personal (*).
87	1	Numérico	Señalar la nacionalidad del empleado (*).
88	1	Alfabético	Marcar el sexo del empleado (*).
89	1	Alfabético	Apuntar el estado civil del empleado (*).
90 a 95	6	Numérico	Proporcionar fecha de actualización del

empleado siguiendo el formato (ddmmaa).

Registro Número 3

COLUMNA	AMPLITUD DE CAMPO	TIPO DE CAMPO	DESCRIPCION
1 a 10	10	Alfanumérico	Anotar el Registro Federal de Causantes del empleado.
11 a 12	2	Numérico	Marcar el número de Registro, en este caso siempre será mayor de 2.
13 a 18	6	Numérico	Anotar la clave de adscripción para lo cual debe utilizar una posición para el nivel 1, dos posiciones para el nivel 2 y tres para el nivel 3. (**).
19 a 20	2	Numérico	Apuntar la clave que corresponde a la carrera (**).
21	1	Numérico	Registrar el tipo de movimiento (*).
22	1	Alfabetico	Señalar la clave de la causa del movimiento (*).
23	1	Numérico	Indicar la categoría (*) solamente incluye profesores de asignatura y ayudantes de profesor.
24	1	Numérico	Apuntar el tipo de plaza (*).
25 a 30	6	Numérico	Facilitar la fecha de inicio de contrato del profesor siguiendo el formato (ddmmaa).

31 a 36	6	Numérico	Proporcionar la fecha de terminación del nombramiento del profesor siguiendo el formato (ddmmaa).
37 a 40	4	Numérico	Anotar la clave de la asignatura excluyendo el número de la carrera (**).
41 a 44	4	Numérico	Señalar la clave que corresponde al grupo.
45 a 48	4	Numérico	Dar el número de salón donde se impartirá la materia.
49 a 54	6	Numérico	Registrar la clave del día que imparte clases (*). Ej: Si imparte clase los lunes, miércoles y viernes se marcaría así !1 3 5 !.
55 a 58	4	Numérico	Indicar la hora de entrada. Ej: Si entra a las 8:30 se codificará 0830.
59 a 62	4	Numérico	Registrar la hora de salida, igual que la de entrada.
63 a 66	4	Numérico	Marcar el número de horas teóricas, utilizando dos campos para las horas y dos para minutos. Ej: si son 6 hrs con 30 mins se codificará: 0630.
67 a 70	4	Numérico	Facilitar el número de horas prácticas igual que con el campo anterior.
71 a 74	4	Numérico	Indicar el número de horas de apoyo, utilizando el formato del campo

anterior.

75 a 88

14

Numérico

Marcar el código programático en base al que otorga la Dirección General de Presupuesto por Programas, utilizando 14 dígitos distribuidos de la siguiente forma:

programa	2	posiciones
subprograma	2	posiciones
dependencia	3	posiciones
subdependencia	2	posiciones
partido	3	posiciones
dígito de control	2	posiciones.

(*) Consultar tabla de claves.

(**) Estas claves se estructuran al elaborar los catálogos auxiliares.

NOTA: Si el profesor imparte más de una asignatura igual o diferente a la registrada en el registro 3, se seguirán las mismas instrucciones de esta y se utilizará el siguiente renglón.

Operación del Sistema .- En esta sección se describen de manera muy general los módulos de los cuales está constituido el sistema.

Una vez que se está en el directorio BH, se puede pedir al sistema operativo que liste los archivos contenidos en el directorio. Hecho esto se deberá observar lo siguiente:

SISTBH	COM
OPCIONES	COM
ALTASINT	COM

ALTASBAT	COM
ACTUALZA	COM
CONSULTA	COM
MODIFICA	COM
DIRPERS	COM
MOVSPERS	COM
RUNCOR	COM
CARRERAS	SBH
MATERIAS	SBH
NIVEL1	SBH
NIVEL2	SBH
NIVEL3	SBH

Para una microcomputadora PC los archivos tendrán la misma estructura, solo que la extensión será para los objetos de EXE.

Una vez que se corra el sistema, este creará tres archivos los cuales serán:

PERSONAL	DAT
PERSONAL	KEY
MOVIMTOS	DAT

El sistema esta constituido por cinco módulos y una opción para finalizar, los que integran su menú principal.

Para ejecutar el Sistema de Banco de Horas se deberá teclear el siguiente comando:

SISTEM

mismo que iniciará la presentación del sistema y mostrará despues el Menú Principal, el cual tendrá la siguiente forma:

SISTEMA BANCO DE HORAS

- 1.- A L T A S
- 2.- ACTUALIZACION
- 3.- MODIFICACIONES
- 4.- CONSULTA
- 5.- INFORMES
- 6.- F I N

TECLEE OPCION DESEADA >> _

El sistema a su vez solicita que se le proporcione el número del módulo que se quiere realizar. Para lo cual deberá teclear el número del módulo que seleccione.

Ejemplo: Si selecciona el módulo de CONSULTA, teclee: "4". En caso de que ya no requiera ejecutar ningún módulo teclee: "5" opción que corresponde al fin del sistema.

A continuación se explica la finalidad de cada uno de los módulos, así como los datos que el usuario debe proporcionar.

MODULO DE ALTAS .- La función de este módulo es la de permitir la creación del banco de datos del sistema. Al teclear la opción para este módulo, se genera un submenú, mismo que pregunta la forma en que se va a crear el banco de datos. Su formato es como sigue:

A L T A S

- 1.- INTERACTIVO
- 2.- POR LOTE
- 3.- REGRESA A MENU PRINCIPAL

TECLEE MODO PARA DAR DE ALTA : _

Si se tecllea la opción "1", el proceso que se ejecuta es en el modo interactivo. Esto indicará al sistema que los datos que van a entrar van a ser vía terminal. Los datos requeridos serán de la siguiente manera:

A L T A S

RFC: _____ FOLIO: _____
NOMBRE: _____ TELEFONO: _____
DOMICILIO: _____
COLONIA: _____ CODIGO POSTAL: _____
FECHA INGRESO: _____ ANTIG UNAM: _____ TIPO PERSONAL: _____
NACIONALIDAD: _____ SEXO: _____ ESTADO CIVIL: _____
FECHA ACTUALIZACION: _____

'FIN'
para terminar.

Al capturar la información, el sistema irá validando automáticamente cada uno de los conceptos, desplegando el tipo de error en el momento en el que este aparece. Cuando existe un error, el sistema automáticamente lo regresa al inicio del campo para así teclear una vez más el dato; este proceso se repite hasta que el dato es correcto.

Una vez que se termina de registrar la forma de datos personales, automáticamente se desplegará una segunda forma, misma que sirve para asignar las materias que el profesor impartirá. La captura de la información ocurre de igual manera que en la forma de datos

personales del empleado. Este proceso se ejecuta indefinidamente para cada una de las materias que se impartirán.

A continuación se muestra el formato de la segunda forma:

MOVIMIENTOS

RFC:xxxxxx

ADSCRIPCION:_____ CAUSA:_____ CATEGORIA:_____ CARRERA :_____
TIPO MOVIMTO:_____ PLAZA:_____
FECHA INICIO:_____ FECHA TERMINO:_____
ASIGNATURA:_____ GRUPO:_____ SALON:_____ DIAS:_____
HORA ENTRADA:_____ HORA SALIDA:_____
HORAS TEORICAS:_____ HORAS PRACTICAS:_____ HORAS APOYO:_____
CODIGO PROGRAMATICO:_____

'FIN'
para terminar.

Cuando ya no se desee capturar más movimientos para el profesor, bastará con poner FIN en el campo de lugar de adscripción, regresándole a la forma de Altas, y así poder dar de alta a otro empleado con sus respectivos movimientos.

Este proceso se repetirá indefinidamente, pero cuando ya no se quieran dar más altas, bastará con teclear FIN en lugar del R.F.C., desplegando a continuación el Menú Principal.

Al escoger la opción "2", el sistema ejecutará el módulo de Altas por Lote, mismo que necesitará que exista en disco el archivo BHDATOS.DAT, conteniendo los datos

capturados según se especificó en la etapa de Codificación.

Una vez que se encontró el archivo de datos capturados, el sistema mostrará en pantalla:

"PROCESANDO BHDATOS.DAT"

el cual indicará que se están creando los archivos de Personal y Movimtos. Este módulo validará la información y en caso de que exista algún error, el registro con la anomalía será escrito en el archivo BHMALAS.DAT, en caso de que este correcto, aparecerá en el archivo BHBUENAS.DAT.

Este módulo producirá un Reporte con el diagnóstico general de todos los datos que se leyeron del archivo primario. Una vez que terminó de procesar datos, el sistema regresará automáticamente al Menú Principal.

Módulo de Actualización .- Este módulo permite asignar nuevas materias al empleado de la dependencia que se le indique, para ello deberá estar dado de alta el profesor; en caso contrario marcará un error. La forma que verifica es la siguiente:

TECLEE RFC A ACTUALIZAR >> _____

Una vez que se verifica la existencia del profesor en el banco de datos, aparece la forma 2 (equivalente a la del módulo de Altos), que funciona de la misma forma

hasta que se teclea 'FIN', finalizando el proceso de este módulo desplegando el Menú Principal.

Módulo de Modificaciones .- En este proceso se realiza tanto la corrección como la cancelación de datos de profesores, así como de movimientos que se tengan que realizar y que hubiesen ocurrido durante la captura de la información.

El formato que presenta la pantalla al correr este módulo es el siguiente:

MODIFICACIONES

- 1.- EN DATOS PERSONALES
- 2.- EN MOVIMIENTOS
- 3.- F I N

TECLEE OPCION >> _

Una vez que se escoge una opción, aparecerá un nuevo letrero pidiendo el modo con que se va a modificar la información:

```
=====
CORRECCION          CANCELACION
  CR                  CN
                   OPCION?_
=====
```

Ya tomado este dato, el sistema pedirá el RFC del empleado al que se le debe realizar el proceso. Una vez encontrado el profesor, desplegará los datos personales en caso de que la opción del primer menú haya sido de "1" o de datos personales en caso que dicha opción haya sido

"2".

En caso de escoger la opción de la corrección en el segundo menú, cada uno de los campos que forman parte de las formas, aparecerán para ser recapturados, si se tecldea "return" inmediatamente al pedir los datos, este campo quedará sin modificación, pero si se tecldea información, esta se validará y en caso de que este sea correcto, sustituirá a la que había antes.

Si se tomó la opción de cancelación, en el momento que se indentifique el registro a cancelar, este se dará de baja del banco de datos.

Una vez que se termina, el sistema nos regresa al primer menú, si se desea regresar a Menú Principal, se teclleará la opción 3.

Módulo de Consulta .- La función de este módulo es permitir la consulta de todas las afectaciones que se le han hecho al Banco de Horas. Después de tecllear la opción "4" del Menú Principal, aparecerá en pantalla el siguiente menú:

C O N S U L T A

- 1.- POR DATOS PERSONALES
- 2.- POR MOVIMIENTOS
- 3.- F I N

TECLEE OPCION >> _

El sistema solicita que se le proporcione la consulta que se quiere realizar; para lo cual deberá teclear el número de consulta que seleccione.

Ejemplo:

Si selecciona por: DATOS PERSONALES inmediatamente el sistema pedirá el RFC que se quiere consultar:

TECLEE EL RFC A CONSULTAR >> _

una vez teclada la información se desglosarán los datos personales del profesor.

Este módulo funciona de manera similar cuando se teclee como opción que se quiere consultar MOVIMIENTOS.

Módulo de Informes .- La función de este módulo es emitir los diferentes reportes que se requieren. Una vez que se haya teclado la opción número "5" que es el que corresponde a este módulo, aparecerá en la pantalla la siguiente lista:

I N F O R M E S

- 1.- DIRECTORIO DEL PERSONAL
- 2.- DETALLADO DE MOVIMIENTOS
- 3.- REGRESA A MENU PRINCIPAL

TECLEE INFORME A OBTENER: _

El sistema solicita que se le proporcione el reporte que se quiere realizar; para lo cual deberá teclear el número del reporte que seleccione. Inmediatamente el sistema preguntará la fecha correspondiente al día en que se esta realizando el proceso o la fecha del último

movimiento que se registró. Acto seguido se pedirá el tipo de informe, pudiendo ser GLOBAL o por NIVELES. Para el tipo de informe por Niveles, este se obtendrá con los datos sorteados por los niveles de Adscripción que tenga el Banco de Datos, en cambio en forma Global el informe que se obtenga saldrá reflejado en forma general, despreciando el nivel de adscripción.

Una vez que se ha terminado el Informe, el sistema automáticamente le regresa al Menú principal, para así poder procesar alguna otra tarea, o bien otro reporte.

Si ya no se quiere seguir trabajando con el Sistema Banco de Horas, se tecleará estando en el Menú Principal la opción "6" opción que corresponde al fin del sistema.

TABLA DE CLAVES (*)

El número de la columna, solo se tomará en cuenta cuando el proceso que se vaya a realizar, sea por medio de archivos. ("creación por lote").

A. TIPO DE PERSONAL

COLUMNA	CLAVE	CONCEPTO
86	1	ADMINISTRATIVO
	2	INVESTIGADOR
	CLAVE	DOCENTE

B. NACIONALIDAD

COLUMNA	CLAVE	CONCEPTO
87	0	MEXICANO
	1	EXTRANJERO

C. SEXO

COLUMNA	CLAVE	CONCEPTO
88	F	FEMENINO
	M	MASCULINO

II. ESTADO CIVIL

COLUMNA	CLAVE	CONCEPTO
89	S	SOLTERO
	C	CASADO
	V	VIUDO
	D	DIVORCIADO
	X	OTRO

F. TIPO DE MOVIMIENTO

COLUMNA	CLAVE	CONCEPTO
21	1	ALTA
	2	BAJA
	3	LICENCIA CON GOCE DE SUELDO
	4	LICENCIA SIN GOCE DE SUELDO

G. CAUSA

COLUMNA	CLAVE	CONCEPTO
22	A	NUEVO INGRESO
	B	REINGRESO
	C	PRORROGA
	D	OTRO NOMBRAMIENTO
	E	RENUMERACION ADICIONAL
	F	AUMENTO DE HORAS
	G	PROMOCION
	H	REANUDACION DE LABORES
	I	INTERINATO
	J	RECLASIFICACION
	K	TRANSFERENCIA
	L	DISMINUCION DE HORAS

M	ENFERMEDAD
N	GRAVIDEZ
O	AÑO SABÁTICO
P	COMISION OFICIAL
Q	REVOCAACION
R	SUSPENSION
S	RENUNCIA
T	DEFUNCION
U	ESTUDIOS EN EL PAIS
V	ESTUDIOS EN EL EXTRANJERO
W	PARTICULAR
X	RESCISION
Y	PRORROGA POR CAMB.DE MATERIA

H. CATEGORIA

COLUMNA	CLAVE	CONCEPTO
23	01	PROFESOR ASIGNATURA A
	02	PROFESOR ASIGNATURA B
	03	AYUDANTE DE PROFESOR A
	04	AYUDANTE DE PROFESOR B

I. PLAZA

COLUMNA	CLAVE	CONCEPTO
24	1	INTERINA
	2	DEFINITIVA

J. DIAS DE LA SEMANA

COLUMNA	CLAVE	CONCEPTO
49 a 65	1	LUNES
	2	MARTES
	3	MIERCOLES
	4	JUEVES
	5	VIERNES
	6	SABADO

Mantenimiento .- Aquí el término de mantenimiento, se utiliza para describir todos los cambios necesarios al programa para que este pueda ser implementado en escuelas y facultades de la universidad.

Este Sistema tiene la característica de ser altamente modificable, tratando de guardar su fisonomía definida en la fase de especificación de los requerimientos, de tal forma, que con solo algunos pocos cambios, el sistema se pueda adaptar a las diferentes dependencias que requieran su utilización.

Así, la primera modificación que debe observarse en la construcción del nuevo sistema, es del módulo de validación, dado que este es único para las necesidades de cada usuario. El módulo es localizado en disco como:

"VALIDA.COB".

Luego, habrá que modificar los archivos auxiliares que contienen los letreros de los distintos niveles de adscripción, localizados como:

"NIVEL1.SBH", "NIVEL2.SBH" y "NIVEL3.SBH"; así como el archivo de materias llamado: "MATERIAS.SBH" y el catálogo de carreras que se encuentra bajo "CARRERAS.SBH". Estos archivos se caracterizan por tener los nombres de los

catálogos auxiliares elaborados por el usuario al momento de solicitar el sistema para su implementación.

Ya que se tienen los archivos auxiliares que utiliza el sistema para la elaboración del reportes, se procede a trabajar en el "diskette" o directorio, dependiendo del equipo donde se localize el sistema, únicamente con el fin de darle mantenimiento al sistema.

Aquí estarán los programas fuentes, y una serie de archivos, que serán compilados para la generación de la versión del Sistema Banco de Horas que se solicitó.

Existe también aquí un pequeño archivo que contiene el nombre de la dependencia, que es parte de los programas del sistema concernientes a los reportes. Este archivo se localiza por el nombre de: "DEPENDENCIA.SBH" y es llamado al momento de compilarse cada uno de los módulos de los reportes. Así el archivo esta formado por:

```
01 ENCA-DEF,  
03 FILLER FIC X(54)  
   VALUE SPACES.  
03 FILLER FIC X(24)  
   VALUE "NOMBRE DE LA DEPENDENCIA".  
03 FILLER FIC X(54)  
   VALUE SPACES.
```

El llamado durante la compilación, por medio de la instrucción COPY DEPENDENCIA.SBH, en el lugar que corresponde dentro del programa fuente.

Todos estos archivos, incluyendo el módulo de validación podrán ser modificados por medio de un editor

de textos.

Una vez que estan hechos todas las modificaciones en los archivos que utiliza el sistema, se deberá correr un proceso que compila todos los módulos del Sistema, este proceso tiene la característica que va llamando los archivos que van siendo requeridos, durante la compilación de cada una de las partes integrantes del sistema de banco de horas.

También se cuenta con programas de utilería que se elaboraron durante el desarrollo del Sistema, estos permiten listar archivos de datos y en su caso corregirlos, a fin de observar cada registro, para en caso de algún problema no determinado, se pueda verificar la correcta estructura de los archivos.

Para la corrección o recuperación de los archivos, se pueden también utilizar estos archivos, solo bastará con hacerles pequeñas modificaciones para que puedan ser aplicados, al problema que se presenta en el momento de su invocación.

EVALUACION DEL SISTEMA

Una vez que el sistema se pone en marcha, es necesario su evaluación, para verificar si se alcanzaron o no, los objetivos planteados originalmente en la primera parte de la fase de Análisis.

Evaluando cada uno de los objetivos tenemos:

a) Este sistema consigue llevar un control de profesores en cualquier escuela que pertenezca a la universidad, ya que permite el manejo tanto de datos personales como de los movimientos que estan formados por asignaturas, grupos, horarios, días, categorías y códigos programáticos. Todo esto para cada una de las personas que labore en la escuela.

b) El sistema permite la consulta directa al banco de datos via terminal, así como la obtención de informes que

reflejan la situación de cada uno de los profesores, además de informar sobre sus datos personales.

c) Cumple con la finalidad de poder ser implementado en diferentes escuelas y facultades, ya que los cambios necesarios para su adaptación, se realizan únicamente en el módulo de validación y en los catálogos auxiliares definidos en el manual del usuario.

d) Respecto al último de los objetivos, se cumple con la posibilidad de poderse transportar fácilmente a otro equipo, y esto se debe básicamente a que el Sistema Banco de Horas está programado en Cobol, que es un lenguaje Estandard y que el sistema está desarrollado modularmente, permitiendo ser adaptado a equipos con problemas de capacidad en memoria principal.

Como desarrollo de la tesis, se puede concluir que fue que los resultados son satisfactorios, pues se alcanzaron los objetivos propuestos para este trabajo.

Actualmente, el Sistema Banco de Horas se encuentra en proceso de implementación hacia algunas dependencias, como la Escuela Nacional Preparatoria, que llevará el sistema a cada uno de sus respectivos planteles. Además está la Facultad de Ciencias que requerirá la creación de nuevos informes, y que gracias a la estructura del sistema, estos reportes podrán ser obtenidos modificando únicamente 2 módulos dentro de uno de los programas que obtienen los reportes, ya que es muy grande su facilidad

de adaptación.>

Finalmente como conclusión se puede considerar al sistema como sigue:

- a) UTIL pues satisface ampliamente las expectativas de muchos usuarios con necesidades semejantes.
- b) BAJO COSTO pues se ajusta a un presupuesto ya establecido. (en este caso el salario percibido durante la elaboración de esta tesis).
- c) SEGURO la protección del sistema contra daños de terceras personas, dependerá del control que se tenga al acceso del equipo por parte del usuario en donde se implemente el sistema.
- d) MODULAR o sea que el sistema se puede modificar fácilmente de acuerdo a nuevos requerimientos.
- e) PORTABLE es decir, el sistema es compatible en varios equipos de cómputo, y por lo tanto es fácilmente intercambiable.

El Sistema se encontrará a disposición de cualquier dependencia, escuela o facultad para su respectivo utilización. Para ello, bastará con dirigirse a la Dirección General de Servicios de Cómputo para la Administración o fin de seguir los pasos necesarios para poderse implementar.

DESCRIPCION DEL EQUIPO UTILIZADO

El sistema utilizado para la elaboración de este programa fué un equipo SISTEMA TRES DE CROMEMCO el cual tiene la particularidad de trabajar bajo CROMIX que es un sistema operativo con muchas características, de Unix V y que fue diseñado específicamente para equipos Cromemco.

Aparte este equipo tiene la posibilidad de poder cargar el sistema operativo CDOS que es un programa utilizado por microcomputadoras Cromemco que manejan hasta 64K de memoria.

Cuando el Sistema tiene a CROMIX como sistema operativo residente, este ocupa 64K de memoria. Cada proceso normalmente requiere 64K Bytes de memoria adicional de los cuales, solo 1K es ocupado por el sistema y lo demás queda disponible al usuario para poder

realizar cada una de sus aplicaciones.

Además, tiene muchas capacidades, las cuales se muestran a continuación:

1. Soporta tareas y usuarios múltiples sobre disco duro y el almacenamiento en discos flexibles de sistemas.
2. Múltiples directorios y subdirectorios jerárquicos.
3. Dispositivos de E/S compatibles, los cuales soportan redireccionamientos de entrada y salida.
4. Versatilidad del Programa Shell para flexibilidad y reconfiguración de la interface del usuario.
5. Sistema de seguridad por 'password', limitando el acceso del sistema y a los archivos que estén protegidos con read, write, append, y atributos de ejecución.
6. Soporte para la fecha y la hora.
7. Numerosos buffers de archivo para una ejecución muy veloz.

El Sistema Operativo Cromix de Cromemco, esta diseñado para operar con dos de siete Tarjetas Cromemco 64KZ Random Access Memory Boards. Una tarjeta 64KZ reside en el banco 0 de memoria y es reservada para el sistema operativo. Esta tarjeta también reside en todos los bancos no usados.

Un sistema mínimo requiere dos tarjetas 64KZ Rom.

Una de ellas es reservada para el sistema operativo y la otra se dedica para ejecutar programas de usuario.

Aunque el sistema con dos tarjetas de memoria puede soportar diferentes usuarios, no puede procesar dos procesos diferentes a la vez, pero esto se logra mediante la implantación de tarjetas 64KZ adicionales.

Una característica que tiene CROMIX es que programas que corren en CDOS en otras computadoras Cromemco puedan ser también ejecutados bajo CROMIX, y esto se logra por medio de un pequeño programa llamado "sim.bin", el cual es buscado por CROMIX en cuanto detecta que un programa que se quiere ejecutar tiene el formato CDOS (compilado y ligado) se pueda ejecutar. Este simulador ocupa aproximadamente 3K de memoria, dejando al usuario el restante de los 64K que se le asigna cuando entra en sesión.

HARDWARE

El SISTEMA TRES con el que se trabajó esto compuesto por unidades de disco que permiten guardar información sobre la superficie de discos magnéticos. Las unidades de disco que tiene el sistema son: 2 unidades para disco flexible de 8 pulgadas, 1 unidad de disco duro con capacidad de 20 Megabytes y dos unidades de disco

flexible de 5 1/4 de pulgada con capacidad de 384 K.

Tiene un procesador de 8 bits Z80A más otro procesador Z80 que trabaja como esclavo del primero y maneja cada uno de los procesos que encuentran en ese momento procesándose y que proviene de cada una de las terminales que están conectadas.

Otro modelo del SISTEM THREE es el que trae una unidad DPU, el cual tiene un procesador de 32 bits (Motorola) junto con un Z80A. Este modelo acepta como sistemas operativos a CROMIX o UNIX V dependiendo de la tarjeta que se monte. Pero bajo Unix no es posible que se puedan ejecutar sistemas en formato Cromix, dando con ello problemas de compatibilidad.

El diagrama general de bloques del SYSTEM THREE utilizado se muestra en la figura 28 al final del apéndice.

A continuación se describirán en forma general cada una de las tarjetas que componen la arquitectura de este sistema.

ZPU. Unidad central de procesamiento con Z-80A.- Se utiliza un microprocesador Z-80A, que es una versión del Z-80, pero que es capaz de utilizar un reloj de 4MHz.

Esta tiene un circuito de salto automático instalado en la tarjeta del ZPU, permitiendo a esto ser usada en un sistema de Bus S-100 sin controles en los

paneles externos de la microcomputadora. Cuando el Sistema es encendido, el hardware del ZPU fuerza un salto automatico a una de las 16 posibles localidades de memoria disponible. En la dirección seleccionada, se encuentra el sistema operativo residente en ROM (RDOS).

1 6 F D C .- Este es un controlador de discos hecho por Cromemco. Esta tarjeta esta provista de un sistema completo para la operación de los floppies, que pueden ser indistintamente de 8 pulgadas y 5 pulgadas, incluye un puerto serie E/S para una terminal de video con interfaz RS-232, una memoria preprogramada ROM, en donde se encuentra guardado un programa monitor llamado RDOS (Resident Disk Operating System), por medio del cual se inicia el sistema (boot); así mismo, posee comandos por medio de los cuales se realiza un diagnostico básico de la memoria y drives. El 16FDC normalmente maneja hasta 4 drives.

6 4 K Z .- La tarjeta Cromemco 64KZ es una memoria dinámica para Lectura/Escritura de 65,536 bytes, compatibles con el bus S-100. Esta tarjeta incorpora una memoria dinámica RAM (Random Access Memory). Opera en un tiempo de acceso máximo de 250 nseg que le permite operar en los sistemas Z80 con un reloj de 4MHz, sin utilizar estados de espera en lo absoluto. esta tarjeta maneja la señal de selección de banco, es posible el aumentar lógicamente la memoria, pues se puede

intercambiar bancos de memoria por software, esta es una razón muy importante, pues se utiliza grandemente en sistemas operativos para multiusuarios.

Controlador de Disco S T D C .- Este es un controlador de alto desempeño y flexibilidad para discos duros (Winchester). Sobre la tarjeta, el microprocesador Z-80A realiza las operaciones del controlador, incluyendo la selección de las unidades de disco. Como es un controlador de disco inteligente, el STDC requiere de una mínima atención por parte del sistema operativo residente. Así, el CPU residente, se olvida de los detalles del acceso del disco, puede continuar ejecutando instrucciones mientras la operación de E/S se esta reolizando.

El STDC habilita la ejecución de sistemas sofisticados a travez del uso de 5 buffers de datos. Cuando un sector es requerido por el sistema operativo, ese sector y todos los otros sectores son leidos del track y puestos en los 'track buffers', el sector deseado es transferido a memoria principal, mientras los sectores restantes son retenidos dentro del STDC hasta que son llamados posteriormente por el sistema operativo. Todas estas características hacen que el acceso a disco se realice a muy alta velocidad.

INTERFACE DE LA IMPRESORA PRI .- La tarjeta de interface de la impresora en paralelo (parallel printer

interface) permite la operación simultánea de impresoras cromemco 3779 o 3703 con la 3355a. La tarjeta PRI, se diseñó para la compatibilidad con el bus S-100.

SOFTWARE

Cuando se inicia una sesión en CROMIX, el sistema le pregunta al usuario su nombre y password, supongamos que el usuario se llama fred, entonces:

```
Login: fred RETURN
```

```
Password: xxxxxx RETURN
```

```
Después el sistema desplegará:
```

```
Logged in fred Jun-24-1986 17:12:15 on tty1
```

```
%
```

Y es este momento se esta en posibilidades de poder utilizar las utilerías y hacer programas bajo CROMIX para lo cual, el sistema nos lo indicará mediante un "%".

La estructura de los archivos esta en forma de ARBOL, y esta es una estructura que el sistema operativo utiliza para guardar archivos. La raiz es el directorio root que siempre esta presente. Para llegar a un archivo, se tendria de recorrer una ruta dada por los directorios. Así, si queremos llegar al archivo DATOS que esta en un tercer nivel o más bien, bajo tres directorios tendríamos que la ruta estaria dada por:

DIR1\DIR2\DIR3\DATOS,

Ya dentro del directorio en donde se va a trabajar, se tienen una serie de comandos y utilerías que nos permiten realizar una serie de tareas sobre la información que se este procesando.

A continuación se presentan ejemplos de algunos comandos que se utilizan bajo Cromix, dejando para el final una descripción general de algunos de los comandos más importantes que se utilizan por lo regular.

El comando d (Directory) nos permite dar el nombre del directorio bajo el cual nos encontramos o bien para cambiarnos a otro directorio. ejemplo:

```
% d          % d\etc
\            % d
%            /etc
            %
```

En la utileria L despliega en orden alfabético los archivos presentes así como todos los subdirectorios que estan en el directorio. por ejemplo:

```
% l
      52   D   1   bin
      13   D   1   cmd
  36,864   1   cromix.sys
      32   D   1   dev
       7   d   1   etc
      32   D   1   letters
       6   D   1   usr
```

El primer paso para crear un directorio se usa el

comando Makdir y tiene la forma de:

```
% makdir /letters
```

Para utilizar el editor de textos de cromenco y trabajar con un archivo que se llama 'archivo' y que se encuentra bajo el directorio letters, bastará con teclear:

```
% screen /letters/archivo
```

El comando Type puede ser utilizado para desplegar los contenidos de un archivo. Este comando puede ser abreviado por ty como se muestra a continuación:

```
% ty /letters/archivo
```

El comando Rename le cambia el nombre a un archivo, con la abreviatura ren, y funciona dejando un espacio entre el el nombre viejo y el nuevo nombre con el que quedará el archivo:

```
% ren antiguo nuevo  
o bien % ren /letters/otro/viejo /letters/otro/nuevo
```

La utilería Del permite borrar un archivo del directorio. Su abreviación es del. También se puede borrar un archivo que se encuentre en otro directorio, por ejemplo:

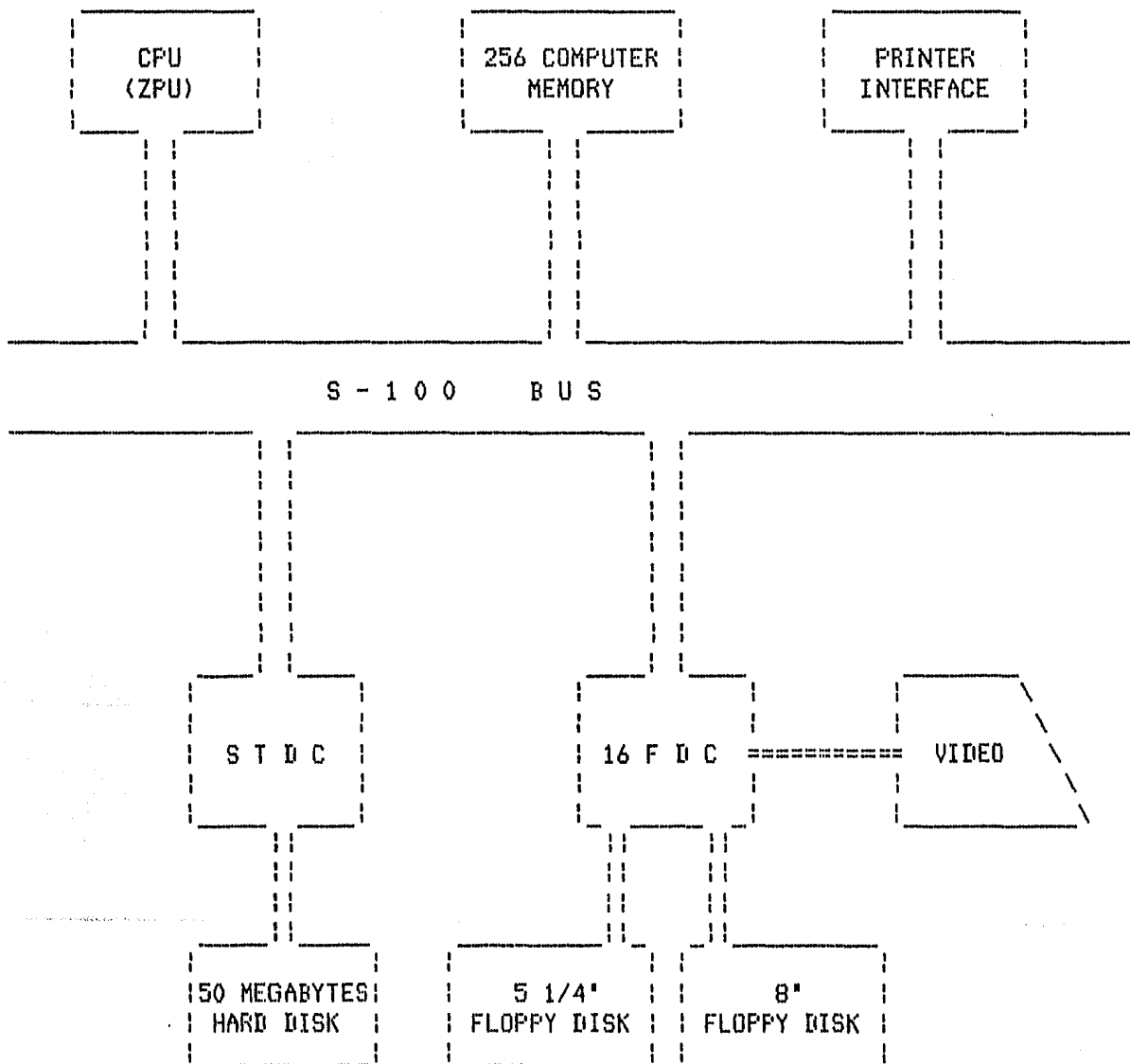
```
% del archivo  
o bien % del /letters/otro/archivo
```

Existen además una serie de programas que no son

intrínsecos del sistema operativo y que deben ser llamados a disco cuando se invocan. Entre ellos se puede citar:

cdoscopy	permite copiar archivos a/de diskettes en CDOS
copy	copia un archivo bajo Cromix
cpTREE	copia un directorio y sus descendientes a otro directorio
create	crea un archivo
delete	remueve un archivo o directorio desde un archivo de sistema
directory	cambia o despliega el directorio presente
find	escuenta archivos
help	despliega el manual del sistema como ayuda
l	lista la información de un archivo.
makdir	crea un directorio
rename	cambia el nombre de un directorio o un archivo
screen	llama al Screen Editor para editar archivos
shell	crea un proceso shell (de comandos)
sim	permite correr programas bajo formato CDOS
sort	sortea o junta archivos
type	manda los contenidos de un archivo de texto a la consola u otro dispositivo de E/S
who	lista a todos los usuarios que están utilizando en ese momento el sistema.

CROMEMCO MODEL CS3/3C



TRANSPORTABILIDAD DEL SISTEMA

Dentro de los objetivos planteados al iniciarse el desarrollo del Sistema Banco de Horas, se encuentra el tener la facilidad de ser implementado en otros equipos; siendo una de las características más importantes de un sistema, la transportabilidad.

La transportabilidad es la propiedad que tiene un sistema de poder ser implementado en otro equipo con un mínimo de cambios dentro de su estructura (software).

Así es que se tomaron en consideración varios lenguajes de programación, y como se dijo anteriormente COBOL ofrecía mayores ventajas en cuanto a la fácil implementación en otros equipos. Esto se debe básicamente a que existe un Estandard, el cual tiene el nombre de COBOL ANSI STANDARD, mismo que esta dividido en 12 Módulos los cuales son:

- 1.- NUCLEUS
- 2.- TABLE HANDLING

- 3.- SEQUENTIAL I/O
- 4.- RELATIVE I/O
- 5.- INDEXED I/O
- 6.- INTERPROGRAM COMMUNICATION
- 7.- LIBRARY
- 8.- COMMUNICATION
- 9.- DEBUG
- 10.- REPORT-WRITER
- 11.- SEGMENTATION
- 12.- SORT/MERGE

Cada módulo tiene 2 niveles de implementación, nombrados NIVEL I y NIVEL II (el cual es un subconjunto del NIVEL I). De acuerdo al Estandar, los tres primeros módulos en la lista, deben ser implementados por lo menos a Nivel I pero los otros nueve módulos pueden o no ser implementados.

De esta forma cada uno de los compiladores diseñados para Cobol se rigen por estas reglas produciendose versiones con bastante similitud y fácil compatibilidad.

Tenemos que Cromemco Cobol establece con respecto al ANSI Standard de lo más importante para el Sistema Banco de Horas, los Niveles Nucleus y Table Handling se incluyen a Nivel II.

-En cuanto a los módulos sobre los manejos de archivos, Cobol Cromemco los incluye a todos a Nivel II excepto que no puede manejar Llaves Multiples Indexadas y un lenguaje especial para el manejo de cintas.

-La Interprogram Communication y los módulos de Librería estan implementados a Nivel I.

-Los módulos de Debug y Report-Writer no están implementados del todo.

-Una extensión que tiene es el permitir el control de la pantalla por medio de SCREEN SECTION, además de opciones especiales para las instrucciones ACCEPT y DISPLAY, así por ejemplo:

```
ACCEPT (12, 35) var WITH AUTO-SKIP  
DISPLAY (16, 57) var ERASE
```

-Se tiene para los archivos secuenciales, la opción:

```
LINE SEQUENTIAL
```

En lo que respecta a COBOL II del sistema operativo RTOS de los equipos B 20, cuenta con las siguientes características:

-Los siguientes módulos están completamente implementados a Nivel 1:

- Segmentacion
- Library
- Debug

-Aparte los módulos que siguen se implementan a Nivel 2 en su totalidad:

- Nucleus
- Table Handling
- Sequential Input and Output
- Relative Input and Output
- Indexed Input and Output
- Inter-Program Communication
- Sort/Merge

-En cuanto al Módulo de Nucleus se tiene que esta implementado completamente. Además tiene unos extensiones respecto al ACCEPT y DISPLAY dado que su formato esta dado por:

```
ACCEPT var [AT {var}] FROM CRT  
DISPLAY var [AT {var2}] UPON {CRT/CRT-UNDER}.
```

-Referente a los módulos Sequential, Relative e Indexed I-O tenemos que estan completamente implementados a nivel 1 y 2 con la extensión de:

LINE SEQUENTIAL es un tipo de archivo adicional

Todas las clausulas FD son opcionales.

-El Debug tiene habilitado el paquete Run-Time-Debug.

En cuanto al MS-COBOL de Microsoft este trabaja bajo el sistema operativo MS-DOS. Este lenguaje esta conformado en base a los requerimientos de la ANSI STANDARD Cobol 1974, además de contener otra gran variedad de extensiones. Entre lo relevante tenemos:

-Se tienen extensiones en la DATA DIVISION (Screen Section) y en la PROCEDURE DIVISION (opciones Accept y Display) según se muestra a continuación:

```
ACCEPT (12, 25) var WITH AUTO-SKIP  
DISPLAY (16,57) var ERASE
```

-Se permite la opción para archivos Indexados la clausula ALTERNATE RECORD KEY así como archivos secuenciales la opción LINE SEQUENTIAL.

-Estan implementadas en su totalidad los módulos de SORT y MERGE además de tener las extensiones que comúnmente no estan en otras versiones de COBOL las siguientes instrucciones:

-EXIST, RENAME, REMOVE, COMMAND, UPCASE, LOCASE y EXCODE.

-Para finalizar se tiene a RM-COBOL de Unix el cual con respecto al ANSI STANDARD tiene las siguientes características:

Estan implementadas a Nivel 1 los módulos: Table Handling, Segmentation, Library Inter-Program Communication

Con Nivel 2 tenemos los módulos: Nucleus, Sequential I/O, Relative I/O, Indexed I/O.

No se implementan el Sort/Merge y Report Writer

-Se hacen modificaciones al ACCEPT y DISPLAY, al añadir especificaciones sobre la misma instrucción referentes a la pantalla.

-No soporta el Compilador la clausula LINE SEQUENTIAL.

-Se Permite la llamada a comandos del sistema operativo.

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

1.- COMPUTERS AND INFORMATION SYSTEMS

Gore & Stubbe.

McGraw-Hill Company.

1984. Second Edition.

2.- DESARROLLO Y ADMINISTRACION DE PROGRAMAS DE COMPUTADORA

Victor Gerez, Mouricio Mier, Rolando Nieva y
Guillermo Rodriguez.

C. E. C. S. A.

1984. Primera Edición.

3.- SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH

Roger S. Pressman.

McGraw-Hill Company.

1982. Primera Edición.

4.- SYSTEM THREE INSTRUCTION MANUAL

Cromemco.

Cromemco Inc.

1983.

5.- CROMIX INSTRUCTION MANUAL

Cromemco.

Cromemco Inc.

1982.

6.- CDOS INSTRUCTION MANUAL

Cromemco.

Cromemco Inc.

1981.

7.- SCREEN EDITOR INSTRUCTION MANUAL

Cromemco.

Cromemco Inc.

1981.

8.- COBOL REFERENCE MANUAL AND USER'S GUIDE

Cromemco.

Cromemco Inc.

1980.

9.- M P F - P C, USER'S GUIDE (MS-DOS VERSION)

Multitech Industrial.

Multitech Industrial Corp.

1985. Primera Edición.

9.- MICROSOFT MS-DOS, USER'S GUIDE OPERATING SYSTEM

Microsoft.

Microsoft Corporation.

1983. Second Edition.

10.- MICROSOFT COBOL REFERENCE MANUAL & USER'S GUIDE

Microsoft.

Microsoft Corporation.

1983.

11.- B 20 SYSTEMS COBOL II REFERENCE MANUAL

Burroughs.

Burroughs Corporation.

1983.

12.- RM-COBOL REFERENCE MANUAL

Onyx.

Onyx Inc.

1984.