



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

39

2E

FACULTAD DE INGENIERIA

SISTEMA NACIONAL DE INFORMACION PARA LA
SUBDIRECCION DE RELACIONES PUBLICAS DEL
INSTITUTO NACIONAL DE BELLAS ARTES

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N :
JOSE REFUGIO FERNANDEZ CORTES
SANDRA MARGARITA MEDINA NAVARRO
RODOLFO NAPOLES FRANCO



DIRECTOR DE TESIS:
M.I. JUAN CARLOS ROA BEIZA

MEXICO, D. F.,

1995

FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A MIS PADRES:

Por ser el apoyo que todo hijo desearía tener y por el amor que siempre me han brindado.

A MI ABUELITA(ANGELITA):

Por ser mi inspiración en toda la carrera.

A MI ESPOSA ALICIA:

Por ser el motivo de mi lucha día con día.

A MIS HERMANOS:

Por haber compartido tantos y tantos años, cada momento de mi vida.

A MIS TIOS Y PRIMOS:

Por la confianza que han tenido en mí.

A MIS AMIGOS:

Por estar conmigo en las buenas y en las malas.

JOSE

A DIOS:

Por todas las bendiciones recibidas y la vida misma.

A MIS PADRES:

Porque gracias a ellos, su dedicación y cariño, entregándonos siempre lo mejor de ellos; puedo llegar a este momento.

A RODOLFO Y AL BEBE:

Con mucho amor, por ser el centro de mi vida.

A MIS HERMANOS Y NENA:

Por todo el cariño y experiencias que hemos y seguimos compartiendo.

A MI ABUELITA Q.P.D., ABUELITO, TIOS Y PRIMOS:

En los que siempre encuentro una sonrisa amiga.

A MIS AMIGOS :

Por estar cerca en todo momento.

SANDRA MARGARITA

A DIOS:

Por la vida, por la sabiduría, por la esperanza,
por el amor y por la oportunidad de haber
obtenido una formación profesional.

A SANDRY:

Por su amor, por su apoyo, por su comprensión
y consejos como compañera de escuela, como
novia y como esposa.

A MIS PADRES:

Por su amor, por el apoyo incondicional en mi
vida personal y profesional y por la educación
que he recibido.

A MI ABUELITO:

Quien fue mi compañero durante toda mi vida
de estudiante y quien me daba aliento cuando
sentía que ya no podía seguir adelante.

A MIS HERMANAS:

Porque siempre contaba con su apoyo y
comprensión en los momentos más difíciles.

A MIS AMIGOS:

Porque no importando cuán difíciles fueran los
momentos o cuán divertidos podíamos pasarla,
siempre estaban dispuestos a compartir cada
momento.

RODOLFO

INDICE

CAPITULO I. FUNDAMENTOS TEORICOS.

I.1. INTRODUCCION A LAS BASES DE DATOS	1
I.1.1. Definición	1
I.1.2. Evolución del concepto de bases de datos	8
I.1.3. Independencia de datos (lógica y física)	12
I.1.4. Arquitecturas de bases de datos	16
I.1.5. Objetivos de una organización de bases de datos	27
I.1.6. Tipos de organización	38
I.1.7. Ventajas de utilizar bases de datos	39
I.1.8. Bases de datos relacionales	45
I.2. DISEÑO DE BASES DE DATOS	49
I.2.1. Etapas de diseño	49
I.2.2. Modelo entidad - relación	60
I.2.2.1. Diagrama entidad - relación	60
I.2.2.2. Entidades y conjunto de entidades	62
I.2.2.3. Relaciones y conjunto de relación	63
I.2.2.4. Tipos de relación	64
I.2.2.5. Llaves primarias	66
I.2.3. Esquemas y subesquemas	68
I.2.4. Normalización	83
I.3. SISTEMAS MANEJADORES DE B.D. PARA COMPUTADORAS PERSONALES	98
I.3.1. Dbase	98
I.3.2. Clipper	99
I.3.3. FoxPro	102
I.3.4. Datacase	103
I.3.5. Paradox	105
I.3.6. Informix SQL	106
I.3.7. R:Base	107
I.3.8. Comparativo de características	108
I.4. PLANEACION DE UN PROYECTO DE PROGRAMACION	114
I.4.1. Definición del problema	114
I.4.2. Desarrollo de una estrategia de solución	124
I.4.3. Planeación del proceso de desarrollo	127
I.4.4. Otras actividades de la planeación	143

CAPITULO II. IDENTIFICACION DE LA PROBLEMÁTICA Y PROPUESTA DE SOLUCIONES.

II.1. ANTECEDENTES	145
II.2. RECOPIACION DE LA INFORMACION	148
II.3. PROBLEMÁTICA ACTUAL	158
II.4. IDENTIFICACION Y ANALISIS DE REQUERIMIENTOS DEL USUARIO	161
II.5. ANALISIS DE LA INFORMACION	163
II.6. ALTERNATIVAS DE SOLUCION	167
II.6.1. Desarrollo con Clipper	167
II.6.2. Desarrollo con FoxPro	169

CAPITULO III. DISEÑO Y DESARROLLO DEL SISTEMA.

III.1. ELECCION DE LA BASE DE DATOS	172
III.1.1. Criterios de evaluación	172
III.1.2. Manejo de datos	175
III.1.3. Soporte multi - usuario	180
III.1.4. Recursos de programación	181
III.1.5. Requerimientos de hardware	183
III.1.6. Conclusión de la evaluación	185
III.2. DESCRIPCION DEL SISTEMA	188
III.2.1. Modulos principales	188
III.2.2. Diagrama de flujo de datos	246
III.2.3. Diagrama entidad - relación	250
III.2.4. Diccionario de datos	252
III.2.5. Normalización de tablas	264
III.3. DESARROLLO DEL SISTEMA	266
III.3.1. Programación	266
III.3.2. Pruebas modulares	393
III.3.3. Pruebas integrales	431

III.4. MANUALES DEL SISTEMA	472
III.4.1. Manual de mantenimiento	472
III.4.1.1. Descripción de archivos del sistema	472
III.4.1.2. Estructura del sistema	477
III.4.2. Manual de usuario	483
III.4.2.1. Instalación del sistema	483
III.4.2.2. Operación del sistema	484
III.4.2.2.1. Programación de eventos	485
III.4.2.2.2. Recintos	491
III.4.2.2.3. Disciplinas	493
III.4.2.2.4. Índices (regenera)	494
III.5. IMPLANTACION Y CAPACITACION	494
III.5.1. Implantación	494
III.5.2. Capacitación	496
CONCLUSIONES	499
GLOSARIO DE TERMINOS	501
BIBLIOGRAFIA	521

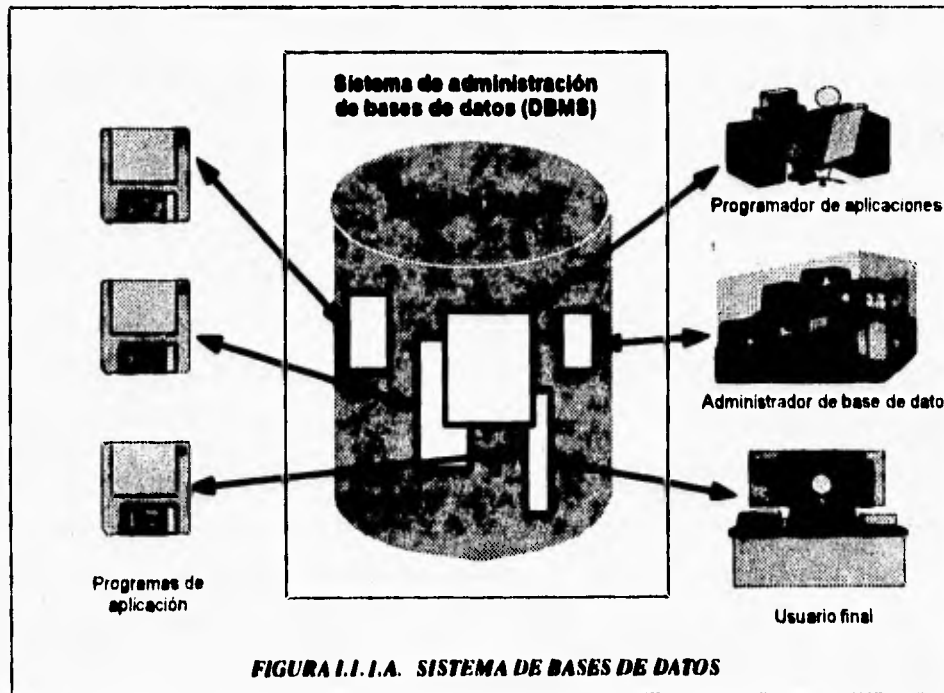
CAPITULO I.

FUNDAMENTOS TEORICOS

1.1.1 BASE DE DATOS

La tecnología de las bases de datos se ha descrito como "una de las áreas de la ciencia de la computación y la información de más rápido desarrollo".

Una base de datos, en esencia, no es más que un sistema de mantenimiento de registros basado en computadores, es decir, un sistema cuyo propósito general es registrar y mantener información. Tal información, puede estar relacionada con cualquier cosa que sea significativa para la organización donde el sistema opera.



La base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados.

Un sistema de bases de datos incluye cuatro componentes principales: datos, hardware, software y usuarios; estos se representan gráficamente en la figura I.1.1.A.

DATOS

Los datos almacenados en el sistema, se dividen en una o más bases de datos. Desde un punto de vista didáctico, es más conveniente suponer que solo hay una base de datos, la cual contiene todos los datos almacenados en el sistema.

Una base de datos, es pues, un conjunto de datos almacenados, y en general, es tanto integrada como compartida. Por compartida se entiende, que partes individuales de la base de datos pueden compartirse entre varios usuarios distintos, en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la base de datos y utilizarla para propósitos diferentes. Tal compartimiento, es en verdad consecuencia del hecho de que la base de datos es integrada. Otra consecuencia del mismo hecho, se advierte en que cualquier usuario específico, por lo general tendrá acceso tan solo a algún subconjunto de la base de datos completa; además, subconjuntos de diferentes usuarios se trasladarán de muy diversas maneras. En otras palabras, diferentes usuarios percibirán de modos muy distintos una base de datos específica.

La palabra compartido, a menudo se amplía para abarcar no solo al compartimiento antes descrito, sino también al compartimiento concurrente, es decir, la oportunidad de que diversos usuarios accesen en realidad la base de datos al mismo tiempo (llamado sistema de usuarios múltiples).

HARDWARE

Se compone de los volúmenes de almacenamiento secundario, donde reside la base de datos, junto con dispositivos asociados como las unidades de control, los canales, etc.

SOFTWARE

Entre la base de datos física y los usuarios del sistema, existe un nivel de software, que a menudo recibe el nombre de sistema de administración de bases de datos o DBMS. Este maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios. Una función general del administrador de la base de datos, por tanto, es proteger a los usuarios de esta, contra los detalles a nivel de hardware. En otras palabras, el administrador de la base de datos, ofrece una vista de la base de datos que está por encima del nivel de hardware y apoya las operaciones del usuario, que se expresan en términos de esa vista de nivel superior.

Comunmente, dentro del DBMS se maneja un lenguaje de definición de datos y un lenguaje de manipulación de datos. El resultado de la compilación del código del DDL es un conjunto de tablas que son almacenadas en un archivo especial llamado diccionario de datos. Un diccionario de datos es un archivo que contiene información referente a los datos. Este archivo se consulta antes de que los datos reales sean leídos o modificados en el sistema de base de datos.

Por manipulación de datos debemos entender la recuperación de información almacenada en una base de datos, la inserción de información nueva y el borrado de información en la base de datos. Un sistema de manipulación de datos (DML), es un lenguaje que permite el acceso o manipulación de datos organizados de acuerdo al modelo de datos. Existen dos tipos básicos de DML.

- De procedimientos. Requiere que el usuario especifique los datos necesarios y la manera de obtenerlos.
- De no procedimientos. Requiere que el usuario especifique los datos necesarios sin especificar la manera de obtenerlos.

USUARIOS

Se consideran tres clases generales de usuarios. La primera la representa el programador de aplicaciones, encargado de escribir programas de aplicación que utilicen bases de datos. Estos programas de aplicación, operan con los datos de todas las maneras usuales; recuperan información, crean información nueva, suprimen o cambian información existente, etc. Los programas en sí, pueden ser aplicaciones convencionales de procesamiento por lotes o programas en línea diseñados para apoyar a un usuario final, que interactúa con el sistema desde una terminal en línea.

La segunda clase de usuario es, el usuario final, que accesa la base de datos desde una terminal. Este puede emplear un lenguaje de consulta que forme parte integral del sistema o recurrir a un programa de aplicación escrito por un usuario programador que acepte ordenes desde la terminal y a su vez formule solicitudes al administrador de base de datos en nombre del usuario final. De cualquier manera, el usuario final, puede realizar, en general, todas las funciones de recuperación, creación, supresión y modificación, aunque

tal vez no sea falso afirmar que la recuperación es la función más común de esta clase de usuario.

La tercera clase de usuario, la representa el administrador de bases de datos. El administrador de la base de datos, es el custodio de los datos de la empresa, o de la parte de éstos con que su sistema está relacionado. Su función es la de controlar la estructura general de los datos. Obsérvese que ser custodio de los datos es muy distinto de ser su propietario. El propietario de los datos suele ser un departamento o un individuo. El administrador es el responsable de la seguridad y el control de los datos. Estos pueden ser utilizados por cualquier persona que tenga autoridad para ello. Además, controlar los datos, no significa que el administrador sabe qué contienen los registros. No obstante, si es necesario ampliar el ítem, solo el administrador puede hacer el cambio. Si un programador de aplicación quiere crear un nuevo tipo de registro o modificar un registro antiguo para incluir nuevos ítems de datos o aumentar el tamaño de un ítem existente, debe pedir permiso al administrador. Este tomará las medidas necesarias para modificar las estructuras de datos de la manera que crea más conveniente para el sistema en conjunto. No se permite que un programador de aplicación o un analista de sistemas que está trabajando en una aplicación altere la estructura general de los datos; solo el administrador y el personal que tenga un punto de vista global del sistema, pueden estar familiarizados con las características en conjunto de los datos.

El administrador de base de datos, tiene una visión general de la organización de los datos. En realidad, el administrador es a menudo un departamento o un grupo más bien que un individuo. La completa comprensión de la base de datos, su organización, sus criterios de diseño, y los requerimientos de muchos usuarios, es demasiado para un solo hombre.

La idea básica en la implantación de una base de datos es la de que los mismos datos deben ser aprovechados para tantas aplicaciones como sea posible. En la base de datos se pretende eliminar la redundancia, por lo que ha sido definida como una colección no redundante de ítems de datos, pero en realidad, en muchas bases de datos se admite cierta redundancia, con el objeto de reducir los tiempos de acceso o simplificar los métodos de direccionamiento. Algunos registros se duplican, para facilitar la reconstrucción de la base en caso de daño accidental. Por lo tanto, hay necesidad de armonizar el grado de redundancia con otras características deseables de la base, de modo, que es preferible hablar de redundancia controlada o redundancia mínima en lugar de no redundancia como criterio de diseño. En otros términos, en una base de datos bien diseñada se evita la redundancia perjudicial o innecesaria.

La redundancia no controlada acarrea varios inconvenientes. En primer lugar tenemos el costo adicional del almacenamiento de copias múltiples de los mismos datos. En segundo término, para actualizar por lo menos una parte de las copias redundantes, es preciso recurrir a múltiples operaciones de actualización. La redundancia, por lo tanto, es más costosa en los archivos que exigen una actualización permanente, o peor, en los que es frecuente la inclusión de nuevos datos y la eliminación de los obsoletos. Debido a que las distintas copias pueden hallarse en diferentes estados de actualización, el sistema tiende a proporcionar informaciones contradictorias.

En las grandes operaciones de procesamiento que no cuentan con el apoyo de una base de datos, hay tantos datos redundantes que resulta prácticamente imposible mantenerlos todos en el mismo nivel de actualización. Muchas veces, los usuarios sufren las consecuencias de la incoherencia a que esto da lugar, y dejan de considerar confiable la información que reciben. La incapacidad para mantener homogéneamente actualizados

los datos redundantes, provoca en ocasiones, que los usuarios se encarguen de desacreditar el sistema.

La base de datos debe prestarse a una fácil reestructuración siempre que haya que agregarle nuevos tipos de datos o utilizarla para nuevas aplicaciones. Esta reestructuración no debe originar la necesidad de volver a escribir los programas de aplicación, y en general, no debe ser fuente de trastornos.

A menudo se habla de la independencia de datos como de uno de los atributos destacados de la base de datos. Esta idea implica que los datos y los programas de aplicación que de ellos se sirven son mutuamente independientes, de manera, que unos u otros puedan ser modificados sin afectar a los restantes. En particular, el programador de aplicaciones no debe ser afectado por los cambios que se introduzcan en los datos, en su organización, o en los dispositivos físicos donde se almacenan. En realidad, así como en la práctica son pocas veces totalmente no redundantes, así también son pocas veces completamente independientes.

No obstante, la independencia de datos es uno de los más valiosos argumentos en pro de las bases de datos.

En gran medida, es preocupación principal en la organización de la base de datos, la representación de las relaciones que existen entre ítems de datos y registros, así como también el cómo y el dónde se almacenan los datos.

La seguridad de los datos y la posibilidad de reconstruirlos en caso de falla, son aspectos de gran importancia en el diseño de toda base de datos.

I.1.2 EVOLUCION DEL CONCEPTO DE BASES DE DATOS

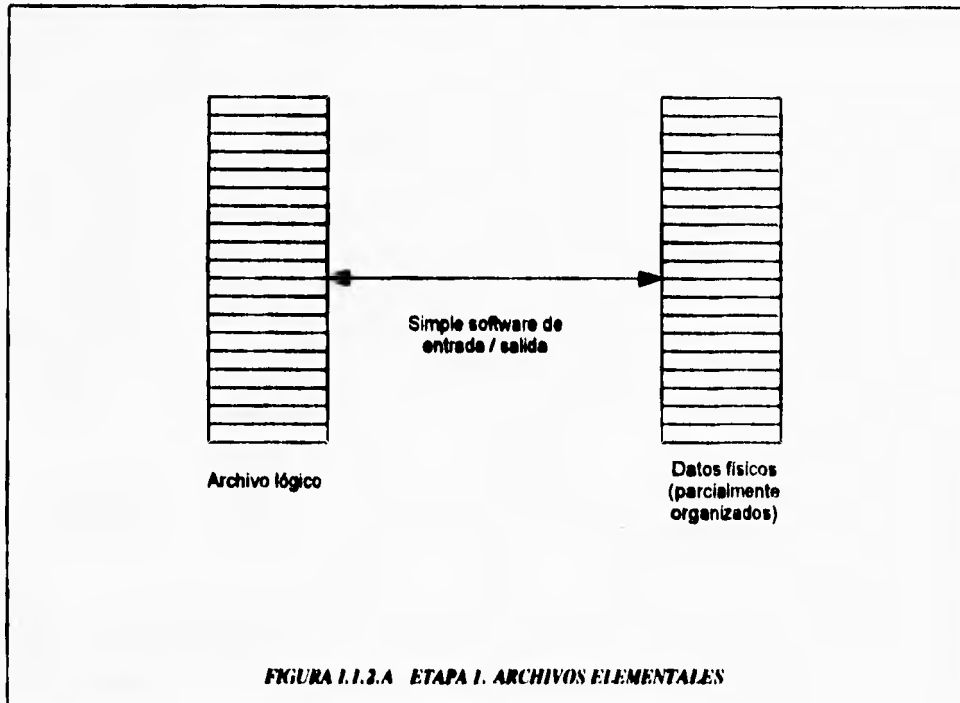
La expresión base de datos comenzó a popularizarse al principio del decenio que se inició en 1969. Antes de esto, se hablaba de archivos y de conjuntos de datos, pero no tenían las características de no redundancia, independencia de datos, interconectividad, protección de seguridad, y en muchos casos, accesibilidad en tiempo real. Como campo comercial, es relativamente nueva; los fabricantes y vendedores no empezaron a ofrecer sistemas de administración de bases de datos hasta mediados de la década de los sesentas.

Antes de que aparecieran las computadoras de la tercera generación (la primera se instaló en 1965), el software ejecutaba las operaciones de entrada/salida de los dispositivos de almacenamiento. La codificación incluida en los programas de aplicación, se encargaba en forma muy elemental, de la organización de los datos, por lo general solo servía a modo de simples archivos secuenciales en cinta. No había independencia de datos. Si se modificaba la organización de los datos o se cambiaban los dispositivos de almacenamiento, el programador estaba obligado a volver a escribir los programas y repetir luego los procesos de compilación y depuración. Para actualizar un archivo, se escribía de nuevo. Se conservaba el original, el padre y también la generación anterior.

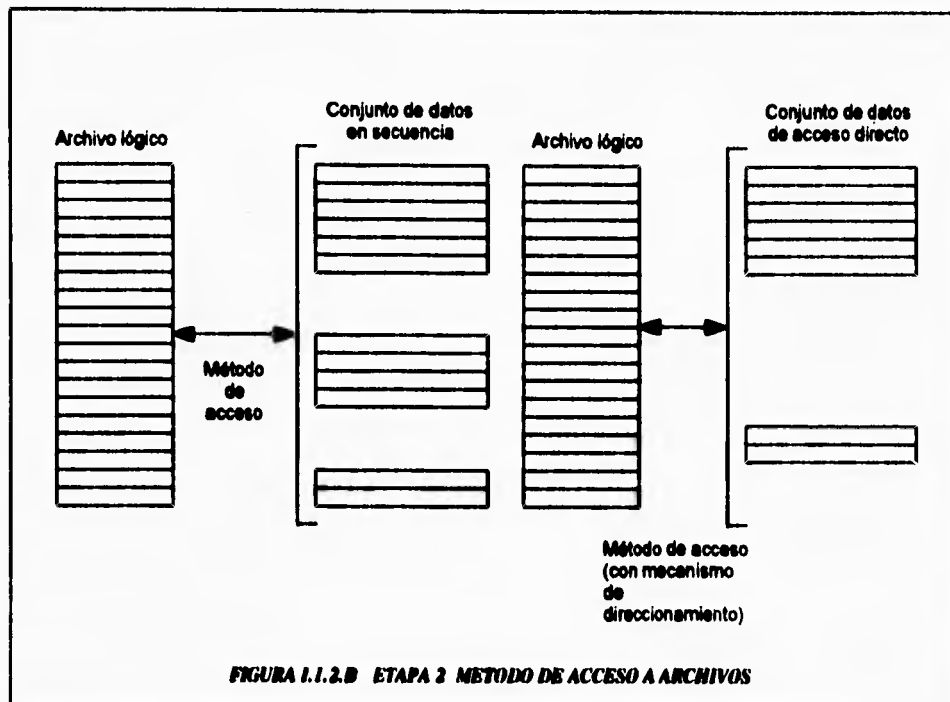
La mayoría de los archivos servían solo para una aplicación. Se originaba de este modo un elevado nivel de redundancia, con varios archivos que en resumen contenían prácticamente los mismos datos.

En la era de la etapa 1 (figura I.1.2.A), se utilizaban también algunos dispositivos de acceso al azar, que permitían al usuario el acceso a cualquiera de los registros almacenados sin tener que explorar forzosamente todo el archivo. Los medios de

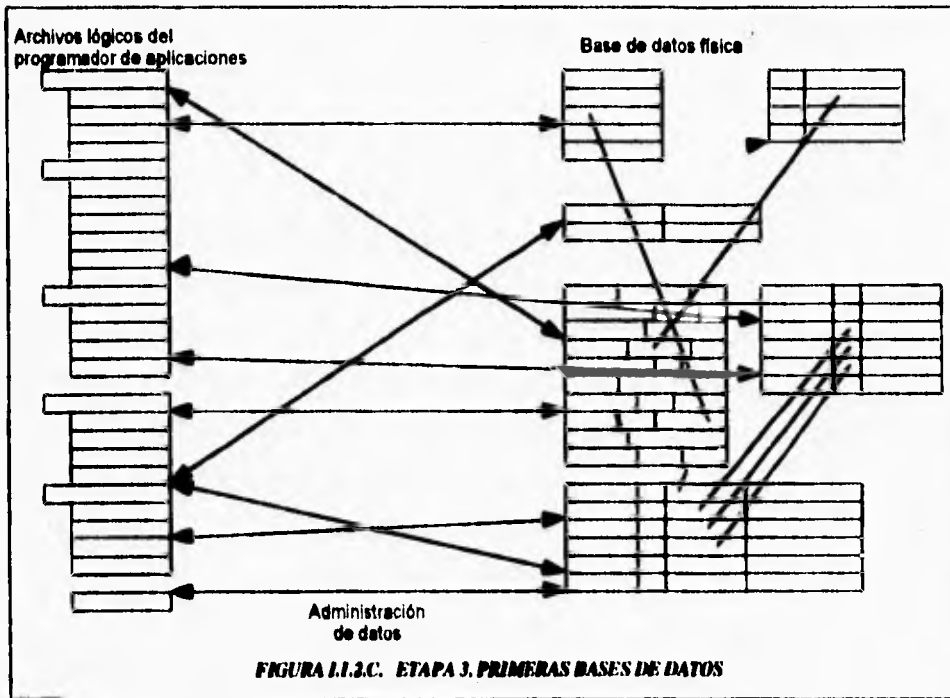
direccionamiento, no obstante, tenían que ser previstos por el programador al escribir su programa. Si se cambiaban los dispositivos de almacenamiento, había que reformar todos los programas.



En la segunda etapa (figura 1.1.2.B), se intentó proteger al programador contra los efectos de los cambios que se introducían en el hardware. El software hizo posible modificar la distribución física de los datos, sin que por ello se alterase su estructura lógica, siempre que no se introdujesen cambios en los contenidos de los registros ni en la estructura fundamental de los archivos. Los archivos utilizados durante esta segunda época, estaban por lo general diseñados, como los de la primera, para una aplicación determinada o para un grupo de aplicaciones muy similares.



En la etapa 3 (figura 1.1.2.C), aparecen las primeras bases de datos, en donde de los mismos datos físicos, se derivan múltiples bases de datos lógicas y en donde los datos pueden utilizarse para diferentes aplicaciones. En esta etapa, el software proporciona medios para disminuir la redundancia, por lo tanto se conserva la integridad de los datos, si se decide cambiar el almacenamiento físico no es necesaria la modificación de aplicaciones, es posible direccionar los datos a nivel campo o grupo. Y por último, es posible utilizar organizaciones de datos muy complejas, sin que esto se refleje en los programas de aplicación.



I.1.3 INDEPENDENCIA DE DATOS

La independencia de los datos puede entenderse mejor si se analiza primero el fenómeno opuesto. La mayoría de las aplicaciones actuales son dependientes de los datos. Esto significa que la manera como los datos se organizan en almacenamiento secundario y la manera como se tiene acceso, dependen de los requerimientos de la aplicación y además, que el conocimiento de la organización de los datos y de la técnica de acceso forma parte de la lógica de la aplicación. Por ejemplo, al decidir, que un archivo particular se almacene en forma secuencial con índices por razones de desempeño; la aplicación debe saber que el índice existe y conocer la secuencia del archivo, de modo que la estructura interna de la aplicación se construirá con base en este conocimiento. Además, la forma de los diversos de acceso y de control dentro de la aplicación dependerá, ante todo, de la interfase presentada por el software secuencial. Se dice que una aplicación como esta es dependiente de los datos porque es imposible cambiar la estructura de almacenamiento (la manera como están registrados los datos físicamente) o la estrategia de acceso (manera como se accesan los datos) sin afectar la aplicación.

Puede concluirse que la provisión de independencia de los datos, es un objetivo esencial de los sistemas de bases de datos. Es posible, definir la independencia de los datos como la inmunidad de las aplicaciones a los cambios de la estructura de almacenamiento y de la estrategia de acceso; lo que implica, desde luego, que las aplicaciones no dependen de ninguna estructura de almacenamiento o estrategia de acceso especial.

En muchos sistemas, los cambios de la estructura lógica general de los datos es permanente, por esta razón, se necesitan dos niveles de independencia de datos: la independencia física y la independencia lógica.

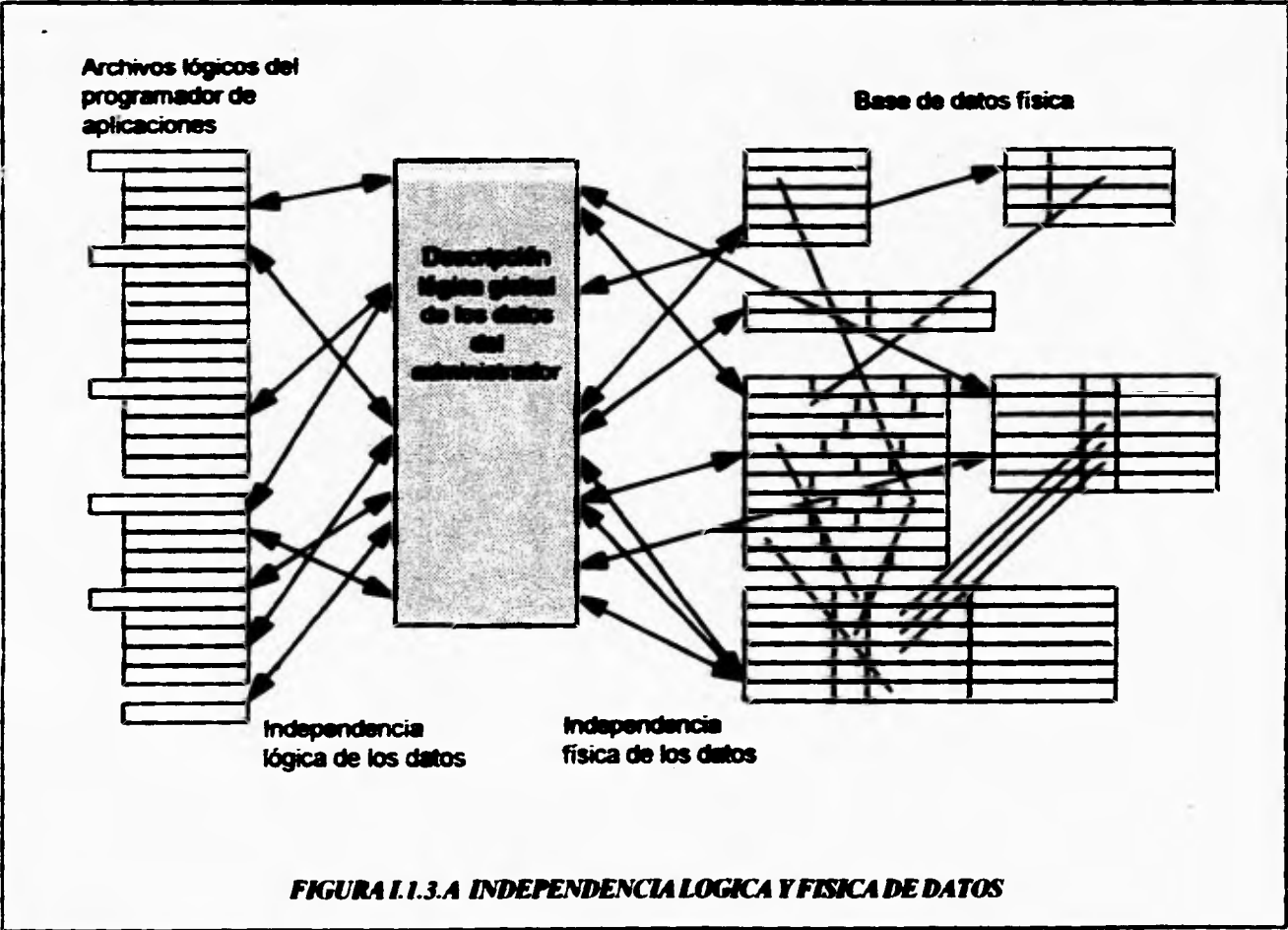
Por independencia lógica entendemos que la modificación de la estructura lógica general no afecta a los programas de aplicación (el cambio, desde luego no debe eliminar ninguno de los datos que el programa necesita).

Por independencia física de datos, entendemos que pueden modificarse la distribución y la organización físicas de los datos sin afectar ni a la estructura lógica general ni a los programas de aplicación.

La figura I.1.3.A, ilustra el concepto de independencia de datos en sus dos niveles, y la figura I.1.3.B, muestra las características de independencia de datos deseables. El bloque que aparece en el centro de la ilustración representa la estructura lógica general de los datos, llamada a menudo vista lógica global de los datos. Esta vista puede ser enteramente distinta de la que ofrece la estructura física y de la propia de los programas de aplicación.

El software de la base de datos, se encargará de convertir la vista que el programador de aplicación tiene de los datos en la vista lógica global y transformará luego, esta vista lógica global en la representación física. El propósito de la estructura que vemos en este diagrama, es permitir la máxima libertad para cambiar las estructuras de los datos sin rehacer mucho de lo ya hecho en la base de datos.

En la figura de características de independencia, se muestran algunos cambios frecuentes en las bases de datos, y se indica si esos cambios pueden ejecutarse sin reestructurar la organización física de almacenamiento, la vista lógica global de los datos o los



programas de aplicación. Las cruces representan los objetivos para el diseño del software de bases de datos moderno.

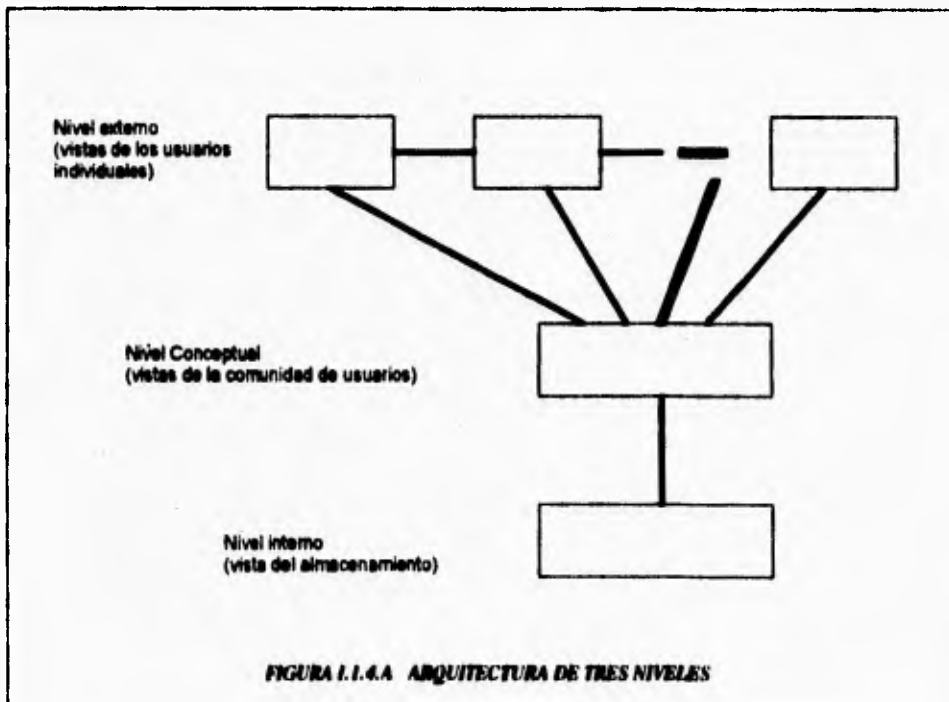
	No hay cambios en los (otros) programas de aplicación	No hay cambio en la descripción lógica global de los datos	No hay cambio en la organización del almacenamiento físico
Se agrega un nuevo programa de aplicación, el que utiliza nuevos tipos de datos	X	X	X
Un programa de aplicación exige una representación modificada de los datos existentes (por ej., punto flotante en lugar de punto fijo)	X	X	X
Se agrega un nuevo programa de aplicación que utiliza los tipos de datos existentes	X		
Se insertan nuevas ocurrencias de registro o se eliminan las viejas	X	X	X
Se mejora la descripción lógica global de los datos, o se crean nuevas relaciones entre los datos	X		
Se consolidan dos bases de datos	X		
Se mejora la organización física de los datos; posiblemente se usan diferentes representaciones	X	X	
Se modifican los métodos de direccionamiento	X	X	
Los datos se mudan a un tipo diferente de volumen	X	X	
Se modifica el software	X	X	
Se modifica el Hardware	X	X	

FIGURA 1.1.3.B INDEPENDENCIA DE DATOS DESEABLES

1.1.4 ARQUITECTURA

Una arquitectura comunmente aceptada, es conocida como arquitectura de tres niveles, la cual fue propuesta inicialmente por ANSI/SPARC, (SPARC, Standars Planning and Requirements Committee; ANSI, American National Standards Institute). En 1975, este grupo propuso una arquitectura estandar, la cual se muestra en la figura 1.1.4.A y que contiene los siguientes tres niveles:

1. Un nivel conceptual, para representar una vista de una comunidad de usuarios
2. Un nivel externo, para representar cualquier número de vistas de usuario
3. Un nivel interno, para representar la forma de almacenamiento de los datos



En términos generales, el nivel interno, es el más cercano al almacenamiento físico, es decir, el que concierne a la manera como los datos de almacenan en realidad; el nivel externo, es el más cercano a los usuarios, es decir, el que atañe a la manera cómo cada usuarios ve los datos; y el nivel conceptual, es un nivel de mediación entre los otros dos.

En otras palabras, habrá muchas vistas externas, cada una compuesta por una representación mas o menos abstracta de alguna parte de la base de datos, y habrá una sola vista conceptual, compuesta también por una representación abstracta de la base de datos en su totalidad. Hay que recordar que la mayoría de los usuarios no se interesan en la base de datos total, sino tan solo en alguna parte limitada. Así mismo, habrá una sola vista interna que representa la base de datos total tal como está almacenada.

El manejador de la base de datos basado en la arquitectura de tres niveles, debera ofrecer lenguajes para definir la base de datos conceptual, las vistas externas y la representación de almacenamiento de los datos. El software manejador de base de datos, debe ser capaz de convertir los comandos de las vistas externas del usuario en comandos del nivel de almacenamiento conceptual

Un aspecto importante de la arquitectura de tres niveles, es el modelo que define las bases de datos en el nivel conceptual. Como otros modelos de datos, la arquitectura de tres niveles debe permitir al usuario modelar las empresas de manera natural. Y una vez más, en la generalidad, el modelo de datos en el nivel conceptual, debe modelar una variedad de empresas usuarias. Para lograrlo, el nivel conceptual soporta un modelo de datos, y el modelo de usuario de la empresa debe convertirse en el modelo conceptual de datos. La arquitectura de tres niveles sin embargo, no restringe el nivel externo al mismo modelo de datos tal como se usa para definir el esquema conceptual. En tales

arquitecturas, es posible definirlo, por ejemplo, para definir la empresa usando un modelo de datos, por decir el relacional, y para proveer el esquema externo usando otros modelos de datos (ver figura 1.1.4.B).

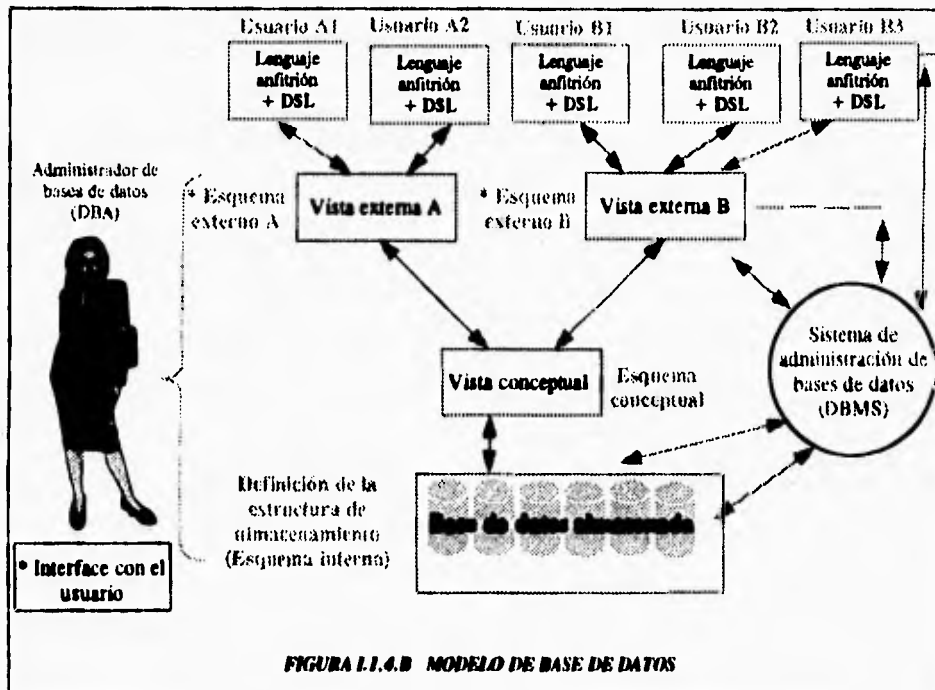


FIGURA 1.1.4.B MODELO DE BASE DE DATOS

Generalmente las arquitecturas de tres niveles que soportan varios modelos de datos en la interfase externa no están disponibles comercialmente. La falta de una solución general para la conversión de comandos entre cualquier modelo de datos, en los niveles conceptual y externo limita el rango de vistas externas que pueden ser soportadas por un esquema conceptual determinado. De aquí, que hay restricciones en el rango de vistas que puede proveer una interfase externa. La mayoría de los manejadores de bases de datos, soportan vistas externas en los siguientes casos:

- La vista externa usa el mismo modelo de datos que el esquema conceptual

- La vista externa es un subconjunto del esquema conceptual
- Algunos registros en la vista externa contiene items que computacionalmente se derivan de más de un registro en el esquema conceptual.

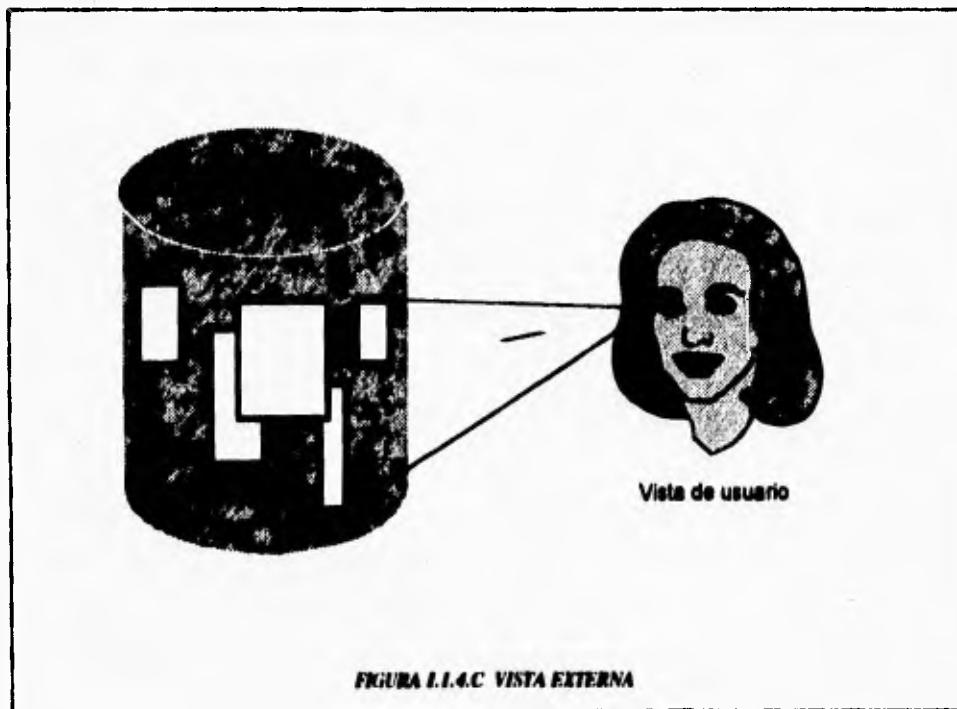
Cada usuario tiene un lenguaje a su disposición. Para el programador de aplicaciones, se trata de un lenguaje convencional de programación como COBOL o PL/I; para el usuario de una terminal, se trata de un lenguaje de consulta o un lenguaje de propósito especial "hecho a la medida de sus necesidades" y apoyado por un programa de aplicación en línea. Para nuestros propósitos, lo importante del lenguaje del usuario es que incluye un sublenguaje de datos (DSL), es decir, un subconjunto del lenguaje total que concierne a los objetos y a las operaciones de la base de datos. Se dice del sublenguaje de datos que está implícito en un lenguaje anfitrión. Un sistema específico puede incluir múltiples lenguajes anfitrión y numerosos sublenguajes de datos.

En principio, cualquier sublenguaje de datos es en realidad una combinación de dos lenguajes: un lenguaje de definición de datos (DDL) que permite la definición o descripción de los objetos de la base de datos (tal como los percibe el usuario), y un lenguaje de manipulación de datos (DML) que apoya el manejo o procesamiento de esos objetos.

Hay que recordar, uno, las definiciones son completamente externas al programa de aplicación y se escriben en un DDL, que no se asemeja en absoluto al lenguaje anfitrión del usuario; y dos, la manipulación se lleva a cabo mediante proposiciones de llamado a subrutinas estándar (suministradas como parte del DBMS) también es por tanto, ajena al marco de referencia del lenguaje anfitrión. En otras palabras, en la mayoría de los sistemas actuales, el sublenguaje de datos y el lenguaje anfitrión, están acoplados con soltura. Un sistema estrechamente acoplado, ofrece un conjunto más uniforme de

recursos para el usuario, pero implica desde luego, más esfuerzo por parte de los diseñadores y los productores del sistema.

Considerando de nuevo la arquitectura; un usuario individual, a menudo solo se interesa en alguna parte de la base de datos como ya hemos mencionado, y la vista que tenga el usuario de esta parte, casi siempre es un poco abstracta cuando se compara con la manera en que esten físicamente almacenados los datos. La vista de un usuario individual en términos de ANSI/SPARC, se llama vista externa. Por lo tanto, la vista externa, es el contenido de la base de datos tal como lo ve un usuario específico.



Cada vista externa, se define por medio de un esquema externo, el cual se compone en esencia de las definiciones de cada uno de los diversos tipos de registros externos de esa vista externa; el esquema externo, se escribe usando la parte de DDL del sublenguaje de datos. Ese DDL se conoce como DDL externo (figura I.1.4.C).

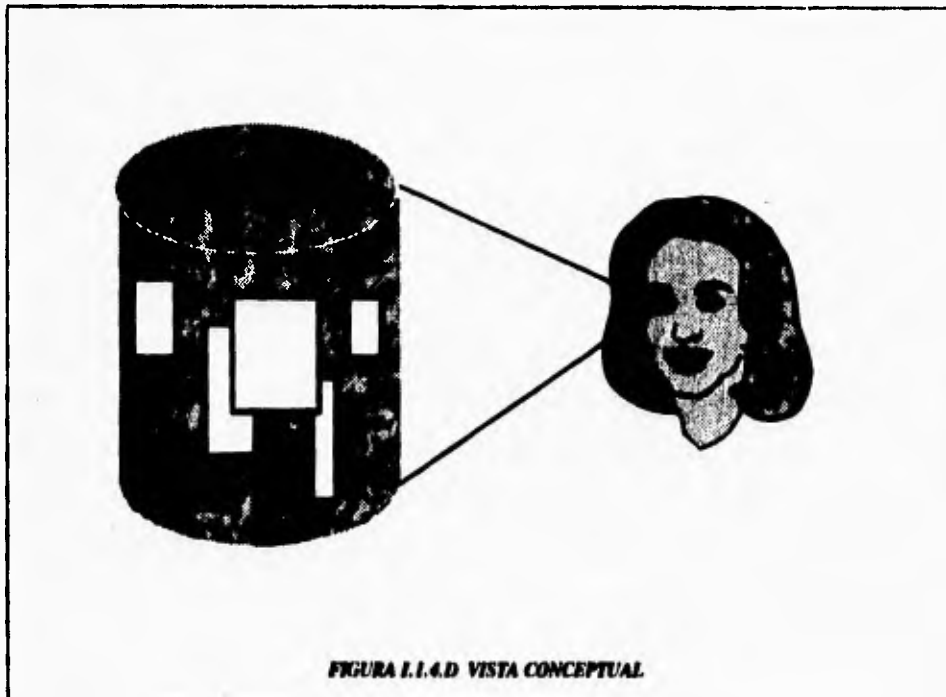
Ahora, examinemos el nivel conceptual. La vista conceptual, es una representación del contenido total de información de la base de datos, una vez más en forma relativamente abstracta, en comparación con la forma en la cual se almacenan los datos físicamente. Puede ser también, bastante diferente de la manera en que cualquier usuario específico vea los datos. En términos generales, se pretende que sea una vista de los datos "como son en realidad" y no como los usuarios están obligados a verlos por las restricciones del lenguaje particular o del hardware que utilizan (ver figura I.1.4.D).

Si se desea lograr independencia de los datos, las definiciones de los registros conceptuales incluidas en el esquema conceptual, no deben incluir ninguna consideración sobre la estructura de almacenamiento ni sobre la estrategia de acceso (deben ser definiciones de contenido de información únicamente). Así pues, no debe haber ninguna referencia a representaciones de campos almacenados, secuencias físicas, índices, direccionamiento por dispersión o cualquier otro detalle de almacenamiento/acceso. Por lo tanto, si de esta manera, el esquema conceptual se hace en realidad independiente de los datos, los esquemas externos que se definen en términos del esquema conceptual, también serán con mayor razón independientes de los datos.

Cada vista externa, se define por medio de un esquema externo, el cual se compone en esencia de las definiciones de cada uno de los diversos tipos de registros externos de esa vista externa; el esquema externo, se escribe usando la parte de DDL del sublenguaje de datos. Ese DDL se conoce como DDL externo (figura I.1.4.C).

Ahora, examinemos el nivel conceptual. La vista conceptual, es una representación del contenido total de información de la base de datos, una vez más en forma relativamente abstracta, en comparación con la forma en la cual se almacenan los datos físicamente. Puede ser también, bastante diferente de la manera en que cualquier usuario específico vea los datos. En términos generales, se pretende que sea una vista de los datos "como son en realidad" y no como los usuarios están obligados a verlos por las restricciones del lenguaje particular o del hardware que utilizan (ver figura I.1.4.D).

Si se desea lograr independencia de los datos, las definiciones de los registros conceptuales incluidas en el esquema conceptual, no deben incluir ninguna consideración sobre la estructura de almacenamiento ni sobre la estrategia de acceso (deben ser definiciones de contenido de información únicamente). Así pues, no debe haber ninguna referencia a representaciones de campos almacenados, secuencias físicas, índices, direccionamiento por dispersión o cualquier otro detalle de almacenamiento/acceso. Por lo tanto, si de esta manera, el esquema conceptual se hace en realidad independiente de los datos, los esquemas externos que se definen en términos del esquema conceptual, también serán con mayor razón independientes de los datos.



La vista conceptual, es por tanto, un vista del contenido total de la base de datos, y el esquema conceptual, es una definición de esa vista. Pero, no hay que suponer que el esquema conceptual es un conjunto de definiciones muy semejantes a las definiciones sencillas de registros de un programa. Las definiciones del esquema conceptual, pretenden incluir numerosas características adicionales, tales como los controles de autorización y los procedimientos de validación.

Es necesario recalcar, que en la mayoría de los sistemas existentes, la vista conceptual es más que una sencilla unión de todas las vistas de los usuarios individuales, tal vez con la adición de algunos procedimientos sencillos de validación y autorización. Pero parece no

haber duda de que los sistemas de bases de datos del futuro, seran mucho más complejos en cuanto a su nivel conceptual.

El tercer nivel de la arquitectura, es el interno, la vista interna, es una representación de nivel muy bajo de la base de datos en su totalidad; se compone de múltiples ocurrencias y de múltiples tipos de registros internos.

Registro interno, es el término que ANSI/SPARC utiliza para la construcción llamada registro almacenado; la vista interna, por tanto, aún se mantiene a un paso del nivel físico, ya que no atañe a registros físicos o bloques, ni a ninguna restricción específica de dispositivos tales como capacidades de cilindros o pistas. La vista interna, se describe por medio del esquema interno, el cual no solo define los diversos tipos de registros almacenados, sino que también especifica cuáles índices existen, de qué manera se representan los campos almacenados, en qué secuencia física se hallan almacenados los registros, etc. El esquema interno se escribe usando otro lenguaje de definición de datos: el DDL interno.

Existen dos tipos de correspondencia en el sistema, uno entre el nivel externo y el nivel conceptual, y el otro entre los niveles conceptual e interno.

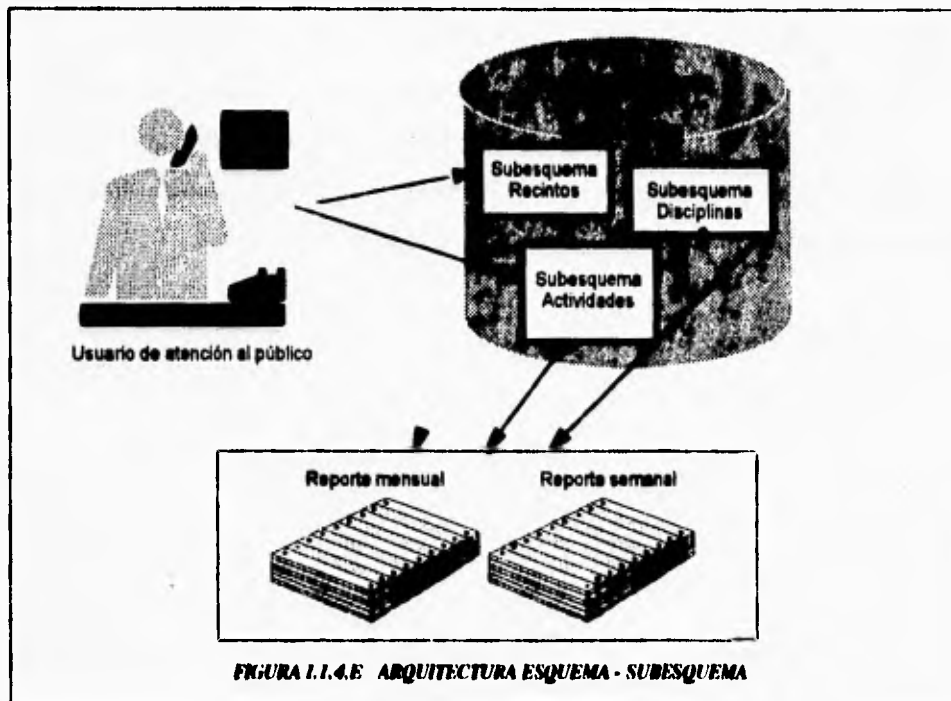
La correspondencia conceptual/interna, define la correspondencia entre la vista conceptual y la bases de datos almacenada; especifica de qué manera corresponden los registros conceptuales y los campos, con sus contrapartes almacenadas. Si se cambia la estructura de la base de datos almacenada, es decir, si se efectúa un cambio en la definición de la estructura de almacenamiento, la correspondencia conceptual/interna debe cambiarse en reciprocidad para que el esquema conceptual no varíe. En otras

palabras, los efectos de esas modificaciones deben estar contenidos bajo el nivel conceptual para que pueda lograrse la independencia de los datos.

La correspondencia externa/conceptual, define la correspondencia entre una vista externa específica y la vista conceptual. En general, puede existir la misma clase de diferencias entre estos dos niveles y entre la vista conceptual y la base de datos almacenada; por ejemplo, los registros pueden tener diferentes tipos de datos, los registros pueden ordenarse en secuencias distintas, etc. Pueden existir cualquier número de vistas externas, un gran número de usuarios puede compartir una vista externa específica. Por otra parte algunos sistemas permiten que la definición de una vista externa se exprese en términos de otras, en vez de requerir siempre una definición explícita de la correspondencia con el nivel conceptual.

ARQUITECTURA ESQUEMA-SUBESQUEMA

La arquitectura esquema-subesquema fue propuesta en 1971 por DBTG (Data Base Task Group) del comité de CODASYL. Se distingue entre la definición de la base de datos y la definición de aquella parte de la base de datos que está disponible para ciertos programas en particular (se muestra en la figura 1.1.4.E).



El esquema define la base de datos entera, y el subsquema define la vista del usuario. Generalmente el subsquema es un subconjunto del esquema que usa el mismo modelo de datos. Cada programa debe tener un subsquema, sin embargo, más de un programa puede compartir un subsquema.

El término de lenguaje manipulador de datos (DML) se usa frecuentemente para referirse a los comandos que hacen posible el acceso a la base de datos. Estos comandos están implícitos en un programa y son usados para transferir datos entre el programa y la base de datos.

Versiones anteriores del CODASYL, no separaban las vistas internas y conceptuales, por lo tanto estas estaban definidas en un mismo esquema. Desarrollos posteriores, dividieron las definiciones físicas del esquema y propusieron desarrollar un DSDL (Data Storage Definition Language) separado para definir estructuras de almacenamiento. El esquema contendría solamente la definición lógica de datos del usuario.

I.1.5 OBJETIVOS DE UNA ORGANIZACION DE BASES DE DATOS

1. VERSATILIDAD PARA LA REPRESENTACION DE RELACIONES

Diferentes programadores requieren diferentes archivos lógicos. Algunas bases de datos, comprenden un complejo entretreído de relaciones. el método de organización debe tener capacidad para representar éstas relaciones y acomodar los posibles cambios en el futuro. Puede haber muy poco parecido entre el archivo lógico que se presenta a un programador de aplicaciones y el almacenamiento físico de los datos.

2. DESEMPEÑO

Las bases de datos diseñadas para ser usadas por operadores de terminal, deben asegurar un tiempo de respuesta adecuado para el diálogo entre el hombre y la terminal. Además, el sistema de bases de datos debe tener capacidad para manejar un adecuado caudal de transacciones. En los sistemas en que el volumen de tráfico es reducido, el caudal de transacciones, no tiene por qué imponer muchas restricciones al diseño de la base de datos. En cambio, en los sistemas de alto volumen de tráfico, el caudal de transacciones, tiene una gran influencia sobre la organización del almacenamiento físico.

En los sistemas diseñados solo para el procesamiento por lotes, el tiempo de respuesta no tiene una gran significación y el método de organización física se elegirá teniendo a la vista la mayor eficiencia con ese tipo de procesamiento.

3. COSTO MINIMO

Con el fin de mantener bajo el costo hay que elegir técnicas que minimizan las necesidades totales de almacenamiento. Apelando a éstas técnicas, la presentación física de los datos en el almacén puede ser muy distinta de la representación que usa el programador. La conversación entre ambas representaciones se hace por medio del software, o posiblemente del hardware o de la microprogramación. Siempre debe llegarse a una concordancia entre el costo de los algoritmos de conversación y la economía de almacén.

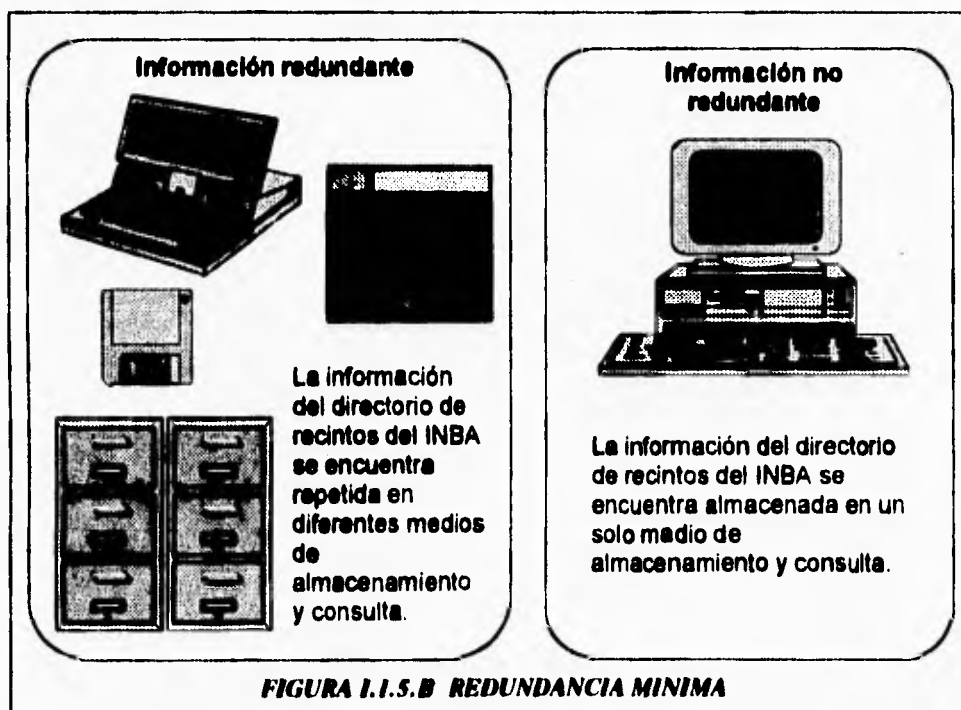


El costo por bit de almacenamiento, está disminuyendo rápidamente gracias al progreso tecnológico, mientras que no ocurre lo mismo con el costo de la programación. Hay por lo tanto una necesidad creciente de mantener sencilla la programación de aplicación, y las organizaciones lógicas de datos deben diseñarse con este objetivo.

Ver figura I.1.5.A

4. REDUNDANCIA MINIMA

Antes de la aparición de las técnicas de base de datos, existía un altísimo nivel de redundancia en los sistemas de procesamiento de datos.



Debería ser un objetivo de la organización de una base de datos el de eliminar los valores de datos redundantes, siempre que resulte económico, y controlar las incoherencias a que pueden dar lugar esos valores redundantes. Pero si hay una sola copia de ítems de datos que son necesarios para diversos fines, es preciso almacenar y mantener las relaciones pertinentes. Por esta razón, entre otras, puede no ser siempre razonable eliminar toda la redundancia en un sistema. Ver figura I.1.5.B

5. CAPACIDAD DE BUSQUEDA

Hay una creciente demanda para sistemas que sean capaces de atender consultas o



FIGURA I.1.5.C CAPACIDAD DE BUSQUEDA

producir informes que no han sido previstos detalladamente en la época del diseño.

Uno de los objetivos de la organización es, pues, el de lograr capacidad para la búsqueda rápida y flexible. La aptitud para la rápida exploración de una base de datos con criterios de búsqueda variados es onerosa con los actuales hardware y software. Ha resultado demasiado costosa en muchos sistemas que de otro modo la habrían incluido en su diseño. Con el avance de la tecnología, la capacidad para la exploración rápida y flexible será un objetivo más accesible en el futuro.

Ver figura I.1.5.C

6. INTEGRIDAD

Ocasionalmente, se producirán fallas de hardware y diversos tipos de accidentes. El almacenamiento de los datos y los procedimientos de actualización e inserción deben asegurar que el sistema pueda recuperarse de éstas contingencias sin daño para los datos. Toda instalación debe garantizar la integridad de la información que almacena. Además de proteger los datos contra posibles problemas del sistema deben incluirse también procedimientos de chequeo que aseguren que los valores de los datos se ajusten a ciertas reglas prescritas de antemano. Estas pruebas pueden verificar las relaciones que existen entre varios valores de datos.

7. RESERVA Y SEGURIDAD

Los datos albergados en una base deben ser conservados con seguridad y reserva. La seguridad de los datos se refiere a la protección de éstos contra el acceso accidental o

intencional por parte de personas no autorizadas y contra su indebida destrucción o alteración.

La reserva se refiere al derecho de los individuos y organismos para determinar por sí mismos cuándo, cómo y en qué medida se permitirá la transmisión a terceros de la información que les concierne.

Aunque la tecnología necesaria para asegurar este secreto está estrechamente relacionada con la de la seguridad. En gran medida es un problema social. Para preservar el secreto de los datos personales, se necesitan recursos que escapan a la tecnología.

El analista de sistemas que se hace responsable de la seguridad, debe estar familiarizado con todas las particularidades del sistema, porque éste puede ser atacado con fines ilícitos desde muchos ángulos. A veces, se presta mucha atención a alguno de los aspectos del problema mientras se descuidan otros.

Hay siete aspectos esenciales para la seguridad de la base de datos:

1. La base de datos debe estar protegida contra el fuego, robo y otras formas de destrucción.
2. Los datos deben ser re-construibles (ya que nunca estamos exentos de accidentes).
3. Los datos deben poder ser sometidos a procesos de auditoría.
4. El sistema debe diseñarse con controles a prueba de intromisiones.
5. Los usuarios de la base de datos, deben ser sometidos a un proceso de identificación antes de tener acceso a ella.
6. El sistema debe ser capaz de verificar que cada acción ha sido autorizada.
7. Las acciones de los usuarios deben ser supervisadas, de tal modo que pueda descubrirse cualquier acción indebida.

Ver figura I.1.5.D



8. INTERFACE CON EL PASADO

Quando un organismo se decide a instalar un nuevo software de base de datos, es importante que éste pueda trabajar con los programas y procedimientos existentes y que los datos ya almacenados puedan ser convertidos en nuevas formas, ya que se ha invertido mucho dinero en los programas, procedimientos y datos existentes.

Es esencial, no obstante, que no se frene el avance de la tecnología de la informática por prestar una atención excesiva a esta compatibilidad con el pasado.

9. INTERFACE CON EL FUTURO

Más importante es la interface con el futuro. En el futuro, los datos y los medios de almacenamiento cambiarán de muchas maneras. Pero los cambios han resultado extremadamente costosos para los usuarios del procesamiento de datos. El enorme costo de cambios aparentemente triviales, ha frenado seriamente el desarrollo de nuevas aplicaciones. El costo ha sido el resultado de tener que re-escribir programas de aplicación, convertir datos y resolver demasiados problemas menores introducidos por los cambios. Uno de los objetivos más importantes en el diseño de una base de datos será, pues el de planearla de manera que se le pueda modificar sin necesidad de tener que alterar los programas de aplicación en uso; es decir, debemos tomar muy en cuenta, la independencia tanto física como lógica de los datos.

10. AFINACION

Cuando es indispensable asegurar el acceso en tiempo real de los datos, los usuarios del sistema se interesan en el tiempo de respuesta del mismo. En los sistemas de procesamiento por lotes, el interés se traslada al caudal de transacciones que el sistema admite o al tiempo que se tarda en despachar la carga de trabajo. Estos factores dependen del tiempo necesario para acceder a los datos necesarios y dependen también de la organización de éstos y de su localización en las unidades de almacenamiento. La

diferencia entre una organización adecuada y una inadecuada, se refleja en la enorme diferencia de tiempos de respuesta y de caudales de transacciones.

En ocasiones, por lo tanto, resulta necesario ajustar y hasta cambiar fundamentalmente la organización para almacenar los datos después que el sistema ha entrado en servicio y se han aclarado las pautas de uso.

En muchos casos, el uso de la base de datos evoluciona continuamente, a medida que más personas se van familiarizando con ella y se crean más programas de aplicación.

El ajuste de la organización del almacenamiento de datos, con el objeto de mejorar su desempeño se convierte así, en un proceso continuo.

Este proceso de ajuste de la base de datos, se llama afinación (*tuning*). Frecuentemente, el proceso de tuning ha servido para marcar algunos puntos importantes por ejemplo la diferencia de lo que es una aplicación rentable y lo que no lo es. El responsable de la afinación de la base de datos es el administrador o su grupo.

Sin un software apropiado, la afinación suele incurrir en costos inadmisibles en lo que se refiere al mantenimiento y la prueba de programas.

La correcta afinación tiene dos requisitos:

1. Independencia física de datos
2. Medios para supervisar automáticamente el uso de la base de datos con el fin de que se puedan hacer los ajustes necesarios.

11. MIGRACION DE DATOS.

Algunos datos se usan con mucha frecuencia y otros solo raramente. Es deseable almacenar los datos de uso frecuente de manera que resulte fácil y rápido acceder a ellos. Los datos de uso ocasional se almacenarán, en cambio, de manera que resulte más económico. En una oficina, la información que se usa diariamente se halla en los archiveros de las secretarías; la información que solo se usa accidentalmente se guardará probablemente en archivos muertos, de modo que no estorben y su almacenamiento no cueste mucho. El equivalente del sótano en la computadora podría ser la cinta magnética, mientras que los datos de uso frecuente se hallarán en discos, de modo que se les pueda leer siempre en una fracción de segundo.

A medida que cambia la popularidad de un conjunto de datos, será conveniente mudarlos dentro del almacén a posiciones más o menos accesibles, de acuerdo con su actividad. Este proceso de ajustar el almacenamiento de los datos de acuerdo con su popularidad se llama migración de datos. En algunos sistemas se hace automáticamente. En otros, la operación está a cargo de los programadores del sistema o del administrados de la base de datos. En ocasiones se le considera como parte del proceso de afinación de la base de datos.

12. SIMPLICIDAD.

Los medios que se utilizan para representar la vista general de los datos deben ser concebidos de manera simple y nítida. En muchos sistemas se utilizan punteros en la representación lógica con el fin de patentizar las relaciones que existen entre los diferentes ítems. Un problema que surge con el uso de los punteros lógicos es que, a

medida que se agregan más y más relaciones entre los ítems de datos, la colección de punteros resulta tan complicada que se hace difícil representar la vista lógica general de la base de datos con suficiente claridad. Aparecen punteros que indican punteros de punteros. En algunos casos, el abuso de los punteros múltiples en la representación de los datos para el usuario resulta engañoso. No se necesita, en realidad, tanta complejidad como la que se encuentra en ciertas estructuras lógicas de datos.

I.1.6 TIPOS DE ORGANIZACION

Existen tres aspectos en la organización de las bases de datos, estos tres aspectos son la organización de archivos, la organización lógica global de los datos, y la organización física del almacenamiento. Estos tres tipos de organización son a menudo muy diferentes.

La organización de archivos se preocupa de la vista de los datos tal como los percibe el programador de aplicaciones. El programador puede, por ejemplo, ver los archivos como un registro maestro con registros de detalle subordinados. El programador describe su propia vista de los archivos en su programa de aplicación.

La organización lógica global de los datos se preocupa por la organización general de la base de datos de la cual pueden derivarse múltiples organizaciones de archivos. Es la vista lógica de los datos, completamente diferente de la organización física del almacenamiento. Esta organización se describe apelando a un lenguaje para la descripción de datos, el cual es una parte del software (programa) de administración de datos.

La organización física del almacenamiento tiene por objeto la representación y la distribución física de los datos y la organización de éstos en las unidades de almacenamiento, es decir, tal y como quedan grabados en dichas unidades. Se ocupa de índices, punteros, cadenas, y otros medios de localización física de los registros y de las áreas de desborde y las técnicas utilizadas para insertar registros nuevos y borrar los que han perdido vigencia.

I.1.7 VENTAJAS DE UTILIZAR BASES DE DATOS

Podemos partir del hecho de que un sistema de bases de datos proporciona a toda empresa o institución un control centralizado de sus datos de operación, los cuales constituyen uno de los activos más valiosos. Esto contrasta con la situación que prevalece actualmente en algunas empresas, donde a menudo cada aplicación tiene sus propios archivos, de modo que los datos de operación se hallan muy dispersos y, por lo tanto, es probable que sean difíciles de controlar.

Lo anterior implica que en toda aquella empresa en la que se utilice un sistema de bases de datos debe existir una persona específica cuya responsabilidad central sea controlar los datos de operación; a esta persona se le denomina administrador de bases de datos y requiere tener un elevado nivel técnico y capacidad de entender e interpretar los requerimientos administrativos a nivel gerencial.

Las ventajas de utilizar bases de datos son las siguientes:

PUEDE REDUCIRSE LA REDUNDANCIA.

En sistemas que no usan bases de datos, cada aplicación tiene sus propios archivos. Esto a menudo origina enorme redundancia en los datos almacenados, así como desperdicio resultante del espacio de almacenamiento; por ejemplo, una aplicación de personal y otra de registros a educativos pueden poseer cada una un archivo que contenga información de departamentos de empleados, estos dos archivos pueden integrarse (para eliminar la redundancia) si el administrador de la base de datos está consciente de los requerimientos

de información para ambas aplicaciones, es decir, el administrador de la base de datos tiene el control global necesario.

No se pretende dar a entender que toda la redundancia por fuerza habrá de eliminarse. A veces hay sólidas razones comerciales o técnicas para mantener múltiples copias de los mismos datos. Sin embargo, en un sistema de bases de datos, la redundancia debe controlarse, es decir, el sistema debe estar al tanto de la redundancia y asumir la responsabilidad de propagar las actualizaciones.

PUEDE EVITARSE LA INCONSISTENCIA.

Esto tiene mucho que ver con el punto anterior. Supóngase que un hecho específico del mundo real, por ejemplo, el empleado E3 trabaja en el departamento D8, se representa por dos entradas distintas en la base de datos, y que el sistema no está al tanto de esta duplicación (en otras palabras, la redundancia no está controlada). Habrá entonces algunas ocasiones en que las dos entradas no concuerden (es decir, cuando una y solo una de ellas se haya actualizado). En tales circunstancias se dice que la base de datos es inconsistente.

Desde luego, una base de datos que se halle en estado de inconsistencia puede proporcionar información incorrecta o contradictoria.

No hay duda de que si el hecho específico se representa por una sola entrada (es decir, si la redundancia se elimina), tal inconsistencia no puede ocurrir. Por otra parte, si la redundancia no se suprime, pero se controla, entonces se puede garantizar que la base de datos no será inconsistente, para ello el usuario deberá asegurar que todo cambio hecho a una de las dos entradas se efectúe de manera automática en la otra. Este proceso se

denomina propagación de actualizaciones, donde el término actualización se utiliza para abarcar todas las operaciones de creación, supresión y modificación.

LOS DATOS PUEDEN COMPARTIRSE.

No sólo significa que las aplicaciones existentes pueden compartir los datos de la base de datos, sino también que es factible desarrollar nuevas aplicaciones que operen con los mismos datos almacenados. En otras palabras, las necesidades de datos de las nuevas aplicaciones pueden atenderse sin tener que crear nuevos archivos. Esto se logra debido a que el administrador de la base de datos tiene un panorama general de los requerimientos de la empresa. Ver figura I.1.7.A



PUEDEN HACERSE CUMPLIR LAS NORMAS ESTABLECIDAS.

Con un control central de la base de datos, el administrador de la base de datos puede garantizar que se cumplan todas las formas aplicables a la representación de los datos. Las normas aplicables pueden comprender la totalidad o parte de lo siguiente: normas de la compañía, de instalación, departamentales, industriales, nacionales o internacionales. Es muy deseable unificar los formatos de los datos almacenados como ayuda para el intercambio o migración de datos entre sistemas.

PUEDEN APLICARSE RESTRICCIONES DE SEGURIDAD.

Al tener jurisdicción completa sobre los datos de operación, el administrador de la base de datos puede:

- (a) asegurar que el único medio de acceder la base de datos sea a través de los canales establecidos.
- (b) definir controles de autorización para que se apliquen cada vez que se intente el acceso a datos sensibles.

Diferentes controles puedan establecerse para cada tipo de acceso (recuperación, modificación, supresión, etc.) a cada parte de la información de la base de datos. Es importante hacer notar que sin dichos controles la seguridad de los datos, en realidad, corre mayor peligro en un sistema de base de datos que en un sistema tradicional, en donde los archivos se encuentran dispersos.

Ver figura I.1.7.B

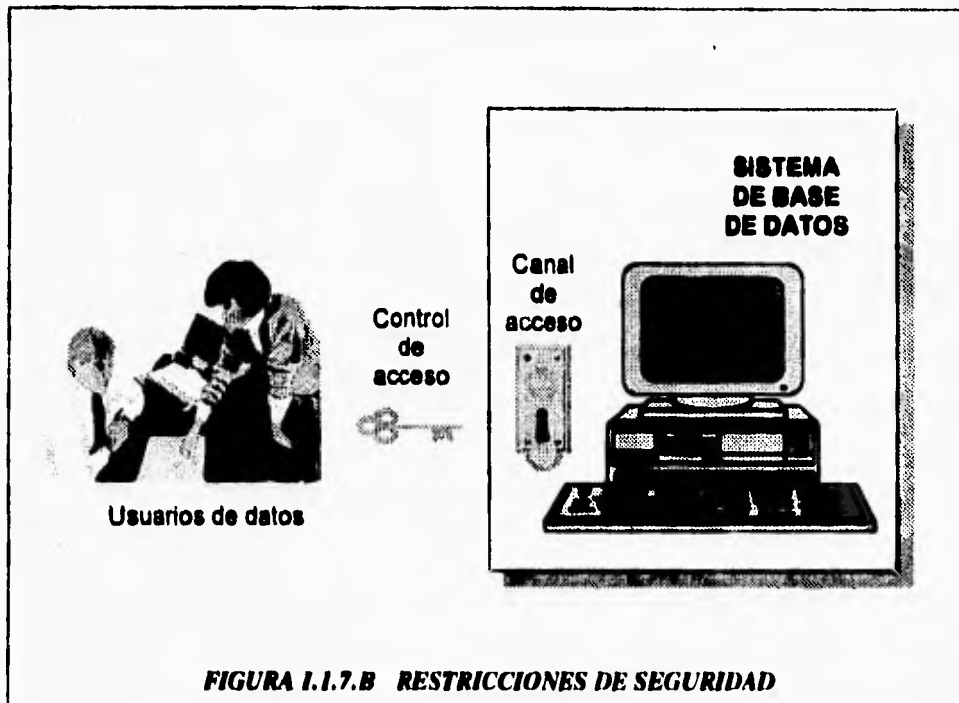


FIGURA 1.1.7.B RESTRICCIONES DE SEGURIDAD

PUEDE CONSERVARSE LA INTEGRIDAD.

La finalidad de la integridad es garantizar que los datos de la base de datos sean exactos. La inconsistencia entre dos entradas que representan al mismo hecho es un ejemplo de falta de integridad (que por supuesto, sólo ocurre si existe redundancia en los datos almacenados). Aun cuando la redundancia se elimine, empero, la base de datos puede contener aún datos incorrectos; por ejemplo, un empleado puede aparecer como si hubiera trabajado 200 horas semanales, o la lista de números de empleado de un departamento específico puede incluir el número de un empleado inexistente. El control centralizado de la base de datos ayuda a evitar estas situaciones en la medida de lo

posible, y puede permitir al administrador de la base de datos definir procedimientos de validación que habrán de ejecutarse cada vez que se intente una operación de actualización (entendiéndose por actualización las operaciones de modificación creación y supresión). Es conveniente señalar que la integridad de los datos es más importante en un sistema de bases de datos que en un sistema de archivos privados, precisamente porque el primero se comparte y porque sin procedimientos de validación adecuados es posible que un programa con errores genere datos incorrectos que afecten a otros programas que utilicen esa información.

PUEDEN EQUILIBRARSE LOS REQUERIMIENTOS CONTRADICTORIOS.

Cuando se conocen los requerimientos globales de la empresa, en contraste con los requerimientos de cualquier usuario individual, el administrador de la base de datos puede estructurar el sistema de base de datos para brindar un servicio que sea el mejor para la empresa en términos globales. Por ejemplo, puede elegirse una representación de los datos almacenados que ofrezca rápido acceso a las aplicaciones más importantes a costa de un desempeño de menor calidad en otras aplicaciones.

I.1.8 BASES DE DATOS RELACIONALES.

El análisis de datos debe estar basado en un modelo formal que cumpla con los siguientes objetivos:

1. Puede ser usado para identificar los requerimientos de usuario y presentarlos de forma tal que sean fácilmente de entender para los usuarios así como para los programadores de sistemas.
2. Que pueda ser fácilmente convertido a una implementación técnica.
3. Que proporcione reglas y criterios para la estructuración de la base de datos.

El modelo relacional fue uno de los primeros en cumplir estos tres objetivos, que son, las características esperadas en un buen diseño. Tales relaciones se dice que están en forma normal.

Estructura básica.

Para lograr el primer objetivo (identificar los requerimientos del usuario) el modelo debe servir como un medio de comunicación entre los usuarios y los departamentos de sistemas, dándoles una interface que puedan entender fácilmente. La independencia de esta interface para su implementación en computadora es importante. Tal independencia elimina la temprana consideración de restricciones impuestas por dispositivos físicos, de este modo se producen especificaciones de usuario que enfatizan las necesidades lógicas de los usuarios.

El modelo relacional utiliza tablas para proporcionar esta interface, en el cual el modelo de datos se especifica en un conjunto de tablas o relaciones. Normalmente, una especificación completa consiste de varias relaciones.

La representación tabular de relaciones satisface el primer objetivo del análisis de datos.

El segundo objetivo (la conversión a una implementación física), también es satisfecha por el modelo relacional.

Un acercamiento obvio, es implementar directamente en una maquina el modelo relacional. Para hacer esto, se debe tener disponible, en un sistema de cómputo, un sistema manejador de bases de datos que soporte el modelo relacional. La declaración directa de un conjunto particular de relaciones se puede hacer utilizando el lenguaje de definición proporcionado por el sistema manejador de bases de datos.

Obviamente, no es necesario tener un sistema manejador de bases de datos relacional para poder usar el modelo relacional para especificar requerimientos de usuario.

Una especificación relacional, también se puede convertir a otras estructuras físicas. Por ejemplo, suponiendo que se va a utilizar un sistema de archivos convencional para la implementación. En este caso, cada relación (tabla) se puede convertir en un archivo, cada columna a un campo y cada renglón a un registro. Por lo tanto se satisface el segundo objetivo.

El tercer objetivo es tener la capacidad de juzgar la calidad del diseño.

El modelo relacional proporciona el siguiente criterio para buenas estructuras de datos lógicas:

- 1. Cada hecho debe ser almacenado una sola vez en la base de datos.**
- 2. La base de datos debe ser consistente con sus operaciones.**
- 3. La base de datos debe ser inmune a los cambios.**

El hecho de almacenar únicamente una vez los datos, no solamente elimina la redundancia sino que también da consistencia a la base de datos. El tercer criterio es una consecuencia del ambiente en el que se encuentra la base de datos. Normalmente, el ambiente está cambiando constantemente y en consecuencia, la base de datos debe ser rediseñada constantemente para seguir los cambios de los requerimientos de los usuarios.

El modelo relacional proporciona también lenguajes para acceder las relaciones. Tales lenguajes deben satisfacer ciertos criterios, de los cuales, dos de los más importantes son su poder selectivo y su interface natural entre el modelo y los usuarios.

Los lenguajes relacionales más recientes, se concentraron en el poder selectivo. Codd definió las bases de los lenguajes relacionales como álgebra relacional y cálculo relacional. El cálculo relacional es una forma de predicados de cálculo diseñado especialmente para el modelo relacional y es usado para medir el poder selectivo de los lenguajes relacionales.

Codd también planeó el lenguaje ALPHA, que está basado en el cálculo relacional, así como en el álgebra relacional, el cual mostró ser relacionalmente completo. El álgebra

relacional se puede ver como un lenguaje de procedimientos para procesar relaciones, construyendo nuevas relaciones a partir de relaciones existentes. Ambos lenguajes contienen expresiones que se refieren a relaciones, pero éstas expresiones son comúnmente orientadas matemáticamente y no tienen una interface adecuada para un usuario práctico. Se han propuesto otros lenguajes que se adhieren a todos los principios del modelo relacional, pero al mismo tiempo mejoran la interface con el usuario.

1.2 DISEÑO DE BASES DE DATOS

1.2.1 ETAPAS DEL DISEÑO

Desde de la introducción de los sistemas manejadores de datos ha habido una evolución gradual en el proceso del diseño de bases de datos. Existen dos propuestas a las que se les denomina "Top-down" y "Bottom-up"; son muy comunes las combinaciones de estas dos propuestas, y de hecho, las definiciones de lo que constituye un diseño puramente top-down o bottom-up son muy divergentes.

Fases y pasos del diseño Top-Down

El proceso de diseño se puede visualizar en tres fases:

- **Modelado de datos**
- **Diseño lógico de la base de datos**
- **Diseño físico de la base de datos**

La fase de modelado de datos, se puede realizar a través de toda la organización o de una división de la organización, y pueden considerarse todos sus procesos productos, requerimientos, etc. A este nivel, solamente se requiere del usuario, los requerimientos generales, y no es necesario llegar a un gran detalle.

El diseño lógico de la base de datos, se puede realizar en una aplicación básica o puede estar centrada alrededor de una entidad especialmente importante del negocio, tal como los empleados o los productos; para esta fase, los elementos de datos individuales se

deben definir normalmente en vistas de usuarios, pantallas y formatos de reporte requeridos por los usuarios del sistema.

Finalmente, el diseño físico, implementará la base de datos bajo uno o más sistemas manejadores de bases de datos que proporcionen ventajas de desempeño; este nivel de diseño deberá estar orientado a aplicaciones específicas y requiere del usuario criterios de uso y rendimiento.

Las tres partes antes mencionadas, se pueden dividir en los siguientes siete pasos:

1. **Modelo de datos**
 - **Desarrollo del modelo entidad-relación**
 - **Desarrollo del modelo conceptual de datos**
 - **Definición de los elementos de datos llave**
2. **Diseño lógico de la base de datos**
 - **Definición de los elementos de datos no llave**
 - **Normalización de las entidades de datos**
3. **Diseño físico de la base de datos**
 - **Análisis de la utilización de las entidades de datos**
 - **Desarrollo del modelo físico de datos**

Modelo de Datos

El objetivo de esta fase, es desarrollar una estructura, la cual servirá como diseño a largo plazo o como guía para los detalles que vendrán más tarde. Es necesario, conocer por ejemplo, la importancia de los vendedores de la empresa y de ser importante, como son identificados dichos vendedores individualmente, y cuál es su relación en la empresa con otras cosas tales como contratos, pagos, ordenes, recibos, etc. Es especialmente

importante, tener esta visión global cuando se está desarrollando una aplicación que solamente afecte información de vendedores, de lo contrario, información importante para la organización, puede estar subordinada a información más importante dentro de la aplicación. El desarrollo del modelo de datos, se puede obtener en tres pasos, y puede proporcionar una estructura a largo plazo para el diseño.

Modelo Entidad-Relación

Un modelo entidad-relación, es un modelo que consiste en entidades del negocio, y sus relaciones. Lo veremos más a detalle en incisos posteriores.

Desarrollo del Modelo Conceptual de Datos

El siguiente paso en el modelo de datos, es convertir el modelo entidad-relación, en un modelo que esté más orientado a las estructuras de datos. El modelo conceptual de datos, es un modelo que consiste en entidades de datos y sus relaciones.

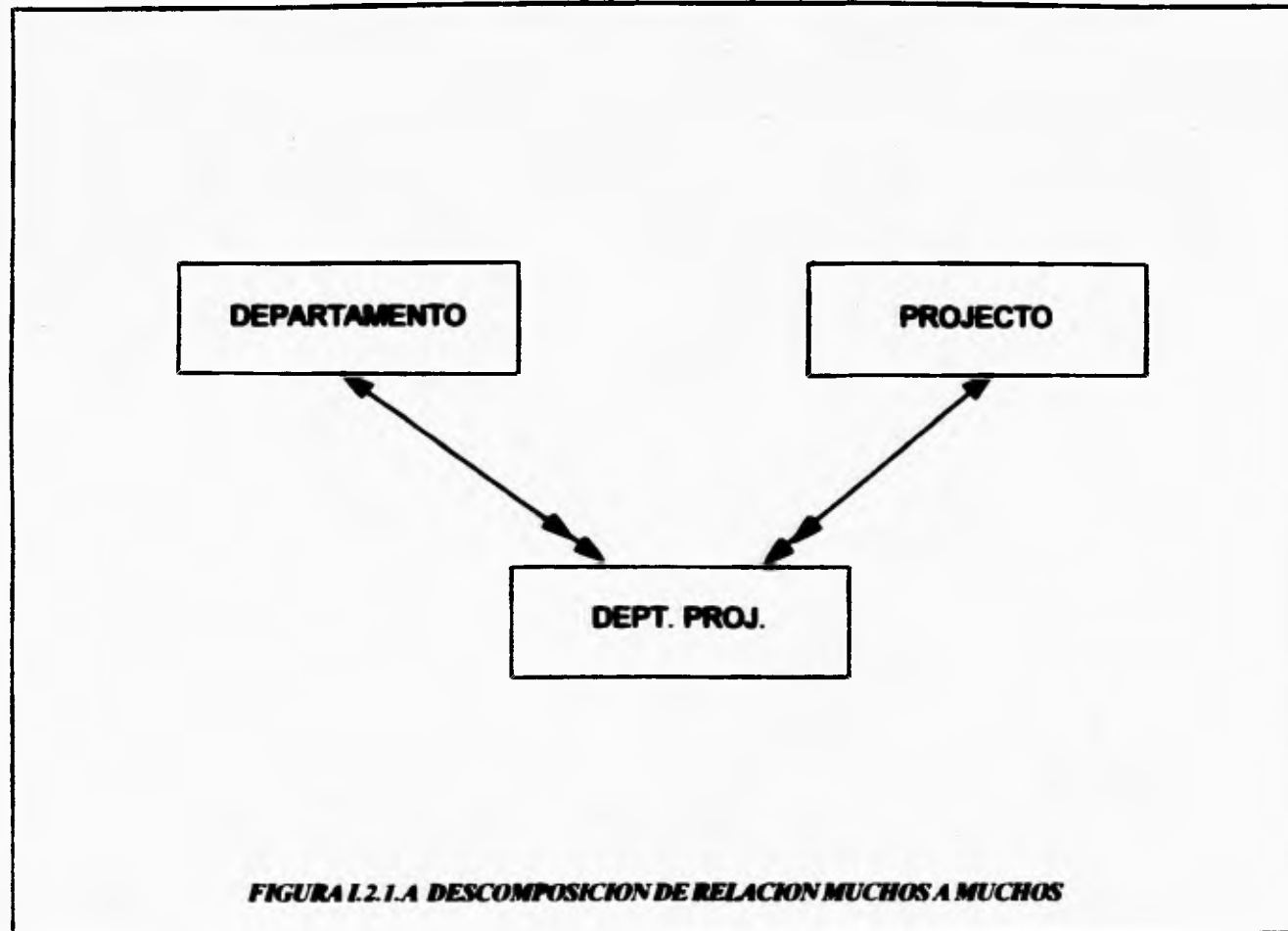
Una entidad de datos, es una colección de datos que contiene, entidades del negocio o relaciones. Dado que ambas, las entidades de negocio y las relaciones, están documentadas por un modelo entidad-relación, este puede ser convertido en el modelo conceptual de datos, por medio de un proceso en línea recta.

Existen tres tipos distintos de entidades de datos:

- **Medulares**
- **Asociaciones**
- **Características**

El primer paso para crear el modelo conceptual de datos, es crear una entidad de datos medular para cada entidad de negocio en el modelo entidad relación. Dado que la entidad de negocio es, por definición, algo de lo que queremos mantener datos, necesitaremos una entidad de datos para esos datos. Teniendo identificados los datos medulares, podemos llevar las relaciones uno a uno y uno a muchos del modelo entidad relación hacia el modelo conceptual de datos. Sin embargo, las relaciones muchos a muchos se deben descomponer. Para cada relación muchos a muchos se crea una asociación de entidad de datos, hay una relación uno a muchos entre cada dato medular y su asociación; por ejemplo, si había una relación muchos a muchos entre un departamento y un proyecto en el modelo entidad-relación, la porción correspondiente del modelo conceptual de datos, debe verse como en la figura 1.2.1.A.

De aquí que el proceso, es una conversión directa del modelo entidad-relación, después de que esto se ha completado, se puede añadir al modelo conceptual, información más detallada y orientada a los datos. Esto se hace examinando cada una de las entidades de datos que han sido definidas para ver los elementos de datos o grupos de elementos repetitivos. Si se encuentra alguno, se divide en una característica separada de la entidad de datos. Hay una relación uno a muchos entre la entidad de datos de la cual fue extraída la información repetitiva y la nueva característica formada. Es decir, un elemento o grupo de elementos repetitivos, son con frecuencia una función de si hay una necesidad de datos relacionados con el tiempo, por ejemplo, un dato histórico. Mientras las características son las entidades de datos más difíciles de definir en esta etapa, y las destinadas a ser aumentadas posteriormente, son también las menos difíciles de aumentar.



Definición de elementos dato llave

La fase del modelo de datos, termina con la identificación de cada entidad de datos, las llaves primarias y las llaves foráneas. Las llaves primarias se deben identificar primero y así las llaves foráneas pueden ser determinadas de las relaciones existentes en el modelo conceptual de datos.

Las llaves primarias varían entre entidades medulares, asociaciones o características. Las llaves medulares, vienen directamente de los identificadores de las entidades de negocio. La asociación generalmente tiene una llave que consiste en llaves de las dos entidades de datos que están asociadas (número de departamento concatenado con el número de proyecto en cualquier orden).

Generalmente, una característica tiene una llave primaria, la llave de la entidad de datos de donde se tomó el grupo repetitivo, más otro identificador.

Las llaves foráneas, reflejan las relaciones del modelo conceptual de datos. Para relaciones uno a uno, una de las entidades de datos debe contener la llave de la otra: ambas pueden tenerla, pero es innecesario. Para relaciones uno a muchos, la entidad de datos en los muchos lados de la relación, debe contener la llave de la entidad de datos en uno de los lados. La llave de una entidad de datos contenida en otra entidad de datos, se conoce como llave foránea en la entidad de datos en la cual está contenida.

DISEÑO LOGICO DE BASES DE DATOS

El objetivo de esta fase, es producir entidades de datos normalizadas conteniendo todos los elementos de datos relevantes. Para lograr esto, se deben identificar todos los elementos de datos para una entidad de datos dada, o todos aquellos que necesitan para

cierta aplicación. Al tener la última aproximación, las entidades de datos pueden estar incompletas, a pesar de que los datos que se conocen estén normalizados. Hasta aquí estamos en el modelo conceptual, no en el modelo físico, por lo tanto adherir elementos a la última hora no desencadena ningún problema. Y aunque este modelo esté físicamente implementado, los elementos adicionales se pueden aumentar al sistema por medio de un sistema manejador de bases de datos que nos facilitará la adición, o se puede aumentar como un objeto físico separado (archivo, tabla, segmento, etc.).

Definición de los Elementos Datos No Llave

Para determinar los elementos no llave, se necesita nueva y más detallada información de la comunidad de usuarios. Generalmente, vendrá de un nivel operativo de la organización, y consistirá en documentación de vistas de usuario: reportes, pantallas de salida, etc.

Cada elemento dato es identificado y colocado en la entidad de datos más apropiada. Si no existe una entidad de datos apropiada, se deberá crear una y aumentarla en el modelo conceptual de datos. Estas entidades de datos nuevas pueden ser de dos tipos: secundarias (entidades medulares o características adicionales). Por entidades medulares secundarias entendemos una entidad de datos que se sostiene a sí misma, es decir, que se necesita para cierto proceso que tiene pocos elementos de datos, normalmente de naturaleza estática. Este es el tipo de entidades de datos, que se convertirían en una tabla de referencia cuando estén físicamente implementados.

Una pregunta frecuente en el proceso de análisis de las vistas de usuario es acerca de que si los datos derivables deben ser guardados. Hasta aquí es el modelo conceptual, como ya mencionamos, no la implementación física, y esta es una pequeña razón para no

guardarlos. Pero, existen dos razones para guardarlos. La primera, documentación de esos elementos tal como aparecen en los reportes, por ejemplo, puede ser necesitada si los elementos no se guardan en cualquier base de datos física. Segundo, si hay documentación en este momento, puede ser analizada cuidadosamente como parte del diseño físico de la base de datos, hasta la pregunta de que hacer con ellos, que sería un punto esencial en el rendimiento.

Normalización de Entidades de Datos

En el diseño "bottom-up" de bases de datos, la normalización es un proceso clave. Y en el procedimiento "top-down", si el paso prioritario se hace correctamente, las entidades de datos serán normalizadas. Sin embargo la normalización puede ser utilizada para chequeo final, y en todo caso es importante entender que significa normalización y por qué es deseable en el diseño de bases de datos.

Normalizar, significa tener las entidades de datos en su forma más simple, con los elementos dato agrupados dentro de grupos separados lógicamente. Esto maximiza la flexibilidad y minimiza los problemas en términos de restricciones del negocio e inconveniencias técnicas. Entendiendo esto y siendo capaces de diferenciar entre restricción de negocio e inconvenientes técnicos, es de primordial importancia considerar la desnormalización por razones de rendimiento.

DISEÑO FÍSICO DE DATOS

El objetivo de esta fase, es producir un modelo físico, para un sistema de manejo de bases de datos específico basado en características de uso y criterios de rendimiento. La información nueva que se necesita de los usuarios para completar esta fase es, primero,

uso y volumen de datos (frecuencia de transacciones, ocurrencias de entidades de datos) y dos, prioridad de rendimiento (tiempo de respuesta, o prioridades de ejecución batch).

El diseño físico de bases de datos, consiste en dos pasos: el primero, colección y análisis de uso y rendimiento de datos, y entonces el acceso a dichos datos, junto con el sistema manejador de bases de datos; y el segundo se refiere al modelo conceptual de datos normalizado para producir el modelo físico.

Análisis de uso de entidades de datos

Para comenzar el análisis de uso, la primera parte de datos que se necesita, es el número de ocurrencias de cada entidad de datos. Esto, es importante, por ejemplo, si tenemos 200 departamentos y 2000 empleados, sabemos que una transacción que proporciona una lista de empleados para un número de departamento dado, tendrá que hacer un promedio de diez accesos. La mayoría de estos números, serán dados como parte de los requerimientos de la aplicación, pero cuando no se proporcionan, o cuando se proporcionan dos o más números conflictivos, es necesario establecer una resolución.

Después, las especificaciones para cada una de las transacciones y programas batch, deben ser analizadas para determinar para cada una: la frecuencia de acceso para cada entidad de datos y el argumento de búsqueda utilizado para cada acceso. Se suman los resultados de cada programa, para obtener el número de accesos requeridos para en ejecución de programa, y se hace lo mismo por cada entidad de datos, para obtener la frecuencia de accesos por cada argumento de búsqueda. El análisis del programa también proporciona información acerca de las afinidades de acceso, es decir, las entidades de datos que son usadas al mismo tiempo. Esta información sobre la frecuencia es importante, para obtener un rendimiento óptimo en el paso final.

Desarrollo del modelo físico de datos

Antes de seleccionar el sistema manejador de base de datos, hay algunas consideraciones de rendimiento y son los que deben tomarse en cuenta para cualquier sistema. Dos de estos, datos que se derivan y la denormalización, tienen que ser discutidos brevemente. Sin embargo, en este punto la información requerida está finalmente disponible para tomar decisiones inteligentes.

La decisión de calcular los datos derivables cuando se requieren y almacenarlos o calcularlos cuando se necesiten, en ocasiones representa una minimización esencial en el costo total. Los factores que deben tomarse en cuenta son:

- **cuántos accesos adicionales tomaría calcularlos (al momento de una actualización)**
- **cuántos datos adicionales tendrían que ser almacenados**
- **cuántos accesos adicionales se necesitarían para calcularlos al momento que se necesiten**
- **Las frecuencias relativas de la afectación de actualizaciones contra la frecuencia de necesidad**

Es necesario, determinar por cada caso que decisión produce el menor costo total. Sin embargo, puede haber otras prioridades adicionales, que deben ser consideradas. Por ejemplo, podría tener un precio y un costo total adicional almacenar algunos datos derivables para tener mucho mejor tiempo de respuesta para una transacción de consulta adicional.

Similarmente, desnormalización, es una pregunta sobre el rendimiento. Sin embargo, en este caso, la primera pregunta que hacer sería: ¿es la desnormalización causa de una restricción del negocio? Si lo es, no debemos denormalizar. Por otra parte, si no lo es, el alcance de las inconveniencias técnicas, debe tomar peso como un beneficio de rendimiento potencial para reducir los accesos. Existen dos posibilidades de desnormalización, que pueden ser consideradas. Por ejemplo, teniendo un item name en item y en order item, podemos eliminar item por estar combinado con order item, o podemos mantener item y repetir name en order item para salvar los accesos a esas transacciones que necesitan item names por orders. Esto último, proporciona mejor rendimiento a un pequeño costo adicional, y no causaría una restricción de negocio.

Finalmente, se debe considerar el bloqueo potencial. En el paso anterior determinamos el número de ocurrencias y la frecuencia de acceso para cada entidad de datos. Bloquear una entidad con un número de ocurrencias bajo y un número de accesos alto, causa una potencial contención de acceso. Si se da este caso, la solución más común es incrementar el número de ocurrencias de la entidad de datos.

Teniendo listo el analisis anterior y habiendo realizado algunos posibles cambios a las entidades de datos, se puede desarrollar un modelo para un sistema manejador de bases de datos seleccionado. Las reglas varian dependiendo de cual sistema manejador de bases de datos se seleccionó.

I.2.2 MODELO ENTIDAD RELACION

Este modelo se basa en una percepción del mundo real, y consiste en un conjunto de elementos básicos denominados "entidades" y de las "relaciones" que existen entre dichos elementos.

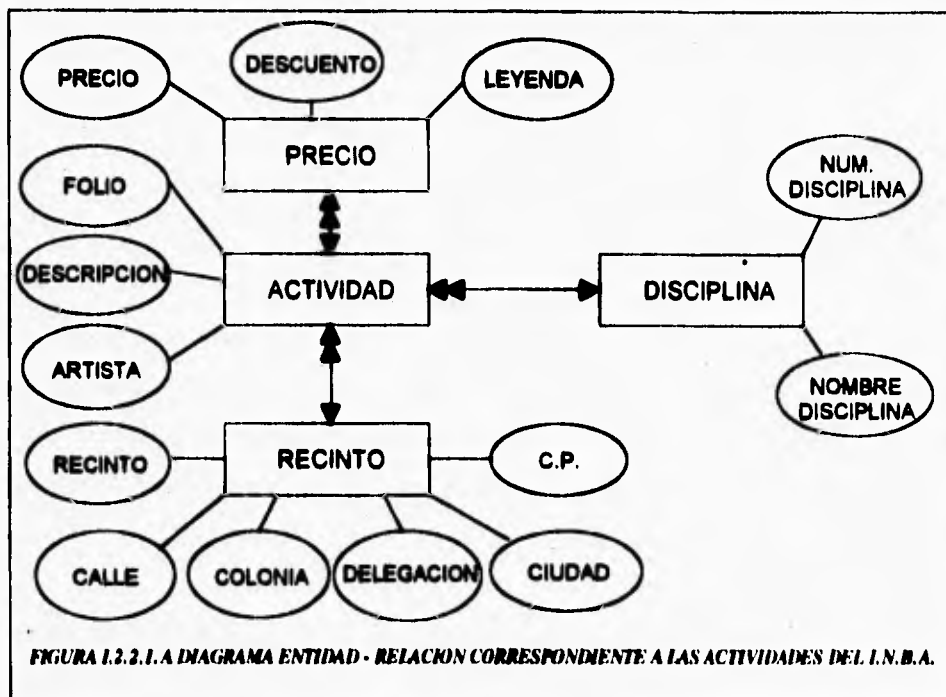
El objetivo de este modelo es el facilitar el diseño de bases de datos permitiendo la especificación de un esquema, dicho esquema representa la estructura lógica de la base de datos.

I.2.2.1 DIAGRAMA ENTIDAD RELACION.

El modelo entidad relación, representa las "entidades" y las "relaciones" por medio del diagrama Entidad - Relación (E-R), éste diagrama es la mejor forma de expresar la vista global de la base de datos.

Existe una terminología desarrollada para realizar los diagramas E-R.

En un diagrama E-R, las entidades se representan con un rectángulo, y las relaciones se representan con líneas con punta de flecha en los extremos. Por ejemplo, en la figura I.2.2.1.A ACTIVIDAD, DISCIPLINA, PRECIOS y RECINTO son entidades, y las flechas representan las relaciones entre las entidades. Una punta de flecha en un extremo y dos puntas de flecha en el otro extremo indica que el tipo de relación entre las entidades es de uno a muchos, es decir, en el diagrama podemos ver que en un recinto se pueden llevar a cabo más de una actividad, y que más de una actividad pueden pertenecer a una disciplina y una actividad puede tener varios precios.



Las entidades y las relaciones poseen propiedades, dichas propiedades pueden ser expresadas en términos del valor de sus atributos. Opcionalmente estas propiedades se pueden representar en el diagrama E-R por medio de círculos y los valores de éstos se expresan en los arcos que van de la entidad a su atributo correspondiente. Por cada función de una aplicación se dibuja un diagrama de E-R para describir los requerimientos de datos.

1.2.2.2 ENTIDADES Y CONJUNTOS DE ENTIDAD.

Una entidad es un objeto o elemento real que se puede distinguir de cualquier otra entidad. Por ejemplo, El lago de los cisnes con un número de folio específico es una entidad, ya que el número de folio la identifica en forma única.

Un conjunto de entidad contiene entidades del mismo tipo. Por ejemplo, el conjunto de las actividades coordinadas por el I.N.B.A. puede ser definido como el conjunto de entidad denominado actividades. De la misma manera el conjunto de entidad Recintos debe representar el conjunto de todos los recintos en los que se llevan a cabo las actividades coordinadas por el I.N.B.A.

Una entidad se representa por un conjunto de atributos. Los posibles atributos del conjunto de entidad Recintos son: Recinto, calle, colonia, delegación, ciudad, código postal, clave lada, teléfono 1, teléfono 2. Los posibles atributos para el conjunto de entidad disciplinas son: Número de disciplina, nombre de la disciplina. Para cada atributo hay un conjunto de valores permitidos, a este conjunto se le llama dominio del atributo; por ejemplo, el dominio del atributo recinto debe ser el conjunto de cadenas de letras de una longitud específica, de manera similar, el dominio del atributo número de disciplina debe ser el conjunto de todos los números enteros positivos.

Formalmente, un atributo es una función que mapea a un conjunto de entidad dentro de un dominio. De esta forma cada entidad esta descrita por un conjunto de parejas, de donde la entidad disciplinas está descrita por el conjunto:

$\{(Número\ de\ disciplina, l), (nombre\ de\ disciplina, danza)\}$

mediante el cual la entidad define a la entidad llamada danza con un número de disciplina l.

1.2.2.3 RELACIONES Y CONJUNTOS DE RELACION.

Una relación es una asociación entre varias entidades, por ejemplo, se definió una relación que asocia a la actividad El lago de los cisnes con la disciplina Danza, ésta relación implica que El lago de los cisnes pertenece a la disciplina danza.

Un conjunto de relación está formado por un conjunto de relaciones del mismo tipo.

Formalmente, es una relación matemática sobre $n \geq 2$ conjuntos de entidad. Si E_1, E_2, \dots, E_n son conjuntos de entidad, entonces, un conjunto de relación R es un subconjunto de:

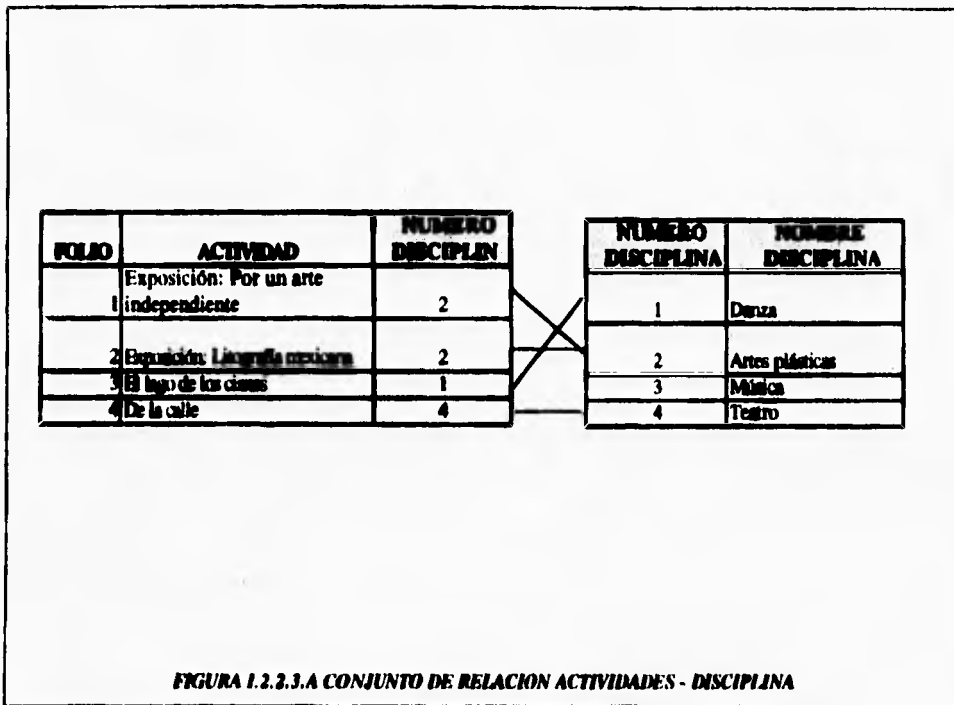
$$\{(e_1, e_2, \dots, e_n) \mid (e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n)\}$$

donde (e_1, e_2, \dots, e_n) es una relación.

En la figura 1.2.2.3.A se define la relación de las entidades Actividades - Disciplinas para denotar la asociación entre las actividades y disciplinas existentes.

Esta relación es un ejemplo de conjunto de relación binaria, es decir, que solamente se involucran dos conjuntos de entidad. Muchos de los conjuntos de relación en una base de datos son binarios. Cuando existen relaciones de más de dos entidades, es muy aconsejable que estas relaciones se descompongan en relaciones del tipo binario para facilitar el modelado de la base de datos.

La función que juega una entidad en una relación se denomina rol. Los roles son normalmente implícitos y normalmente no se especifican, sin embargo, se utilizan cuando el manejo de una relación necesita ser clarificada; por ejemplo, cuando el conjunto de entidad no está bien definido.



1.2.2.4 TIPOS DE RELACION.

Un esquema implementado por el modelo E-R define algunas restricciones que deben conformar el contenido de la base de datos. Una de estas restricciones es el tipo de relación entre entidades, éstas relaciones pueden comprenderse como un mapeo cardinal que expresa el número de entidades a la cual otra entidad puede ser asociada por medio de una relación.

Este tipo de mapeos son los mas utilizados en la descripción de conjuntos de relación binaria, además, algunas veces ayudan a la descripción de conjuntos de relación que involucran más de dos conjuntos de entidad.

Para un conjunto de relación binaria entre los conjuntos de entidad Actividades - Disciplinas, el mapeo cardinalidad debe ser alguno de los siguientes:

Uno a uno

Uno a muchos

Muchos a uno

Muchos a muchos

En la figura 1.2.2.1.A podemos observar que la relación entre las entidades Actividad y Recintos es muchos a uno, ya que diferentes actividades se pueden presentar en el mismo recinto.

Adicionalmente existen dependencias entre entidades; por ejemplo, la entidad precio depende de la entidad actividad y si la entidad actividad se eliminara, entonces también todas las entidades precio relacionadas con la entidad actividad deberán ser eliminadas.

Por lo anterior la entidad dominante es Actividad, mientras que Precio es la entidad subordinada.

En notación matemática se puede definir de la siguiente manera:

Sea "A" el conjunto de entidad dependiente (Precios) con atributos descriptivos a_1, a_2, \dots, a_n . Sea "B" el conjunto de entidad dominante (Actividades) sobre el cual "A" es dependiente.

Si la llave primaria de "B" consiste de los atributos b1, b2, ..., bs, representamos al conjunto de entidad "A" por una tabla llamada "A" con una columna por cada atributo del conjunto:

$$\{ a1, a2, \dots, ar \} \cup \{ b1, b2, \dots, bs \}$$

Como podemos ver en la figura 1.2.2.1.A, el conjunto de entidad precios posee tres atributos: precio, descuento y leyenda. La llave primaria del conjunto de entidad actividad por medio de la cual precio es dependiente de folio, por lo que podemos expresar al conjunto entidad precio mediante una tabla de cuatro columnas denominadas folio, precio, descuento y leyenda que corresponde a la unión de sus atributos y la llave primaria del conjunto de entidad dominante actividades.

1.2.2.5 LLAVES PRIMARIAS

Es muy importante que durante el modelado de la base de datos se distingan perfectamente las entidades y las relaciones, y la diferencia entre ellas debe ser expresada en términos de sus atributos. Para realizar estas distinciones se debe asignar una llave a cada conjunto de entidad. Esta llave se forma con un conjunto de uno o más atributos y nos permite identificar de manera única a una entidad en el conjunto de entidad.

Por ejemplo, el atributo folio del conjunto de entidad actividades, nos permite identificar y distinguir de forma única a una entidad de cualquier otra entidad del conjunto, por esta razón el atributo folio es una llave.

En algunos casos en los que la llave está formada por una combinación de dos o más atributos, la llave puede llegar a ser demasiado grande, por lo que se deberá minimizar, a éste tipo de llave se le llama llave candidata.

Es posible que una entidad no posea la cantidad suficiente de atributos para poder formar una llave primaria. por ello, ésta deberá ser dependiente de la entidad dominante que posea la llave primaria. esto se relaciona con la dependencia existencial que se mencionó en el punto 1.2.2.4.

1.2.3 ESQUEMAS Y SUBESQUEMAS.

Si la función de la base de datos fuese meramente la de almacenar datos, su organización resultaría por demás simple. En efecto, la mayoría de sus complejidades surgen por el hecho de que también tiene que indicar las relaciones que existen entre los diversos ítems de datos que almacena.

Antes de intentar la descripción de las representaciones físicas de las relaciones entre datos, debemos considerar la manera cómo los usuarios de la base (usuarios terminales y programadores de aplicaciones) ven esas relaciones.

Hay una diversidad de maneras de representar lógicamente las relaciones entre los datos. Algunas son buenas. Otras resultan confusas y conducen a errores de interpretación. Las hay que son demasiado limitadas, por cuanto no permiten la representación de algunas de las relaciones que existen en la realidad. Algunas son demasiado rígidas porque no permiten que se expanda y cambie la vista de los datos, en la medida que la base misma debería de variar y desarrollarse. El presente punto tratará de explicar las representaciones lógicas de los datos.

Reconocida la capacidad del software de administración de datos para separar la organización física de los datos de la vista que tiene de ellos el usuario, u organización *lógica*, la vista del usuario debería, al menos en teoría, formularse sin preocupación alguna por la representación física. La vista que el usuario tiene de los datos debería adoptar siempre la forma que resulta más conveniente para él y sus asociados (en el presente y en el futuro), quedando a cargo del software de administración de datos el hacer la *traducción* entre esta organización lógica y la organización física, cualquiera que sea la elegida, teniendo en cuenta la eficiencia del sistema.

En la práctica, la industria de la computación no ha llegado aún a desarrollar un software de administración perfecto, de modo que siempre tendrá que llegarse a soluciones de compromiso entre las descripciones lógicas y físicas, como veremos.

ESQUEMAS.

Es preciso describir la organización de los datos de una manera formal. Las descripciones, lógica y física, de la base de datos son indispensables para el software de administración con el fin de extraer de la base de datos que pide el usuario.

Llamamos esquema a la descripción lógica de la base de datos.

El esquema es un diagrama de los tipos de datos que se usan. Proporciona los nombres de las entidades y sus atributos y especifica las relaciones que existen entre ellos. Es un marco en el que se inscriben los valores de los ítems de datos. Como en los tableros de los aeropuertos que indican las horas de llegada y partida, el esquema permanece, mientras que los valores inscritos en él varían de un momento a otro. La figura 1.2.3.A podría ser considerada como un esquema si se removieran de su interior los valores de atributo.

Cuando el marco del esquema se completa con valores de ítems, como en la figura 1.2.3.A entonces hablamos de una instancia del esquema.

Debemos distinguir también entre tipo de registro e instancia de un registro. Cuando hablamos de un registro personal nos referimos en realidad a un tipo de registro. Con él no hay aún valores asociados. Como el esquema, es un marco en el que pueden

inscribirse valores de datos específicos. En cambio es más correcto llamar registro al registro personal de Juan Pérez. Pero aún no hay necesariamente un conjunto de valores de datos permanente asociado con ese registro : los valores de los datos pueden variar de un momento a otro. En un momento dado, por ejemplo, el registro personal de Juan Pérez puede tener inscritos los valores que aparecen en la primera línea de la figura I.2.3.A. Hablamos entonces de una instancia de ese registro.

Hay lugar para una distinción similar en los que se refiere a los ítems de datos, los agregados de datos, etc. Usamos términos tales como ítem de datos SALARIO, cuando en realidad SALARIO significa un tipo de ítem de datos, y empleamos la expresión registro PROVEEDOR cuando en realidad significa tipo de registro, etc.

RELACIONES Y REFERENCIAS CRUZADAS.

Los esquemas se presentan a menudo a modo de diagramas de bloques. La figura I.2.3.B ilustra un esquema típico.

Las líneas llenas que unen ciertos bloques representan relaciones. Así el registro ORDEN-DE-COMPRA está conectado con registros RENGLON-DE-COMPRA, que son los que forman la orden de compra. El registro PROVEEDOR está conectado con los registros COTIZACION, en los que se indican los artículos que el proveedor puede suministrar y las cotizaciones que ha presentado.

Las líneas discontinuas representan referencias cruzadas. NOMBRE-DE-ARTICULO y DETALLES-DE-LOS-ARTICULOS no aparecen en el registro RENGLON-DE-COMPRA sino en un registro separado, ARTICULOS. De igual modo, NOMBRE-DEL-

PROVEEDOR y DIRECCION-DEL-PROVEEDOR no se hallan en el registro ORDEN-DE-COMPRA sino en el registro independiente PROVEEDOR. De esta manera es posible evitar la duplicación de PROVEEDOR y DETALLES-DE-LOS-ARTICULOS en cada RENGLON-DE-COMPRA. Obsérvese que el diagrama continuaría representando la información completa aunque se suprimiesen las líneas discontinuas.

NOMBRE DEL ATRIBUTO FORMA DE REPRESENTACION	CLAVE PRIMARIA				CLAVES SECUNDARIAS				
	N5 NUMERO DE EMPLEADO	AV NOMBRE	B1 SEXO	N2 CALIFICACION	N6 FECHA	N3 DEPARTAMENTO	N2 CODIGO DE CAPACITACION	AV TITULO	N4 SALARIO
VALOR DE ATRIBUTO	53730	JONES, ALBERTO	1	03	100335	044	73	CONTADOR	2000
REGISTRO, SEGMENTO O TUPLA	28719	LARRONDO, JOSE	1	05	101019	172	43	PLOMERO	1800
	53550	GOMEZ, EMILIA	0	07	090938	044	02	AUXILIAR	1100
	79632	GUTIERREZ, PEDRO	1	11	011132	090	11	CONSULTOR	5000
	15971	GLAS, FERNANDO	1	13	021242	172	43	PLOMERO	1700
	51883	NEURINGER, JUAN	1	03	091130	044	73	CONTADOR	2000
	36453	BARREIRO, PASCUAL	1	08	110941	044	02	AUXILIAR	1200
	41618	RUSS, PATRICIA	0	07	071235	172	07	INGENIERO	2500
	61903	PANZA, VICTOR	1	11	011030	172	21	ARQUITECTO	3700
	72921	TABOADA, CAROLINA	0	03	020442	090	93	PROGRAMADOR	2100

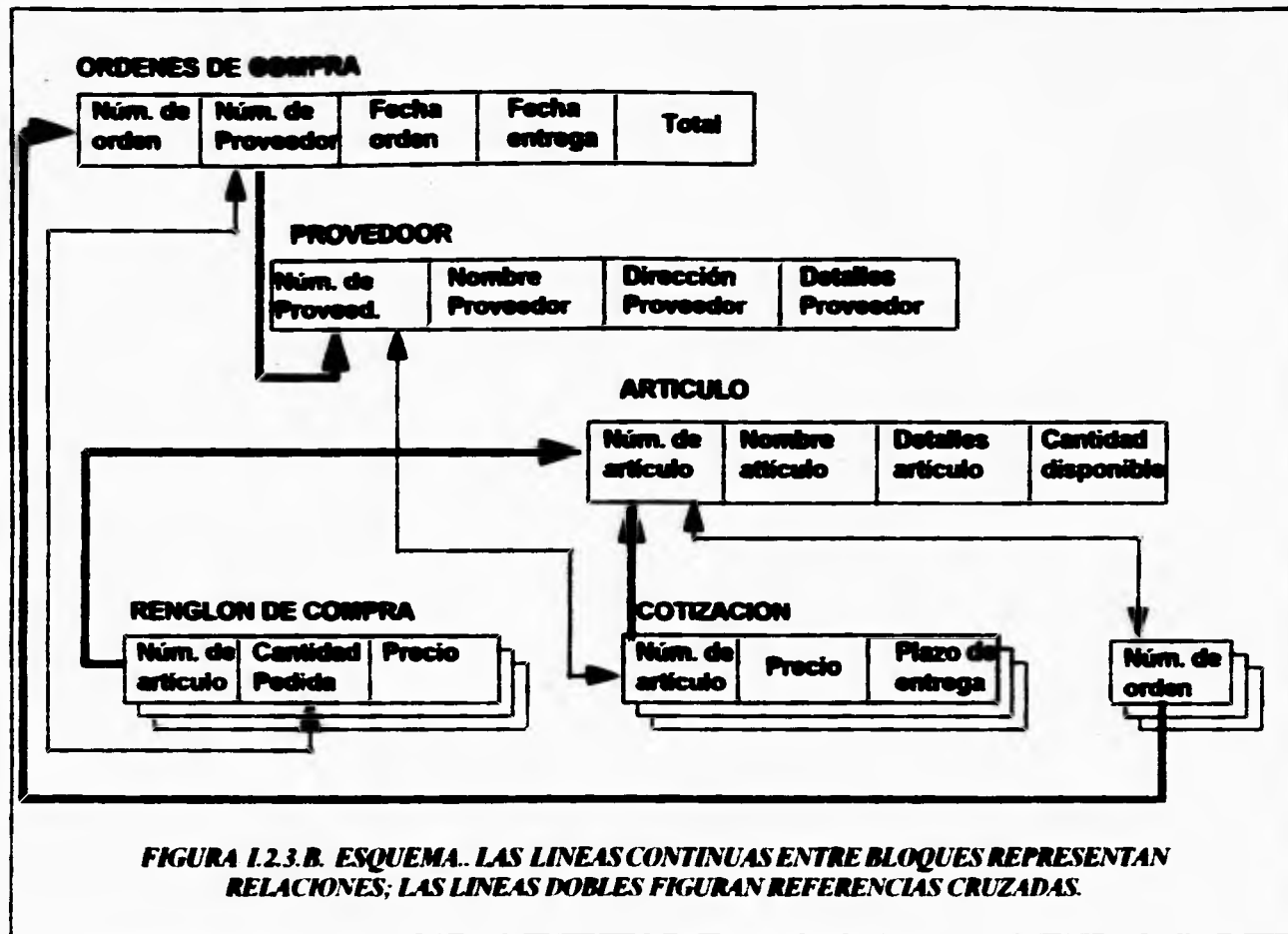
IDENTIFICADOR DE ENTIDAD

CONJUNTO DE VALORES DE UN ITEM DE DATOS (UN DOMINIO)

ALGUNOS ATRIBUTOS SON IDENTIFICADOS DE ENTIDAD EN OTROS ARCHIVOS

N 5 = Decimal, punto fijo, 5 digitos
 AV = Alfanumérico, longitud variable
 B 1 = Binario, 1 bit

FIGURA L2.3.A CARACTERISTICAS DE UNA TABLA



Las relaciones (líneas continuas) indicadas en el esquema agregan una información que no es inherente a los ítems de datos indicados en el esquema. Por ejemplo, no hay nada en el registro ORDEN-DE-COMPRA de la figura 1.2.3.B que diga qué artículos y en qué cantidades se compran. Esta información se completa sólo cuando se vinculan los registros RENGLON-DE-COMPRA con los registros ORDEN-DE-COMPRA que correspondan. En algunas bases de datos los mismos dos tipos de registro pueden estar vinculados mediante relaciones múltiples diferentes.

Las referencias cruzadas, en cambio, no aportan nueva información. El registro ORDEN-DE-COMPRA no contiene el nombre del proveedor ni su dirección, pero sí el número del proveedor, de modo que aprovechando esto pueden recabarse el nombre y la dirección del proveedor del archivo PROVEEDOR. Si se suprimiera la línea discontinua entre PROVEEDOR-NUM y el registro PROVEEDOR no se perdería información alguna. Las líneas de referencia cruzada en el esquema indican vinculaciones existentes en el archivo que permiten localizar los datos más fácilmente.

SUBESQUEMAS.

El término esquema se aplica al diagrama general de todos los tipos de ítems de datos y de registros almacenados en la base de datos. El término subesquema, en cambio, se refiere a la vista que tiene el programador de aplicaciones de los datos que utiliza. Del mismo esquema se pueden derivar muchos subesquemas.

La figura 1.2.3.C muestra los subesquemas correspondientes a dos programas de aplicación. Cada programador tiene su propia vista de los datos, pero ambos derivan su subesquema del esquema de la figura 1.2.3.B. El registro MAESTRO-DE-ORDEN-DE-COMPRA del programador A contiene ahora NOMBRE-DEL-PROVEEDOR y

DIRECCION-DEL-PROVEEDOR, mientras que el registro **DETALLE-DE-ORDEN-DE-COMPRA** contiene **NOMBRE-DE-ARTICULO**, **CANTIDAD-PEDIDA** y **PRECIO**.

El registro **DETALLE-DE-ORDENES-PENDIENTES** del programador B contiene ítems de datos tomados de tres diferentes registros de la figura I.2.3.B.

Los programadores de aplicación no tienen por qué conocer la totalidad del esquema. En cambio, el administrador de datos debe asegurar que los esquemas que aquéllos usan son derivables del esquema. El software de administración de datos deriva los datos de los subesquemas de los datos del esquema, automáticamente, y los pone a disposición del programa de aplicación.

TRES DESCRIPCIONES DE DATOS

Ni los esquemas ni los subesquemas reflejan la forma como los datos se almacenan físicamente. Tenemos así, en total, tres distintas descripciones de los datos :

1. El subesquema : el diagrama de una porción de los datos, orientado a satisfacer uno o más programas de aplicación, esto es, una organización de archivos del programador.
2. La descripción lógica global de la base de datos o esquema : el diagrama lógico de la base de datos completa. Es ésta la vista general de los datos como los contempla el administrador o los analistas de sistemas que usan toda la base.
3. La descripción de la base de datos : el diagrama de la distribución física de los datos en los dispositivos de almacenamiento. La vista de los datos que tienen los programadores de sistemas y los diseñadores de sistemas que se ocupan del desempeño y de cómo se ubican los datos en el hardware, de cómo se les localiza, y que técnicas de compactación se emplean.

Hay a menudo un cuarto punto de vista : el del usuario de terminal, el que tiende a ser una persona que conoce poco o nada acerca de las técnicas del procesamiento de datos.

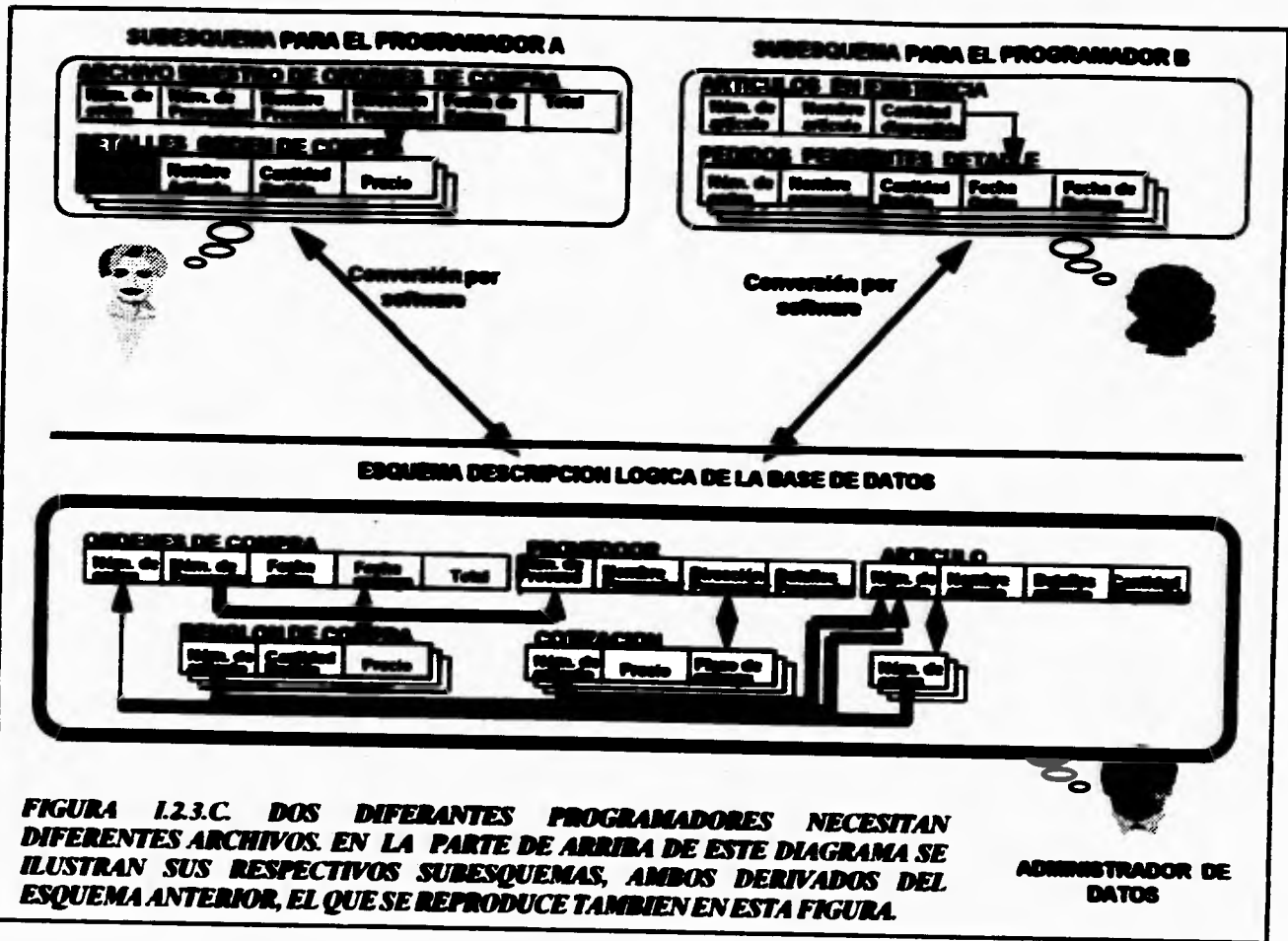


FIGURA 1.23.C. DOS DIFERENTES PROGRAMADORES NECESITAN DIFERENTES ARCHIVOS. EN LA PARTE DE ARRIBA DE ESTE DIAGRAMA SE ILUSTRAN SUS RESPECTIVOS SUBESQUEMAS, AMBOS DERIVADOS DEL ESQUEMA ANTERIOR, EL QUE SE REPRODUCE TAMBIEN EN ESTA FIGURA.

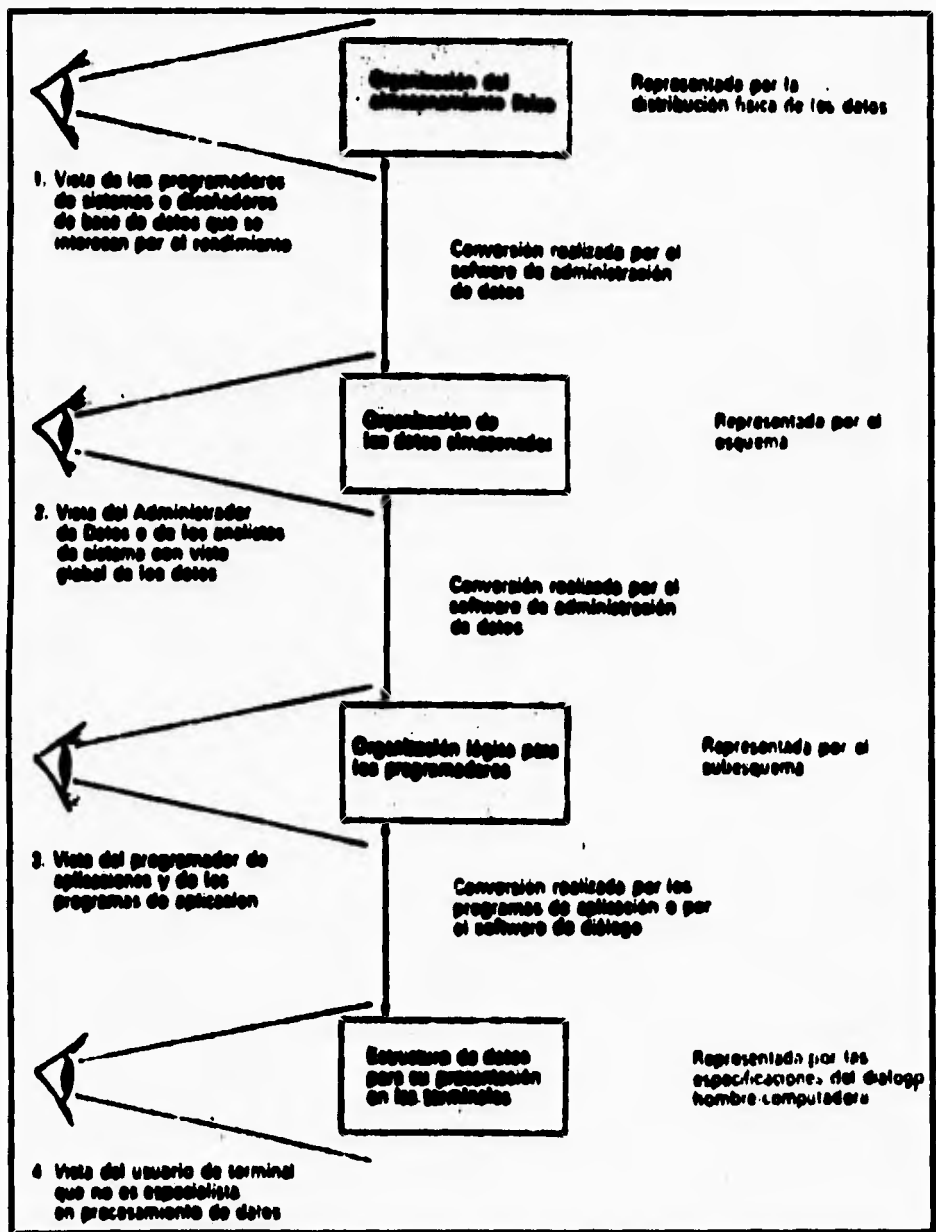


FIGURA 1.2.3.D. VISTAS DE DATOS DE 4 DIFERENTES USUARIOS.

La vista de los datos que el sistema le presenta debe asemejarse tanto como sea posible a la que es inherente al trabajo del usuario. El modo de presentación de esta vista depende del diseño del diálogo hombre-terminal que utiliza.

La figura I.2.3.D resume estas cuatro distintas vistas de los datos.

Nótese que el sistema de administración de la base de datos no recurre a estas vistas múltiples por que sí. Ocurre que cada vista puede ser modificada, si resulta necesario, sin afectar las otras. Sólo así es posible evitar que cualquier cambio cause trastornos mayores. En los sistemas que carecen de esta facilidad, el intento de introducir alguna modificación aparentemente trivial en el contenido o la organización de los datos ha demostrado resultar casi siempre inadmisiblemente oneroso, especialmente en las bases de datos complejas.

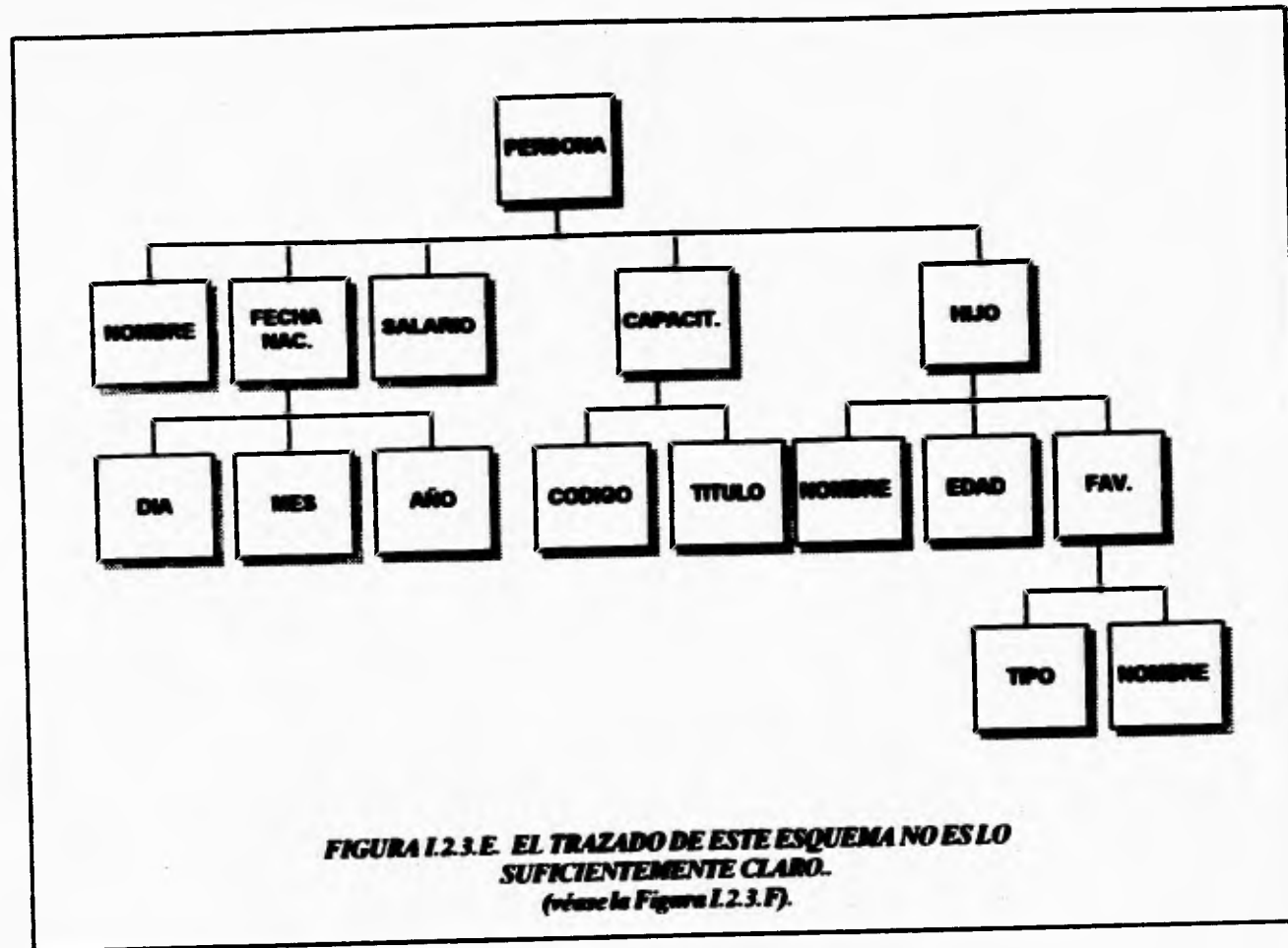
COMO DIBUJAR UN ESQUEMA

Un esquema mal dibujado más que aclarar las cosas las confunde. La figura I.2.3.E muestra un tipo de esquema tal como aparece en muchas publicaciones. Exhibe varios aspectos oscuros. Primero, el nombre de una parte del esquema es un bloque idéntico al que contiene el nombre de un ítem de datos. SALARIO y CAPACITACION, por ejemplo, aparecen el uno al lado del otro en bloques idénticos, pero SALARIO en un ítem de datos y CAPACITACION no. CAPACITACION es el nombre de un grupo de ítems de datos que contiene CODIGO y TITULO. Segundo, en el diagrama no se distinguen los atributos de las entidades. FECHA-DE-NACIMIENTO sale de un grupo de bloques similar a HIJOS. FECHA-DE-NACIMIENTO es simplemente un atributo de PERSONA, mientras que HIJOS es una entidad separada. HIJOS podría almacenarse en

un archivo independiente de PERSONA, pero no ocurre lo mismo con FECHA-DE-NACIMIENTO. Tercero, el diagrama no da indicación alguna de cuáles son relaciones de uno a uno y cuáles de uno a muchos. Sabemos que una PERSONA puede tener muchos HIJOS y que un HIJO puede tener más de un animal FAVORITO de modo que éstas son relaciones de uno a muchos.

Pero una PERSONA tiene una sola FECHA-DE-NACIMIENTO, de modo que ahora la relación es de uno a uno. No sabemos si una PERSONA puede tener más de una CAPACITACION. Es éste un hecho importante acerca de la base de datos, pero el diagrama no indica nada.

Por otra parte, el diagrama no aclara qué grupos de ítems de datos son tipos de registro ni cuáles son tipos de agregados de datos. La diferencia entre un tipo de registro y un tipo de agregado de datos está en que puede haber un número ilimitado de ocurrencias del primero, pero sólo un número limitado del segundo. El agregado de datos se direcciona comúnmente a través del registro de que es parte, mientras que el registro debe tener sus propios medios de direccionamiento. Algunos sistemas importantes de bases de datos no hacen diferencia entre los agregados de datos y los registros. Por ejemplo los sistemas de administración de bases de datos IMS y CICS de IBM consideran a todo grupo de ítems de datos individualmente direccionable (campos) como segmentos. Cuando, como en los productos basados en el CODASYL, ocurre que se distingue entre registro y agregado, el esquema debe indicar la diferencia claramente.



1. El diagrama debe distinguir claramente los nombres de registro, los de ítems de datos y otros.
2. Debe quedar clara la reunión de ítems de datos en agregados de datos.
3. Debe aclararse la distinción entre agregados de datos y registros.
4. Deben mostrarse los identificadores de registro.
5. El diagrama debe indicar claramente qué relaciones son simples y cuáles complejas.
6. Las relaciones deben diferenciarse de las referencias cruzadas.
7. Las relaciones entre registros deben nominarse o numerarse.
8. No deben utilizarse nombres duplicados.

FIGURA 1.2.3.F REGLAS PARA DIBUJAR UN ESQUEMA. (Compárense las Figuras 1.2.3.E y 1.2.3.G)

Un defecto más del diagrama de la figura 1.2.3.E está en que algunos bloques llevan nombres idénticos.

Nosotros sugerimos que los diagramas de esquema se tracen respetando las reglas que se dan en la figura 1.2.3.F.

El diagrama de la figura 1.2.3.E puede dibujarse de muchas otras formas, sin dejar de satisfacer las reglas anteriores. La figura 1.2.3.G muestra una de esas formas.

Los registros aparecen en bloques de línea llena, mientras que los bloques correspondientes a los agregados de datos repetitivos se dibujan con línea discontinua; los nombres de los registros y los agregados de datos se anotan fuera de los bloques. Los nombres de los ítems de datos van dentro de los bloques correspondientes. Los nombres

identificadores de registro se subrayan. Las flechas dobles que indican relaciones de uno a muchos no son estrictamente necesarias, pues la separación entre niveles en este diagrama indica precisamente esa relación. Pero ellas suelen resultar indispensables en diagramas más complejos.

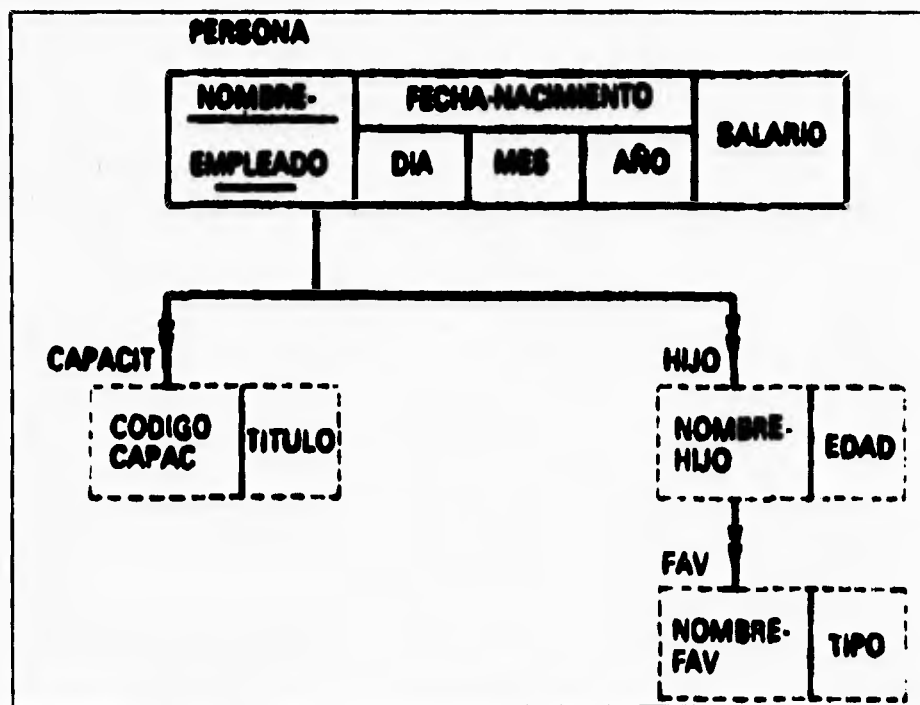


FIGURA 1.3.G. DIBUJO DEL ESQUEMA MODIFICADO.

I.2.4 NORMALIZACION.

La normalización es una técnica que se ha desarrollado para obtener estructuras de datos eficientes. El concepto de normalización fue introducido por [CODD 74] y fue pensado para aplicarse a sistemas relacionales, sin embargo, tiene aplicaciones mas amplias. La normalización es la expresión formal del modo de realizar un buen diseño. Provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información.

Antes de definir o describir el proceso de normalización, es de suma importancia abarcar ciertos conceptos que facilitan la comprensión de dicho proceso.

Una relación es no reductible si es imposible separarla en varias relaciones de grado mas bajo por medio de operaciones de proyección, de manera tal, que se pueda reconstituir la relación original al unir las relaciones producto. Una relación no reductible se le conoce como relación elemental.

Un atributo B de una relación es funcionalmente dependiente del atributo A, si en todo momento cada valor de A está asociado con uno y sólo uno de los valores de B. De otra manera, existe siempre una asociación simple o condicional de A a B. Lo que significa que B esta en función de A y se puede representar como una tabla de dos columnas.

La notación que generalmente se utiliza para indicar que B es funcionalmente dependiente de A en la relación R es:

R.A. ---> R.B

Por otro lado si B no es funcionalmente dependiente de A en la relacion R, se denota como:

$$R.A. \not\rightarrow R.B$$

La definición anterior se puede generalizar a una colección de atributos en R, se dice que B_1, \dots, B_n , es funcionalmente dependiente a otra colección de atributos en R, A_1, \dots, A_m , si siempre cada lista de atributos $R[A_1], R[A_2], \dots, R[A_m]$ está asociada a una y sólo una lista de atributos $R[B_1], R[B_2], \dots, R[B_n]$.

En donde $R[A_1]$ significa un valor del atributo A_1 correspondiente a una fila de la relación.

Una dependencia trivial (en donde E es un subconjunto de D) es de la forma:

$$R.D \twoheadrightarrow R.E$$

El atributo o colección de atributos B, en una relación R, es funcionalmente dependiente de forma completa de un atributo o colección de atributos A en R, si B es funcionalmente dependiente de A, pero no es funcionalmente dependiente de ningún otro subconjunto de A. Esto es equivalente a tener una asociación simple o condicional. En otras palabras, el atributo o colección de atributos de A en R si existe una asociación simple o condicional de A a B, en donde toda asociación de cualquier subconjunto propio de A a B es compleja.

La notación usual para indicar que B es funcionalmente dependiente de forma completa de A en R es:

$$R.A \twoheadrightarrow R.B$$

Por otro lado

$$R.A \neq \Rightarrow R.B$$

significa que B no es funcionalmente dependiente de forma completa de A en R.

La dependencia transitiva se puede definir como sigue:

Supóngase que A,B y C son tres distintos atributos de una colección de atributos en una relación R, y que las siguientes dependencias ocurren:

$$R.A \twoheadrightarrow R.B$$

$$R.B \not\rightarrow R.A$$

$$R.B \twoheadrightarrow R.C$$

entonces

$$R.A \twoheadrightarrow R.C$$

$$R.C \not\rightarrow R.A$$

Se dice que C es transitivamente dependiente en forma estricta de A en la relación R, en el caso:

$$R.C \twoheadrightarrow R.B$$

y B y C son transitivamente dependientes de A en la relación R.

Al atributo o colección de atributos C de una relación R para el cual cualquier otro atributo o colección de atributos D de R es funcionalmente dependiente de C, es denominado determinante.

En el proceso de datos los conceptos de claves o llaves y valores de llaves son fundamentales, por lo que el modelo relacional debe incorporar facilidades para el manejo de llaves. De hecho, los sistemas relacionales tienen reglas especiales para éstas.

Una llave puede estar formada por uno o más atributos de las relaciones.

Una llave posible o candidata, es una combinación de atributos de cualquier n-ada de la tabla, que distingue en forma exacta(única) a esa n-ada de cualquier otra de la tabla. En otras palabras, la llave candidata debe ser única. En los sistemas relacionales siempre es posible formar esa llave única, ya que no existen relaciones duplicadas. Otra regla vigente es, que si de la llave candidata se elimina un atributo, se pierde la propiedad de identificación unívoca que ésta nos permite, pues la llave no puede contener atributos redundantes.

Resumiendo se define como llave candidata K a un atributo o conjunto de atributos de una relación R, si cumple con las siguientes propiedades:

1. En cada n-ada de R el valor de K identifica de manera única esa n-ada (identificación única). Como corolario se puede establecer que cada atributo de R es funcionalmente dependiente de K.

2. En el caso en donde K es una colección de atributos, ninguno de los atributos de K pueden eliminarse sin violar la condición 1 (No redundancia). Como corolario cada subconjunto propio de la colección de atributos K, es funcionalmente independiente de cualquier otro subconjunto propio de atributos de la colección K. En otras palabras, existe un mapeo de varios a varios entre cualquier par diferente de subconjuntos propios de la colección de atributos K.

Una relación puede poseer diversas llaves candidatas. Una de las llaves candidatas siempre es designada como llave primaria. La forma usual de la designación consiste en que no se permiten valores indefinidos o nulos en los atributos que la constituyen.

Un atributo primo de una relación R, es un atributo que pertenece al menos a una llave candidata de R. Análogamente un atributo de una relación R que no pertenece a ninguna

llave candidata de R, es denominado un atributo no primo. A los atributos que conforman una llave primaria se les conoce como atributos llave.

Un atributo de una relación R_1 , es una llave externa si no es llave primaria de R_1 , pero sus valores son valores de la llave primaria de alguna otra relación R_2 . Las llaves primarias junto con las llaves externas, proveen de un mecanismo de visualización de las asociaciones entre relaciones.

Si una relación R contiene varios atributos que son funcionalmente dependientes de otros atributos en R, puede ser dividida en varias relaciones de menor grado, de tal manera que esas relaciones producto pueden ser unidas para reconstruir la relación R original.

Ahora podemos describir la normalización de relaciones, este proceso es de suma importancia sobre todo en la definición de relaciones que son consistentes con un modelo especificado, de manera que no tengan dependencias indeseables que causen problemas en la manipulación de la base de datos, es decir, cuando se crean, modifican o eliminan los valores de los datos, a dichos problemas se les denomina anomalías en las operaciones de almacenaje.

Codd [Codd 71] originalmente definió los niveles de normalización llamados primera forma normal (1FN), segunda forma normal (2FN) y tercera forma normal (3FN). Hablando de manera general cada relación normalizada está en la 1FN, algunas relaciones en 1FN son también 2FN y estas son un subconjunto de las relaciones 3FN. En resumen, es posible adelantar que las relaciones 3FN son las mejores por que estas no permiten la mayoría de las inconsistencias de la información de la base de datos.

El proceso de normalización, permite convertir cualquier relación en un conjunto semánticamente equivalente de relaciones 3FN. La intención es implementar una base de datos que evite propiedades indeseables como las anomalías en las operaciones sobre la base de datos.

A las relaciones que se caracterizan por tener en cada atributo un valor atómico, es decir, que no es posible descomponerlo, se les denomina relaciones en la primera forma normal (1FN). Es posible visualizar este tipo de relaciones en una tabla, de manera tal, que en toda intersección fila y columna de dicha tabla se tiene un solo valor, nunca un conjunto de valores. Los valores indefinidos se permiten sólo en los atributos que no constituyen la llave primaria de la relación.

Por otro lado, una relación que tiene valores múltiples en alguna intersección fila y columna, no está en 1FN y se le conoce como relación no normalizada.

La definición de Codd plantea, una relación R está en segunda forma normal (2FN), si está en primera forma normal (1FN), y cada atributo no primo de R es funcionalmente dependiente en forma completa de toda llave candidata de R.

Ahora se sabe que la definición anterior no maneja satisfactoriamente las relaciones con llaves candidatas traslapadas. Por lo que propuso [SCHMID 75] la siguiente definición:

Una relación esta en segunda forma normal si está en 1FN, y todo atributo en el complemento de una llave condidata K es funcionalmente dependiente en forma completa de K.

Una colección de relaciones C está en segunda forma normal óptima con respecto a una relación T si se cumple con:

1. Todas las relaciones en C están en 2FN.
2. Las relaciones en C retienen toda la información esencial en la relación T . es decir, T puede recobrase por la unión de las relaciones en C .
3. No hay una colección más pequeña con las propiedades 1 y 2.

Una relación R está en tercera forma normal (3FN) si está en la segunda forma normal y todo atributo no primo de R es dependiente no transitivamente de cada llave candidata de R , una vez más, la definición anterior no maneja satisfactoriamente el caso en que una relación que posee dos llaves candidatas traslapadas, por lo que se propuso una definición alternativa:

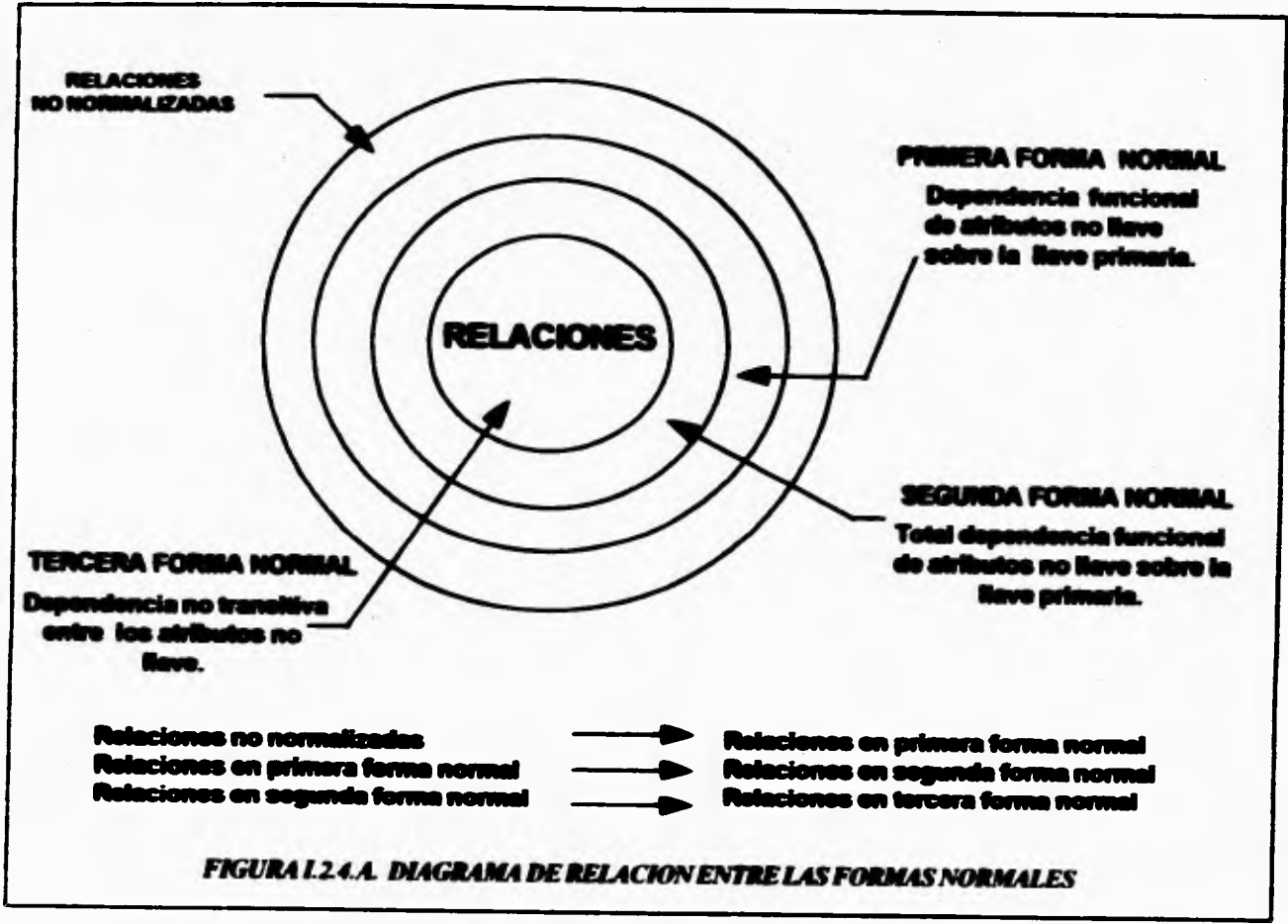
Una relación en 2FN está en 3FN, si cada atributo en el complemento de una llave candidata es dependiente no transistiva de esa llave candidata.

De manera similar, al caso de las relaciones 2FN, existe una noción para 3FN óptimas.

Sea C_2 una colección de relaciones en 2FN óptimas, y C_3 una colección de relaciones en 3FN obtenidas por proyecciones de las relaciones en C_2 . Entonces la colección C_3 es una forma normal (3FN) óptima con respecto a C_2 si:

1. Todas las relaciones en C_3 están en 3FN.
2. Las relaciones en C_3 retienen toda la información esencial en C_2 , es decir, la colección C_2 puede recobrase por unión de relaciones en C_3 .
3. Ninguna relación en C_3 contiene dos atributos que sean transitivamente dependientes de forma estricta en alguna relación en C_2 .
4. No hay colección de relaciones más pequeñas que tenga las propiedades 1,2 y 3.

En la figura 1.2.4.A, se muestra la relación que hay entre las formas normales.



EJEMPLO DE TABLA.

Un ejemplo de una tabla, es la tabla de FACTURAS, que se muestra en la figura 1.2.4.B.

Núm.Factura	Cliente	Fecha	No.reporte	Monto	Surtida
112233	1024	05-01-94	123	1234.56	.T.
112234	2541	05-01-94	233	2000.50	F.
112235	6886	05-02-94	235	1005.50	.T.

FIGURA 1.2.4.B.

En esta tabla existen 6 columnas y 3 renglones. Núm.Factura es un atributo que consiste de un valor numérico utilizado para identificar una orden particular. Así que, '112233' es el valor del atributo, tomado del dominio de Núm.de facturas (en donde el dominio quizás este limitado por el rango de 100000 a 999999), el cual es utilizado para identificar una factura del negocio. El esquema de esta tabla es:

FACTURAS(Num..Factura, Cliente, Fecha, No.reporte, Monto, Surtida)

EJEMPLO DE LLAVE PRIMARIA

¿ Qué campos escogería para llave primaria en la tabla REP(representante de ventas), que se muestra en la figura 1.2.4.C ?.

Apellido	Nombre	Dirección	Ciudad	Estado	Venta
Juárez	Benito	Reforma 123	Parral	Chihuahua	5,000,000
Fernández	Juan	Insurgentes 12	Jalapa	Veracruz	2,000,000
Pérez	José	Hidalgo 234	Toluca	Edo.México	3,500,000

FIGURA 1.2.4.C.

Como primera proposición quizás elegimos REP (Apellido + Nombre) como llave primaria compuesta. Pero en una gran Compañía quizás existan 2 personas con el nombre de Pérez José. La solución tal vez sería generar otra columna con un campo secuencial que cuente el número de nombres duplicados, sumando este campo a la llave para asegurarnos que sea una llave única. Otra posible solución es agregar la columna de la dirección a la llave primaria. Y una mejor solución sería crear una nueva columna con la identificación del vendedor. Esta es simple y única y se garantiza desde que nosotros asignamos el número a cada persona. Observemos la figura 1.2.4.D :

Tabla REP modificada

Tabla REP modificada					
No.Reporte	Apellido	Nombre	Dirección	Ciudad	Estado
101	Juárez	Benito	Reforma 123	Parral	Chihuahua
102	Fernández	Juan	Insurgentes 12	Jalapa	Veracruz
103	Pérez	José	Hidalgo 234	Toluca	Edo.México

FIGURA 1.2.4.D.

EJEMPLO DE NORMALIZACION

Resumiendo lo anterior y tratando de ejemplificarlo observemos lo siguiente:

NORMALIZACIÓN

El proceso de determinar la correcta localización y función para cada atributo en orden, formulando correctamente el esquema relacional es llamada normalización. Generalmente la normalización descompone relaciones que están incorrectamente construidas en relaciones múltiples correctamente normalizadas. Hay numerosas formas de normalizar, analizaremos las 3 más importantes.

El modelo relacional requiere que todas las tablas al menos, estén en primera forma normal (1FN), lo cual significa que los dominios para todos los atributos deben consistir solo de valores atómicos, evitando grupos repetidos. Esto es normalización estructural o sintáctica.

El concepto de dependencia funcional es central, al determinar la normalización de las relaciones. Por ejemplo consideremos una relación con el siguiente esquema.

FACTURA(Núm.Cliente, Dirección, Ciudad, Estado, C.P., Monto);

Un cliente puede tener varias ordenes en nuestra compañía, creando múltiples tuplas en FACTURAS. Entonces si creamos una nueva factura para el cliente, tendremos todos los datos del cliente repetidos, y si cambia la dirección del cliente tendremos que modificarla en todas las tuplas. Como podemos observar esta relación no esta normalizada.

Primera Forma Normal (1FN)

Para estar en primera forma normal, una relación debe tener atributos con dominios que solo sean valores atómicos. No se puede repetir el conjunto de atributos y no tener atributos con múltiples valores. Por ejemplo, la relación CLASE con el esquema:

CLASE(núm.clase,nombre, instructor);

Si existe una relación ESTUDIANTE, con el esquema;

ESTUDIANTE(núm.estudiante, principal, consejero, salón consejero, {clases});

observamos que en la relación ESTUDIANTE, el atributo clases tiene múltiples valores, por que son todas las clases que lleva el estudiante, esto no está en Primera forma normal. El mismo problema se observa si se tiene una serie de campos del mismo tipo repetidos:

ESTUDIANTE(núm.alumno, principal, consejero, salón consejero, clase1, clase2, clase3)

Observemos el ejemplo anterior en la figura 1.2.4.E , tabla no Normalizada:

ESTUDIANTE						
no.alumno	principal	consejero	salón conse.	clase 1	clase 2	clase 3
1022	81	Pérez	412	101-07	143-01	159-02
4123	84	Ríos	216	210-01	211-02	214-01

FIGURA 1.2.4.E.

Como observamos, esto no esta en Primera forma normal, aquí un atributo tipo esta repetido (clase 1, clase 2, clase 3). Si un estudiante tiene mas de 3 clases, existirá un problema y si tiene menos de 3, algunos valores por default tienen que estar definidos. Poniendo esto en Primera forma normal, tendremos como resultado la siguiente tabla:

ESTUDIANTE(núm.alumno, principal, consejero, salón consejero, clase)

Podemos ver el resultado de la relación en la figura 1.2.4.F, donde la información del estudiante es repetida para cada una de las clases donde el estudiante esta inscrito. No obstante, esto tiene todavía un problema, si un consejero cambia, entonces todas las tuplas de estudiante deben ser leídas y actualizadas. También si un estudiante se da de

baja de todas las clases, toda la información del estudiante y consejeros se perderá. Solo formas mas elevadas de normalización resolverán estos problemas.

ESTUDIANTE				
no.alumno	principal	consejero	salón conse.	clase
1022	81	Pérez	412	101-07
1022	81	Pérez	412	143-01
1022	81	Pérez	412	159-02
4123	84	Ríos	216	210-01
4123	84	Ríos	216	211-02
4123	84	Ríos	216	214-01

FIGURA 1.2.4.F.

Segunda forma normal (2FN)

Cuando una relación se encuentra en segunda forma normal, esta ya esta en primera forma normal y todos los atributos no llave son funcionalmente dependientes de la llave primaria. Cualquier atributo que no conozca esta relación debe ser sacado fuera de la relación y generar otra nueva relación.

Considerando el ejemplo anterior, donde el nuevo esquema esta en primera forma normal y donde núm. estudiante es la llave primaria. Notamos que hay múltiples valores para el atributo clase, para cada valor de núm. de estudiante. Esto significa que clase no es funcionalmente dependiente de la llave primaria y entonces esta relación no esta en segunda forma normal. Es necesario descomponer el esquema de ESTUDIANTE, en dos tablas con los siguientes esquemas:

ESTUDIANTE(núm.alumno, principal, consejero, salón consejero)

INSCRIPCIONES(num_estudiante, clase)

como se muestra la relación en las figuras -I.2.4.G. y I.2.4.H. :

ESTUDIANTE			
no.alumno	principal	consejero	salón conse.
1022	81	Pérez	412
4123	84	Ríos	216

FIGURA I.2.4.G.

INSCRIPCIONES	
no.estudiante	clase
1022	101-07
1022	143-01
1022	159-02
4123	210-01
4123	211-02
4123	214-01

FIGURA I.2.4.H.

Tercera Forma Normal.

Cuando una relación se encuentra en tercera forma normal, esta ya esta en segunda forma normal y debe tener todos sus atributos no primos no transitivamente dependientes de la llave primaria. Esto significa que cada atributo no llave, debe depender de la llave primaria. En el ejemplo anterior, notamos que el salón del consejero, es funcionalmente dependiente del atributo consejero. Entonces el atributo salón del consejero es no primo,

INSCRIPCIONES(num_estudiante, clase)

como se muestra la relación en las figuras -I.2.4.G. y I.2.4.H. :

ESTUDIANTE			
no. alumno	principal	consejero	salón conse.
1022	81	Pérez	412
4123	84	Ríos	216

FIGURA I.2.4.G.

INSCRIPCIONES	
no. estudiante	clase
1022	101-07
1022	143-01
1022	159-02
4123	210-01
4123	211-02
4123	214-01

FIGURA I.2.4.H.

Tercera Forma Normal.

Cuando una relación se encuentra en tercera forma normal, esta ya está en segunda forma normal y debe tener todos sus atributos no primos no transitivamente dependientes de la llave primaria. Esto significa que cada atributo no llave, debe depender de la llave primaria. En el ejemplo anterior, notamos que el salón del consejero, es funcionalmente dependiente del atributo consejero. Entonces el atributo salón del consejero es no primo,

esta es una violación a la tercera forma normal. La solución es mover el atributo fuera de la relación de ESTUDIANTES, como vemos en las figuras I.2.4.I y I.2.4.J

ESTUDIANTE		
no.alumno	principal	consejero
1022	81	Pérez
4123	84	Ríos

FIGURA I.2.4.I.

CONSEJEROS		
consejero	salón	depto.
Pérez	412	42
Ríos	216	42

FIGURA I.2.4.J.

Generalmente, es considerado suficiente, normalizar las relaciones hasta en tercera forma normal, pero existen formas de normalización mas elevadas y por lo tanto mas eficientes. Cada normalización mas elevada asegura que la relación esta en tercera forma normal.

1.3 SISTEMAS MANEJADORES DE BASES DE DATOS PARA COMPUTADORAS PERSONALES.

1.3.1 DBASE IV

Ahora dBase es más fácil de usar, adicional a la línea de comando, el programa ofrece una interface a base de menús denominada Centro de Control. A pesar de que dicho centro de control representa una mejora para usuarios poco familiarizados con sistemas de cómputo, la simple adición de esta interface no hace de dBase un producto amigable con el usuario.

El paquete no ofrece manejo de mouse, el centro de control frecuentemente solicita confirmación para efectuar prácticamente cualquier función, aún cuando ya se han grabado los datos y se oprime la tecla escape para regresar al menú anterior, el centro de control pregunta si esta uno seguro de querer realizar dicha operación.

Cuando uno se ha familiarizado con el centro de control, entonces se obtienen algunas ventajas , entre ellas, la construcción de vistas. Como con muchos otros paquetes, se inicia con la estructura de la base de datos activa, se pueden añadir múltiples bases de datos por medio del menú y pasar de una a otra utilizando las teclas de función. Seleccionando los nombres de los campos en la estructura de las bases de datos, se pueden crear ligas entre ellas, especificar que campos formarán parte de la vista, organizar la base de datos utilizando campos específicos y métodos de ordenamiento. Una vista también puede disparar la actualización de registros en forma múltiple, y el resultado se puede guardar como una vista o como una base de datos nueva.

Cuando se generan reportes, la utilización del mouse sería de gran ayuda, y lo peor del caso es que la interface no ofrece ninguna forma para repositionar texto o campos, no existe tampoco la opción de copiado.

El generador de formatos no soporta el uso de múltiples bases de datos sin que se tengan que realizar algunos trucos para lograrlo.

Normalmente, la mejor forma de trabajar con múltiples bases de datos, es crear primero una vista. Esto funciona sin problemas con el generador de reportes, más no con el generador de formatos.

1.3.2 CLIPPER 5.01

El desarrollo de esta base de datos, está destinado definitivamente para programadores. A pesar de la carencia de las elegantes utilerías de generación de código de otros paquetes, Clipper, ofrece un ambiente de herramientas y capacidades que el programador necesita. Dos de éstas herramientas son: un generador de reportes (conocido como RL) y una utilería para crear y manejar archivos de bases de datos (conocida como DBU), estos programas, están escritos en un lenguaje de programación propio de Clipper y está incluido su código fuente. El programador puede usar este código como referencia o para aumentarlo en sus aplicaciones de base de datos. La utilidad de la base de datos, se valoran especialmente cuando se contruyen estructuras y objetos.

Sin embargo, Clipper está construido para usuarios que entienden de programación y de ambientes interactivos de trabajo importantes particularmente cuando vienen de la realización de tareas de manejo de bases de datos comunes. DBU es de gran ayuda en

este momento. El panel principal de DBU, lista opciones a lo largo de la parte superior de la pantalla, que se pueden ejecutar con las teclas de funciones asignadas. El resto de la pantalla, está ocupada por una representación de la vista de la base de datos activa y consiste en una columna dividida en tres secciones. El primer grupo, despliega los nombres de la base de datos activa. Si se incluye un argumento cuando se invoca al programa DBU, la base de datos o vista especificada aparecerán. La siguiente sección, muestra los índices activos asociados con la base de datos activa, y el grupo inferior despliega los nombres de los campos de la base de datos activa. La columna es un área de trabajo. Si la columna está vacía, simplemente se oprime enter para ver la lista de bases de datos en el directorio actual. Se puede seleccionar, una de esas bases de datos, para cargarla en el área de trabajo o crear una base de datos nueva presionando la tecla para esta función.

Este panel, permite cargar rápida y fácilmente diferentes bases de datos dentro de áreas de trabajo separadas, usar o crear índices, y modificar la estructura de la base de datos. Es apreciada especialmente, la facilidad con que el sistema permite crear diferentes vistas y fijar relaciones.

El generador de reportes, no es tan flexible, ya que no soporta un formato WYSIWYG (lo que ves es lo que obtienes) del reporte, o no permite hacer una vista previa del trabajo del programador.

El programa señala, un nombre y definición de columna. Se teclea en el campo nombre lo que se desea que aparezca en la primer columna, con el título de la columna. El programador también puede definir campos de sumatorias del panel de definición de columnas. Solamente se puede aumentar una columna cada vez. Después de haber

definido las columnas, se pueden especificar grupos y fijar parámetros para un reporte completo.

El poder real de clipper se debe gracias a su código. El compilador de programas, es muy rápido, pero el paquete también tiene otras características interesantes. Una de sus cualidades, es la capacidad de soportar llamadas a bloques de código, pequeños trozos de código ejecutable que se pueden almacenar como variables o argumentos de paso a otros programas. Se puede usar una función {EVAL()} para ejecutar un bloque de código.

Se puede invocar la utileria MAKE para ejecutar las operaciones necesarias de compilación y ligado para obtener todos los archivos en sincronía. Clipper también soporta funciones de lectura y escritura de archivos binarios de DOS. Un apuntador de archivo sigue la secuencia de leer o escribir, o el programador puede fijar el apuntador con la función {SEEK()}.

También es importante el debugger de Clipper, el cual permite al usuario seguir el programa paso a paso o monitorear el estado particular de cada variable, nombre de campos o expresiones. Una ventana de estatus, lista las bases de datos abiertas en todas las áreas de trabajo, así como los valores fijados por todos los comandos {SET}. El debugger, también contiene un histórico de las llamadas a cada subrutina y de las ejecuciones del código del programa. Clipper no es una base de datos novicia, pero los programadores, la pueden encontrar muy conveniente para trabajar.

1.3.3 FOXPRO 2.0

La interface que ofrece este paquete incluye menús, además de una ventana de comandos, de tal forma que se pueden usar los menús o la línea de comando. El paquete soporta el uso del mouse. La ventana de comandos mantiene una historia de las instrucciones utilizadas. Se pueden seleccionar parte de los comandos usados de dicha historia y copiarlos en las aplicaciones.

Para crear pantallas de captura se pueden abrir las bases de datos necesarias y establecer las relaciones entre ellas. La ventana muestra gráficamente las áreas de trabajo disponibles. Se puede seleccionar una de las áreas disponibles y abrir una base de datos oprimiendo un botón de comando con el mouse. Fox Pro proporciona una lista de las bases de datos que se pueden escoger. Después de cargar las bases de datos que se necesitan, se resaltan las opciones y se selecciona un comando para establecer las relaciones. Fox pro, ofrece una caja de dialogo con una lista de campos llave.

Cuando se ha diseñado la pantalla de captura, el generador de pantallas, salva dicha información. El programador, no debe preocuparse de abrir las bases de datos y después fijar las relaciones. El generador de pantalla, empieza con un panel vacío en el cual se puede teclear texto o colocar los campos. Se pueden crear también botones y teclas de comando, y listas en forma de menú. Y se pueden unir piezas de código a cualquier objeto, incluyendo los campos. Del menú principal, se pueden designar código que corra antes o después de la pantalla de captura. Cuando se genera código para la pantalla que se ha diseñado, se pueden ligar otras pantallas a la primera, lo cual ahorra tiempo una vez que se haya creado una biblioteca de pantallas generales. Simplemente se añade una pantalla de control al tutorial para navegar a través de los registros, añadiendo un botón

de comando para procesar datos de entrada y asociándole código para actualizar las áreas correspondientes a la base de datos.

Se puede guardar una vista como si fuese un programa y si un formato de reporte es parte de la vista, se puede obtener una impresión de esta directamente desde la línea de comando invocándola con un comando DO y utilizando el nombre del archivo como un argumento.

I.3.4 DATAEASE

Cuando se trabaja con dataease, se siente que el programa nos lleva de la mano llevándonos paso a paso. Esto puede hacer que las cosas sean fáciles pero en ocasiones llegar a desesperar.

Si se ha trabajado con otras bases de datos, no se debe esperar que dataease trabaje como alguna de éstas. Al arrancar Dataease, la entrada al sistema es una pantalla de logon. Es necesario indicar el nombre de la base de datos con la cual se trabajará, y si es un nombre que no aparece en la lista, el programa preguntará si se desea crear una nueva base de datos. Además se solicita un nombre de usuario y una clave de acceso, si se desea éstas dos últimas se pueden dejar en blanco para dejar inactiva esta opción.

En el menú principal, las opciones se muestran junto con un número de opción, se pueden utilizar las teclas de cursor para seleccionar la opción deseada u oprimir el número de la opción, el programa no soporta el uso de mouse. Normalmente la primera acción que se realiza con una nueva base de datos, es crear una tabla, las cuales se denominan formas en este programa. Al momento que se crea una forma, paralelamente

se está creando una pantalla de captura y la estructura de la tabla. Se inicia con una pantalla en blanco en la cual se tecléa el texto. Cuando se oprime la tecla F10 para insertar información en un campo, en ese momento, aparece el contenido del campo que acaba de ser tecléado. Esta secuencia, es la forma típica en la que trabaja Dataease. La interface está formada por menús, teclas de función, y formatos en blanco para ser llenados.

La forma utilizada para definir los campos, muestra algunas de las funciones más poderosas que ofrece Dataease, ya que no se requiere programación para realizar esto. Una vez que se han diseñado las formas, se pueden ligar unas con otras utilizando una opción del menú, simplemente se especifican las formas que se desean ligar y se marcan los campos a través de los cuales se establecerá la liga.

La estructura de Dataease, hace que la generación de reportes sea especialmente fácil. Se designa una tabla primaria para ser utilizada y aparece una forma con los nombres de sus campos. A continuación se seleccionan con la barra espaciadora los nombres de los campos que deben aparecer en el reporte, Dataease permite teclear palabras claves en la lista de los campos para que se puedan agrupar campos seleccionados o acumular información estadística.

El programa es fácil de utilizar, pero a final de cuentas uno se siente condenado al ambiente de Dataease. Para generar código es necesario utilizar el editor de Dataease, y este no permite guardar un archivo que contenga errores de sintaxis o variables no definidas. Estas restricciones hacen que el programa sea poco amigable a los programadores, pero en cambio lo hace amigable a los usuarios novatos.

I.3.5 PARADOX 3.5

Una de sus características más notables, es la velocidad. Su tiempo de respuesta para editar y desplegar tablas, es verdaderamente rápido.

Se pueden crear vistas y ediciones complejas, utilizando la técnica de query by example (QBE). Es necesario proporcionar operaciones relacionales y filtros que operen sobre los campos que se desean evaluar. Paradox ofrece la función de importar y exportar los formatos más comunes, pero se pueden tener problemas cuando se importa información que está en formato ASCII, cuando se realiza esta función, Paradox requiere que los campos de cada registro del archivo que se importa, estén dispuestos en el mismo orden en el que lo están los campos de la tabla destino.

El lenguaje de manejo de datos de Paradox se llama PAL, siendo este un lenguaje de procedimientos, no es una derivación de SQL. Su repertorio de instrucciones, no incluye las instrucciones de manipulación de datos típicas tales como SEEK_KEY, READ_RECORD, etc. en su lugar el PAL, es un lenguaje que automatiza el ambiente de Paradox, por ejemplo, el comando que utiliza para moverse al siguiente registro, es DOWN, lo que simula el efecto de oprimir la tecla del cursor hacia abajo. Cada comando, reacciona en forma diferente dependiendo del estado en el que se encuentra el ambiente de Paradox, por ejemplo, si se está editando una tabla, y se oprime la tecla de cursor hacia abajo, el cursor se moverá al siguiente campo, más no al siguiente registro; esto hace que la programación en PAL, sea un poco difícil. Sin embargo, si no se desea tener nada que ver con la programación en PAL, se puede instalar el programador personal, que es un generador de aplicaciones, este programa guía al programador, para crear aplicaciones que incluyan pantallas de captura, reportes, etc. y dando como resultado final el código correspondiente en PAL.

1.3.6 Informix SQL

En este paquete, no se encontraran pantallas de colores con menús y ventanas. El paquete, es más bien al estilo de Lotus, en el que algunas pantallas de menús automatizan las funciones más significativas de una base de datos, tales como crear una tabla, definir o modificar un campo, etc. Se incluye un generador de reportes y un sistema para ejecutar archivos de definición de formatos que se diseñan y escriben en una especie de pseudolenguaje. Pero en esencia Informix es un sistema SQL (Structured Query Language). En Informix, los formatos no se construyen moviendo un cursor o un mouse sino que se escribe un archivo de especificación de formatos, que es una especie de definición de pantalla con instrucciones ejecutables. Este archivo está compuesto de cinco partes:

- 1. Sección de base de datos.-** Esta sección identifica sobre que base de datos va a operar el formato.
- 2. Sección de pantalla.-** Se define como estará distribuida la pantalla.
- 3. Sección de tablas.-** Se identifica a cuales tablas se tendrá acceso.
- 4. Sección de atributos.-** Se describen todos los campos que serán desplegados.
- 5. Sección de instrucciones opcionales.-** Se definen las operaciones que serán realizadas en los campos que se seleccionaron.

En el paquete se incluyen una serie de utilerías, como son el BCHECK que verifica la integridad de los índices, el DBLINK y DBLOAD que se utilizan para transportar datos entre Informix y Lotus 123, Dbase o archivos ASCII. Con la utilería DBSCHEMA, se pueden crear comandos de SQL necesarios para construir tablas o bases de datos.

I.3.7 R:BASE 3.1

Cuando se arranca R:Base, aparece una barra de comando a lo largo de la base superior de la pantalla. Se puede utilizar el mouse o una combinación de teclas para realizar la selección de una serie de menús. Sin embargo el soporte de mouse que ofrece el paquete, es muy reducido, por lo que en ocasiones es más cómodo utilizar el teclado. Lo peor del caso, es que R:Base no utiliza barras de desplazamiento, por lo que se requiere de mucho tiempo para navegar utilizando el mouse especialmente cuando se preparan reportes que son más anchos que la pantalla. También sería deseable que cuando se están generando reportes o formatos, los campos se pudieran mover de posición utilizando el mouse para arrastrarlos de una posición a otra.

El primer menú lista las bases de datos disponibles, si alguna de éstas es un archivo de Dbase, esta podrá ser abierta desde este menú, incluyendo sus índices, sin tener que realizar una función de importación.

Después de seleccionar una base de datos, otro menú despliega la lista de bases de datos que están asociadas con la base de datos que se seleccionó. Al seleccionar una tabla, automáticamente se entra en modo BROWSE (consulta y edición en forma tabular). Otras opciones del menú de tablas permiten modificar las características de la estructura de las tablas así como crear nuevas tablas.

Todos los elementos de una base de datos, tales como reportes, formatos, vistas, etc., se encuentran disponibles desde el menú principal.

Se puede crear un archivo de comandos de R:Base con cualquier editor ASCII, el código es compatible con SQL ANSI. El generador de aplicaciones permite crear aplicaciones basadas en menús aún a los usuarios sin experiencia en programación.

Una gran desventaja de R:Base es su lentitud, sin embargo es recomendable para usuarios que no tienen experiencia en programación.

1.3.8 COMPARATIVO DE CARACTERISTICAS.

A continuación incluimos una tabla en donde se pueden observar algunas de las

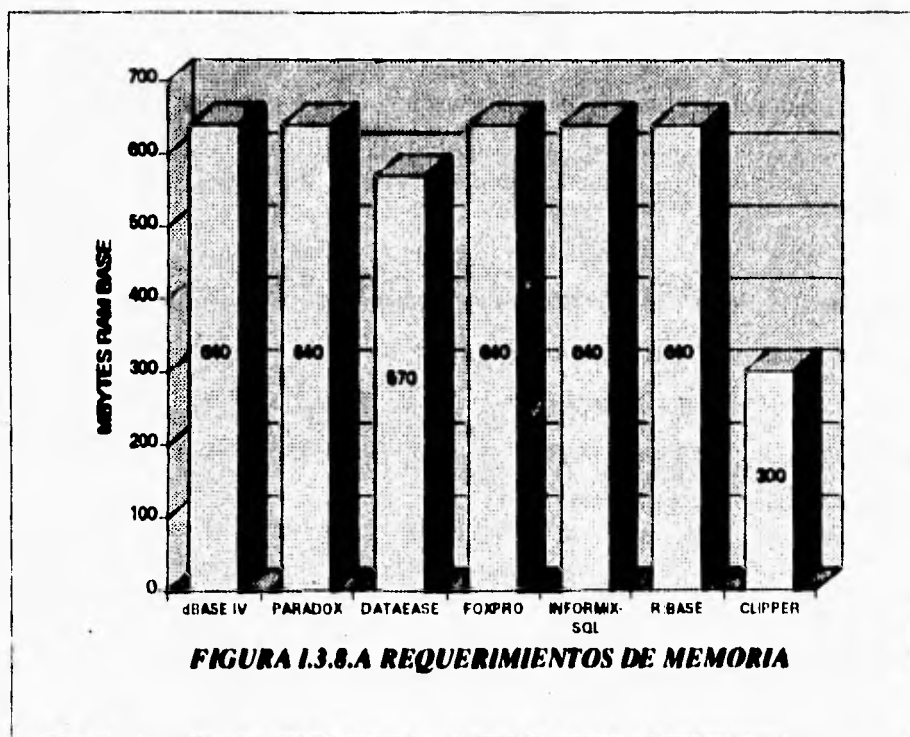
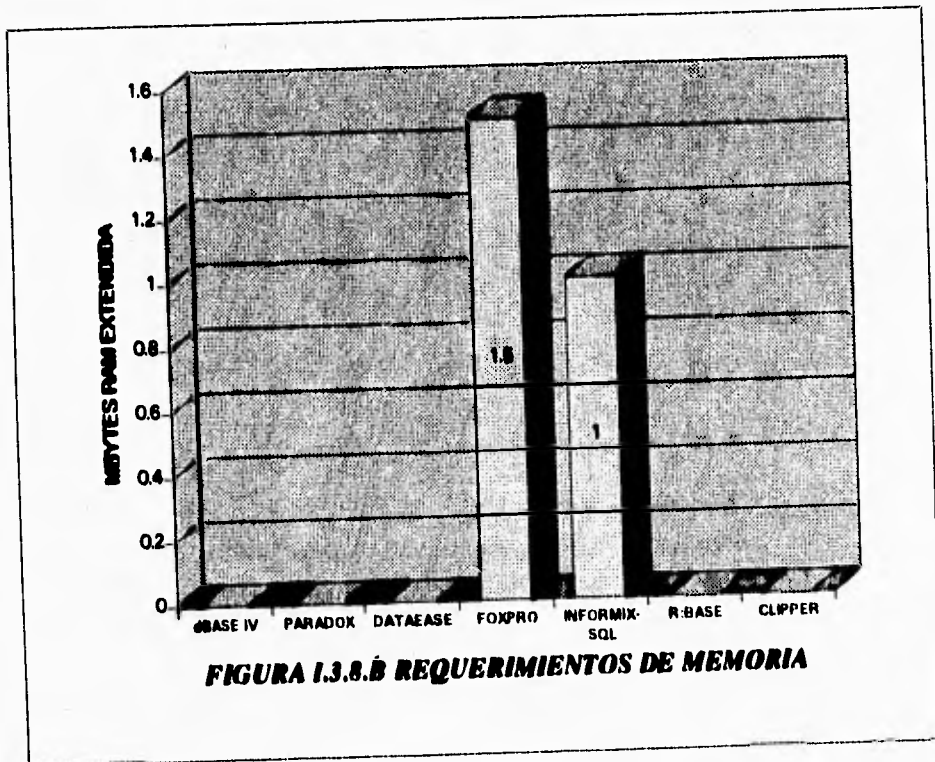


FIGURA 1.3.8.A REQUERIMIENTOS DE MEMORIA

características principales de los manejadores de bases de datos que analizamos en los puntos anteriores. Se incluyen también gráficas en las que se observa más fácilmente las diferencias que para algunos manejadores puede representar una ventaja y para otros una desventaja. Ver figuras 1.3.8.A, 1.3.8.B, 1.3.8.C, 1.3.8.D, 1.3.8.E y 1.3.8.F.



Para obtener las cifras de tiempos de procesamiento, se crearon bases de datos con un total de 4000 registros.

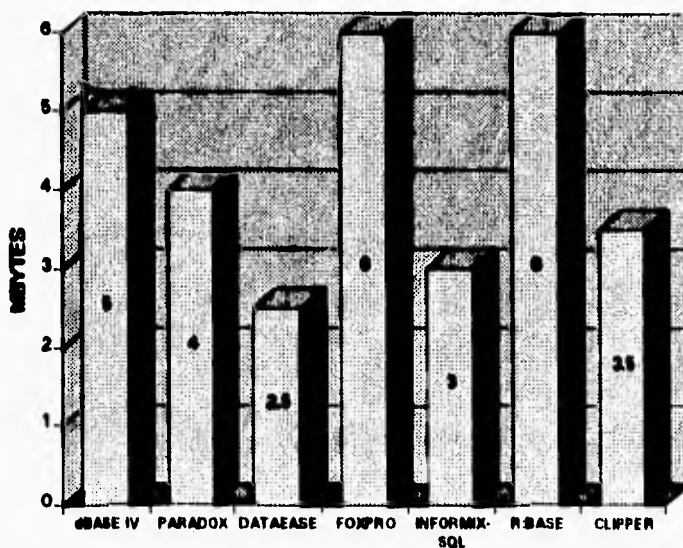


FIGURA 1.3.8.C REQUERIMIENTOS DE ESPACIO EN DISCO DURO

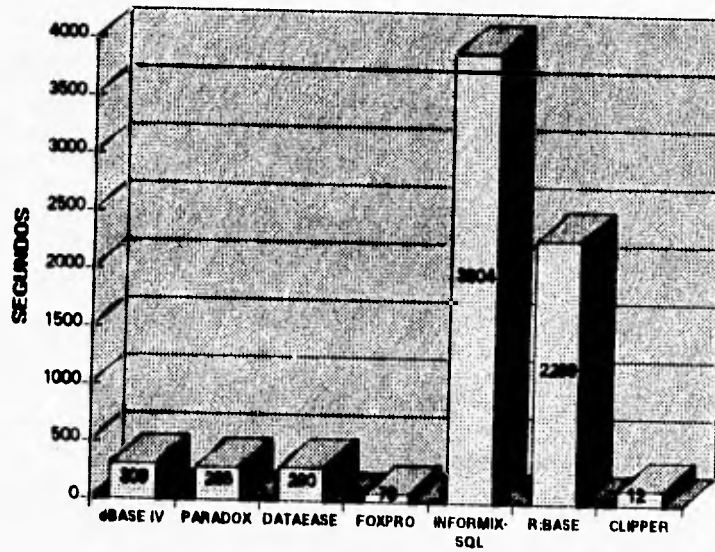


FIGURA 1.3.8.D TIEMPO DE PROCESAMIENTO BATCH

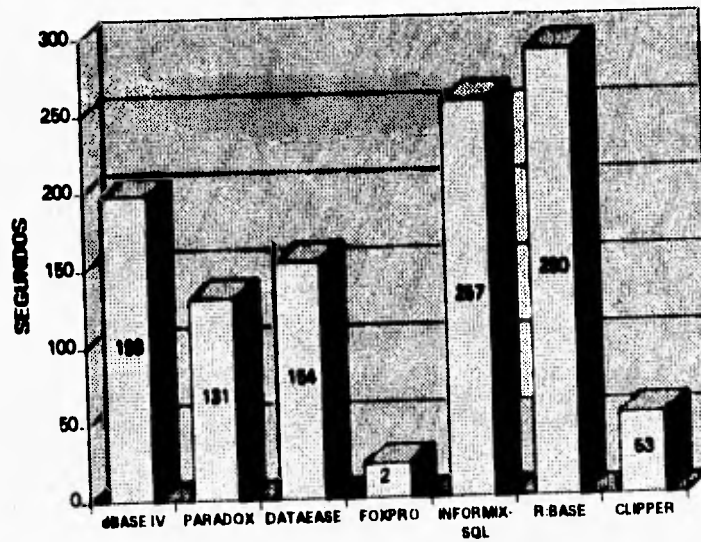


FIGURA 1.3.8.E TIEMPO DE IMPORTACION Y GENERACION DE INDICES

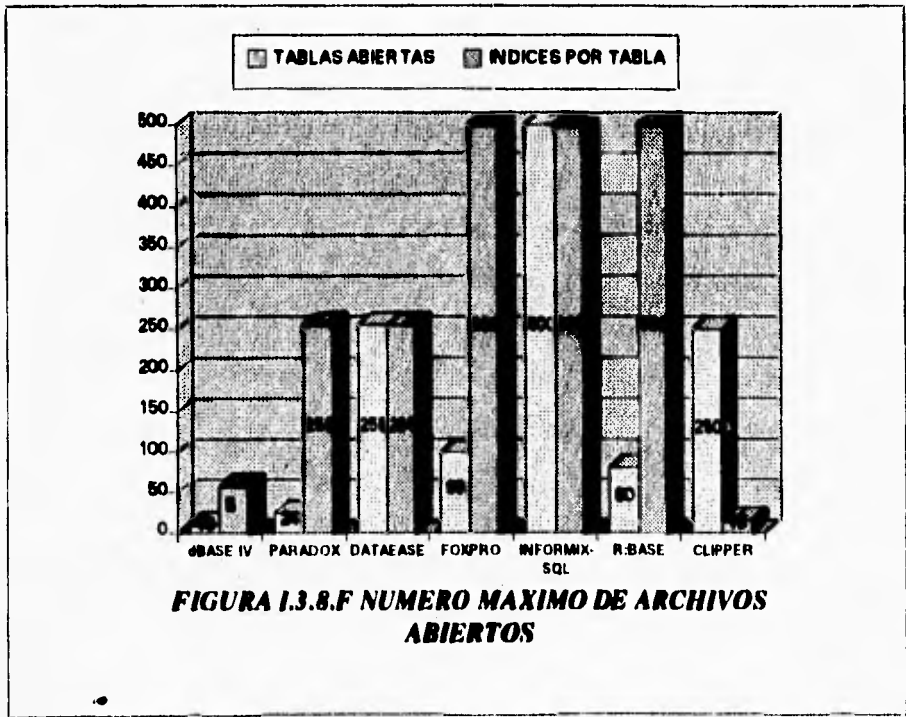


FIGURA 1.3.8.F NUMERO MAXIMO DE ARCHIVOS ABIERTOS

I.4 PLANEACION DE UN PROYECTO DE PROGRAMACION

I.4.1 DEFINICION DEL PROBLEMA

Todo elemento desarrollado por el hombre, primero es una idea en su mente. Los sistemas computacionales, como otros productos de la tecnología, se desarrollan en respuesta a requerimientos detectados. Las fuentes que producen las ideas de productos de programación incluyen las necesidades del cliente generadas externamente, las necesidades internas de la organización, planes de mercadotecnia, y los planes o misiones organizacionales. La mayor parte de las organizaciones que desarrollan productos de programación son muy selectivas al decidir qué productos desarrollarán; no se explotan todas las oportunidades. La decisión de llevar a cabo el proyecto se basa, generalmente, en el resultado de un estudio de factibilidad.

En el figura I.4.1.A de este capítulo, se muestra la secuencia de pasos a seguir para la planeación de un proyecto de programación, mismos que serán explicados a continuación.

El primer paso en la planeación de un proyecto de programación es preparar, en la terminología del cliente, un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. El enunciado definitivo del problema debe de incluir una descripción de la situación actual y de las metas que debe lograr el nuevo sistema.

Para definir el problema es necesario :

1. **Desarrollar un enunciado definitivo del problema por resolver. Incluir una descripción de la situación actual, restricciones del problema y de las metas que se lograrán. El enunciado del problema debe realizarse empleando terminología que sea entendida por el cliente.**
2. **Justificar una estrategia de solución computarizada para el problema.**
3. **Identificar las funciones por realizar, las restricciones, el subsistema del equipo electrónico, el subsistema del producto de programación, y el del personal.**
4. **Determinar los objetivos y requisitos en el nivel del sistema para el proceso de desarrollo y los productos finales.**
5. **Establecer criterios de alto nivel para la aceptación del sistema.**

Para el desarrollo de una estrategia de solución es deseable :

6. **Esbozar varias estrategias de solución, sin considerar las restricciones.**
7. **Realizar un estudio de factibilidad para cada estrategia.**
8. **Recomendar una estrategia de solución, indicando por qué se rechazan las otras.**
9. **Desarrollar una lista de prioridades para las características del producto.**

FIGURA 1.4.1.A. PLANEACION DE UN PROYECTO DE PROGRAMACION.

Hay que tomar en cuenta que el problema del cliente, desde su punto de vista, tal vez sea un problema de nómina, de inventario, o de control del tráfico aéreo, y no problemas de algoritmos de clasificación, o bases de datos relacionales.

La definición del problema requiere de un entendimiento cabal del dominio del problema y del entorno de éste. Las técnicas para obtener este conocimiento, por parte de la persona de planeación, son entrevistas con el cliente, observación de las tareas

problemáticas, y desarrollo de las reales. Esta persona deber ser muy hábil en las técnicas de definición del problema, ya que distintos representantes del cliente tendrán diferentes puntos de vista, sesgos, y prejuicios que influirán en su percepción del área del problema. Además, los representantes del cliente quizás no estén familiarizados con las posibilidades que una computadora pueda ofrecer en su situación, rara vez son capaces de formular sus problemas de modo que sea factible un análisis lógico y algorítmico.

Algunas veces los sistemas computacionales se construyen para aliviar un síntoma y no la causa primaria de un problema. Esto ocurre cuando el problema real se entiende, pero no puede resolverse debido a circunstancias económicas, políticas o sociales, cuando el cliente no es capaz de comunicar el problema real o cuando la persona de planeación no entiende la explicación del cliente sobre el problema.

El segundo paso en la planeación de un proyecto de programación es determinar lo apropiado de una solución computacional. Además de ser eficaz en términos de costo, un sistema computacional debe aceptarse social y políticamente. Para ser eficiente en costo, un nuevo producto de programación debe proporcionar los mismos servicios e información que el sistema antiguo, usando menos tiempo y personal, o proporcionar servicios e información que antes eran inaccesibles. Un sistema que desplace muchos trabajadores puede ser económica y técnicamente posible, pero inaceptable social o políticamente para el usuario.

Habiendo determinado, por lo menos en una forma preliminar, que es apropiada una solución computarizada para el problema, la atención se centra en las funciones de los principales subsistemas del sistema computacional. Un sistema computacional está formado por los subsistemas de personal, equipo y de productos de programación, más las interconexiones entre ellos. El primer subsistema incluye operadores, personal de

mantenimiento y usuarios finales. El segundo comprende el equipo de cómputo y los dispositivos periféricos, y puede tener otros dispositivos como sensores y accionadores para control de proceso, o antenas rastreadoras y radares. El tercer subsistema contiene programas que deben desarrollarse, más programas que ya existen y que pueden emplearse como están o modificándolos.

Las funciones que debe realizar cada subsistema principal se deben identificar, se deben establecer las interacciones entre subsistemas y determinar las restricciones en el desarrollo y operación para cada subsistema principal. Las limitaciones especifican número y tipo de equipos, cantidad y habilidades del personal, y características del producto de programación como funcionamiento, precisión y nivel de confiabilidad. La asignación precisa de funciones entre equipo, programación y personal puede dificultarse durante la planeación preliminar, tal vez sea necesario desarrollar primero un análisis detallado. No obstante, debe intentarse la definición preliminar de las funciones de los subsistemas principales.

METAS Y REQUISITOS.

Dado el enunciado preciso del problema y la indicación de las restricciones que existen para su solución, se pueden formular metas y requisitos preliminares. Las metas son logros por alcanzar; sirven para establecer el marco de referencia para el proyecto de desarrollo del producto de programación. Estas se aplican tanto para el proceso de desarrollo como para los productos finales, y pueden ser cualitativas o cuantitativas :

- **Meta cualitativa para el proceso : el proceso de desarrollo debe de mejorar las habilidades profesionales del personal de control de calidad.**

- **Meta cuantitativa para el proceso :** el sistema se debe entregar en un plazo máximo de 12 meses.
- **Meta cualitativa para el producto :** el sistema debe hacer más interesante el trabajo de los usuarios.
- **Meta cuantitativa para el producto :** el sistema debe reducir el costo de una transacción en un 25%.

Algunas metas se aplican a todos los proyectos y productos. Por ejemplo, todo producto de programación debe de ser útil, confiable, comprensible, y eficiente en costos. Todo proceso de desarrollo debe de producir productos finales a tiempo y dentro de los estimados de costo, y debe permitir que el personal del proyecto aprenda nuevas habilidades. Otras metas, como transportabilidad, entrega anticipada de subsistemas, y facilidad de uso para los no programadores, dependerán de la situación particular.

Los requisitos especifican las capacidades que debe tener un sistema para la solución de un problema. Estos se establecen para la funcionalidad, el rendimiento, el equipo, la programación en el equipo, la programación y las interfaces con el usuario. Los requisitos pueden establecer también estándares de desarrollo y de control de calidad tanto para el desarrollo como para el producto; deben de ser cuantificados siempre que sea posible. Requisitos cuantificados como :

1. La precisión de la fase debe estar dentro de 0.5 grados.
2. La respuesta a interrupciones externas debe de ser de 0.25 segundos máximo.
3. El sistema debe de residir en 50K bytes de memoria principal, excluyendo los archivos residentes.
4. El sistema debe estar operando totalmente el 95% de cada período de 24 horas.

Pueden utilizarse como base para la aceptación del sistema final. Requisitos cualitativos como :

1. La precisión debe ser suficiente para cumplir con el objetivo.
2. El sistema debe producir respuestas en tiempo real.
3. El sistema debe emplear con eficiencia la memoria principal.
4. El sistema debe ser 99% confiable.

Con frecuencia éstos no tienen sentido, pueden producir malos entendidos y desacuerdos entre los clientes y el equipo de desarrollo. Es difícil cuantificar requisitos en la fase de planeación porque, por lo regular, no está claro qué se necesita para resolver el problema, o qué se puede lograr dentro de las restricciones. Sin embargo, se debe realizar un esfuerzo para formular requisitos con significado, y los métodos que se utilizarán para verificar cada requisito.

Las metas y los requisitos de alto nivel se pueden expresar en términos de atributos de calidad que el sistema deberá poseer. Estos atributos de calidad de alto nivel, pueden, a su vez, expresarse en términos de atributos que se pueden obtener en los productos finales. Por ejemplo, la confiabilidad puede significarse en precisión, solidez, integridad y consistencia del código fuente. Cada uno de estos parámetros debe de definirse con cuidado en términos de atributos más específicos del código fuente. Por ejemplo, la precisión puede describirse como la proporción con la que los resultados producidos por el código son suficientemente precisos para satisfacer el uso que de ellos se pretende. Esto se puede traducir en requisitos específicos para cualquier problema en particular. Por ejemplo, "la precisión de la fase debe de ser 0.5 grados" es requisito de precisión que se puede utilizar para establecer la calidad de la programación en un sistema de navegación. La descomposición de atributos cualitativos en características de programas-

fuente se ilustra en la figura 1.4.1.C (BOE76). La figura 1.4.1.B proporciona definiciones para algunos términos empleados en la figura 1.4.1.C(IEE83).

Portabilidad : Facilidad con la que un producto de programación puede ser transferido de un sistema de cómputo a otro o de un ambiente a otro.

Confiabilidad : Capacidad de un programa de realizar una función requerida bajo ciertas condiciones durante un periodo determinado.

Eficiencia : Grado con el que un producto de programación efectúa sus funciones, mediante un mínimo de recursos computacionales.

Exactitud : Especificación cualitativa de ausencia de error. Medida cuantitativa de la magnitud del error, de preferencia expresada como una función del error relativo.

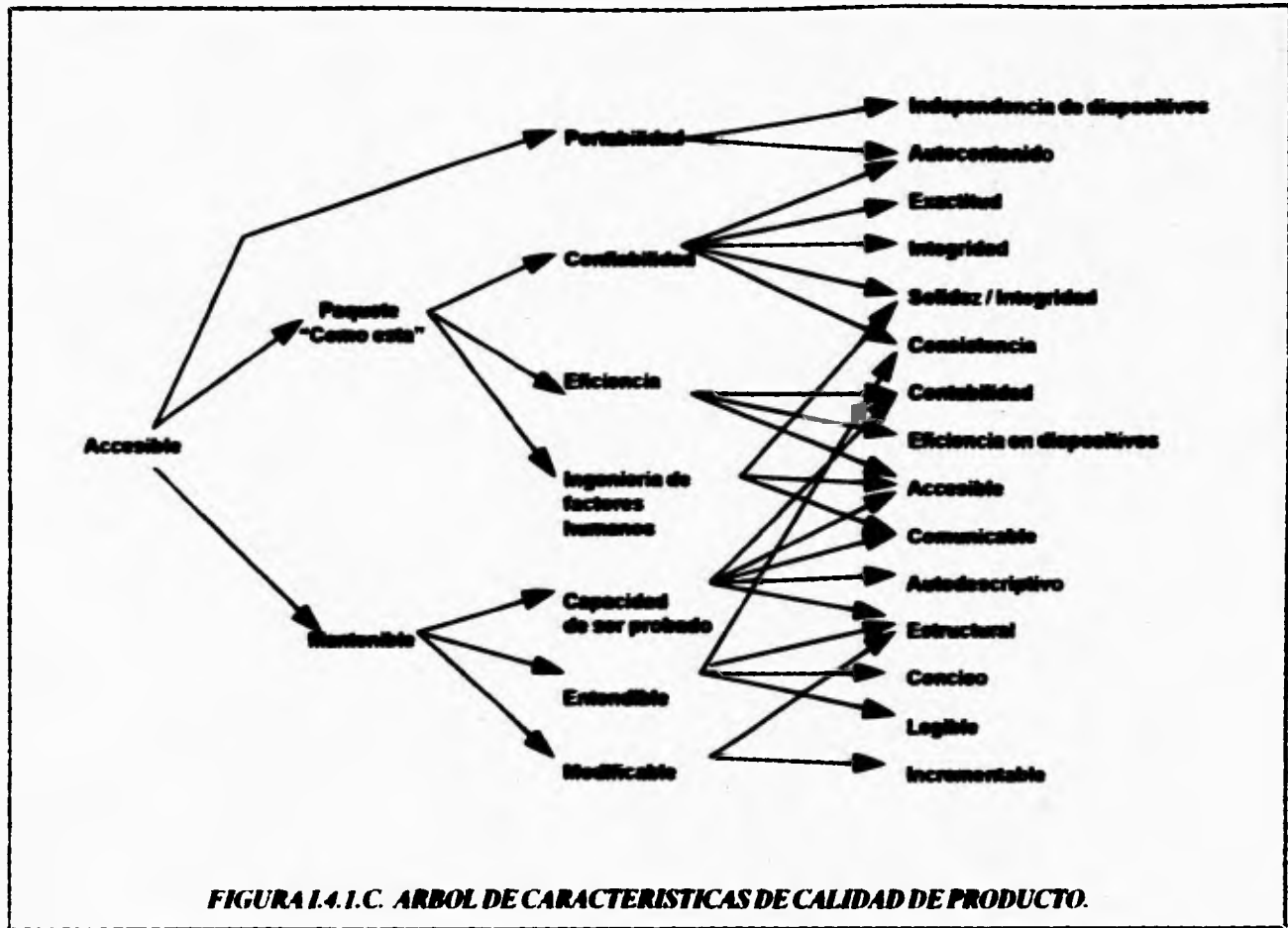
Error : Discrepancia entre una condición o valor calculado y la condición real, específica, o valor correcto teórico.

Solidez : Grado con el que un producto de programación puede continuar operando correctamente, a pesar de la introducción de datos inválidos.

Corrección : Grado en el que un producto de programación está libre de defectos de diseño y de codificación, esto es, libre de fallas. Grado en que un producto de programación cumple los requisitos especificados. Grado en que un producto de programación cumple con las expectativas del usuario.

FIGURA 1.4.1.B. GLOSARIO DE ATRIBUTOS DE CALIDAD (IEE83).

Es importante que los criterios de alto nivel para la aceptación del sistema se definan durante la fase de planeación. La falta de criterios cuantitativos de aceptación establecidos con claridad puede producir graves malentendidos entre el cliente y el encargado de desarrollo. Los criterios de aceptación se deben fijar de tal forma que se puedan verificar por métodos conocidos, como inspección, análisis, o pruebas realizadas



en los productos finales. Cada requisito debe incluir el método que se empleará durante su verificación.

Los planes describen los mecanismos que se ocuparán para lograr las metas y requisitos. Por ejemplo, el objetivo de entregar los productos finales a tiempo se puede expresar en términos de alcanzar logros importantes del proyecto a tiempo. Un logro es un hecho significativo en el ciclo de vida de desarrollo del producto de programación; ejemplos de logros son terminación del análisis de requisitos, terminación del diseño, e integración y prueba con éxito de los componentes del sistema.

Para planear el modo de alcanzar cada logro a tiempo, se deben responder preguntas como las siguientes :

- 1. ¿Cuántos logros son apropiados?**
- 2. ¿Cuándo ocurren?**
- 3. ¿Qué recursos se necesitan para alcanzar cada logro?**
- 4. ¿Quién será el responsable de alcanzarlos?**
- 5. ¿Qué se debe cumplir para obtener cada logro?**
- 6. ¿Exactamente qué significa alcanzar cada logro?**

La consideración de estas preguntas llevará a aspectos como modelos de ciclo de vida, estimación de costos, y cantidad de personal para el proyecto. Las figuras 1.4.1.D y 1.4.1.E presentan algunos factores que deben considerarse cuando se establecen metas y requisitos para los productos de programación y los procesos de su desarrollo.

1. Técnicas de estimación que se utilizarán; precisión requerida
2. Modelo de ciclo de vida, funciones de control, y revisiones
3. Estructura organizacional
4. Nivel de formalidad en especificaciones, planes de prueba, etc.
5. Nivel de verificación y validación
6. Nivel de administración de la configuración requerida
7. Nivel de control de calidad requerido
8. Responsabilidades de seguimiento y mantenimiento
9. Herramientas que se desarrollarán y emplearán
10. Contratación y entrenamiento de personal

FIGURA 1.4.1.D ALGUNOS FACTORES QUE DEBEN CONSIDERARSE EN LA PLANEACION DEL PROYECTO.

1. Nuevas capacidades por proporcionarse
2. Previas capacidades para preservar / mejorar
3. Nivel de complejidad del usuario
4. Requisitos de eficiencia
5. Requisitos de confiabilidad
6. Modificaciones factibles
7. Prioridades de instrumentación y subconjuntos iniciales
8. Requisitos de portabilidad
9. Asuntos de seguridad

FIGURA 1.4.1.E ALGUNOS FACTORES QUE DEBEN CONSIDERARSE AL ESTABLECER METAS DEL PROYECTO.

I.4.2 DESARROLLO DE UNA ESTRATEGIA DE SOLUCION.

La tendencia de utilizar la primera solución que aparece es un problema importante en la integración de software. Una forma de evitarlo es el desarrollar primero una estrategia de solución. Esta no es un plan detallado de solución, sino un enunciado general sobre la naturaleza de las posibles soluciones. Los factores estratégicos incluyen procesamiento por lote o tiempo compartido; base de datos o sistema de archivos; gráficas o texto y procesamiento en tiempo real o en línea. Una estrategia de solución debe considerar todos los factores externos que son visibles para el usuario del producto, y debe redactarse de tal manera que permita caminos alternos para el diseño del producto.

Se deben considerar varias estrategias de solución antes de elegir alguna, aunque las personas que planean deben escoger una o más para poder realizar estudios de factibilidad y estimados de costo preliminares. La estrategia seleccionada proporciona un marco de referencia para el diseño y la instrumentación del producto de programación.

Las estrategias de solución se deben crear sin tomar en cuenta aspectos de factibilidad, puesto que no es posible ser creativo y crítico al mismo tiempo. Por lo regular, una idea no lógica produce otras ideas, y éstas pueden ser muy razonables. A menudo, una composición de las ideas de distintos puntos de vista es la mejor estrategia de solución y ésta aparece sólo después de enumerar todas las soluciones obvias. La generación de ideas se da mejor con grupos de personas con experiencia en técnicas de lluvias de ideas.

La factibilidad de cada estrategia de solución propuesta se puede establecer por el análisis de las restricciones de la solución. Estas establecen las fronteras del espacio de soluciones; el análisis de factibilidad determina si una estrategia de propuesta es posible dentro de dichas fronteras. Una estrategia de solución es factible si las metas y requisitos

del proyecto se pueden satisfacer dentro de las restricciones de tiempo disponible, recursos y tecnología por medio de esa estrategia. Algunas iteraciones y decisiones ponderadas se pueden necesitar para lograr el balance entre factibilidad y restricciones.

Las técnicas para determinar la factibilidad de una estrategia de solución comprenden el estudio de casos, análisis del peor caso, simulación y construcción de prototipos. Un prototipo difiere de un modelo de simulación en que aquél incorpora algunos componentes del sistema real. Las instrumentaciones de prototipos suelen tener una funcionalidad limitada, poca confiabilidad y características de operación pobres. Los prototipos se construyen durante la fase de planeación para examinar aspectos técnicos y para simular despliegues al usuario, formatos de informes y diálogos. El último mecanismo es particularmente útil para obtener una mejor comprensión de las necesidades del usuario.

Cuando se recomienda una estrategia de solución, es muy importante documentar las razones por las que se rechazan otras estrategias. Esto da justificación a la estrategia recomendada, y puede prevenir revisiones equivocadas en fechas posteriores.

Una estrategia de solución debe incluir una lista con prioridades de las características del producto; existen varias razones importantes para establecer dichas prioridades. En algún momento posterior en el ciclo de desarrollo puede necesitarse posponer o eliminar algunas de las capacidades del sistema debido a inconsistencias en los requisitos, cuellos de botella técnicos, o excesos en tiempo o costo. En tales momentos, es esencial disponer de una guía de alto nivel que indique las prioridades de las características fundamentales, de las menos importantes, y de las del tipo "bueno sí". Sin esta guía, un diseñador o programador puede realizar serios errores de juicio, lo que provocaría la insatisfacción del cliente con el producto final. Las prioridades de las características del

producto, también son útiles para indicar la manera en que las capacidades se pueden ir desarrollando en fases dentro de la evolución del sistema. Muchos ingenieros de programación proponen el desarrollo de sistemas como una serie de mejoras sucesivas hacia un sistema núcleo. Las prioridades del producto son útiles en la planeación de las versiones sucesivas que se construirán.

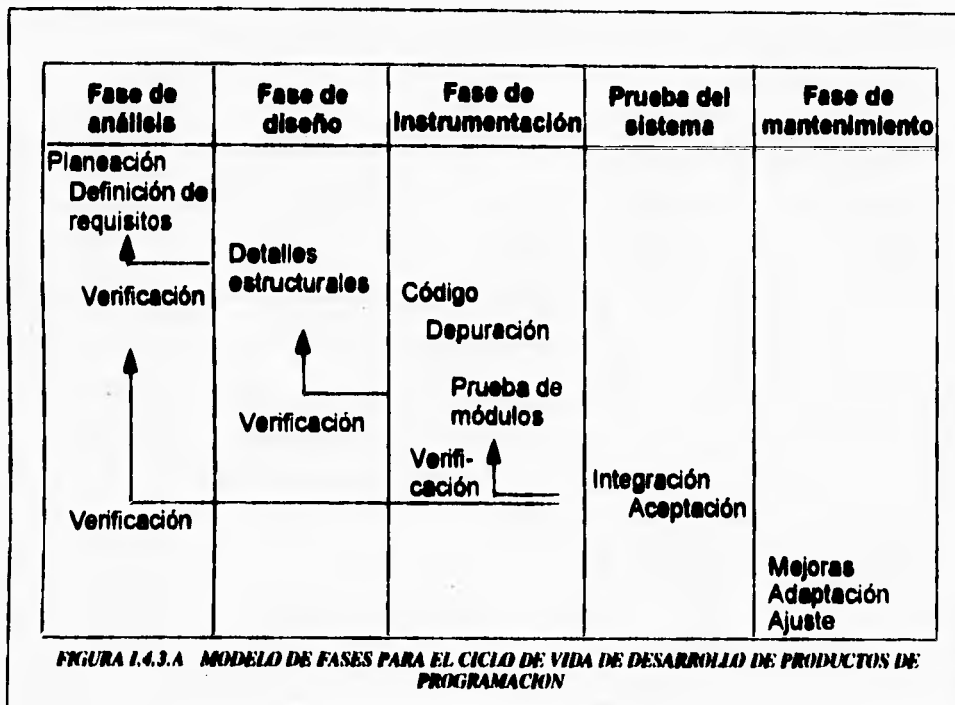
I.4.3 PLANEACION DEL PROCESO DE DESARROLLO

Como primer punto debemos definir el ciclo de vida del producto. El ciclo incluye todas las actividades requeridas para definirlo, desarrollarlo, probarlo, entregarlo, operarlo y mantenerlo. Es esencial definir un modelo de ciclo de vida para cada proyecto de programación, puesto que permite clasificar y controlar las diferentes actividades necesarias para el desarrollo y mantenimiento del producto. Un modelo del ciclo de vida entendido y aceptado por las partes interesadas en el proyecto, mejora la comunicación, permitiendo así una mejor administración, asignación de recursos, control de costos y calidad del producto. Los modelos de ciclo de vida que analizaremos, son los de: fases, costos, prototipos y versiones sucesivas.

MODELO DE FASES DEL CICLO DE VIDA

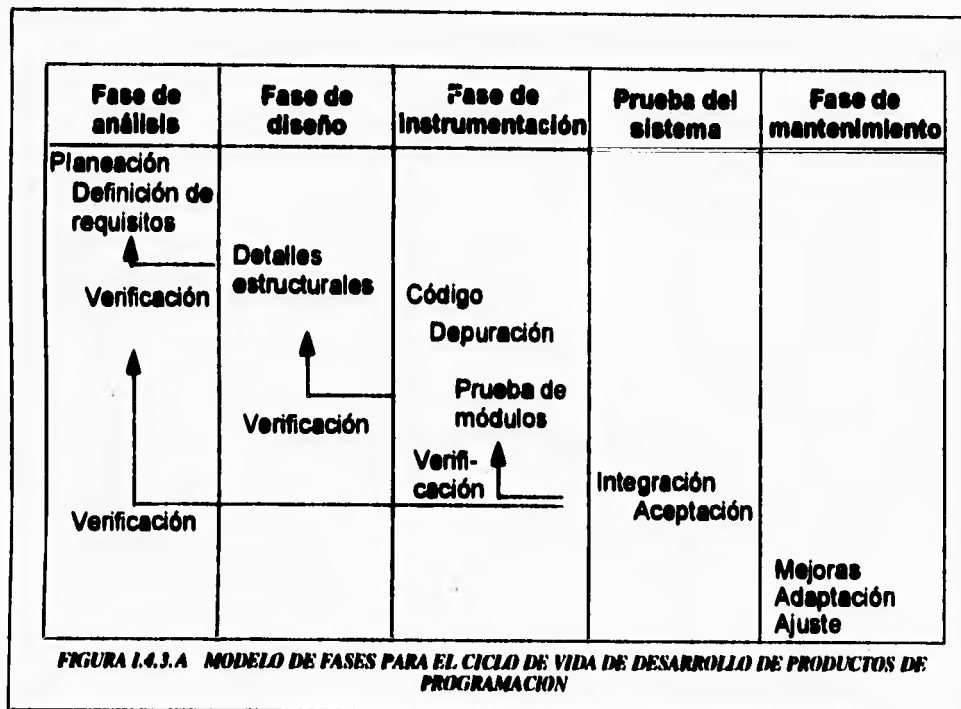
El modelo de fases divide el ciclo de vida del producto de programación, en una serie de actividades sucesivas; cada fase requiera cierta información de entrada, procesos y resultados, cada uno bien definido. Se necesitan recursos para terminar los procesos de cada fase, y cada una de ellas se efectúa mediante la aplicación de métodos explícitos, herramientas y técnicas. Este modelo, se representa en el diagrama de la figura I.4.3.A y esta compuesto por las siguientes fases:

- Analisis
- Diseño
- Instrumentación
- Pruebas
- Mantenimiento



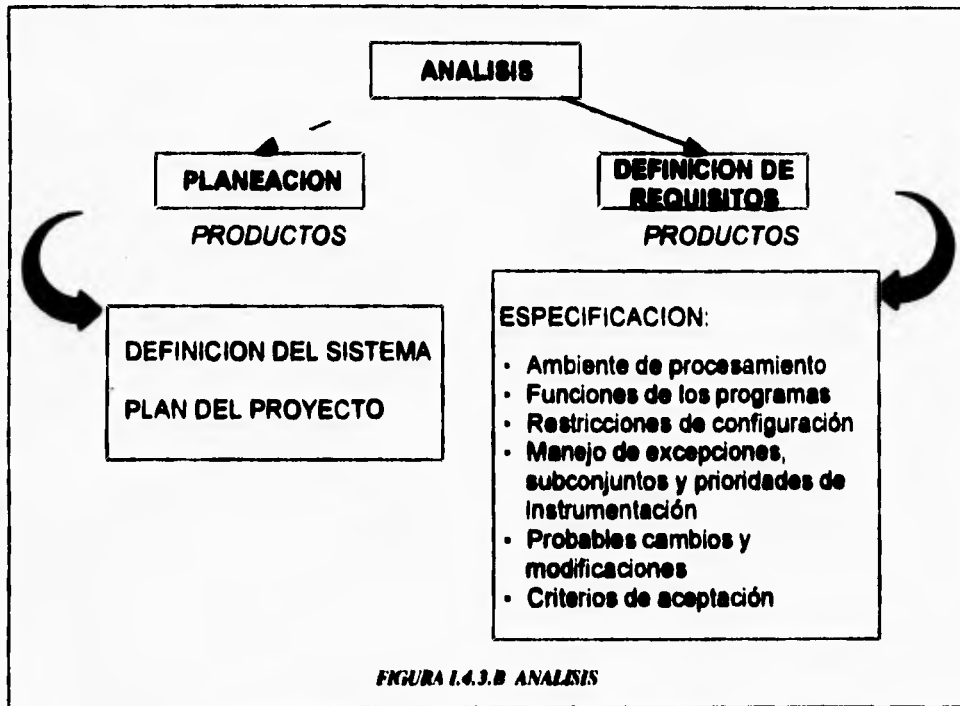
- **Análisis**

Consta de dos subfases (figura 1.4.3.B): planeación y definición de requisitos. Estas subfases, incluyen la comprensión del problema del cliente, estudio de factibilidad, desarrollo de la estrategia de solución recomendada, determinación de los criterios de aceptación, y planeación del proceso de desarrollo. Los productos de la planeación son: la definición del sistema y el plan del proyecto. La definición, por lo regular, se expresa en español o en algún otro lenguaje natural, y puede contener cuadros, figuras, gráficas y ecuaciones de distintos estilos. La notación exacta empleada en la definición, depende mucho del área del problema. Los puntos a seguir en la definición son (figura 1.4.3.C):



- **Análisis**

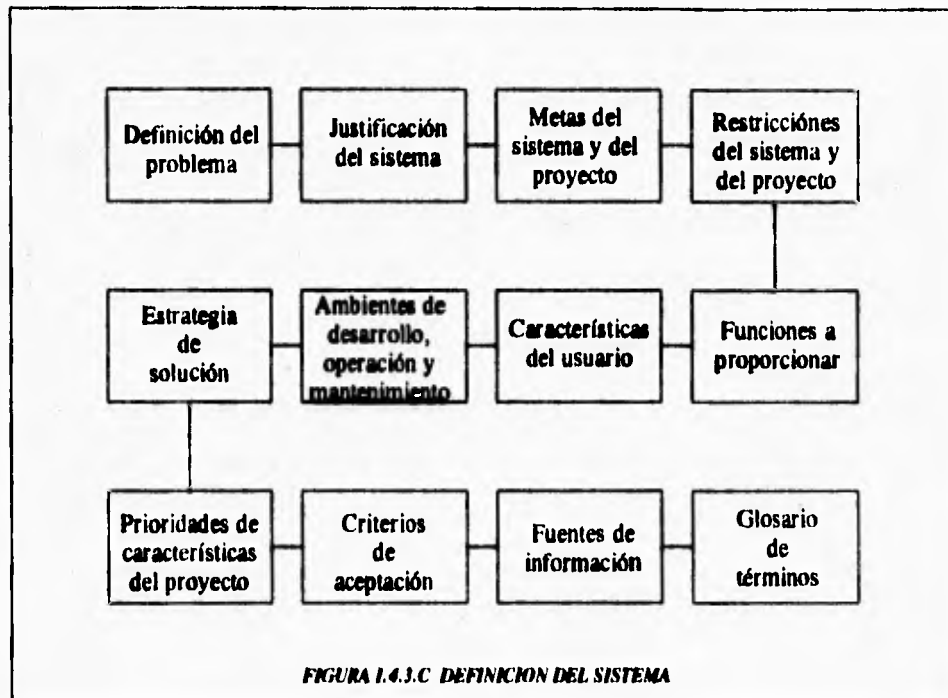
Consta de dos subfases (figura 1.4.3.B): planeación y definición de requisitos. Estas subfases, incluyen la comprensión del problema del cliente, estudio de factibilidad, desarrollo de la estrategia de solución recomendada, determinación de los criterios de aceptación, y planeación del proceso de desarrollo. Los productos de la planeación son: la definición del sistema y el plan del proyecto. La definición, por lo regular, se expresa en español o en algún otro lenguaje natural, y puede contener cuadros, figuras, gráficas y ecuaciones de distintos estilos. La notación exacta empleada en la definición, depende mucho del área del problema. Los puntos a seguir en la definición son (figura 1.4.3.C):



1. Definición del problema
2. Justificación del sistema
3. Metas del sistema y del proyecto
4. Restricciones del sistema y del proyecto
5. Funciones que se proporcionarán (equipo, programación, personal)
6. Características del usuario
7. Ambientes de desarrollo, operación y mantenimiento
8. Estrategia de solución
9. Prioridades para las características del sistema
10. Criterios de aceptación del sistema

11. Fuentes de información

12. Glosario de términos



El plan del proyecto contiene el modelo del ciclo de vida que se utilizará, la estructura organizacional del proyecto, la programación preliminar del desarrollo, estimados preliminares de costos y recursos, así como de personal, herramientas y técnicas que se emplearán, y estándares que se seguirán. Los elementos que se deben incluir en esta subfase (figura 1.4.2.D), son:

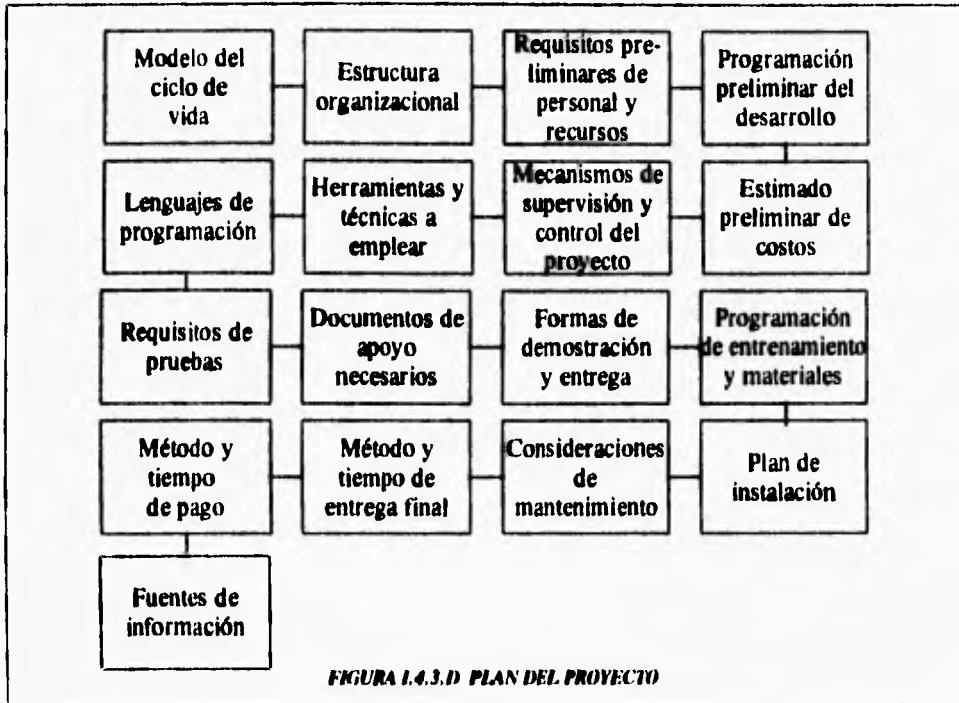
1. Modelo del ciclo de vida
Terminología, logros, productos finales
2. Estructura organizacional

Estructura de administración, de equipos, de distribución de trabajo, definición de puestos

- 3. Requisitos preliminares de personal y recursos**
Programación de personal y recursos
- 4. Programación preliminar del desarrollo**
Gráficas PERT, gráficas de Gantt
- 5. Estimado preliminar de costos**
- 6. Mecanismos de supervisión y control del proyecto**
- 7. Herramientas y técnicas que se emplearan**
- 8. Lenguajes de programación**
- 9. Requisitos de pruebas**
- 10. Documentos de apoyo necesarios**
- 11. Formas de demostración y entrega**
- 12. Programación de entrenamiento y materiales**
- 13. Plan de instalación**
- 14. Consideraciones de mantenimiento**
- 15. Método y tiempo de la entrega final**
- 16. Método y tiempo del pago**
- 17. Fuentes de información**

Durante la fase de planeación, los estimados de costos y la programación del trabajo serán preliminares, ya que normalmente, no es posible realizar estimaciones precisas sin haber realizado algo de diseño, los estimados de costos, por lo tanto, inevitablemente son preliminares. Las prácticas actuales de contratación, requieren que el costo final y la programación se proporcionen durante la fase de planeación. Esta situación, aunada a la naturaleza competitiva del medio, es una de las principales razones de los excesos en costos y las entregas retrasadas de los productos de programación. Reconociendo esta

realidad, muchas organizaciones utilizan una serie sucesiva de estimaciones de costos y programación. Los estimados preliminares, se preparan durante la fase de planeación, su redefinición, se presenta en la revisión preliminar del diseño; el costo y la programación finales se establecen en la revisión final del diseño.



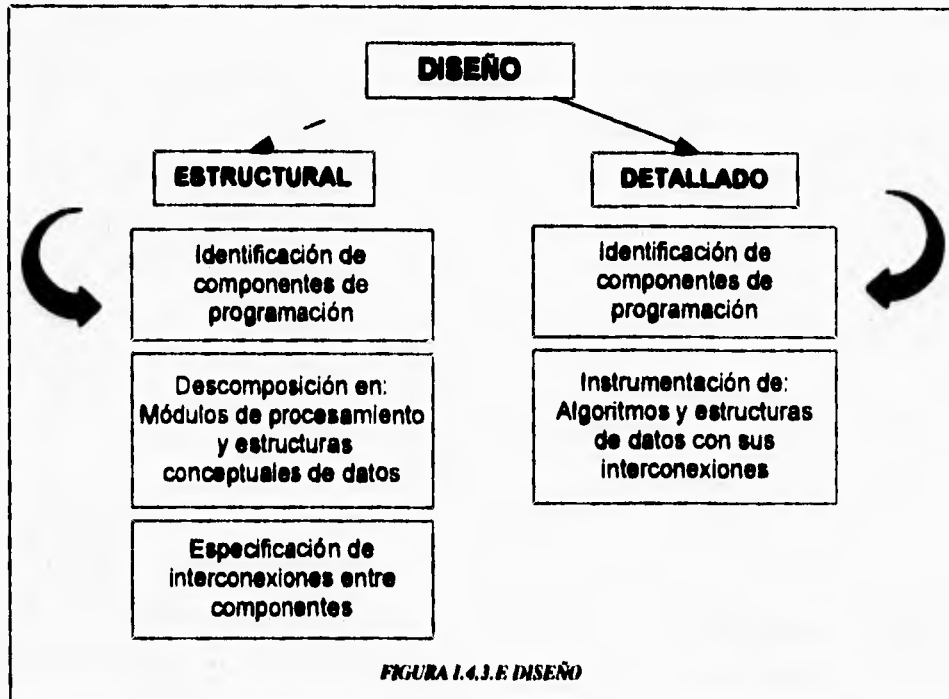
La definición de requisitos, se refiere a la identificación de las funciones básicas del componente de programación en un sistema de equipo, personal, programación. Se pone atención en las funciones y restricciones bajo las cuales se deben desarrollar. La decisión de cómo se instrumentará la programación se retrasa hasta la fase de diseño.

El producto de la definición de requisitos, es una especificación que describe el ambiente de procesamiento, las funciones requeridas de los programas, restricciones de configuración sobre los programas (tamaño, velocidad, configuración de equipo), manejo de excepciones, subconjuntos y prioridades de instrumentación, cambios probables y modificaciones factibles, así como los criterios de aceptación del producto de programación.

- **Diseño**

En el modelo de fases, el diseño de programación, viene después del análisis. El diseño se refiere a la identificación de los componentes de la programación (funciones, flujos de datos y almacenamientos), especificando las relaciones entre ellos, la estructura de la programación, y manteniendo un registro de las decisiones, proporcionando un documento base para la instrumentación. El diseño, se divide en estructural y detallado (figura I.4.3.E).

El diseño estructural, comprende la identificación de los componentes de la programación, su desacoplamiento y descomposición en módulos de procesamiento y estructuras de datos conceptuales, y la especificación de las interconexiones entre componentes. El diseño detallado, se refiere a detalles de cómo empacar módulos de procesamiento, y cómo instrumentar los algoritmos, las estructuras de datos y sus interconexiones.

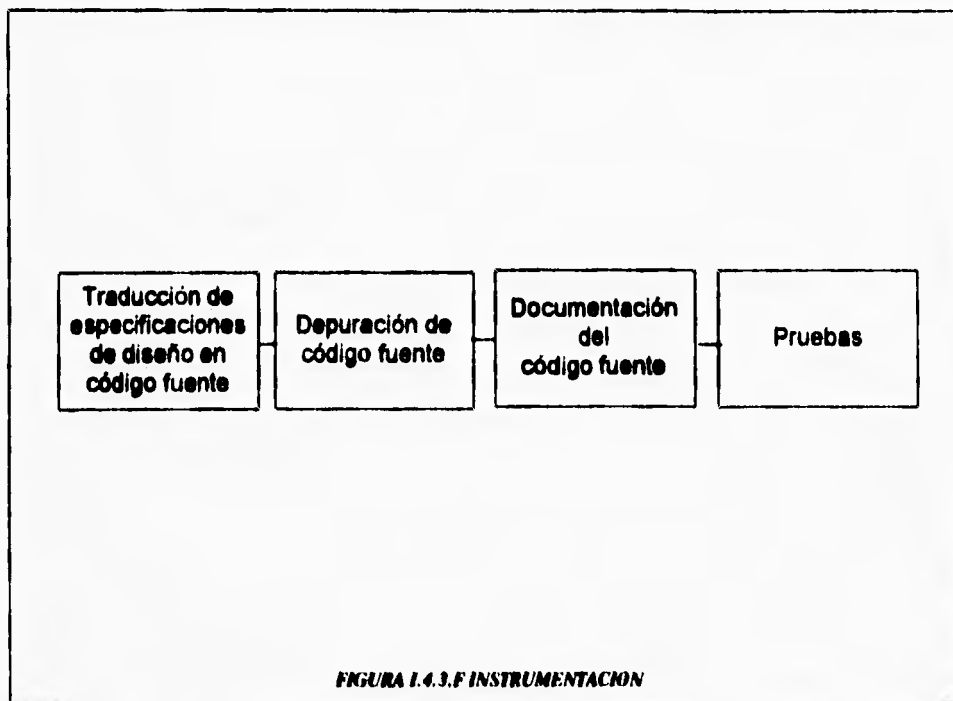


Este diseño, se relaciona con la adaptación del código existente, modificación de algoritmos estandar, invención de nuevos algoritmos, diseño de representaciones de datos, e integración del producto final. Diseño detallado, no es igual que instrumentación. El primero, está muy influido por el lenguaje de programación, pero no tiene que ver con aspectos sintácticos del mismo o con un nivel de detalle como evaluación de expresiones y estatutos de asignación.

- **Instrumentación**

La fase de instrumentación (figura 1.4.3.F) en el desarrollo del producto, incluye la traducción de las especificaciones del diseño en código fuente, así como su depuración,

documentación y pruebas. Los lenguajes de programación modernos, proporcionan muchas características para mejorar la calidad del código fuente, como elementos estructurados, tipos de datos predefinidos o definidos por el usuario, verificación de tipos, reglas flexibles de cobertura, mecanismos para manejo de excepciones, elementos concurrentes, y módulos con compilación separada. Algunas de estas características, se pueden simular en los lenguajes primitivos, mediante un estilo disciplinado de programación.



Los errores descubiertos durante la fase de instrumentación, pueden ser errores en las interfases de datos entre rutinas, errores lógicos en los algoritmos, errores en las estructuras de datos, y de falta de consideración de casos de procesamiento. Además, el

código fuente puede contener: errores de requisitos, que indican alguna omisión de las necesidades del usuario en el documento de requisitos; errores de diseño, que reflejaran una mala traducción de requisitos en especificaciones y por último, errores de instrumentación debidos a una mala traducción de especificaciones en código fuente. Una de las metas principales del modelo de fases para el desarrollo de productos de programación, es la eliminación de errores de requisitos y diseño, antes de iniciada la instrumentación. Es muy caro eliminar errores del analisis y el diseño del código fuente durante la instrumentación y las pruebas.

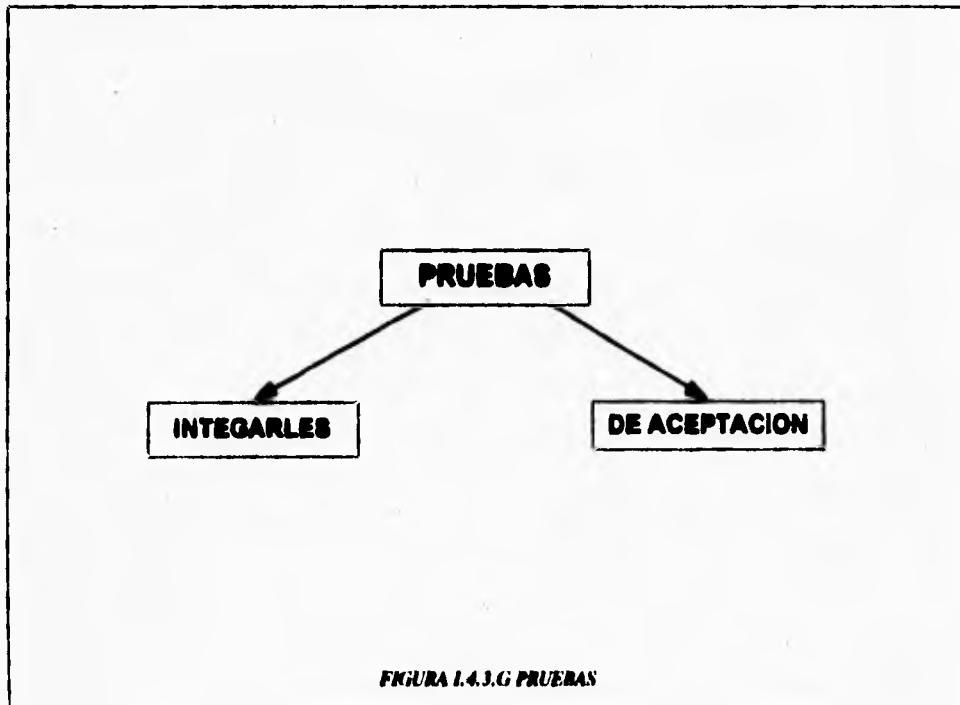
- **Pruebas**

Las pruebas del sistema comprenden dos tipos de actividades: pruebas de integración y de aceptación (figura I.4.3.G). El desarrollo de una estrategia para integrar los componentes de un sistema de programación, en una unidad funcional, requiere una planeación cuidadosa de modo que se disponga de los modulos cuando estos se necesiten. Las pruebas de aceptación, se relacionan con la planeación y ejecución de varios tipos de pruebas para demostrar que el sistema de programación instrumentado, satisface las necesidades establecidas en el documento de requisitos.

Una vez aceptado por el cliente, el sistema de programación se entrega para operación y se inicia la fase de mantenimiento del modelo de ciclo de vida por fases. Las actividades de mantenimiento, incluyen mejoras de las capacidades, adaptación a nuevos ambientes de procesamiento, y corrección de fallas del sistema.

El desarrollo de productos de programación, nunca se lleva a cabo como una sucesión suave de actividades como se muestra en la figura de cascada. Existe más interacción y empalme entre las fases de lo que se puede indicar en una representación simple de dos

dimensiones. Sin embargo, el modelo de fases de vida es válido para el proceso de desarrollo en situaciones donde es posible redactar un conjunto razonablemente completo de especificaciones para el producto de programación, al principio del ciclo de vida. Esto sucede cuando los encargados del desarrollo han evolucionado previamente sistemas similares.



MODELO DEL COSTO DE ALGÚN PROYECTO.

El costo del proyecto es la suma de los costos incurridos en cada fase, y éstos, a su vez, incluyen los costos de la realización de los procesos y preparación de los documentos de

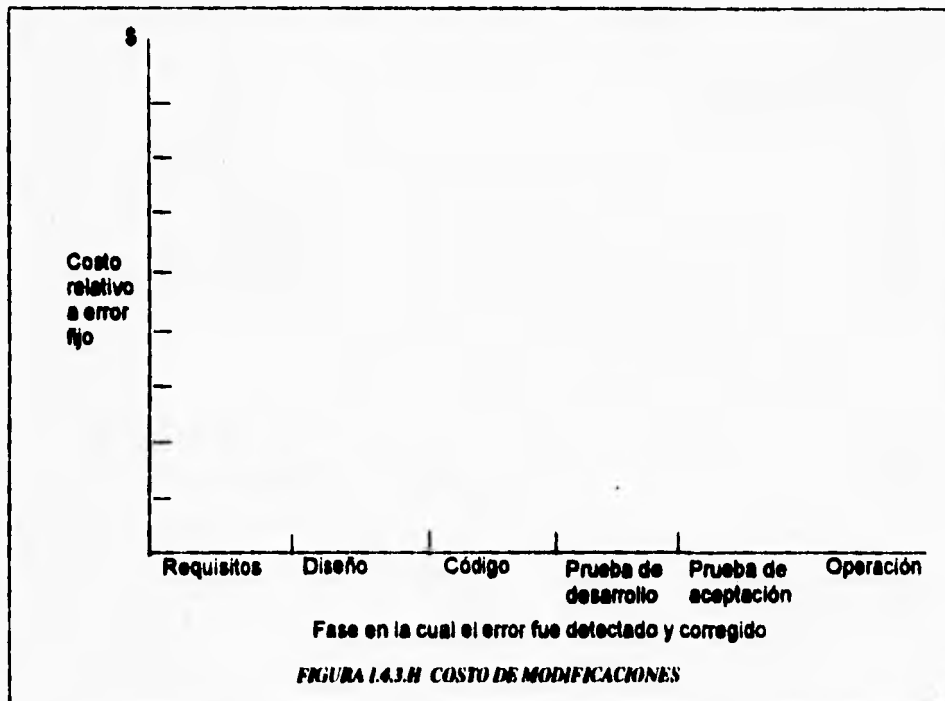
esa fase más los costos de verificación de la consistencia de estos productos con los de las fases previas. Las modificaciones y correcciones a los productos de las fases previas son necesarias, dado que durante el desarrollo de la fase actual se encontrarán imprecisiones, inconsistencias y omisiones en sus productos y además, porque los requisitos, programación, prioridades o presupuesto del cliente pueden cambiar, hecho que producirá modificaciones.

Dado este punto de vista del ciclo de vida de un producto, no es difícil entender por qué es tan caro efectuar modificaciones o correcciones a la especificación de requisitos para la producción de software o documentos del diseño cuando se están en fases posteriores. No solo es necesario modificar los documentos, sino que todos los productos intermedios deben actualizarse y en cada fase siguiente se necesitan coordinar más gente y más detalles. La figura 1.4.3.H ilustra el costo relativo de hacer un cambio en función de la fase en la que el cambio se hace. Se puede observar que es prácticamente 100 veces más costoso hacer un cambio a los requisitos durante la prueba del sistema, que hacerlo durante su definición, y que la figura se presenta con una escala semi logarítmica por lo que la línea recta indica un crecimiento exponencial en el costo.

MODELO DE PROTOTIPO PARA EL CICLO DE VIDA.

Un prototipo es una representación o modelo del producto de programación, que a diferencia de un modelo de simulación incluye componentes del producto real. Por lo regular, un prototipo tiene un funcionamiento limitado en cuanto a capacidades, confiabilidad o eficiencia. Hay varias razones para desarrollar un prototipo; una, es ilustrar los formatos de datos de entrada, mensajes, informes y diálogos al cliente; éste es un mecanismo adecuado para explicar opciones de procesamiento y tener un mejor

entendimiento de sus necesidades. La segunda razón, es para explorar aspectos técnicos del producto propuesto. Con frecuencia, una decisión importante del diseño, depende por ejemplo, del tiempo de respuesta del controlador de un dispositivo o de la eficiencia de un algoritmo de clasificación: en estos casos el prototipo puede ser la mejor o única manera de resolver el problema. La tercera razón para desarrollar un prototipo, es cuando el modelo de análisis ->diseño-> instrumentación, es inadecuado.



El modelo de fases se aplica cuando se puede redactar un conjunto razonablemente completo de especificaciones al inicio del ciclo de vida. Pero algunas veces, no es posible definir el producto sin un desarrollo exploratorio, y en ocasiones no es clara la manera de proceder para la mejora del sistema hasta que no se instrumenta y evalúa una versión.

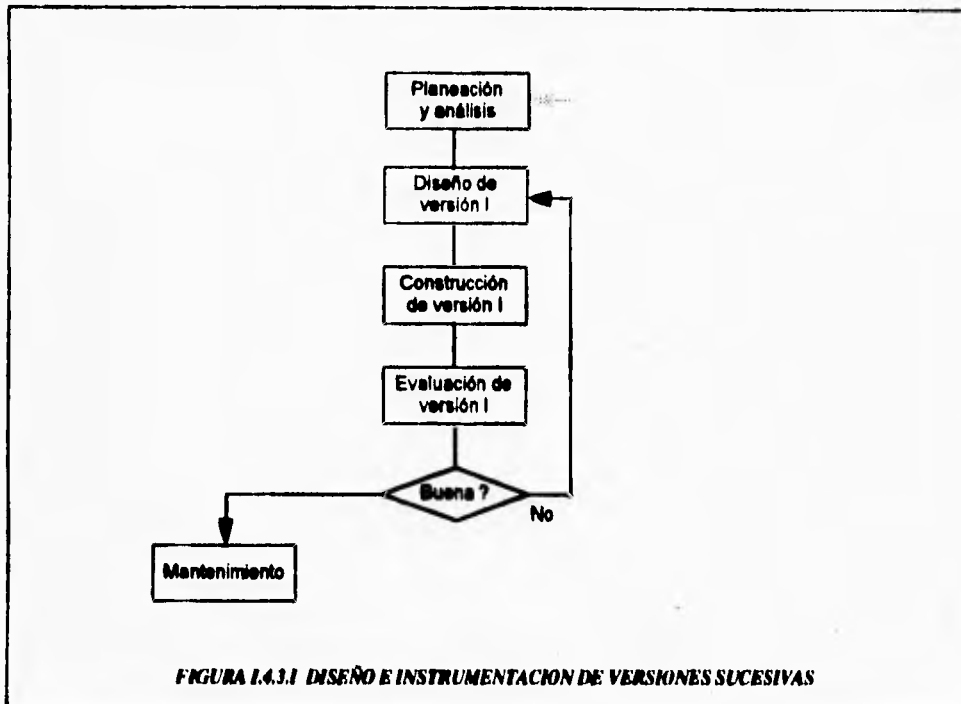
El prototipo por desarrollar en un proyecto, depende de la naturaleza del producto. Se pueden desarrollar nuevas versiones de un producto ya existente con el modelo de fases y sin ningún prototipo. El avance de un producto totalmente nuevo, probablemente requiera de prototipo durante las fases de planeación y análisis, o el producto se puede desarrollar como una serie de diseños e instrumentaciones.

MODELO DE VERSIONES SUCESIVAS

Cada versión, es un sistema funcional y capaz de realizar trabajo útil. La figura I.4.3.I ilustra la fase de análisis seguida por el diseño de instrumentación de versiones sucesivas en un proceso iterativo. Las líneas punteadas, indican que al conseguir la versión I, puede necesitarse más análisis antes de diseñar la versión I+1.

Ahora bien en la figura I.4.3.J, las versiones I a la N del producto, se diseñan antes de cualquier actividad de instrumentación. En este caso, las características de cada diseño sucesivo, serán planeadas durante la fase de análisis. Las líneas punteadas, indican que la instrumentación de cualquier versión puede demostrar la necesidad de mayor análisis y diseño antes de la puesta en marcha de la última versión.

En realidad, el ciclo de desarrollo de un producto de programación, es una combinación de los distintos modelos presentados. Las organizaciones y proyectos especiales pueden adoptar alguno de estos modelos en particular; sin embargo, ciertos elementos se encuentran en todo proyecto de programación.



El conjunto mínimo de documentos que deben generarse durante el desarrollo, incluye, una definición de requisitos, especificación del diseño, plan de pruebas y manual del usuario. La definición de la documentación, programación, logros y revisiones, se debe realizar durante la fase de planeación. Por último hay que recordar que contar con un modelo, aunque este no sea el más adecuado, es mejor que no tener ninguno.

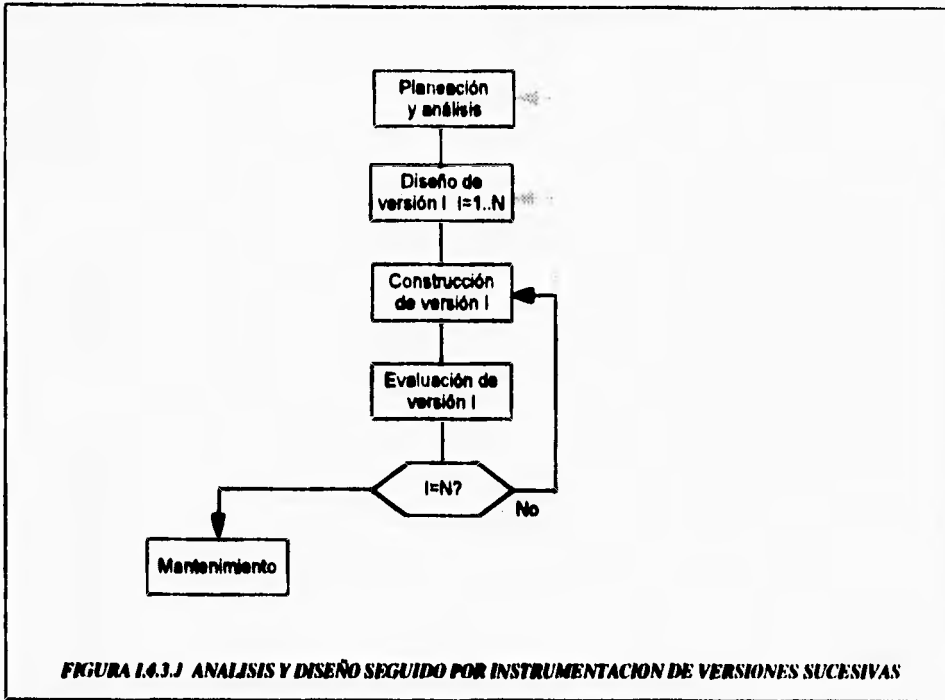


FIGURA 1.4.3.J ANALISIS Y DISEÑO SEGUIDO POR INSTRUMENTACION DE VERSIONES SUCESIVAS

I.4.4 OTRAS ACTIVIDADES DE LA PLANEACION

PLANEACIÓN PARA LA ADMINISTRACIÓN DE LA CONFIGURACIÓN Y EL CONTROL DE CALIDAD.

Se refiere al control de cambios en los productos finales, registro y control de su situación, y mantenimiento de la biblioteca de programas, que es el depósito central de la información del proyecto. El control de calidad desarrolla los estándares del proyecto y supervisa que se respeten, lleva a cabo auditorías de los procesos y productos, y realiza las pruebas de aceptación.

Durante la fase de diseño, los procedimientos de administración de configuración y control de calidad se emplean para controlar y supervisar los requisitos de las especificaciones del diseño, así como para verificar el cumplimiento de los estándares. Durante la fase de instrumentación y pruebas se supervisan los requisitos, las especificaciones de diseño y el código fuente, y en la fase de pruebas, se realizan las pruebas de aceptación. Estas actividades se deben considerar durante la fase de planeación, y se les debe asignar los recursos necesarios para que se realicen con éxito.

PLANEACIÓN PARA VERIFICACIÓN Y VALIDACIÓN EXTERNAS.

La verificación asegura que los productos estén completos y sean consistentes con otros y con las necesidades del cliente. Así, una organización externa puede verificar que las especificaciones de diseño estén completas y que sena consistentes con la definición del sistema y las especificaciones del software, que el código fuente sea consistente con el diseño y los requisitos. La validación se relaciona con la calidad del software en su medio de operación, y suele consistir en la planeación y realización de pruebas. En los proyectos con verificación y validación externa, las pruebas las efectúa la organización independiente, lo que produce sistemas de programación de alta calidad, sin embargo, su

costo puede ser de hasta 25% del costo total del desarrollo, y con base en esto deben analizarse los beneficios.

PLANEACION DE LAS HERRAMIENTAS Y TECNICAS ESPECIFICAS DE CADA FASE.

Para el desarrollo de la especificación de requisitos, el diseño estructural y detallado, y el código fuente, se pueden emplear herramientas automatizadas, notaciones especializadas y técnicas modernas. Lo mismo sucede en las pruebas de módulos, sistema y aceptación. Para controlar el desarrollo se utilizan herramientas de administración. El empleo de estas herramientas, técnicas y notaciones necesita tiempo para su uso y entrenamiento. Esto debe ser anticipado durante la fase de planeación.

OTRAS ACTIVIDADES

Son la preparación de estimados preliminares de costos para el desarrollo del producto, el establecimiento de una programación del proyecto, del personal requerido, de estimados preliminares de recursos de cómputo y del personal necesario para operar y mantener el sistema.

CAPITULO II.

**IDENTIFICACION DE LA PROBLEMÁTICA Y PROPUESTAS
DE SOLUCION.**

II.1 ANTECEDENTES

La Subdirección de Relaciones Públicas del Instituto Nacional de Bellas Artes, emite dos documentos en los que se indica la programación de actividades que coordina o que se presentan en los recintos del I.N.B.A. (Instituto Nacional de Bellas Artes), y del I.N.A.H. (Instituto Nacional de Antropología e Historia); uno de estos documentos se denomina programación mensual y el otro programación semanal.

Ver figura II.1.A

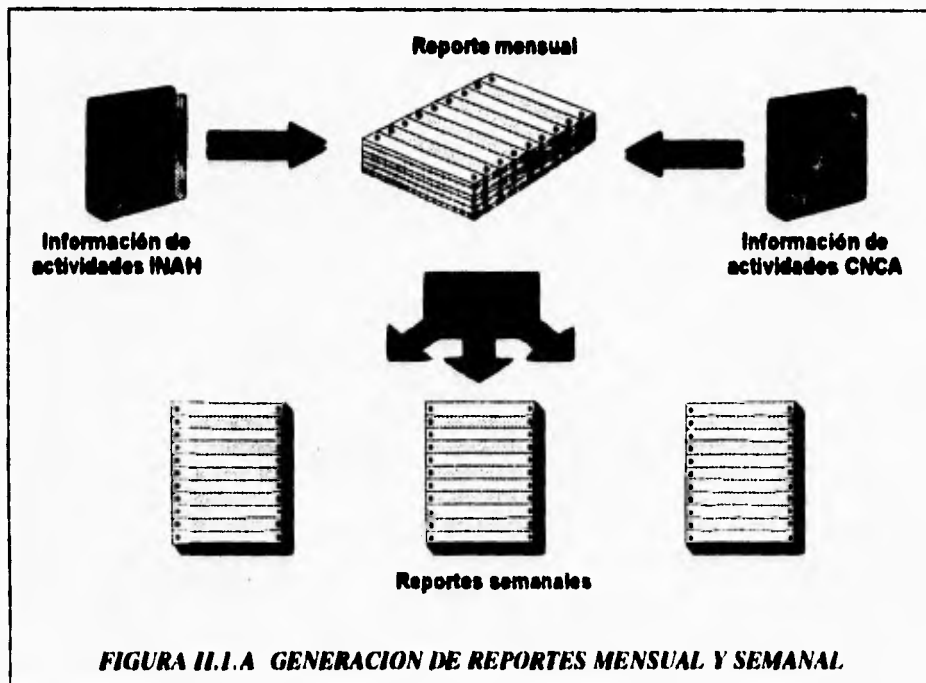


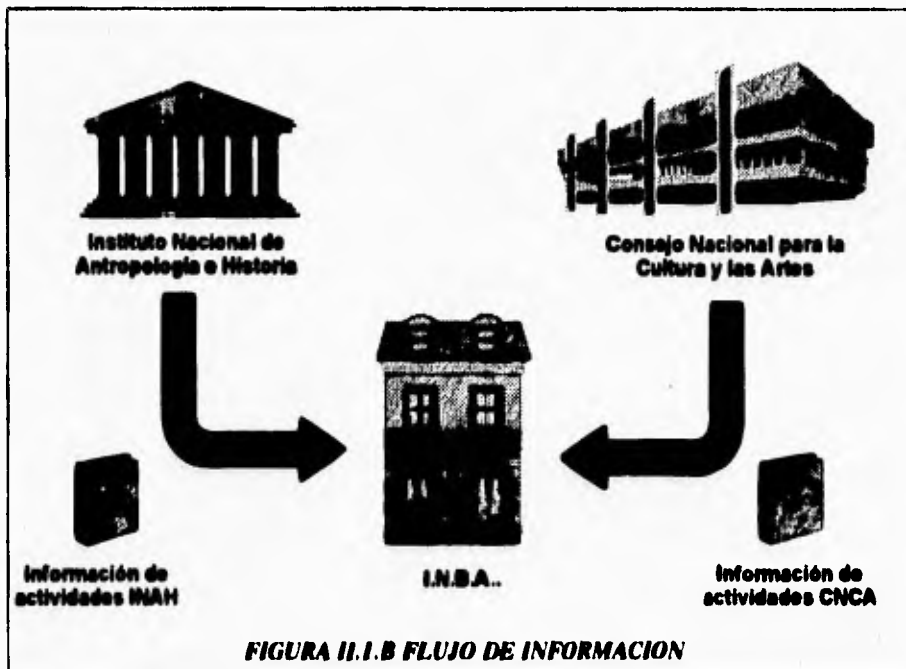
FIGURA II.1.A GENERACION DE REPORTES MENSUAL Y SEMANAL.

Estos documentos se envían a diversos funcionarios de las diferentes áreas de difusión y las diferentes divisiones del Consejo Nacional para la Cultura y las Artes, del I.N.A.H. y del mismo I.N.B.A.

Entre las actividades de la Subdirección de Relaciones Públicas, está la de proporcionar información relacionada a las actividades culturales que organizan los Institutos mencionados en el párrafo anterior, así como la información referente a los recintos en los que se llevan a cabo las actividades.

La Subdirección de Relaciones Públicas del I.N.B.A. recibe la información de las actividades de las áreas de promoción y / o relaciones públicas del I.N.A.H.

Ver figura II.1.B



Esta subdirección cuenta con una línea telefónica específicamente para proporcionar dicho servicio, y adicionalmente se planeo contratar una línea de lada 800.

La información que contiene el reporte de la programación mensual puede ser recibida por la subdirección varios meses, semanas ó días antes del inicio del mes en cuestión, e incluso, se llega a recibir durante el transcurso del mes. La preparación del reporte se inicia a principios de la segunda quincena de cada mes y se inicia la distribución durante los últimos dos o tres días antes del inicio del mes.

Esta función la realiza esta subdirección desde sus inicios de operación. Se cuenta con tres personas para la recopilación de información, análisis, y preparación de los reportes y para proporcionar información al público.

Estos reportes los utilizan las áreas de comunicación e informes del I.N.B.A. e I.N.A.H.

Dentro de la estructura del Instituto Nacional de Bellas Artes existe un departamento de sistemas, al cual se le solicitó en varias ocasiones el desarrollo de un sistema para la subdirección de relaciones públicas, con el cual se facilitara la emisión de los reportes. Se tuvo una entrevista con el personal de dicho departamento de sistemas y comentaron que no tenían el tiempo suficiente para desarrollar el sistema. Por esta razón se solicitó a empresas externas el desarrollo de éste sistema.

En la figura II.1.C se puede ver la estructura jerárquica del I.N.B.A. en la cual se puede apreciar la existencia del departamento de sistemas mencionado.

II.2 RECOPIACION DE LA INFORMACION.

Las técnicas que utilizamos para obtener información del usuario fueron las de entrevistas y aplicación de cuestionarios.

Se aplicó un cuestionario al responsable de la subdirección y otro a las personas que realizan las actividades necesarias para poder obtener los productos deseados.

A continuación se incluyen los cuestionarios que se aplicaron para este fin.

Cuestionario aplicado al subdirector de relaciones públicas del I.N.B.A.

1. ¿ Cuales son las funciones que desean automatizar ?

La captura de las actividades de la programación mensual y semanal

La corrección (edición) de las actividades

La impresión de los reportes mensual y semanal

La consulta de información

2. ¿ Existe un procedimiento para cada una de las funciones a automatizar ?

Se podría decir que sí, ya que siempre se realiza el mismo procedimiento para cada una de las funciones, aún cuando dichos procedimientos no son oficiales.

3. ¿ Cuantas personas están designadas para desarrollar esas actividades ?

Tres personas, para recopilar y capturar la información así como para preparar los reportes y ofrecer servicio de atención al público a través de la línea telefónica.

4. ¿ Cada persona realiza una función específica ?

Todas realizan todas las actividades, es decir, conocen y realizan todo el proceso.

5. ¿ Cuantas de esas personas conocen los procedimientos y cuantas de ellas los siguen?

Todos conocen los procedimientos, que como ya mencioné no son oficiales, sin embargo, comunmente esos procedimientos son seguidos por todos.

6. ¿ Cuales son los problemas que se presentan durante la elaboración de los productos a obtener ?

Los reportes se realizan en una máquina de escribir y dado que el formato es tabular la elaboración resulta tediosa y frecuentemente se tienen errores.

El tiempo que toma la elaboración de los reportes es demasiado largo, El reporte mensual puede tomar hasta más de una semana y el semanal de dos a tres días.

Frecuentemente se presentan cambios en la información que se recibió para la programación mensual lo que implica tener que reproducir nuevamente dos o más hojas del reporte.

Cuando la información se recibe con mucha anticipación, esta no se puede procesar inmediatamente, ya que toda la información debe estar ordenada cronológicamente.

Esto implica que se acumule el trabajo que se podría adelantar y se podría disminuir la carga de trabajo en fechas pico.

7. ¿ Como afectan estos problemas en la obtención de los productos ?

La actividad que no le gusta realizar a las personas asignadas es la elaboración de los reportes, lo cual aumenta el tiempo de elaboración de los mismos y reduce la calidad del producto.

Si se presentan demasiados cambios a la programación uno o dos días antes del inicio del mes, se retrasa la emisión del reporte mensual.

Ver figura II.2.A

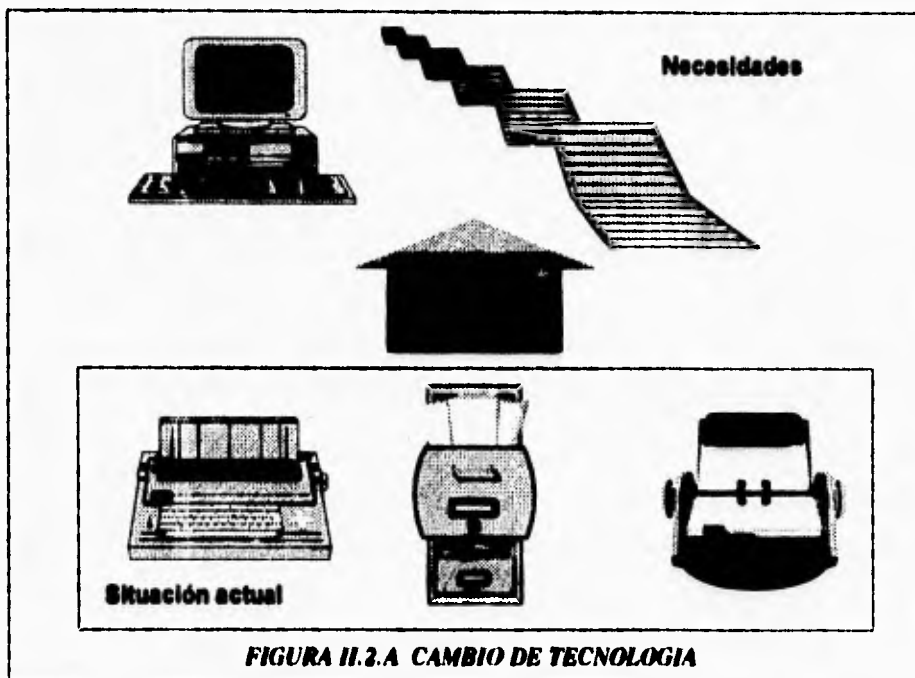


FIGURA II.2.A CAMBIO DE TECNOLOGIA

A continuación se anexa el cuestionario que se aplicó a las personas que elaboran el producto.

Este cuestionario engloba las respuestas que propocionaron las tres personas que elaboran los reportes. En la mayoría de las repuestas coincidieron en varios puntos, sobre todo en las cosas que no les agradan.

1. ¿ Con cuanta anticipación reciben la información ?

Varía mucho, hay actividades de las que se tiene conocimiento dos o tres meses antes, pero esto no representa una ventaja pues no se puede adelantar nada ya que se debe

tener toda la información para ordenarla cronológicamente y entonces proceder a la elaboración física de los reportes. En otros casos la información se recibe uno o dos días antes, esto no afecta si no se ha realizado la página en que debería quedar (cronológicamente), pero si ya se hubiera realizado se tendrá que repetir esa página y quizá dos o tres páginas más, dependiendo de cuanta información se tenga que insertar.

2. ¿ Cuanto tiempo toma elaborar un reporte mensual ?

Puede tomar de cinco a nueve días, dependiendo de la temporada.

3. ¿ Cuanto tiempo toma elaborar un reporte semanal ?

De dos a tres días. Este tiempo puede variar y toma más tiempo cuando hay actividades que se presentan varios días a la semana y / o más de una vez en el mismo día.

4. ¿ Cuantas horas al día dedican para la elaboración del reporte ?

Diez horas, pero esto no es todos los días del mes, solo en la última y penúltima semana del mes, que es cuando se tiene que preparar el reporte mensual y además el semanal correspondiente. Durante las dos primeras semanas se dedica hasta menos de la mitad de ese tiempo al día.

5. ¿ Que es lo que menos le gusta del proceso de elaboración del reporte ?

Que se presenten cambios en las actividades, o que se cancelen ya que se tiene que repetir trabajo que ya se realizó.

En el caso del reporte semanal, como se considera que por cada hora y fecha se tiene una actividad, no importando que se trate de la misma, se tiene que repetir la misma información referente a una actividad más de una vez, por ejemplo, en la temporada del Lago de los Cisnes, si se presenta tres días en una semana, la información de esta

actividad tiene que aparecer tres veces en el reporte y lo único que cambia es la fecha y / o hora.

6. Si tuviera un sistema de cómputo para elaborar los reportes, ¿cuales son las características que desearía para ingresar la información de las actividades ?

Que la información la pueda ingresar conforme la vaya recibiendo para no tener que esperar hasta una semana antes de que se tenga que imprimir el reporte.

Que pueda modificar desde una letra hasta un párrafo de la descripción general del reporte.

Que si una actividad se presenta más de una vez, solo tenga que ingresar la información general una sola vez e indicar en que días y horas se presenta.

Que pueda eliminar la descripción de toda una actividad con solo indicarlo.

No tener que analizar la información para organizarla en forma cronológica, ya que esto nos toma bastante tiempo y si nos llegamos a equivocar al realizar este paso, tendremos que repetir varias páginas del reporte para que cada actividad quede en el lugar correcto.

Que la información que se ingresó la pueda consultar para no tener que buscarla entre tantas hojas, pues esto toma mucho tiempo cuando se da servicio al público por teléfono.

Después de aplicar estos cuestionarios, tuvimos varias entrevistas con estas personas, ya que algunas respuestas no eran muy claras o concretas y en algunos casos por medio de ejemplos nos hicieron saber lo que deseaban o como les afectaban algunas situaciones en el desarrollo de su trabajo.

Durante estas entrevistas también obtuvimos la información que se incluye en los reportes mencionados así como a través de que información se realizan búsquedas en los reportes para proporcionar información al público.

A continuación se describe la información que es incluida en cada reporte y los elementos de búsqueda utilizados.

Reporte mensual:

- **Nombre o descripción de la actividad.**
- **Nombre de la disciplina a la que pertenece la actividad.**
- **Nombre del (los) artista(s), director, autor, expositor, etc., que participa(n) en la actividad.**
- **Nombre del recinto en el que se llevarán a cabo las actividades**
- **Domicilio del recinto en el que se llevarán a cabo las actividades, se incluye calle, colonia, ciudad, código postal.**
- **Teléfono del recinto en el que se llevarán a cabo las actividades (dos números telefónicos como máximo), en el formato 999-99-99, se incluye la clave lada.**
- **Fechas en las que se presentan las actividades. Para el caso de cursos y talleres, se incluye la fecha de inicio y la fecha de terminación.**
- **Horarios en los que se presentan las actividades. Para el caso de cursos y talleres, se incluye la hora de inicio y la hora de terminación.**
- **Horario en el que se encuentra abierta la taquilla, el museo, etc., según sea el caso.**
- **Precios disponibles, puede ser más de uno**
- **Descuentos disponibles, puede ser más de uno**
- **Comentarios sobre los precios y/o descuentos disponibles, puede ser más de uno**
- **Observaciones generales**

Las longitudes máximas utilizadas en cada una de los inciso anteriores son las siguientes:

Nombre / descripción de la actividad	Ilimitado
Nombre de la disciplina	30 caracteres
Nombre de artistas	Ilimitado
Nombre del recinto	40 caracteres
Dirección	40 caracteres
Colonia	40 caracteres
Ciudad	40 caracteres
Código postal	5 caracteres
Clave lada	3 caracteres
Número telefónico	9 caracteres
Fecha de inicio	8 caracteres
Fecha de terminación	8 caracteres
Hora de inicio	5 caracteres
Hora de terminación	5 caracteres
Precios	9 caracteres
Descuentos	3 caracteres
Comentarios de precios y descuentos	30 caracteres
Observaciones generales	Ilimitado.

Reporte semanal:

- **Nombre o descripción de la actividad.**
- **Nombre de la disciplina a la que pertenece la actividad.**
- **Nombre del recinto en el que se llevarán a cabo las actividades**

- **Domicilio del recinto en el que se llevarán a cabo las actividades, se incluye calle, colonia, ciudad.**
- **Fechas en las que se presentan las actividades. Para el caso de cursos y talleres, se incluye la fecha de inicio y la fecha de terminación.**
- **Horarios en los que se presentan las actividades. Para el caso de cursos y talleres, se incluye la hora de inicio y la hora de terminación.**
- **Horario en el que se encuentra abierta la taquilla, el museo, etc., según sea el caso.**
- **Observaciones generales**

Las longitudes máximas utilizadas en cada una de los incisos anteriores son las siguientes:

Nombre / descripción de la actividad	Ilimitado
Nombre de la disciplina	30 caracteres
Nombre del recinto	40 caracteres
Dirección	40 caracteres
Colonia	40 caracteres
Ciudad	40 caracteres
Fecha de inicio	8 caracteres
Fecha de terminación	8 caracteres
Hora de inicio	5 caracteres
Hora de terminación	5 caracteres
Observaciones generales	Ilimitado.

Catálogo de recintos:

Nombre del recinto

Calle en la que se ubica

Colonia

Delegación a la que pertenece

En que ciudad se encuentra

Código postal correspondiente

Clave lada, si es que se encuentra en el interior de la República

Dos números telefónicos

Las longitudes máximas utilizadas en cada uno de los incisos anteriores son las siguientes:

Nombre	40 caracteres
Calle	40 caracteres
Colonia	40 caracteres
Delegación	40 caracteres
Ciudad	40 caracteres
Código postal	5 caracteres
Clave lada	3 caracteres
Teléfono (1)	9 caracteres
Teléfono (2)	9 caracteres

Los elementos que se utilizan como llave para realizar búsquedas de información son los siguientes:

Nombre / descripción de la actividad

Nombres de artistas

Nombre de la disciplina

Nombre del recinto

Fechas de presentación

II. 3 PROBLEMATICA ACTUAL

Analizando los cuestionarios y la información obtenida en las entrevistas pudimos encontrar la problemática que a continuación describimos.

- **Los reportes se realizan en máquina de escribir y son en formato tabular lo que hace difícil la elaboración de los reportes.**
- **Debido a que es muy común que se presenten cambios en la programación, semanas o días antes de la fecha en que se encontraban programadas algunas actividades, el reporte debe ser preparado una semana antes de que inicie el mes en cuestión, aún cuando la información se haya recibido varias semanas o meses antes, esto es para evitar un alto número de modificaciones al reporte.**
- **Uno de los mayores problemas se presenta cuando ya se tiene la última versión del reporte, y poco antes de que se soliciten todas las copias necesarias se presente un cambio en la programación, ya que para resolver esto, se tendrán que reemplazar por lo menos dos o tres páginas del reporte, apreciándose una discontinuidad en alguna de las páginas dependiendo del tipo de cambio que se presentó.**
- **Una vez que ya se tienen todas las copias de la última versión del reporte, si se presenta algún cambio, simplemente se anexa una hoja en donde se indican los cambios de última hora.**
- **El tiempo que toma el desarrollo del reporte es muy alto, ya que como se edita en máquina de escribir, es un formato tabular, y además la longitud de cada columna**

siempre varía para cada actividad, se vuelve un trabajo excesivamente lento y tedioso.

- Hay información que se tiene que escribir más de una vez, lo que puede causar problemas e inconsistencia de datos.
- La calidad del producto final no es la que se desearía obtener, ya que no se puede hacer gran cosa con una máquina de escribir.
- En lo que se refiere a la búsqueda de información para dar el servicio de atención telefónica, cuando la búsqueda no es por fecha y hora de presentación, ésta toma demasiado tiempo, ya que la información se encuentra ordenada por fecha y hora de presentación.
- En cuanto a los datos de los recintos, también resulta lenta la búsqueda del detalle de datos cuando se busca por colonia, delegación, etc., ya que más de un recinto cumple con la condición de búsqueda.

Los puntos que se desean mejorar son los siguientes:

- Poder consultar con mayor rapidez por nombre de la actividad, nombre de artistas, y por fecha, ya que de esta forma se ofrecería un servicio más rápido, dando una mejor imagen al instituto.
- Poder consultar con mayor rapidez el detalle de los recintos utilizando diferentes argumentos de búsqueda.

- **Reducir el impacto que causan los cambios o cancelaciones de última hora.**
- **Eliminar la inconsistencia de datos**
- **Mejorar la calidad del producto final.**
- **Reducir costos en las llamadas por lada 800.**
- **Mejorar la imagen de estos servicios de la subdirección de relaciones públicas del I.N.B.A.**

II.4 IDENTIFICACION Y ANALISIS DE REQUERIMIENTOS DEL USUARIO

Según lo que hemos podido observar en los últimos dos puntos, se pueden identificar los siguientes requerimientos:

1. Tener un sistema de captura de información referente a actividades culturales, el cual facilite la labor de captura y edición de información y por consiguiente se pueda iniciar dicha captura con mayor tiempo, eliminando pagos de tiempo extra, y posibles retrasos así como evitar la captura de una misma información más de una vez, eliminando la inconsistencia de datos.

2. Tener un sistema de modificación de información, que facilite la labor de realizar correcciones y cambios a la información que se había recibido inicialmente, sin tener que capturar el ítem en cuestión en su totalidad.

3. Tener un sistema de consultas que permita recuperar la información deseada en forma ágil y sencilla, ofreciendo así un servicio más rápido y reduciendo los costos del servicio por lada 800.

4. Que a partir de la información que se captura, modifica y elimina, se emitan dos reportes, uno mensual, que se puede considerar contendrá la información global para el mes en cuestión y otro semanal, que será un extracto del reporte mensual, con la facilidad de que si existiera algún cambio de última hora, éste no represente un impacto para la obtención a tiempo y la calidad de los productos finales.

5. Tener un sistema que reduzca el tiempo de preparación de los reportes.

6. Mejorar la calidad de los reportes.

7. El usuario solicita que se cumplan los siguientes puntos específicos:

- **Desarrollar un sistema para computadora personal**
- **Que la información esté almacenada en una base de datos**
- **Que se tengan instalados en dicha computadora personal únicamente programas ejecutables, las bases de datos necesarias así como los archivos de índices correspondientes**
- **Que en la base de datos no se manejen campos tipo MEMO (campos de longitud variable).**
- **Que el sistema que se desarrolle pueda operar bajo sistema operativo MS-DOS desde la versión 3.30 y posteriores**
- **Que los requerimientos de hardware no excedan las características que a continuación se indican ya que es la configuración típica con la que se cuenta en el I.N.B.A. para proporcionar este tipo de servicio.**

Computadora:	Compatible con IBM-PC
Procesador:	Intel 80286
Memoria RAM:	640KB
Espacio en disco duro:	20 MB
Monitor:	TTL, RGB, VGA

8. El usuario solicita que de preferencia el desarrollo se realizara utilizando el compilador de Clipper.

II.5 ANALISIS DE LA INFORMACION

En este punto analizaremos la información obtenida en el punto II.2 con la finalidad de obtener una visión general de los datos que maneja el sistema, los documentos a partir de los cuales se obtienen todos los datos así como de las funciones que se tendrán disponibles y que datos requiere cada una de ellas.

DATOS.

Iniciaremos realizando un inventario de datos, con la finalidad de conocer cuales son todos los datos que se utilizarán a nivel global en el sistema. Los datos son los siguientes:

DATOS
Nombre de la disciplina
Nombre del recinto
Calle del recinto
Colonia del recinto
Delegación del recinto
Código postal del recinto
Ciudad del recinto
Clave lada del recinto
Teléfono del recinto
Nombre de la actividad
Fecha de presentación
Hora de presentación
Nombre de artista
Precio
Descuento
Leyenda sobre precio y/o descuento
Observaciones
Horario de taquilla

DOCUMENTOS.

A continuación enlistamos los documentos que se manejan en la subdirección de Relaciones Públicas del I.N.B.A., a partir de estos documentos se obtiene toda la información necesaria para emitir los productos de mayor interés del sistema.

DOCUMENTOS
REPORTE MENSUAL DE ACTIVIDADES
REPORTE SEMANAL DE ACTIVIDADES
CATALOGO DE DISCIPLINAS
CATALOGO DE RECINTOS

FUNCIONES.

A continuación presentamos todas las funciones que debe realizar el sistema y cuales son los datos que se requieren en cada una de ellas.

FUNCION O PROCEDIMIENTO	INFORMACION REQUERIDA
CAPTURA DE DISCIPLINAS	NOMBRES DE LAS DISCIPLINAS
CAPTURA DE RECINTOS	NOMBRE, CALLE, COLONIA, DELEGACION, C.P., CIUDAD, CLAVE LADA, TELEFONO

DOCUMENTOS.

A continuación enlistamos los documentos que se manejan en la subdirección de Relaciones Públicas del I.N.B.A., a partir de estos documentos se obtiene toda la información necesaria para emitir los productos de mayor interés del sistema.

DOCUMENTOS
REPORTE MENSUAL DE ACTIVIDADES
REPORTE SEMANAL DE ACTIVIDADES
CATALOGO DE DISCIPLINAS
CATALOGO DE RECINTOS

FUNCIONES.

A continuación presentamos todas las funciones que debe realizar el sistema y cuales son los datos que se requieren en cada una de ellas.

FUNCION O PROCEDIMIENTO	INFORMACION REQUERIDA
CAPTURA DE DISCIPLINAS	NOMBRES DE LAS DISCIPLINAS
CAPTURA DE RECINTOS	NOMBRE, CALLE, COLONIA, DELEGACION, C.P., CIUDAD, CLAVE LADA, TELEFONO

CAPTURA DE ACTIVIDADES	NOMBRE DE LA ACTIVIDAD, NOMBRE DE LA DISCIPLINA, FECHAS Y HORAS DE PRESENTACION, NOMBRES DE LOS ARTISTAS, NOMBRE DEL RECINTO, PRECIOS Y DESCUENTOS, OBSERVACIONES, HORARIO DE TAQUILLA.
CONSULTA DE DISCIPLINAS	DISCIPLINAS DISPONIBLES
CONSULTA DE RECINTOS	NOMBRE, CALLE, COLONIA, DELEGACION, C.P., CIUDAD, CLAVE LADA, TELEFONO
CONSULTA DE ACTIVIDADES	NOMBRE DE LA ACTIVIDAD, NOMBRE DE LA DISCIPLINA, FECHAS Y HORAS DE PRESENTACION, NOMBRES DE LOS ARTISTAS, NOMBRE DEL RECINTO, PRECIOS Y DESCUENTOS, OBSERVACIONES, HORARIO DE TAQUILLA.
MODIFICACION DE DISCIPLINAS	NOMBRE DE LA DISCIPLINA A MODIFICAR
MODIFICACION DE RECINTOS	NOMBRE, CALLE, COLONIA, DELEGACION, C.P., CIUDAD, CLAVE LADA, TELEFONO
MODIFICACION DE ACTIVIDADES	NOMBRE DE LA ACTIVIDAD, NOMBRE DE LA DISCIPLINA, FECHAS Y HORAS DE PRESENTACION, NOMBRES DE LOS ARTISTAS, NOMBRE DEL RECINTO, PRECIOS Y DESCUENTOS, OBSERVACIONES, HORARIO DE TAQUILLA.
REPORTE MENSUAL DE ACTIVIDADES	NOMBRE DE LA ACTIVIDAD, NOMBRE DE LA DISCIPLINA, FECHAS Y HORAS DE PRESENTACION, NOMBRES DE LOS ARTISTAS, NOMBRE, CALLE, COLONIA, CIUDAD, C.P., CLAVE LADA Y TELEFONOS DEL RECINTO, PRECIOS Y DESCUENTOS, OBSERVACIONES, HORARIO DE TAQUILLA.
REPORTE SEMANAL DE ACTIVIDADES	NOMBRE DE LA ACTIVIDAD, NOMBRE DE LA DISCIPLINA, FECHAS Y HORAS DE PRESENTACION, NOMBRE, CALLE, COLONIA Y CIUDAD DEL RECINTO, OBSERVACIONES.

REPORTE DE RECINTOS	NOMBRE, CALLE, COLONIA, DELEGACION, C.P., CIUDAD, CLAVE LADA, TELEFONO
REPORTE DE DISCIPLINAS	NOMBRES DE LAS DISCIPLINAS
ELIMINACION DE ACTIVIDADES INDIVIDUALES	NOMBRE DE LA DISCIPLINA A ELIMINAR
ELIMINACION DE ACTIVIDADES POR NOMBRE DEL MES EN QUE SE PRESENTA	NOMBRE DEL MES DE ACTIVIDADES A ELIMINAR

Se pueden identificar tres elementos de datos principales:

- Referente a la programación de actividades.
- Referente a datos de recintos
- Referente a disciplinas

Podemos observar que para cada uno de estos puntos se pueden tener las siguientes funciones:

- Altas
- Bajas
- Cambios
- Consultas
- Reportes

REPORTE DE RECINTOS	NOMBRE, CALLE, COLONIA, DELEGACION, C.P., CIUDAD, CLAVE LADA, TELEFONO
REPORTE DE DISCIPLINAS	NOMBRES DE LAS DISCIPLINAS
ELIMINACION DE ACTIVIDADES INDIVIDUALES	NOMBRE DE LA DISCIPLINA A ELIMINAR
ELIMINACION DE ACTIVIDADES POR NOMBRE DEL MES EN QUE SE PRESENTA	NOMBRE DEL MES DE ACTIVIDADES A ELIMINAR

Se pueden identificar tres elementos de datos principales:

- Referente a la programación de actividades.
- Referente a datos de recintos
- Referente a disciplinas

Podemos observar que para cada uno de estos puntos se pueden tener las siguientes funciones:

- Altas
- Bajas
- Cambios
- Consultas
- Reportes

II.6 ALTERNATIVAS DE SOLUCION

Para dar solución a los requerimientos del usuario en base al análisis anterior presentamos varias opciones de solución:

- **Desarrollar un sistema de bases de datos utilizando el compilador clipper.**
- **Desarrollar un sistema de bases de datos utilizando FoxPro. Si se opta por esta alternativa se cubrirían los requerimientos del usuario, con las siguientes condicionantes:**

El sistema no respondería tan rápido como lo haría si se tuviera disponible memoria extendida.

No se cubriría la preferencia por Clipper.

II.6.1 DESARROLLO CON CLIPPER.

Equipo requerido:

Computadora personal compatible con IBM-PC con las siguientes características mínimas:

Procesador:	Intel 80286
Memoria RAM:	512KB
Espacio en disco duro:	80 MB
Monitor:	TTL, RGB, VGA

Impresora, cualquiera de las siguientes:

De matriz de puntos con las siguientes características:

15 pulgadas de ancho de carro

Compatible con Epson o IBM graphics.

Velocidad de 180 c.p.s. mínimo

Laser con las siguientes características:

6 a 8 páginas por minuto

1 MB de memoria RAM

Impresión en formato Portrait y Landscape.

Recursos humanos:

Una persona que quede como administrador de la base de datos para proporcionar el mantenimiento de la base de datos, índices y respaldos de información.

Una persona que recopile la información a ser procesada.

Una persona para que capture la información conforme se va recibiendo.

Una persona para proporcionar el servicio de atención a público por medio de vía telefónica.

Estas tres últimas actividades pueden ser efectuadas por una misma persona, por lo que con dos personas se cubren los recursos humanos necesarios.

Ventajas:

- **Los requerimientos de hardware son mínimos**

- No es necesario contar con el compilador, ya que solo se tendrán archivos ejecutables, bases de datos e índices.
- Las bases de datos son 100% compatibles con el formato de dbase.
- El tamaño de los archivos ejecutables es más que aceptable.
- Las utilerías de clipper para creación y consulta de bases de datos y generación de reportes y formatos de etiquetas ocupan muy poco espacio en disco

Desventajas:

- La utilería DBU no es muy amigable para efectuar consultas complejas.

Comentarios adicionales:

Si se opta por esta alternativa se cubrirían todos los requerimientos del usuario, incluyendo la preferencia de utilizar Clipper, aún cuando este último requerimiento no es indispensable.

II.6.2 DESARROLLO CON FOXPRO

Equipo requerido:

Computadora personal compatible con IBM-PC con las siguientes características mínimas:

Procesador:	Intel 80386
Memoria RAM:	1MB
Memoria Extendida	1.5 MB
Espacio en disco duro:	80 MB
Monitor:	TTL, RGB, VGA

Impresora, cualquiera de las siguientes:

De matriz de puntos con las siguientes características:

15 pulgadas de ancho de carro

Compatible con Epson o IBM graphics.

Velocidad de 180 c.p.s. mínimo

Laser con las siguientes características:

6 a 8 páginas por minuto

1 MB de memoria RAM

Impresión en formato Portrait y Landscape.

Recursos humanos:

Una persona que quede como administrador de la base de datos para proporcionar el mantenimiento de la base de datos, índices y respaldos de información.

Una persona que recopile la información a ser procesada.

Una persona para que capture la información conforme se va recibiendo.

Una persona para proporcionar el servicio de atención a público por medio de vía telefónica.

Estas tres últimas actividades pueden ser efectuadas por una misma persona, por lo que con dos personas se cubren los recursos humanos necesarios.

Ventajas:

- El tiempo de respuesta de foxpro en consultas y generación de índices es excelente.
- No es necesario tener instalado el paquete para ejecutar los programas.
- Si se tiene instalado el paquete, el usuario puede realizar consultas adicionales no contempladas en el sistema.

Desventajas:

- El tamaño de los archivos ejecutables es muy grande.
- Si se quiere tener instalado el paquete, este ocupa demasiado espacio en disco.
- Los requerimientos de hardware son mayores y si no se cubren todos, se pierde la ventaja de la velocidad de foxpro.

Comentarios adicionales:

- No se cubriría la preferencia por Clipper.

CAPITULO III.

DISEÑO Y DESARROLLO DEL SISTEMA

DISEÑO Y DESARROLLO DEL SISTEMA

III.1 ELECCION DEL MANEJADOR DE BASES DE DATOS

En esta sección realizaremos un análisis para seleccionar el manejador de bases de datos con el cual se realizará el desarrollo e implementación de la solución propuesta.

Analizaremos las características más importantes de los siete manejadores de bases de datos para computadoras personales que revisamos en el punto I.4.1. Se asignará una calificación a cada manejador en función del soporte que proporcionan las características que se analizan para cubrir los requerimientos del usuario.

Aún cuando entre los requerimientos del usuario no está el que el sistema tenga la capacidad de operar en ambiente multi-usuario, se tomará en cuenta, para este análisis, esta característica, con la finalidad de que si en el futuro surgiera la necesidad de operar el sistema en una red, el sistema se pueda adaptar fácilmente.

III.1.1 CRITERIOS DE EVALUACION

En este punto se indican las características que se consideran para la selección del manejador de bases de datos, el cual cumpla con los requerimientos del usuario y proporcione al programador las facilidades para desarrollar el sistema en el menor tiempo posible.

Las fuentes de información utilizadas para realizar esta evaluación son las siguientes:

Revistas de computación

Manuales de los manejadores de bases de datos

Experiencias de usuarios

Experiencias de diseñadores y programadores

Los productos analizados son los siguientes:

dBASE IV	ASHTON TATE CORP.
FOXPRO	FOX SOFTWARE INC.
R:BASE	MICRORIM INC.
PARADOX	BORLAND INTERNATIONAL
CLIPPER	NANTUCKET
INFORMIX - SQL	INFORMIX
DATAEASE	DATAEASE INTERNATIONAL

Los aspectos a evaluar son los siguientes:

- **Tipos de campos disponibles**
- **Formatos de los que se pueden importar datos**
- **Requerimientos de hardware**
- **Número máximo de tablas abiertas**
- **Número máximo de índices por tabla**
- **Número máximo de renglones por tabla**
- **Número máximo de campos por renglón**
- **Longitud máxima de renglón**
- **Funciones multi-usuario**
- **Herramientas integradas**

Los siete manejadores de bases de datos que se analizan se encuentran en el mismo rango de precio, según revistas de computación todos tienen un costo de \$795.00 dólares americanos, aun cuando este precio puede variar en el mercado.

Cada producto irá acumulando puntos según se vaya evaluando cada aspecto, se le asignará mayor número de puntos a aquel producto que cubra mejor los requerimientos del usuario.

En las tablas que a continuación presentamos utilizaremos la siguiente notación:

- dB:** dBase
- P:** Paradox
- DE:** Dataease
- FP:** Foxpro
- I:** Informix SQL
- R:** R:Base
- C:** Clipper
- REQ. USR.:** Requerimiento de usuario
- 1.** No necesario
- 2.** Deseable
- 3.** Necesario

La calificación para cada manejador se asignará sumando la puntuación de acuerdo a los requerimientos del usuario.

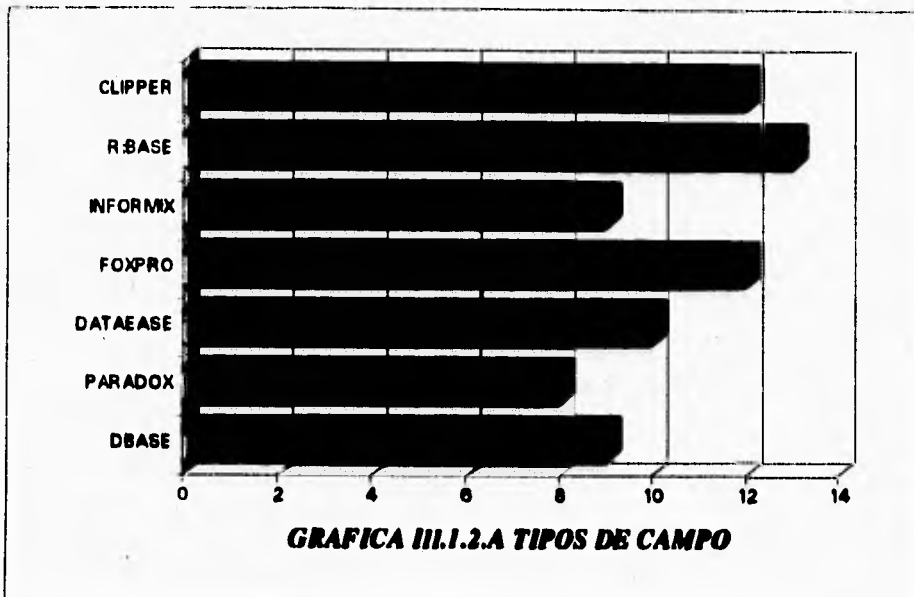
III.1.2 MANEJO DE DATOS

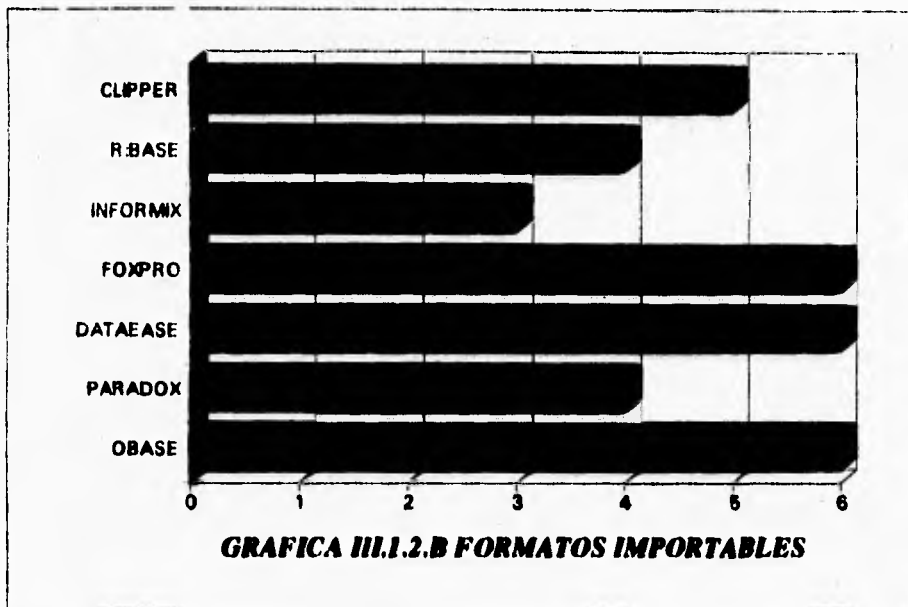
TIPO DE CAMPO	R	F	D	P	L	C	E	REQ. USU.
ENTERO	X		X	X	X	X	X	1
PUNTO FLOTANTE		X	X	X	X	X	X	3
FECHA	X	X	X	X	X	X	X	3
HORA			X			X		1
TEXTO	X	X	X	X	X	X	X	2
LONGITUD VARIABLE	X			X		X	X	3
CALIFICACION	9	8	10	12	9	13	12	

En cuanto a tipos de campos disponibles en los manejadores, podemos ver que R:Base es el más poderoso en este caso, quedando en segundo lugar Foxpro y Clipper, y en tercer lugar Dataease. Analizando la tabla, podemos ver que de los manejadores que obtuvieron los tres mejores lugares, solamente los de los dos primeros lugares cubren los requerimientos del usuario. Por lo que Dataease no califica. Ver gráfica III.1.2.A

Con respecto a los formatos de los cuales se desea importar información son el formato ASCII, en primer lugar, esporádicamente del formato SDF, y se desearía tener la opción de exportar archivos de Lotus (formato WK1). Ver gráfica III.1.2.B

TIPO DE CAMPO	R	F	D	P	L	C	E	REQ. USU.
ASCII	X	X	X	X	X	X	X	3
SDF	X		X	X			X	2
WK1	X	X	X	X		X		1
CALIFICACION	6	4	6	6	3	4	5	





En lo que se refiere al manejo de archivos, evaluaremos el número máximo de tablas e índices que se pueden tener abiertos en forma simultánea, así como el número máximo de registros por tabla, campos por registro y longitud máxima de registro. Se asignará mayor calificación a los manejadores que tengan mayor capacidad.

	10	24	255	99	ILIMIT	80	250
MAX. # TABLAS ABIERTAS							
CALIFICACION	1	1	2	1	3	1	2

	57	255	255	ILIMIT	ILIMIT	ILIMIT	15
INDICES x TABLA							
CALIFICACION	1	2	2	3	3	3	1

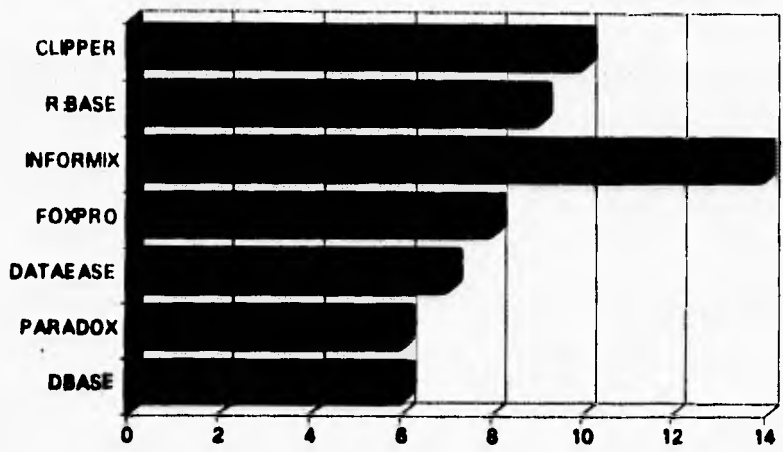
# REG. x TABLA (MILLONES)	1000	256	0.000255	1000	ILIMIT	ILIMIT	1000
CALIFICACION	2	1	1	2	3	3	2

# CAMPOS x REGISTRO	255	255	255	255	ILIMIT	400	1000
CALIFICACION	1	1	1	1	3	1	2

LONG. MAX. DE REG. (Bytes)	4000	4000	4096	4000	32768	4096	65534
CALIFICACION	1	1	1	1	2	1	3

CALIF. GLOBAL	6	6	7	8	14	9	10
---------------	---	---	---	---	----	---	----

Ver gráfica III.1.2.C



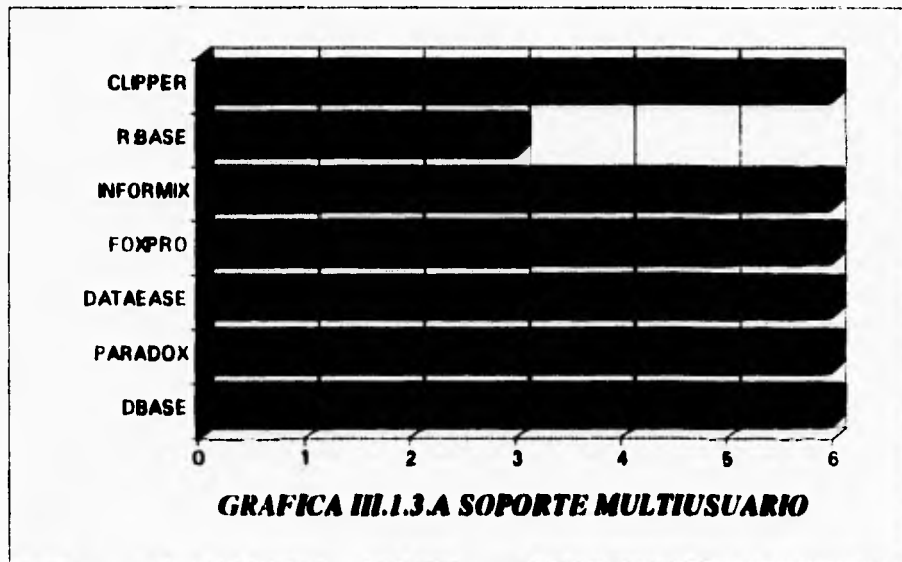
GRAFICA III.1.2.C MANEJO DE ARCHIVOS

III.1.3 SOPORTE MULTI-USUARIO

TABLAS	X	X	X	X	X	X	X	3
REGISTROS	X	X	X	X	X		X	3
CALIFICACION	6	6	6	6	6	3	6	

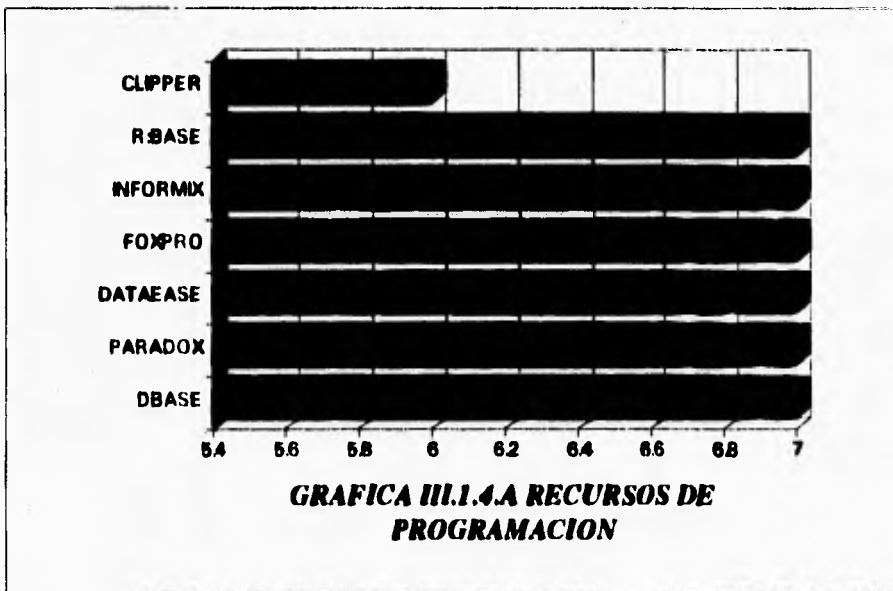
Dentro de las necesidades del usuario no está la de operar en ambiente multi-usuario o de red, sin embargo consideramos que un buen manejador de bases de datos debe proporcionar las herramientas necesarias para poder operar en un sistema multi-usuario o de redes. Consideramos que las características básicas a evaluar son que los manejadores permitan realizar bloqueos a nivel tabla y a nivel registro cuando menos.

Ver gráfica III.1.3.A



III.1.4 RECURSOS DE PROGRAMACION.

PROGRAMACION	A	P	M	PP	V	B	C	REQ. USU.
GENERADOR DE FORMATOS	X	X	X	X	X	X		1
GENERADOR DE REPORTE	X	X	X	X	X	X	X	3
LENGUAJE DE BASE DE DATOS	X	X	X	X	X	X	X	3
CALIFICACION	7	7	7	7	7	7	6	



Normalmente la mayoría de los manejadores de bases de datos incluyen herramientas para auxiliar al usuario, e incluso al programador, en la elaboración de pantallas de

captura de datos; formatos para impresión de reportes y etiquetas, así como un lenguaje de manejo de datos.

De los manejadores evaluados seis cubren los apoyos más deseables por los usuarios poco experimentados en el uso de estos sistemas. Gráfica III.1.4.A

III.1.5 REQUERIMIENTOS DE HARDWARE

Para el caso de los requerimientos de hardware, se calificará con mayor puntuación a los manejadores que operen con la menor cantidad de recursos disponibles, se consideran memoria RAM base, memoria RAM extendida o expandida, aún cuando sea considerada como "recomendable" ya que si el producto no tiene a su disposición dicha memoria adicional, su rendimiento disminuye, y el espacio en disco requerido para la instalación del paquete.

REQUERIMIENTOS DE HARDWARE	M.	P.	DE	PP	L	R	C
RAM (BASE) KBytes	640	640	570	640	640	640	300
CALIFICACION	1	1	2	1	1	1	3

REQUERIMIENTOS DE HARDWARE	M.	P.	DE	PP	L	R	C
RAM (XMS) MBytes				1.5	1		
CALIFICACION	3	3	3	1	1	3	3

REQUERIMIENTOS DE HARDWARE	M.	P.	DE	PP	L	R	C
DISCO (MBytes)	5	4	2.5	6	3	6	3.5
CALIFICACION	2	2	3	1	3	1	3

REQUERIMIENTOS DE HARDWARE	M.	P.	DE	PP	L	R	C
CALIF. GLOBAL	6	6	8	3	5	5	9

Ver gráfica III.1.5.A



III.1.6 CONCLUSION

Acumulando las calificaciones obtenidas en cada punto se tienen los siguientes resultados (gráfica III.1.6.A):

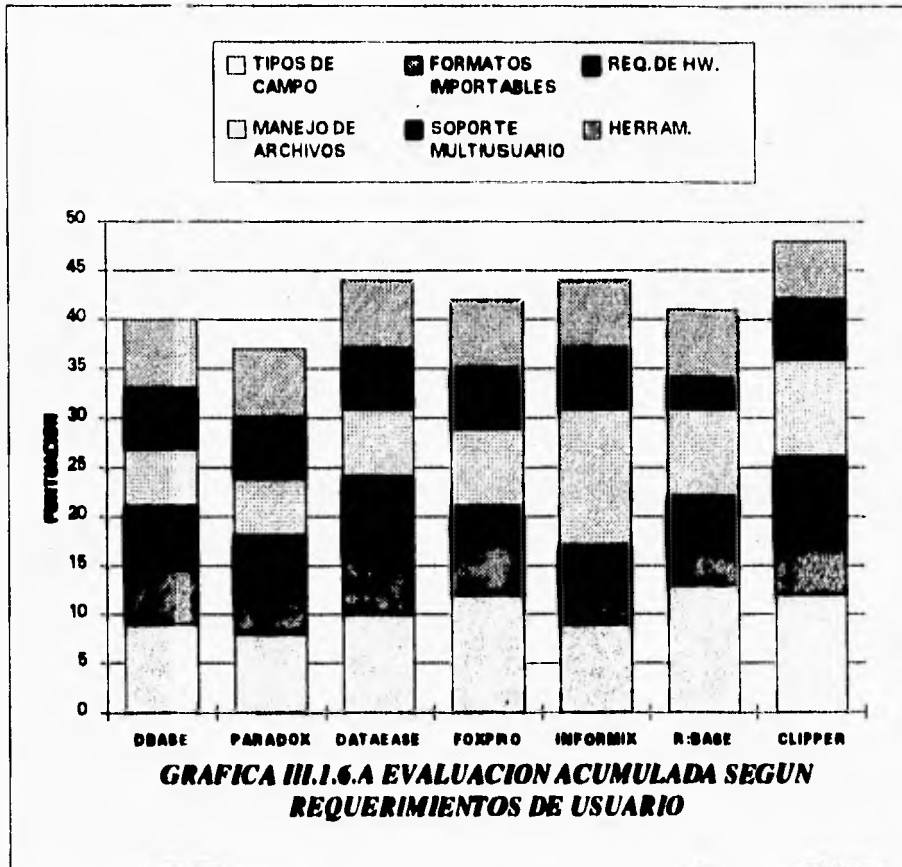
CALIFICACION GLOBAL	40	37	44	42	44	41	48
---------------------	----	----	----	----	----	----	----

Globalizando los resultados podemos ver que de acuerdo a los requerimientos del usuario, que los manejadores quedaron en las siguientes posiciones:

1. Clipper
2. Informix
3. Dataease
4. Foxpro
5. R:Base
6. dBase IV
7. Paradox

Analizando las tablas de calificaciones podemos observar que las características que presentan mayor variación en las calificaciones entre los manejadores son las siguientes:

- Valores máximos.
- Requerimientos de hardware
- Tipos de campos disponibles



Valores máximos

En específico, la longitud de registro máxima que pueden soportar la mayoría de los manejadores está en el rango de 4000 bytes, mientras que Informix sube a 32000 y Clipper hasta 65000, lo cual marca una gran diferencia.

Requerimientos de hardware

En cuanto a memoria RAM base la mayoría de los manejadores requieren 640 KB para poder operar, mientras que Clipper baja hasta los 300 KB.

Otro punto es en las memorias extendida y expandida, solamente Foxpro e Informix recomiendan el uso de estos tipos de memoria adicional, la cual ayuda a que el manejador gane velocidad con respecto a otros manejadores, y al no contar con dicha memoria se ve reducida drásticamente esa ventaja.

En cuanto al espacio de disco duro requerido, en general podríamos considerar que una media de espacio requerido pudiera ser de aproximadamente 4 MB por lo cual se considera que R:Base y Foxpro consumen demasiado espacio para su instalación.

Tipos de campos disponibles.

El último punto en el que se nota una gran variación entre las calificaciones es en los tipos de campos que se tienen disponibles en los diferentes manejadores, aquellos manejadores que carecen de campos de punto flotante y de longitud variable se consideran con calificación muy baja.

Podemos concluir, que en base a los requerimientos indicados por el usuario y a las características que ofrece cada manejador de base de datos, el manejador idóneo para desarrollar la aplicación es el compilador Clipper.

III.2.1 DESCRIPCION DE FUNCIONES DE LOS MODULOS DEL SISTEMA

A continuación se describe cual es la función que realizará cada uno de los módulos que integran el sistema de base de datos.

MODULO PROGRAMACION

El módulo programación estará formado por cinco submódulos (figura III.2.1.A):

- **Captura**
- **Consultas**
- **Modificación**
- **Reportes**
- **Eliminación**

SUBMODULO DE CAPTURA

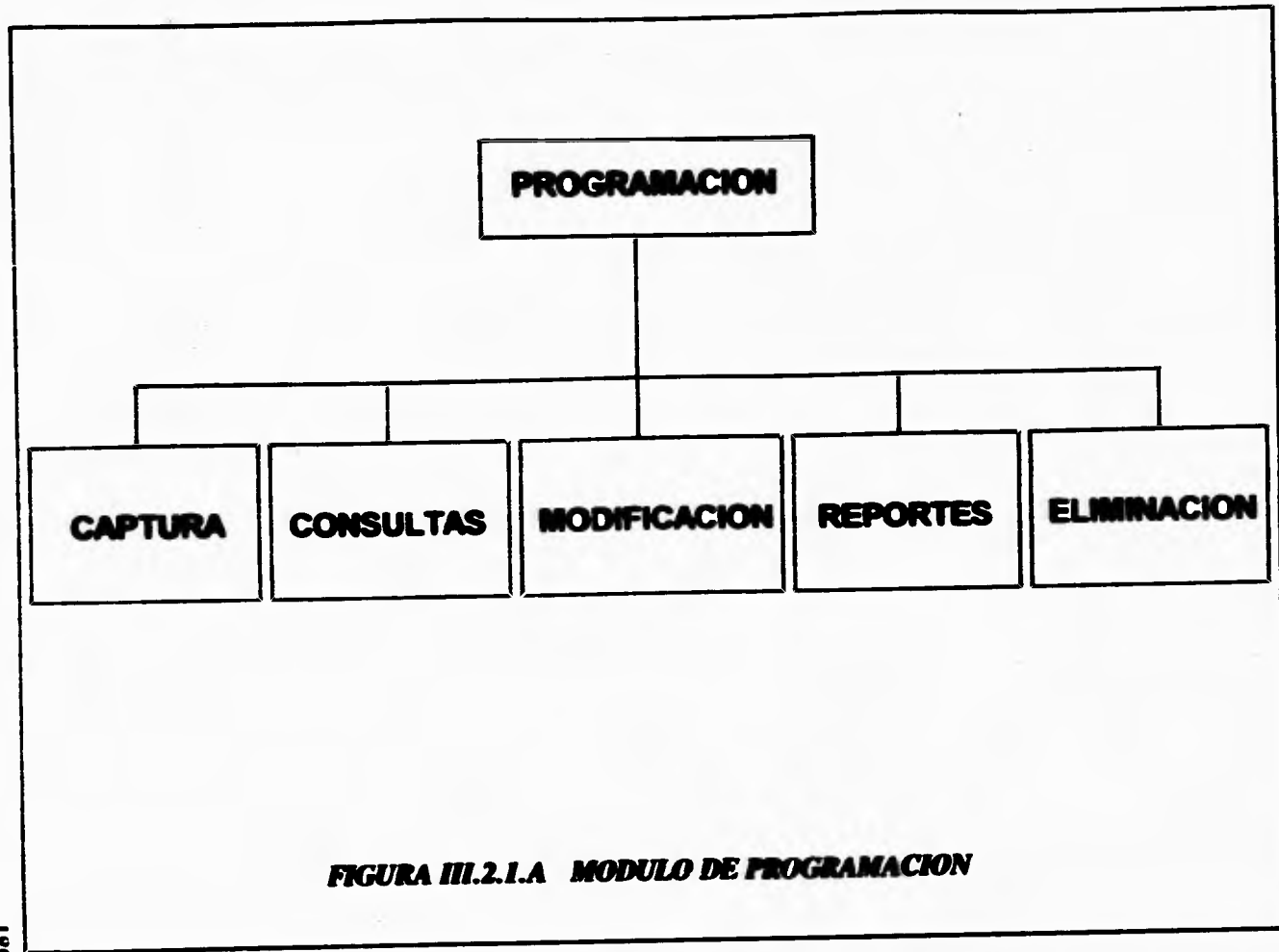
Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.B):

Solicitará el nombre de la disciplina a la que pertenece la actividad. Una vez proporcionado el nombre de la disciplina, se deberá validar su existencia, de lo contrario se debe indicar que esa disciplina no existe y que deberá ser dada de alta o en su defecto indicar otra disciplina que ya haya sido dada de alta. Este campo es obligatorio.

Solicitará las fechas y horarios en los que se presenta la actividad. Para el caso específico de "Cursos y Talleres" deberá solicitar fecha de inicio y terminación del curso o taller así como la hora de inicio y de terminación. Se podrán dar más de una fecha y hora de presentación para cada actividad. Estos campos son obligatorios.

Solicitará el nombre o descripción de la actividad. Este campo es obligatorio.

Solicitará el (los) nombre(s) del (de los) artista(s), director(es), expositor(es), etc.



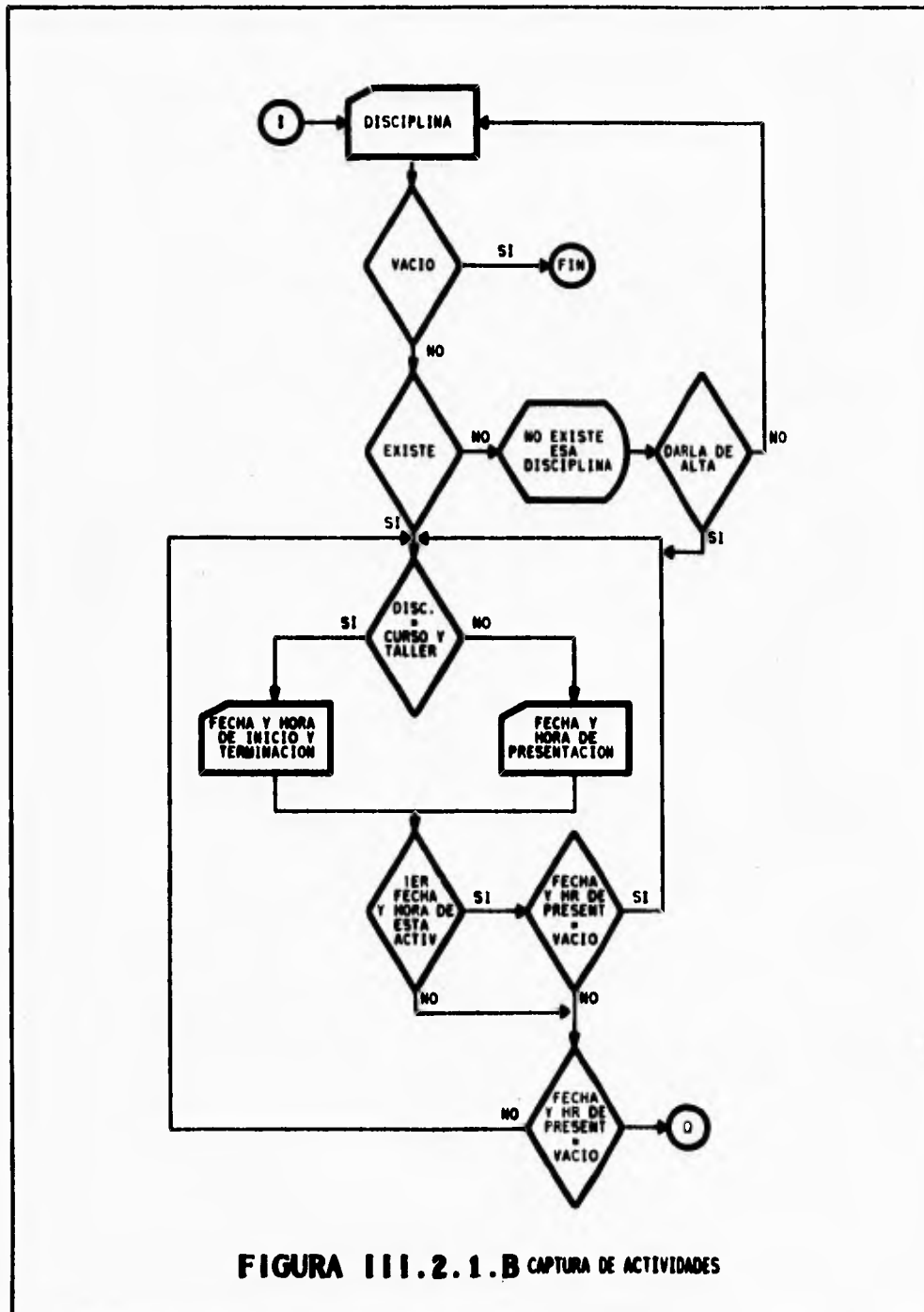


FIGURA III.2.1.B CAPTURA DE ACTIVIDADES

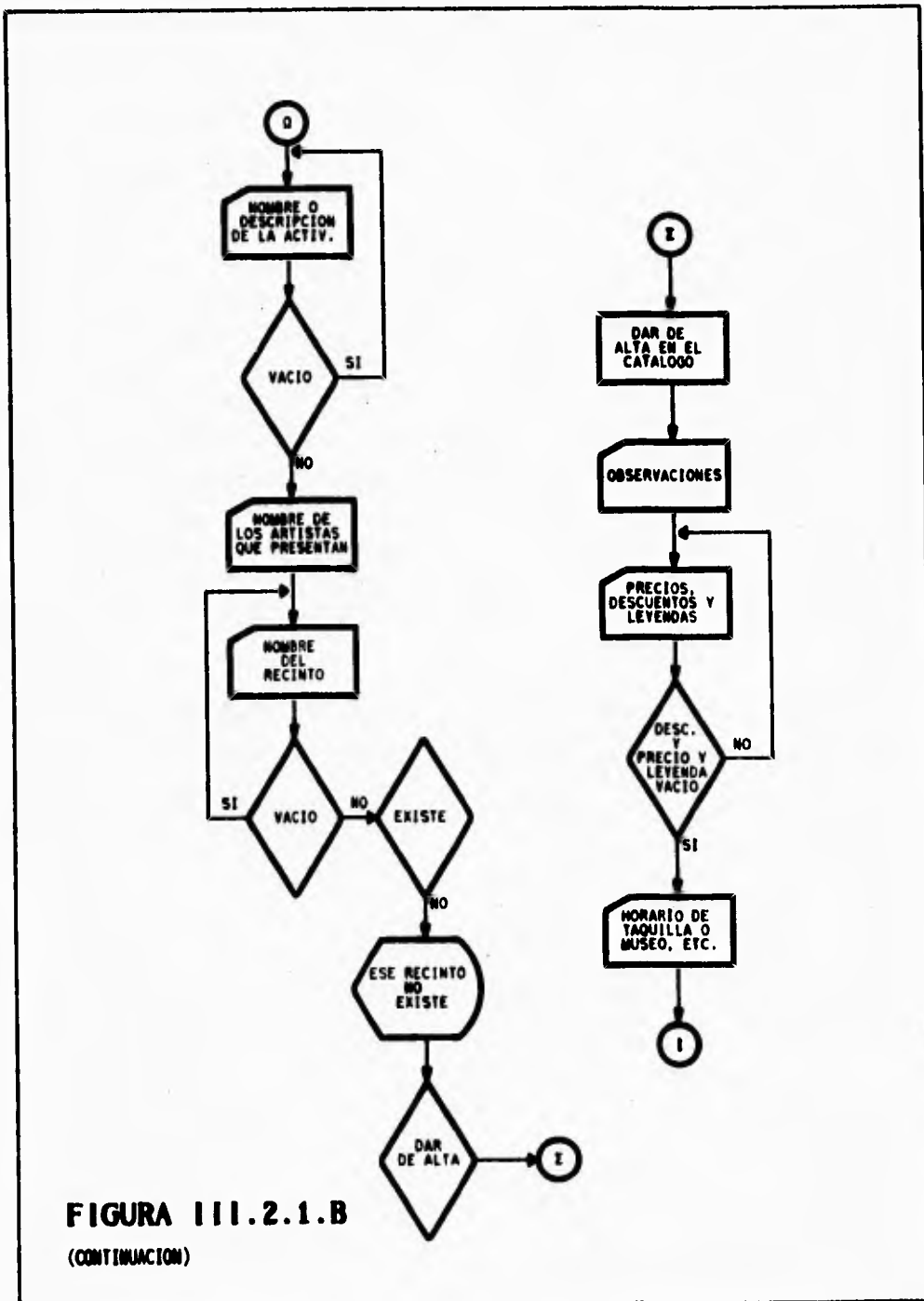


FIGURA III.2.1.B
(CONTINUACION)

Solicitará el nombre del lugar en donde se llevará a cabo la actividad, se deberá validar su existencia, de lo contrario se debe indicar que ese recinto no existe y que deberá ser dado de alta o en su defecto indicar otro recinto que ya haya sido dado de alta. Este campo es obligatorio.

Solicitará observaciones referentes a la actividad, como pueden ser estilo, algún comentario referente a la actividad, etc. Este campo es opcional.

Solicitará el precio, porcentaje de descuento y una leyenda alusiva al precio o descuento. Se podrán dar mas de un precio, descuento y/o leyenda para cada actividad. Estos campos son opcionales.

Solicitará horario de taquilla, museo, etc. Este campo es opcional.

Al terminar de proporcionar la información requerida, el programa borrará la pantalla y pedirá nuevamente los datos para otra actividad a capturar. Para salir de éste submódulo, el usuario deberá dejar en blanco el nombre de la disciplina u oprimir la tecla [ESC] para regresar al menú anterior.

SUBMODULO DE CONSULTAS DE PROGRAMACION

El submódulo de consultas está integrado por los siguientes submódulos:

- Consulta por disciplina
- Consulta por actividad
- Consulta por lugar
- Consulta por rango de fecha de presentación
- Consulta por artista
- Consulta por hora de presentación

- Consulta por disciplina y fecha de presentación
- Consulta por disciplina y lugar de presentación

Ver figura III.2.1.C

CONSULTA POR DISCIPLINA

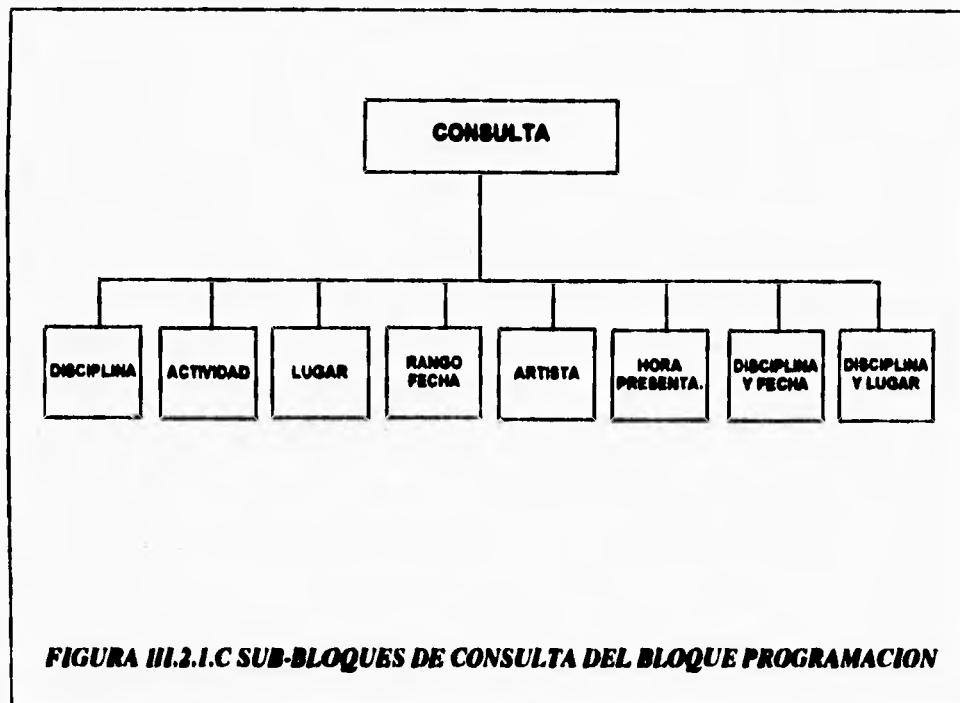
Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.D):

Solicitará el nombre de la disciplina a la que pertenece la actividad. Una vez proporcionado el nombre de la disciplina, se deberá validar su existencia, de lo contrario se debe indicar que esa disciplina no existe e indicar otra disciplina que esté dada de alta o en su defecto regresar al menú anterior dejando el nombre de la disciplina en blanco u oprimiendo la tecla [ESC]. Si la disciplina proporcionada existe, entonces se desplegarán en la pantalla los nombres de las actividades que pertenecen a dicha disciplina (por razones de espacio en pantalla, se desplegarán únicamente los primeros 240 caracteres en cuatro renglones de 60 caracteres cada uno) y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que pertenezcan a la disciplina indicada, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad.

El programa deberá llevar el control de paginación dependiendo de la cantidad de información disponible para la actividad. Al llegar al final de la información, el programa

pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

FIGURA III.2.1.C



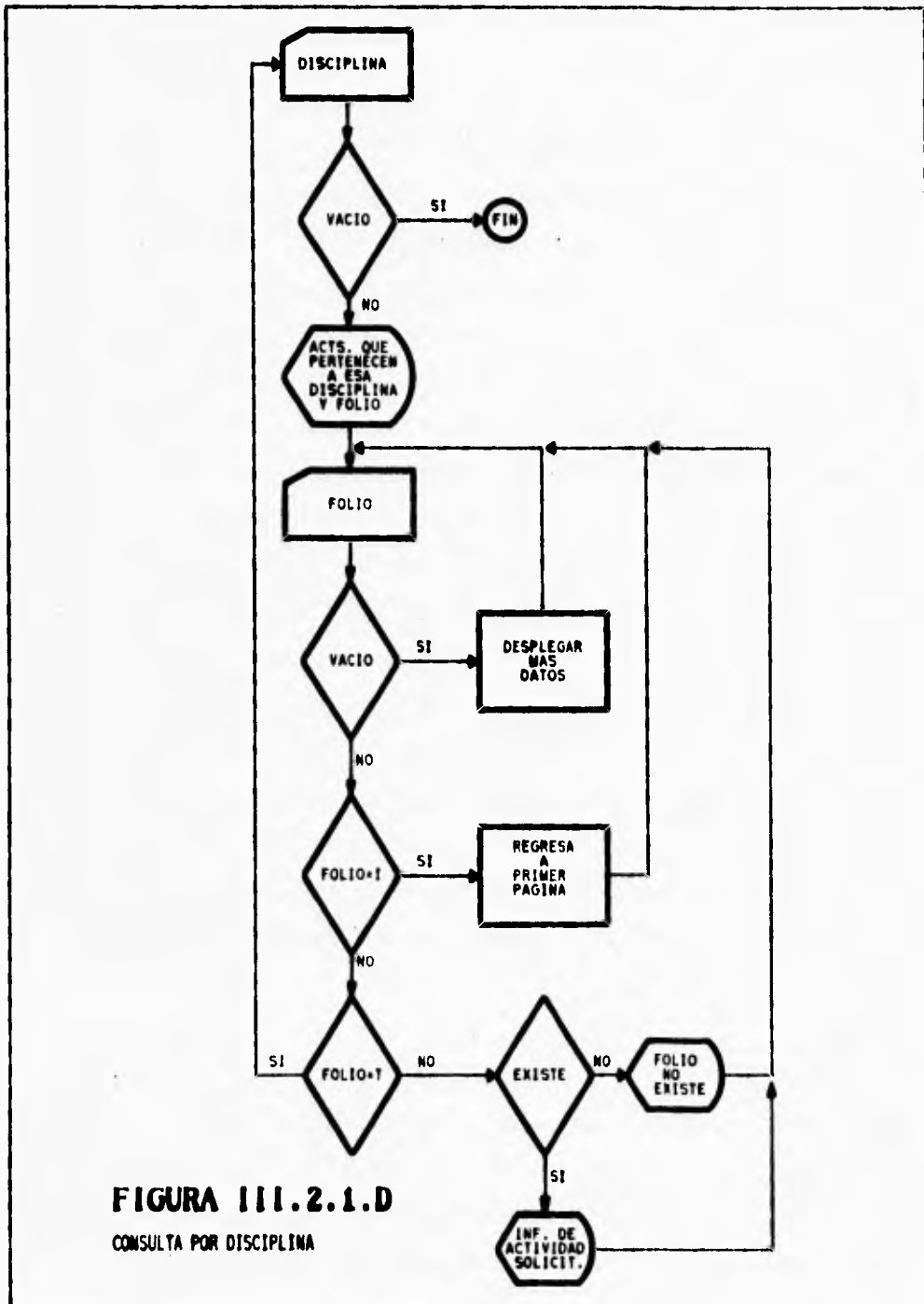


FIGURA III.2.1.D
CONSULTA POR DISCIPLINA

CONSULTA POR ACTIVIDAD

Al entrar a este submódulo, se realizarán las siguientes funciones(figura III.2.1.E):

Solicitará el nombre de la actividad, (se solicitarán únicamente los primeros 240 caracteres, el usuario podrá proporcionar el inicio del nombre de la actividad o una fracción de este, no importando la posición en la que se encuentre dentro del nombre) que se desea consultar. Una vez proporcionado el nombre de la actividad, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades que cumplen con el nombre o fracción del nombre solicitado (por razones de espacio en pantalla, se desplegaran únicamente los primeros 240 caracteres en cuatro renglones de 60 caracteres cada uno) y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que cumplan con el nombre indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad. El programa deberá llevar el control de paginación dependiendo de la cantidad de información disponible para la actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

CONSULTA POR LUGAR

Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.F):

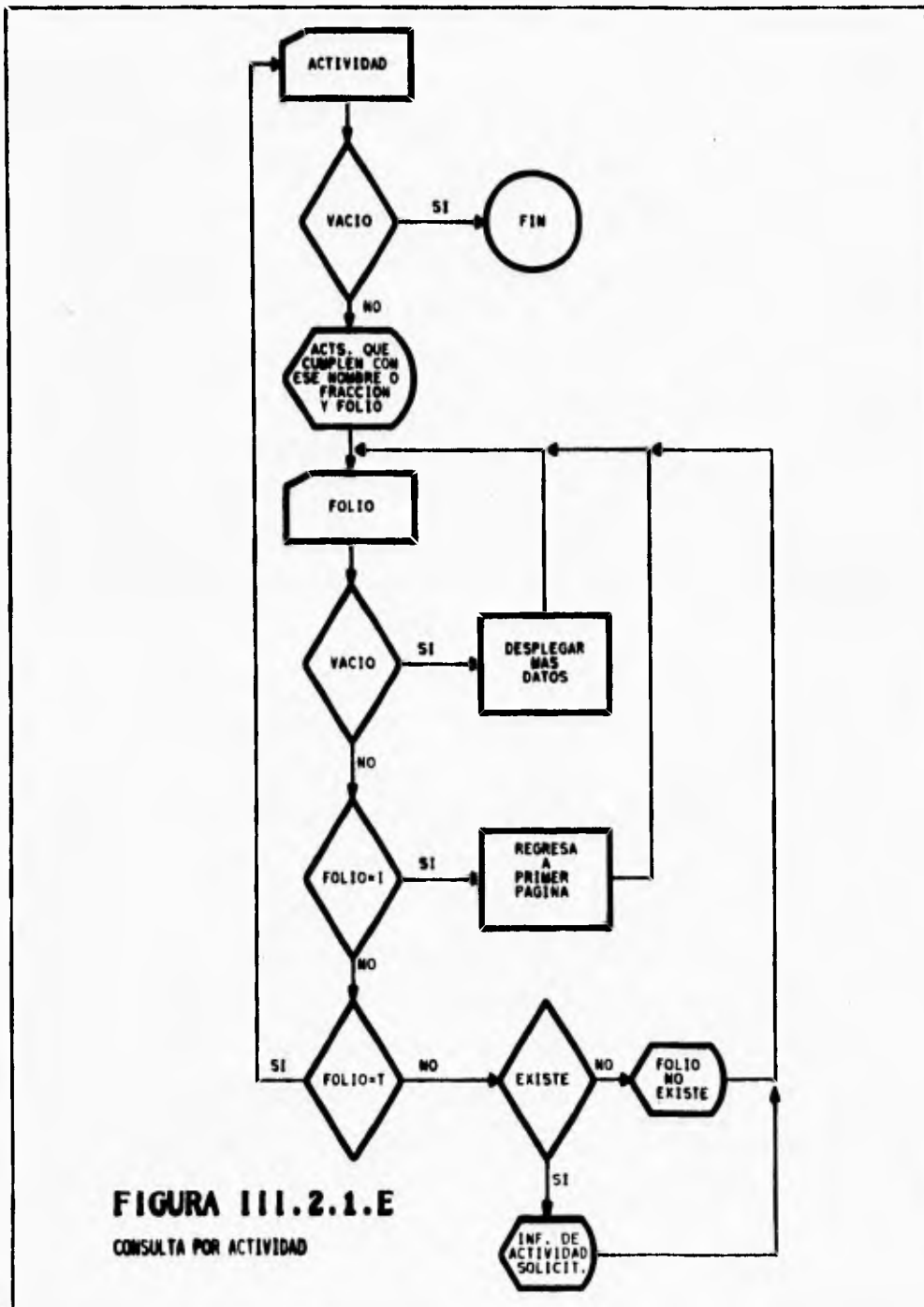
Solicitará el nombre del lugar, el cual servirá como condición de búsqueda para consultar las actividades que se efectúen en ese recinto. Una vez proporcionado el nombre del lugar, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los

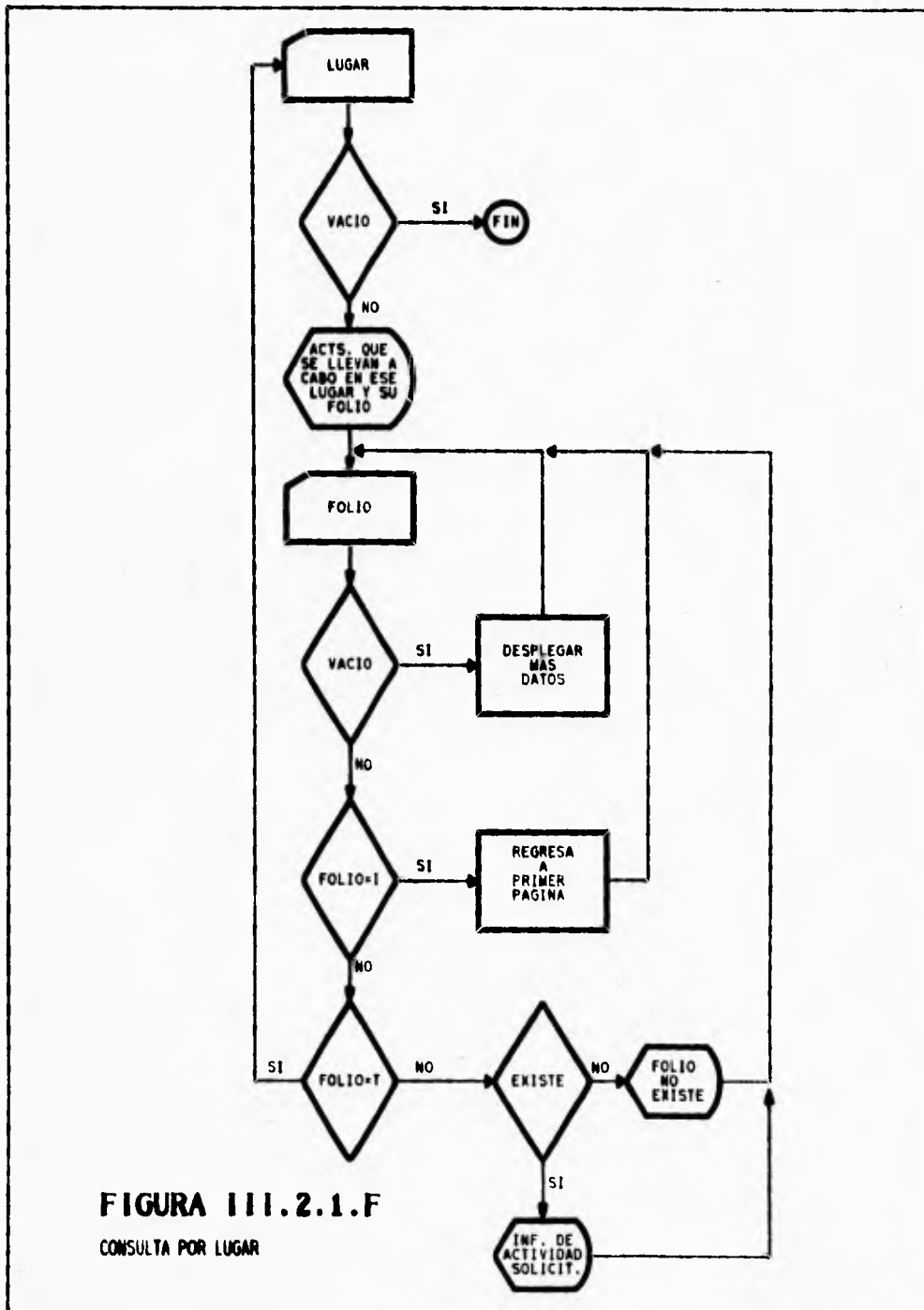
nombres de las actividades que se llevarán a cabo en dicho lugar y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que se efectúen en el lugar indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad. El programa deberá llevar el control de paginación dependiendo de la cantidad de información disponible para la actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

CONSULTA POR RANGO DE FECHA

Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.G):

Solicitará un rango de fecha, el cual servirá como condición de búsqueda para consultar las actividades que se efectúen en esos días. Una vez proporcionado el rango de fecha, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades que están programadas durante esas fechas y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que se realicen en el rango de fecha indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes.





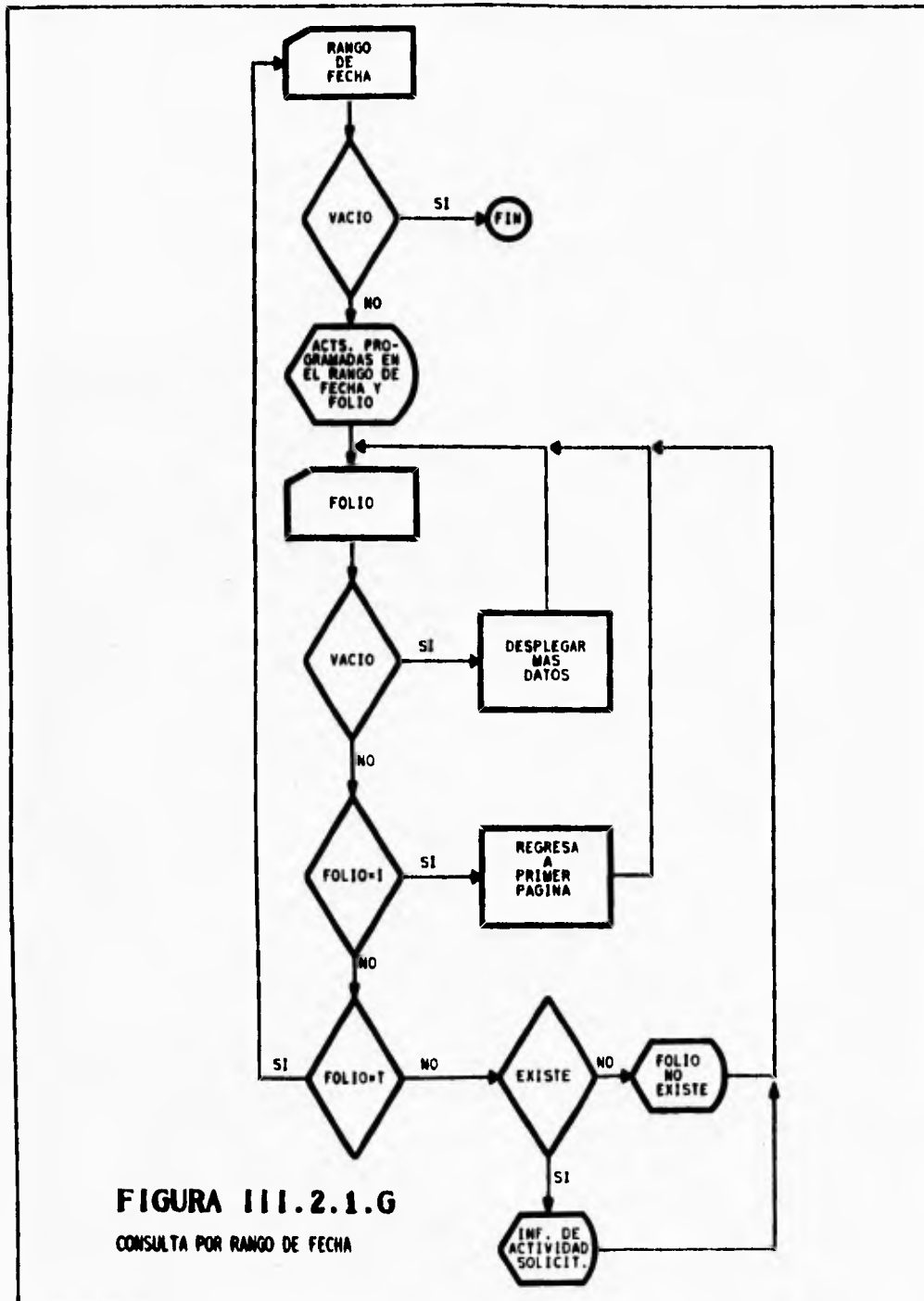


FIGURA III.2.1.G
CONSULTA POR RANGO DE FECHA

Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad. El programa deberá llevar el control de paginación dependiendo de la cantidad de información disponible para la actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

CONSULTA POR ARTISTA

Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.H):

Solicitará un nombre del artista, el cual servirá como condición de búsqueda para consultar las actividades en las que dicho artista participa. Una vez proporcionado el nombre del artista, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades en que participa dicho artista y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que se realicen en el rango de fecha indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desea consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes.

Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad.

El programa deberá llevar el control de paginación dependiendo de la cantidad de información disponible para la actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

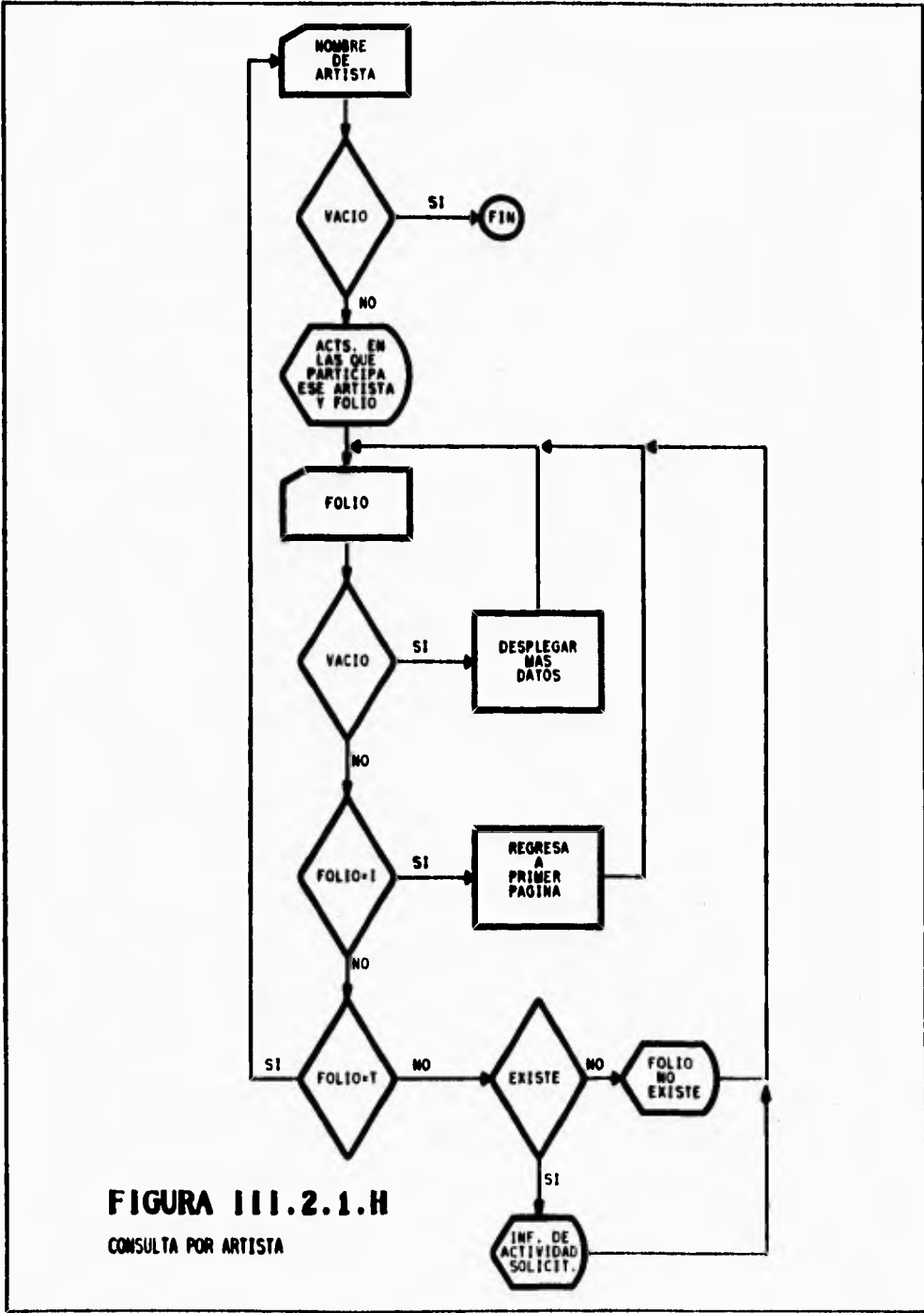


FIGURA III.2.1.H
CONSULTA POR ARTISTA

CONSULTA POR HORA

Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.I):

Solicitará un rango de hora, el cual servirá como condición de búsqueda para consultar las actividades que se efectúen en esos horarios. Una vez proporcionado el rango de hora, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades que están programadas durante esos horarios y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que se realicen en el rango de hora indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad. El programa deberá llevar el control de paginación dependiendo de la cantidad de información disponible para la actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

CONSULTA POR DISCIPLINA Y RANGO DE FECHA

Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.J):

Solicitará un nombre de disciplina y un rango de fecha, los cuales servirán como condición de búsqueda para consultar las actividades que pertenezcan a esa disciplina y que se efectúen en esos días. Una vez proporcionados la disciplina y el rango de fecha, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades que pertenezcan a la disciplina indicada y que están programadas durante esas fechas y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que se realicen en el rango de fecha

indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad. El programa deberá llevar el control de paginación dependiendo de la cantidad de información disponible para la actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

CONSULTA POR DISCIPLINA Y LUGAR

Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.K):

Solicitará el nombre de la disciplina y del lugar, los cuales servirán como condición de búsqueda para consultar las actividades que pertenezcan a esa disciplina y que se efectúen en ese recinto. Una vez proporcionado el nombre de la disciplina y del lugar, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades que pertenezcan a la disciplina indicada y que se llevarán a cabo en dicho lugar y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que se efectúen en el lugar indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea consultar se desplegará en la pantalla toda la información que fue capturada para dicha actividad. El programa deberá llevar el control de paginación dependiendo de la cantidad de

información disponible para la actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a consultar.

SUBMÓDULO DE MODIFICACION DE PROGRAMACION

Al entrar a este submódulo, se realizarán las siguientes funciones (figura III.2.1.L):

Solicitará el nombre de la actividad, (se solicitarán únicamente los primeros 240 caracteres, el usuario podrá proporcionar el inicio del nombre de la actividad o una fracción de este, no importando la posición en la que se encuentre dentro del nombre) que se desea modificar. Una vez proporcionado el nombre de la actividad, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades que cumplen con el nombre o fracción del nombre solicitado (por razones de espacio en pantalla, se desplegarán únicamente los primeros 240 caracteres en cuatro renglones de 60 caracteres cada uno) y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que cumplan con el nombre indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea modificar, se desplegará el nombre de la disciplina a la que pertenece la actividad. Si se modifica el nombre de la disciplina, se deberá validar su existencia, de lo contrario se debe indicar que esa disciplina no existe y que deberá ser dada de alta o en su defecto indicar otra disciplina que ya haya sido dada de alta. Este campo es obligatorio.

Desplegará para modificación las fechas y horarios en los que se presenta la actividad. Para el caso específico de "Cursos y Talleres" desplegará fecha de inicio y terminación del curso o taller así como hora de inicio y hora de terminación. Estos campos son obligatorios

Desplegará para modificación el (los) nombre(s) del (de los) artista(s), director(es), expositor(es), etc.

Desplegará para modificación el nombre del lugar en donde se llevará a cabo la actividad, si este se modifica se deberá validar su existencia, de lo contrario se debe indicar que ese recinto no existe y que deberá ser dado de alta o en su defecto indicar otro recinto que esté dado de alta. Este campo es obligatorio.

Desplegará para modificación observaciones referentes a la actividad. Este campo es opcional, por lo tanto puede quedar modificado o en blanco.

Desplegará para modificación los precios, porcentajes de descuento y leyendas alusivas al precio o descuento. Estos campos son opcionales.

Desplegará para modificación el horario de taquilla, museo, etc. Este campo es opcional. Al terminar de modificar la información requerida el programa borrará la pantalla y pedirá nuevamente los datos para otra actividad a modificar. Para salir de éste submódulo, el usuario deberá dejar en blanco el nombre de la disciplina u oprimir la tecla [ESC] para regresar al menú anterior.

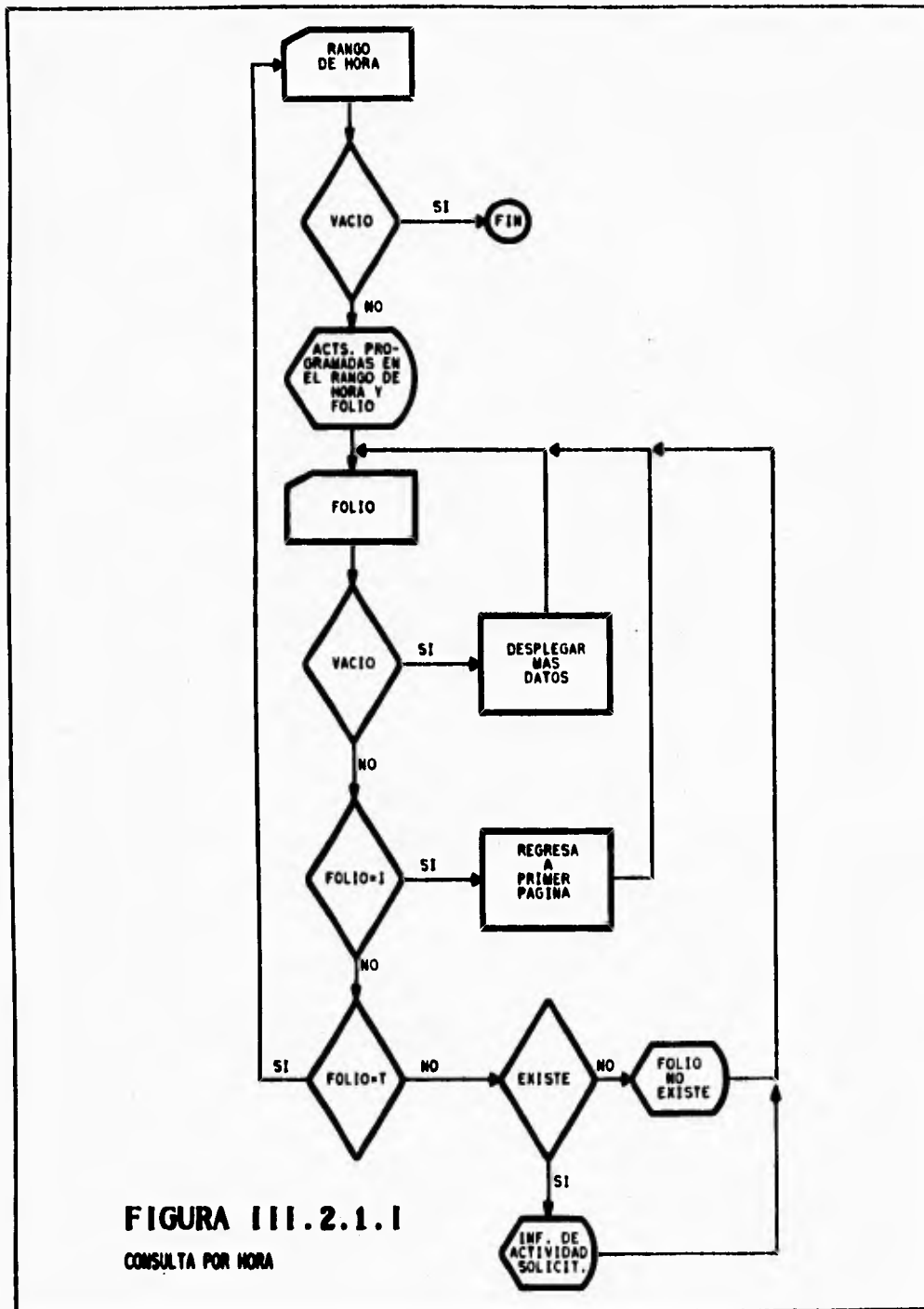


FIGURA III.2.1.1
CONSULTA POR HORA

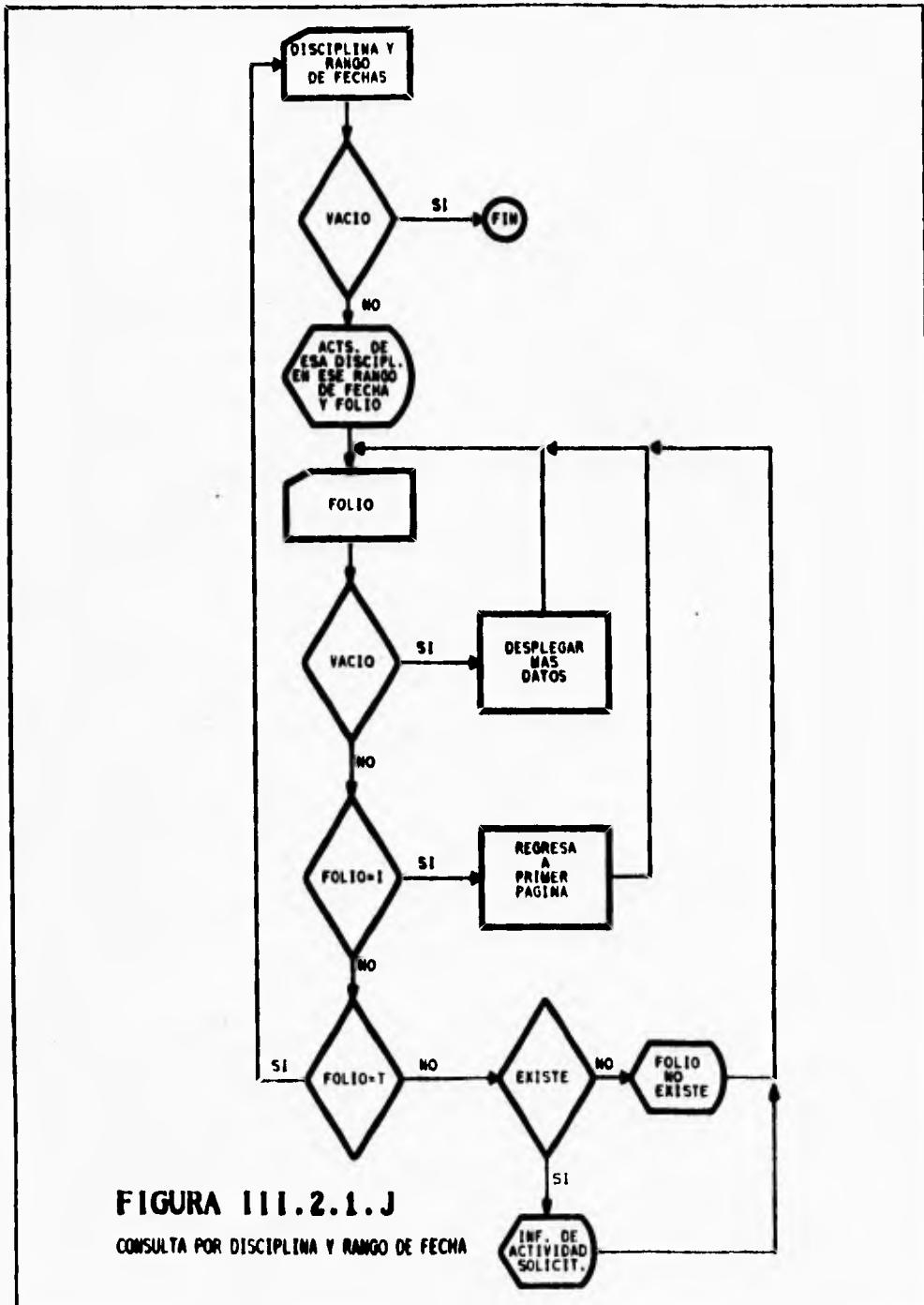


FIGURA III.2.1.J
CONSULTA POR DISCIPLINA Y RANGO DE FECHA

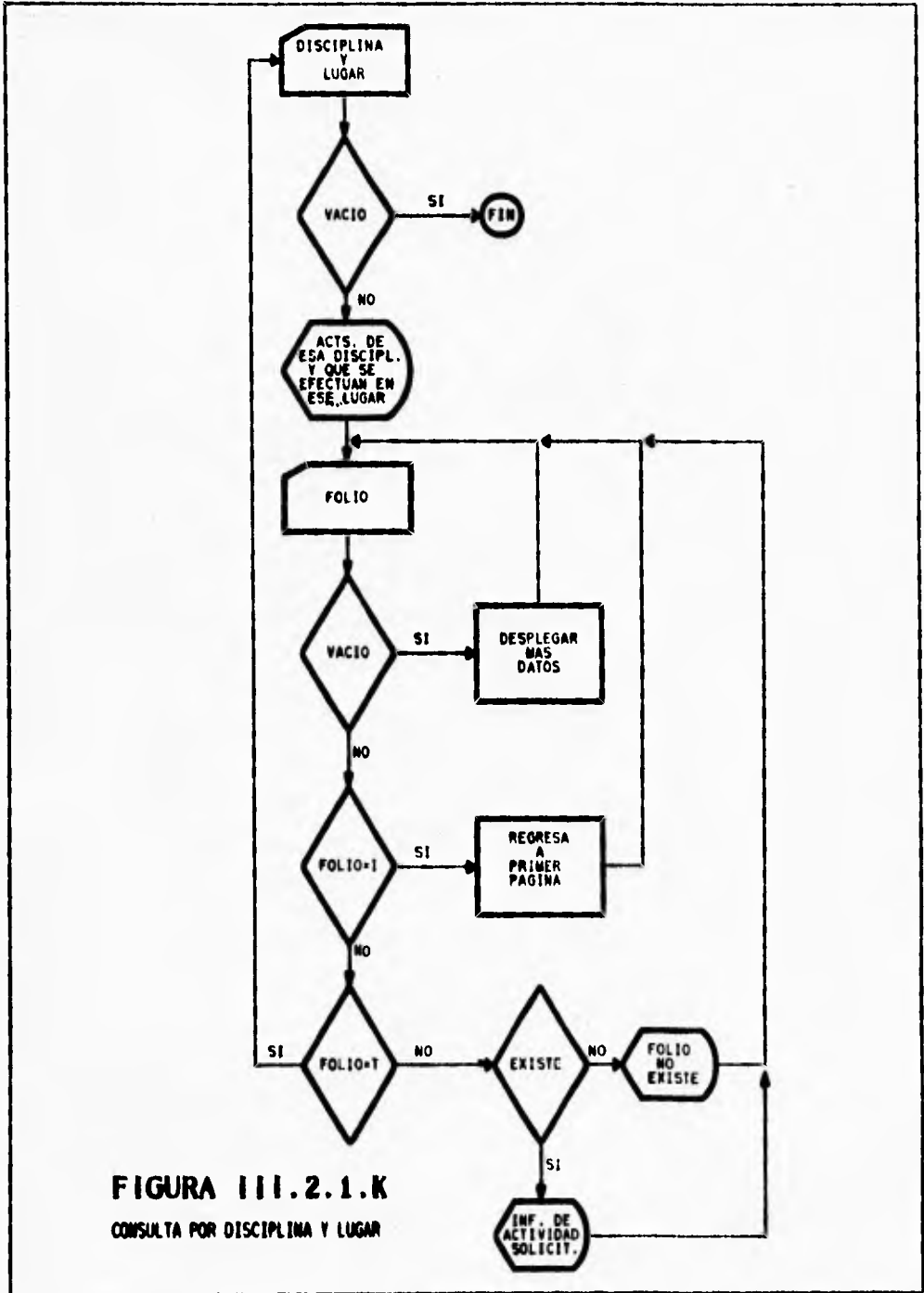
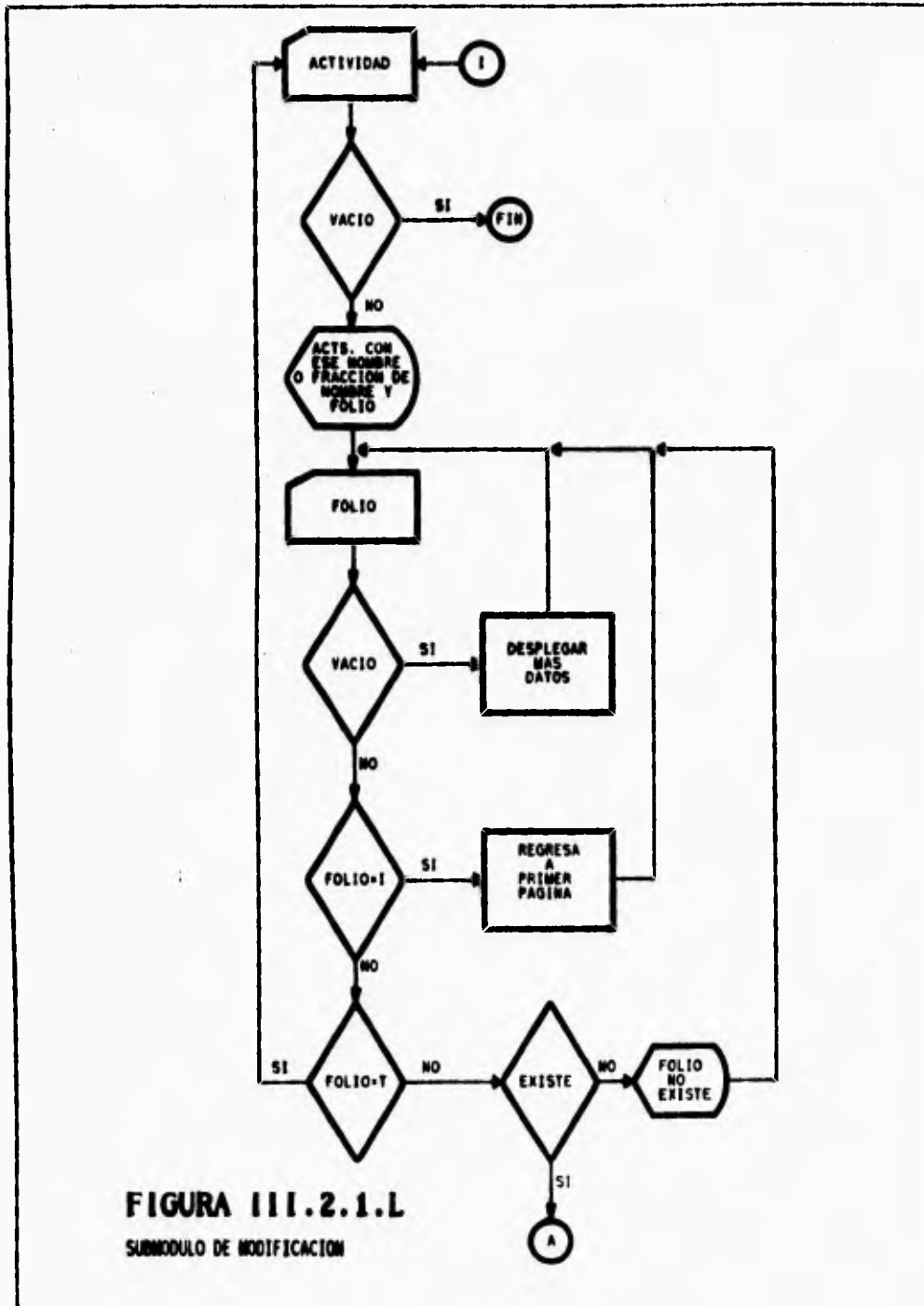


FIGURA III.2.1.K
CONSULTA POR DISCIPLINA Y LUGAR



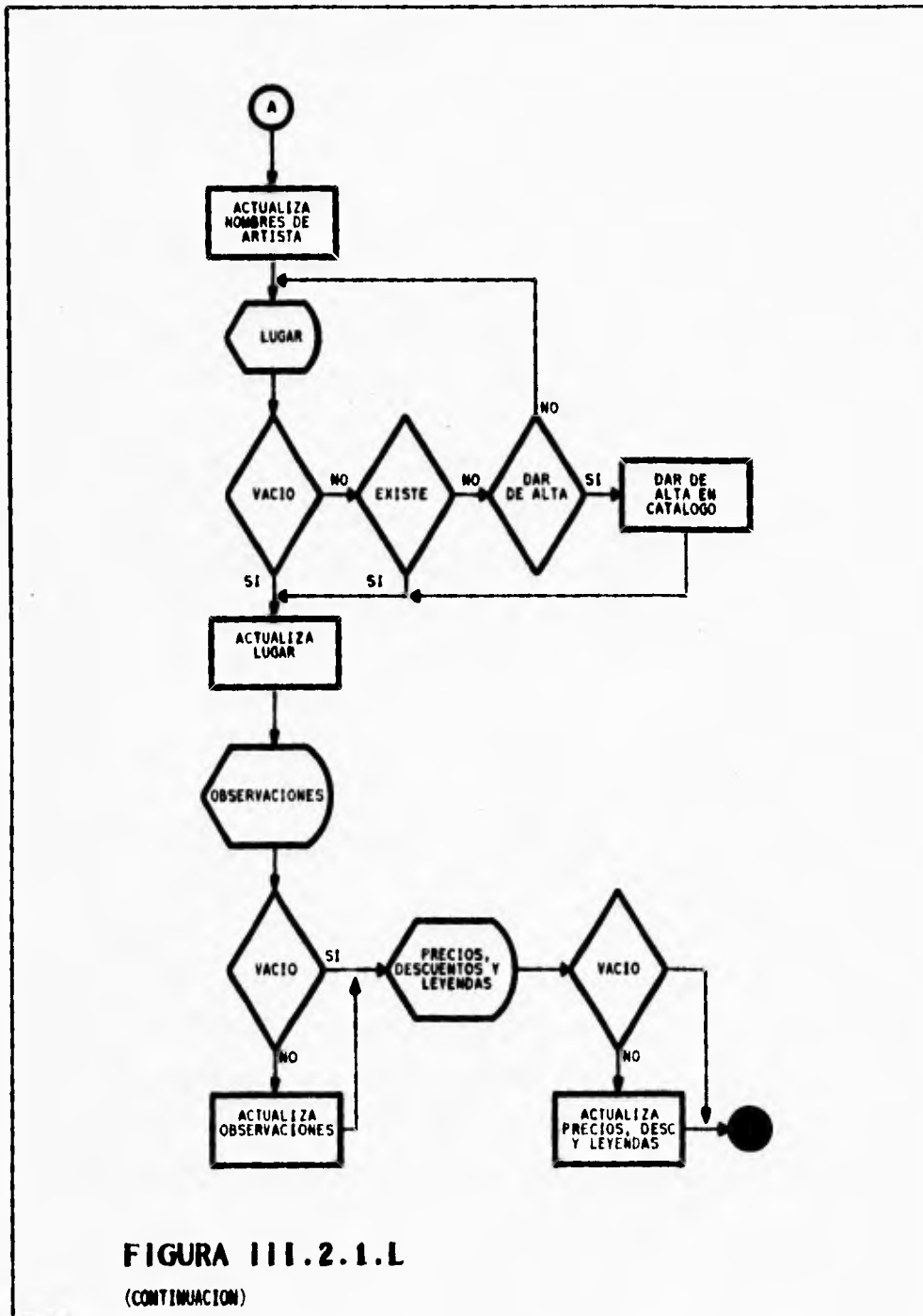


FIGURA III.2.1.L

(CONTINUACION)

SUBMODULO DE REPORTES DEL MODULO PROGRAMACION

Este submódulo está integrado por los siguientes submódulos (figura III.2.1.M):

- **Reporte mensual**
- **Reporte Semanal**

REPORTE MENSUAL

El seleccionar este submódulo, el programa solicitará el nombre del mes del cual se desea imprimir la programación así como el nombre de la disciplina (figura III.2.1.N).

En el reporte se imprimirá toda la información de cada actividad que pertenezca a la disciplina indicada y que se lleven a cabo en el mes indicado. Las actividades estarán ordenadas cronológicamente por fecha y hora de presentación. En el encabezado de cada página del reporte se deberá imprimir el tipo de reporte, el nombre del mes y de la disciplina en cuestión, así como la fecha de emisión del reporte.

REPORTE POR RANGO DE FECHA

El seleccionar este submódulo, el programa solicitará el día de inicio y día en que finaliza el período a imprimir (figura III.2.1.O).

En el reporte se imprimirá toda la información de cada actividad que se lleve a cabo dentro del rango de fecha indicado. No se imprimirán los nombres de artistas ni los precios. Las actividades estarán ordenadas cronológicamente por fecha y hora de presentación, indicando en cada actividad a que disciplina pertenece cada una. En el encabezado de cada página del reporte se deberá imprimir el tipo de reporte, el periodo que se imprime y deberán ir foliadas las páginas.

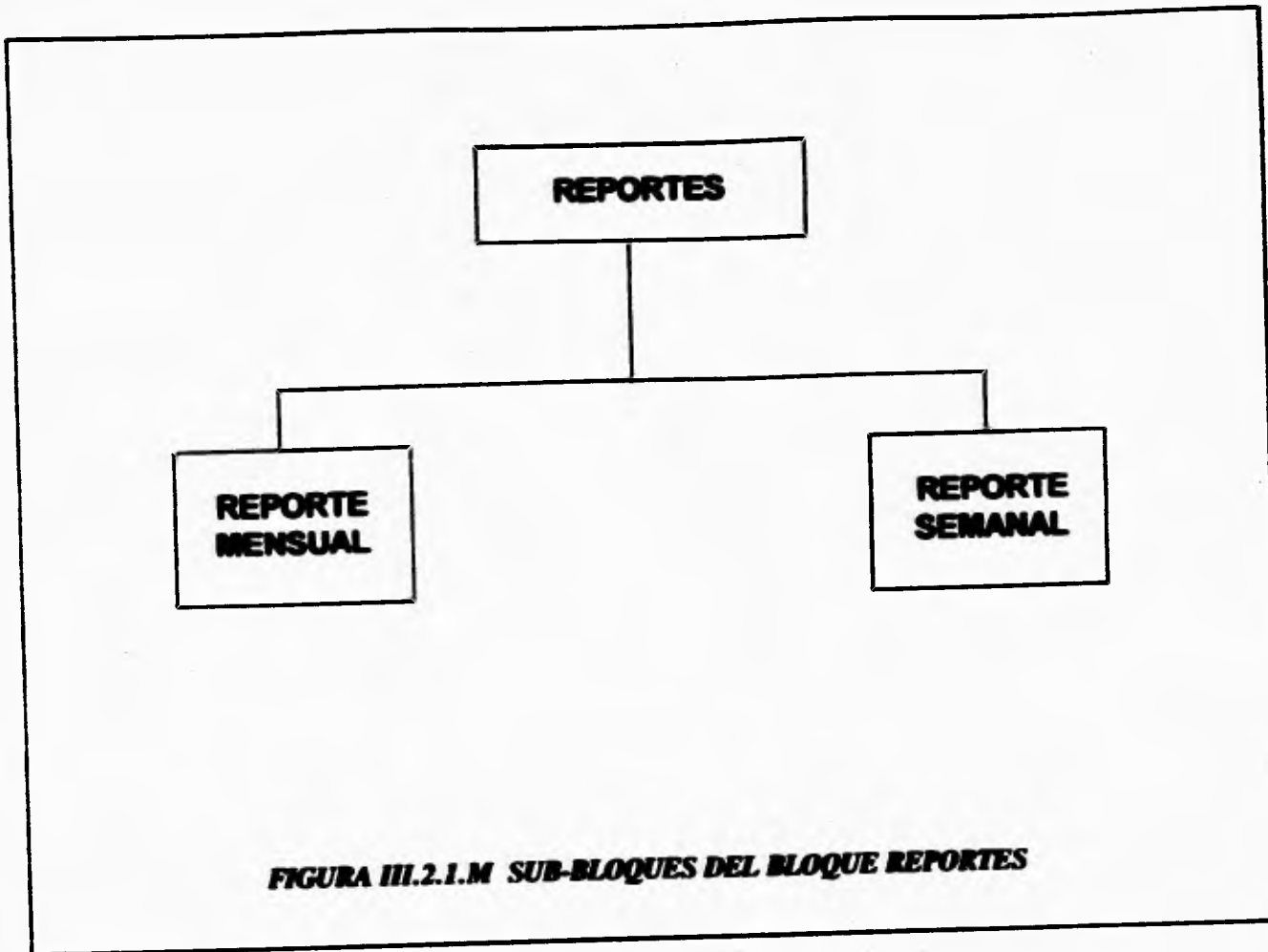


FIGURA III.2.1.M SUB-BLOQUES DEL BLOQUE REPORTES

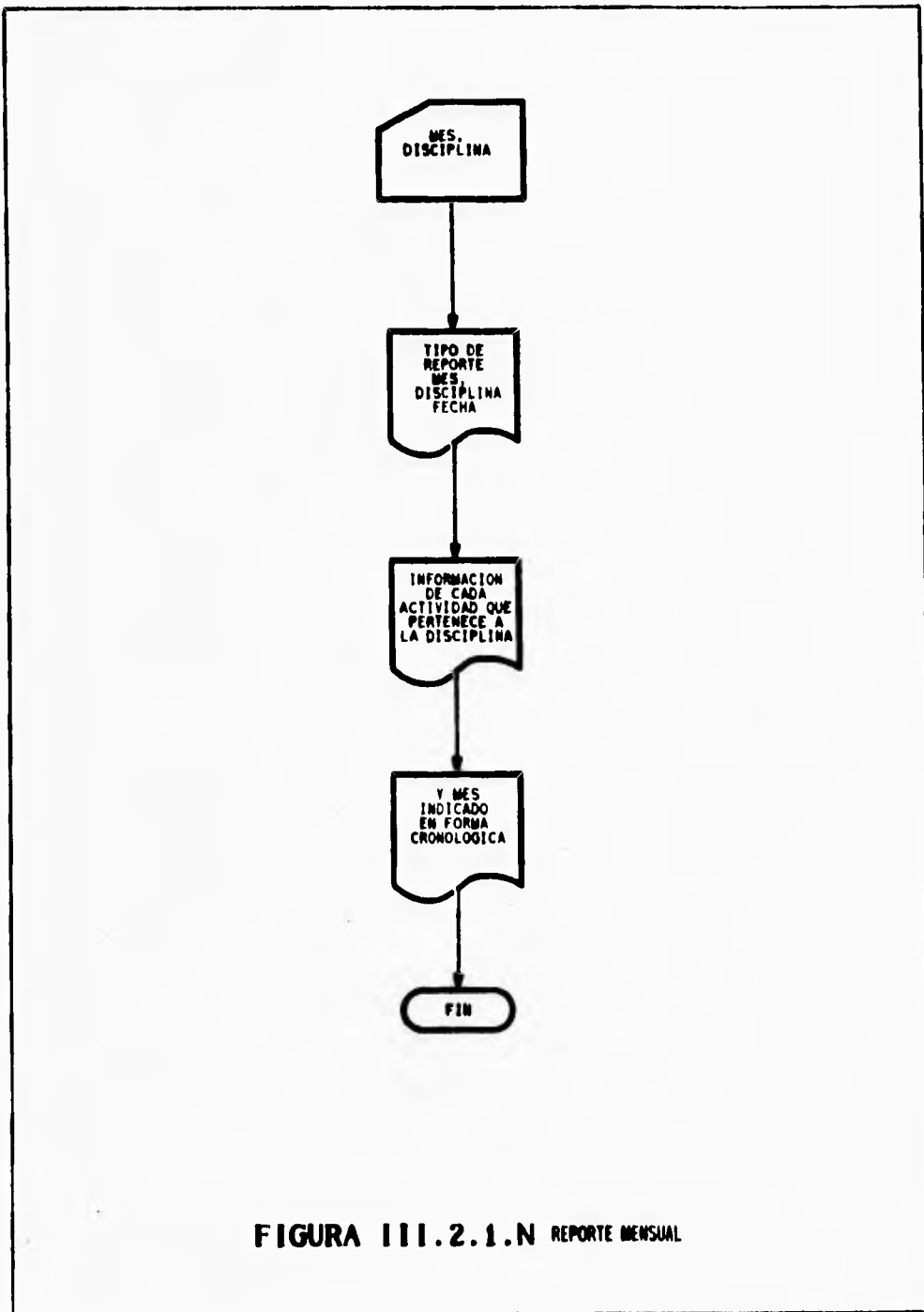


FIGURA III.2.1.N REPORTE MENSUAL

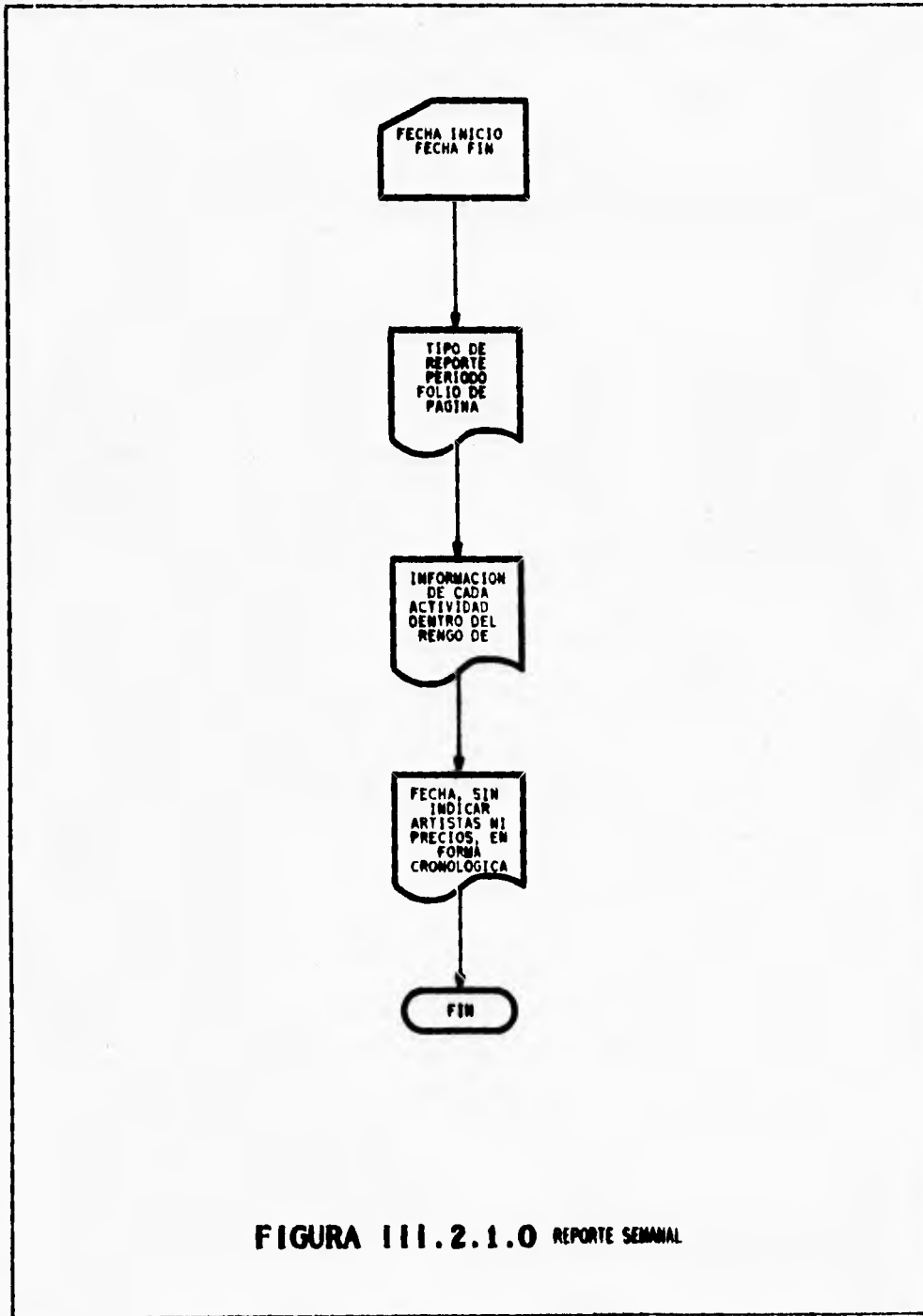


FIGURA III.2.1.0 REPORTE SEMANAL

SUBMODULO DE ELIMINACION DEL MODULO PROGRAMACION

Este submódulo está integrado por los siguientes submódulos:

- **Eliminación de programación mensual**
- **Eliminación de actividad específica**
- **Recuperación de programación mensual**

Ver figura III.2.1.P

La función de este submódulo será poder eliminar actividades que se cancelen, o eliminar la programación de todo un mes, con la finalidad de darle mayor velocidad a los procesos de búsqueda al reducir el número de registros. Al eliminar la información mensual se deberá respaldar dicha información con la finalidad de contar con una historia de los eventos que se han presentado.

ELIMINACION DE PROGRAMACION MENSUAL

El seleccionar este submódulo, el programa preguntará en que unidad de disco (A, B ó C) se respaldará la información que se va a eliminar, una vez seleccionada la unidad de disco el sistema esperará a que se inserte un diskette en la unidad que se especificó, a continuación se pedirá el nombre del mes a eliminar, se eliminará la información referente únicamente al mes en cuestión (figura III.2.1.Q).

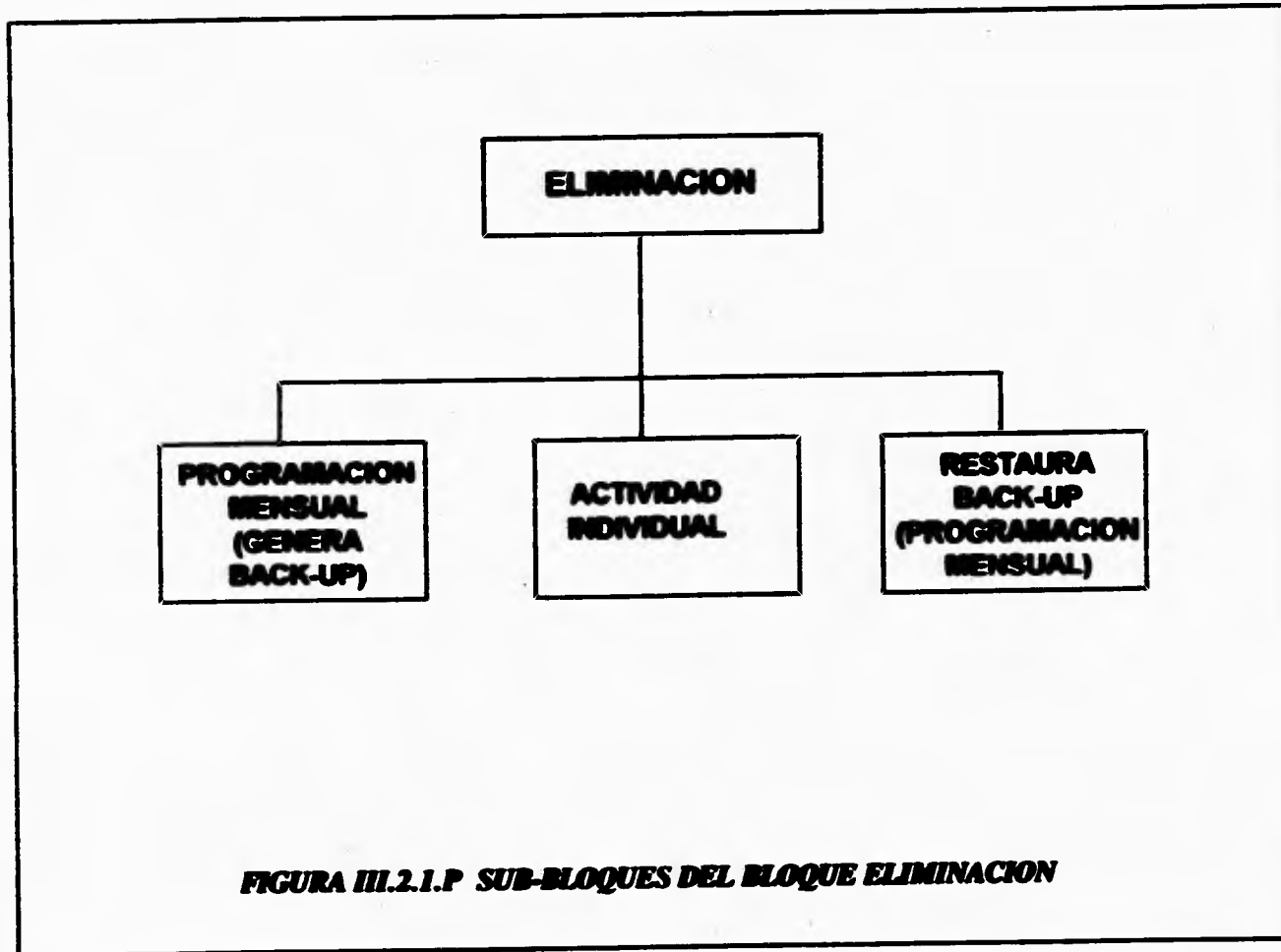
ELIMINACIÓN DE ACTIVIDAD ESPECIFICA

Al seleccionar este submódulo, el programa solicitará el nombre de la actividad, figura III.2.1.R, (se solicitarán únicamente los primeros 240 caracteres, el usuario podrá proporcionar el inicio del nombre de la actividad o una fracción de este, no importando la posición en la que se encuentre dentro del nombre) que se desea eliminar. Una vez proporcionado el nombre de la actividad, se deberá barrer la base de datos desde el inicio para desplegar en la pantalla los nombres de las actividades que cumplen con el nombre o

fracción del nombre indicado (por razones de espacio en pantalla, se desplegarán únicamente los primeros 240 caracteres en cuatro renglones de 60 caracteres cada uno) y al lado izquierdo de éstos un número de folio asociado a cada actividad. Al llenarse la pantalla, o al no encontrar más actividades que cumplan con el nombre indicado, el sistema realizará una pausa para solicitar; a) el número de folio de la actividad que se desea eliminar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado a la actividad que se desea eliminar el sistema preguntará si se está seguro que se desea eliminar dicha actividad. Al llegar al final de la información, el programa pedirá al usuario que oprima la tecla [ENTER] para regresar a la pantalla que solicita los folios de actividades a eliminar.

RECUPERACION DE PROGRAMACION MENSUAL ELIMINADA

Al seleccionar este submódulo, (figura III.2.1.S), el programa solicitará el nombre de la unidad en la que se insertará el diskette que contiene la información a recuperar. Cuando se haya especificado la unidad, el sistema insertará los registros correspondientes y regenerará en forma automática los archivos de índices correspondientes.



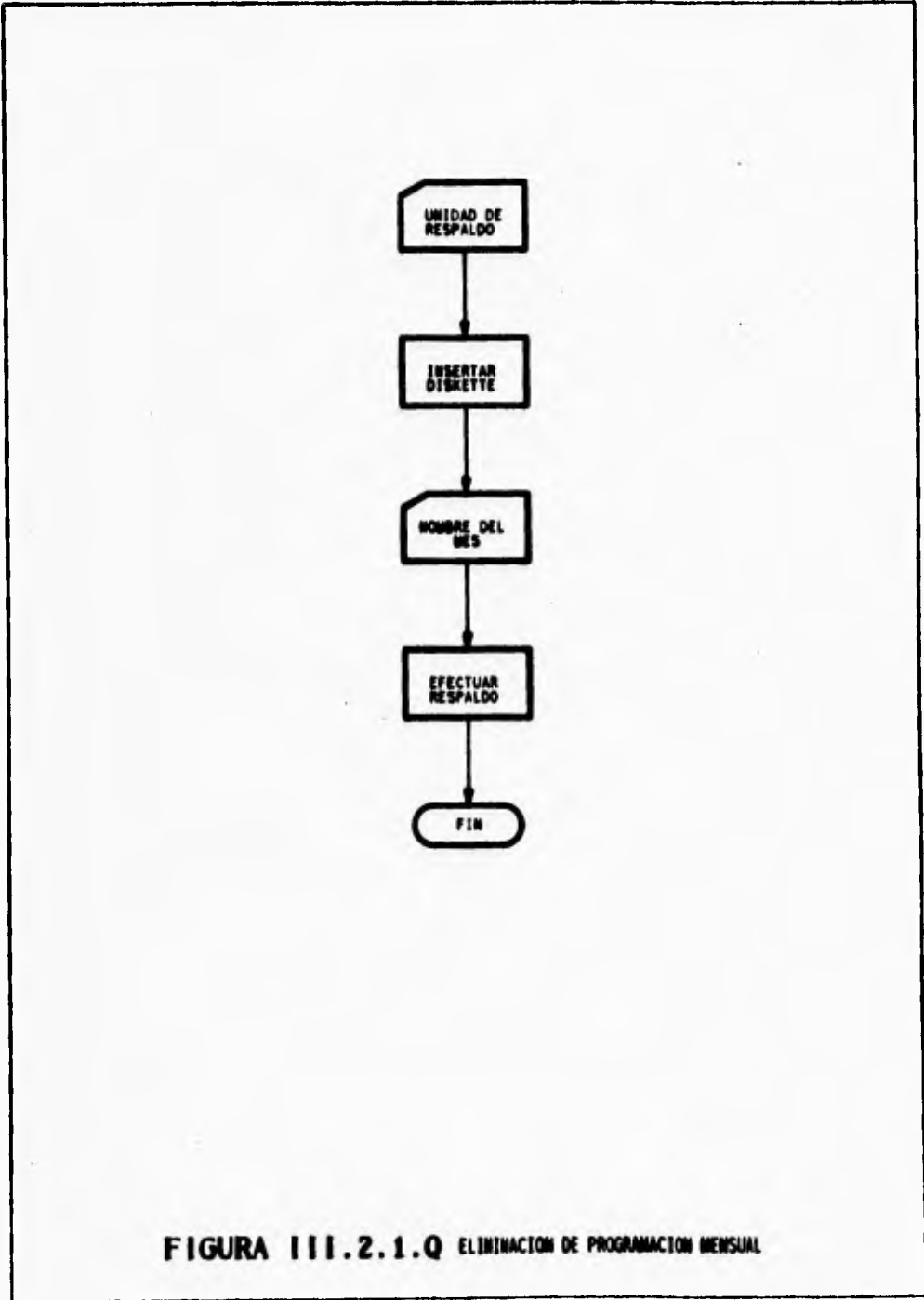


FIGURA III.2.1.Q ELIMINACION DE PROGRAMACION MENSUAL

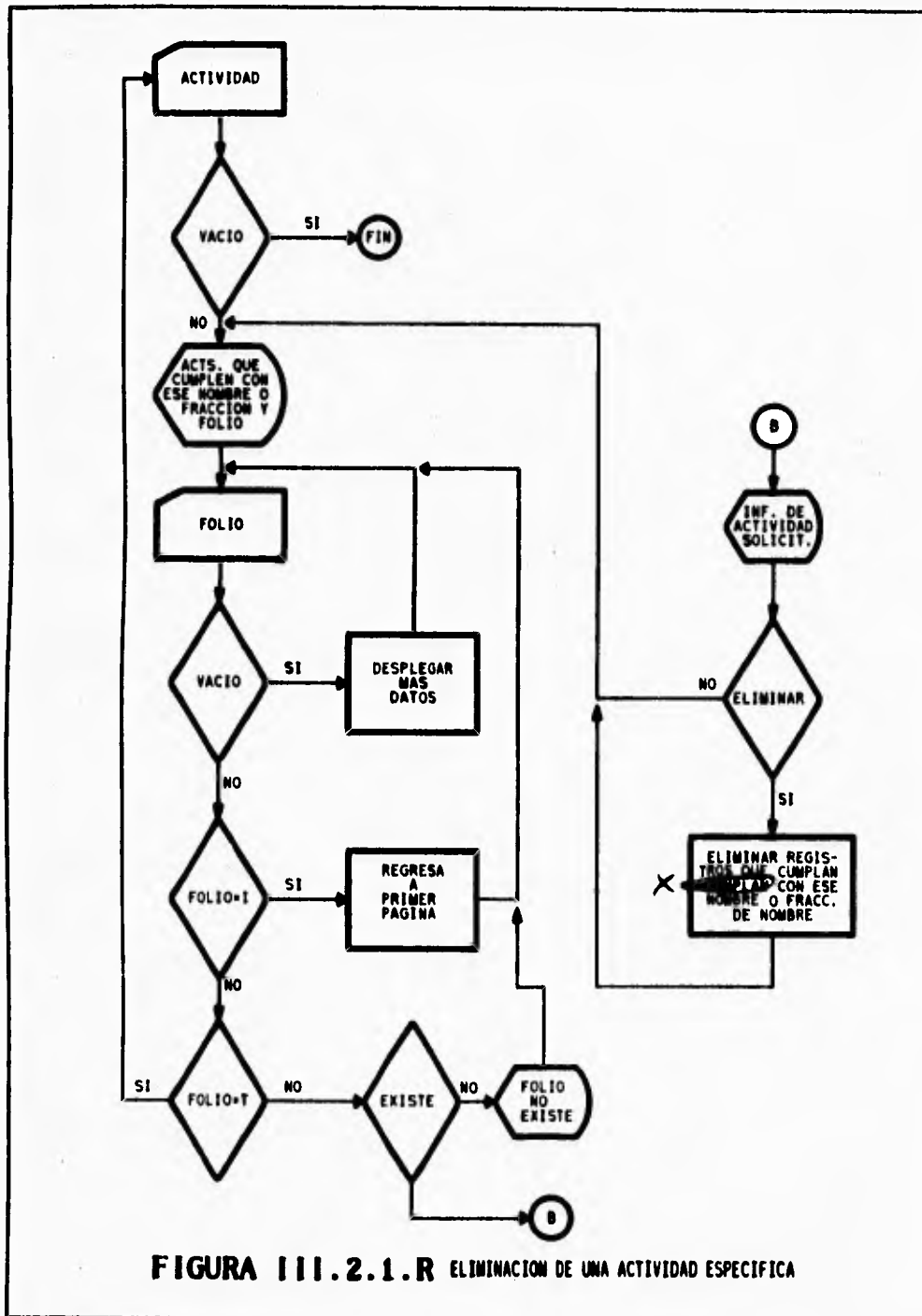


FIGURA III.2.1.R ELIMINACION DE UNA ACTIVIDAD ESPECIFICA

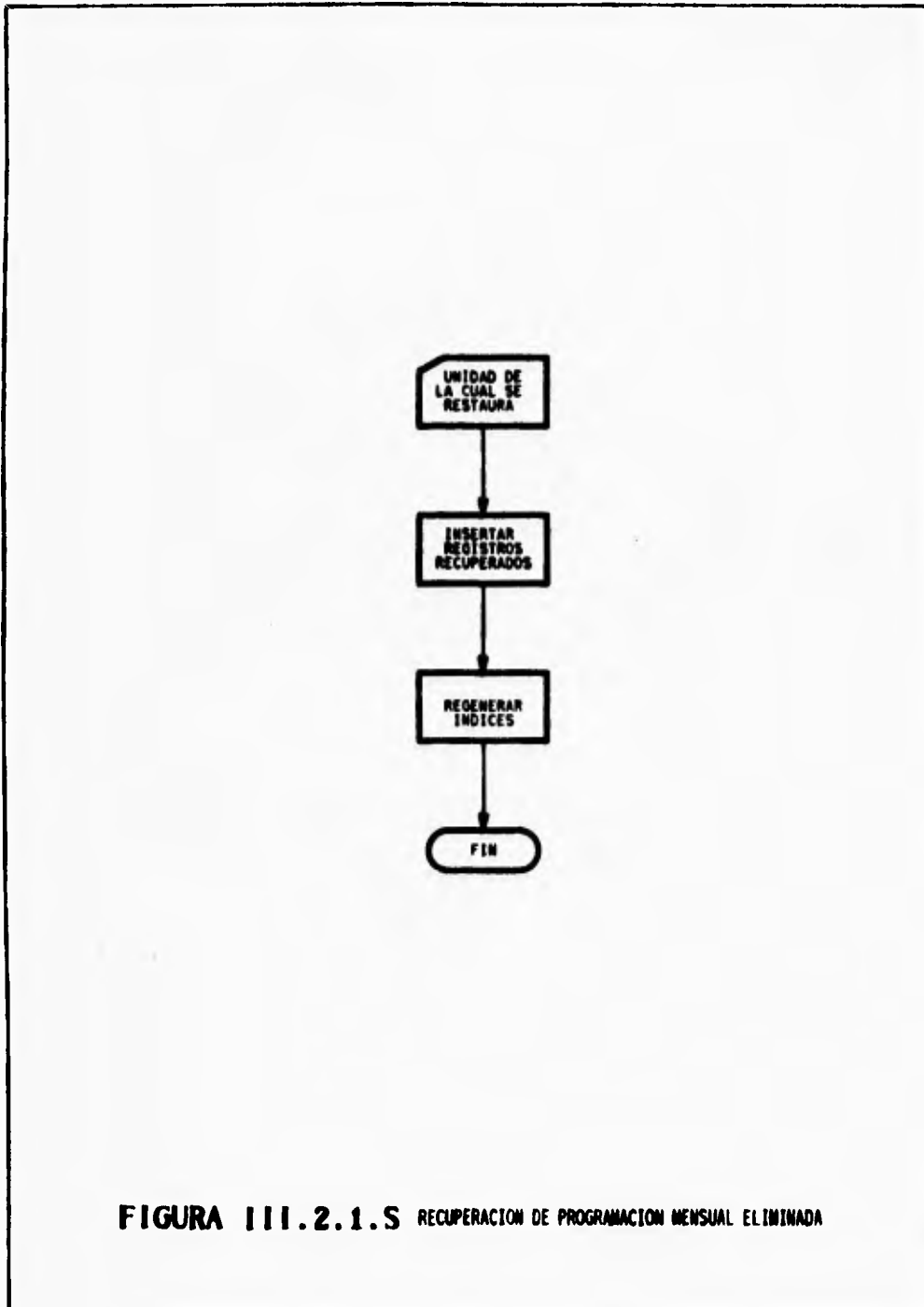


FIGURA III.2.1.S RECUPERACION DE PROGRAMACION MENSUAL ELIMINADA

MODULO DE RECINTOS.

Este submódulo está integrado por los siguientes submódulos:

- Altas
- Consulta
- Cambios
- Bajas
- Reportes

Ver figura III.2.1.T

ALTAS DE RECINTOS

Este submódulo solicitará el nombre del recinto que se dará de alta (figura III.2.1.U), validando que no se repita para garantizar la unicidad, solicitará además, calle, colonia, delegación, ciudad, C.P., clave lada y dos números de teléfono, ninguno de los campos es obligatorio, a excepción del nombre del recinto.

CONSULTA DE RECINTOS

Este submódulo está integrado por los siguientes submódulos:

- Desplegar todos los nombres
- Detalle por nombre
- Por calle
- Por colonia
- Por delegación
- Por ciudad
- Por código postal

Ver figura III.2.1.V

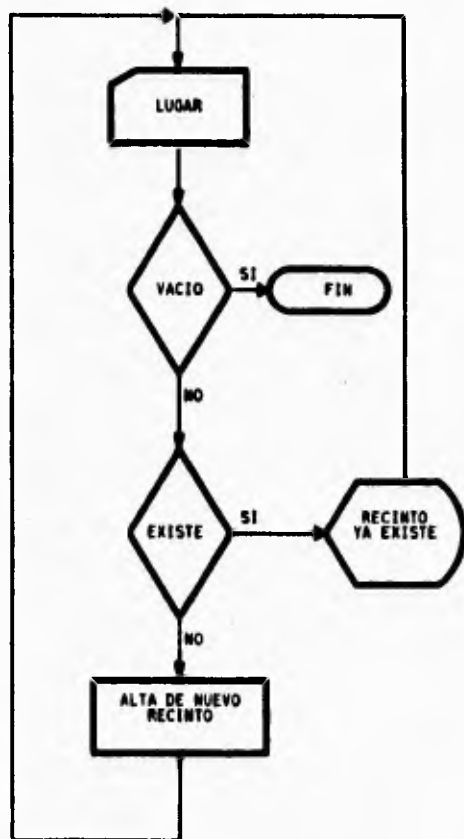
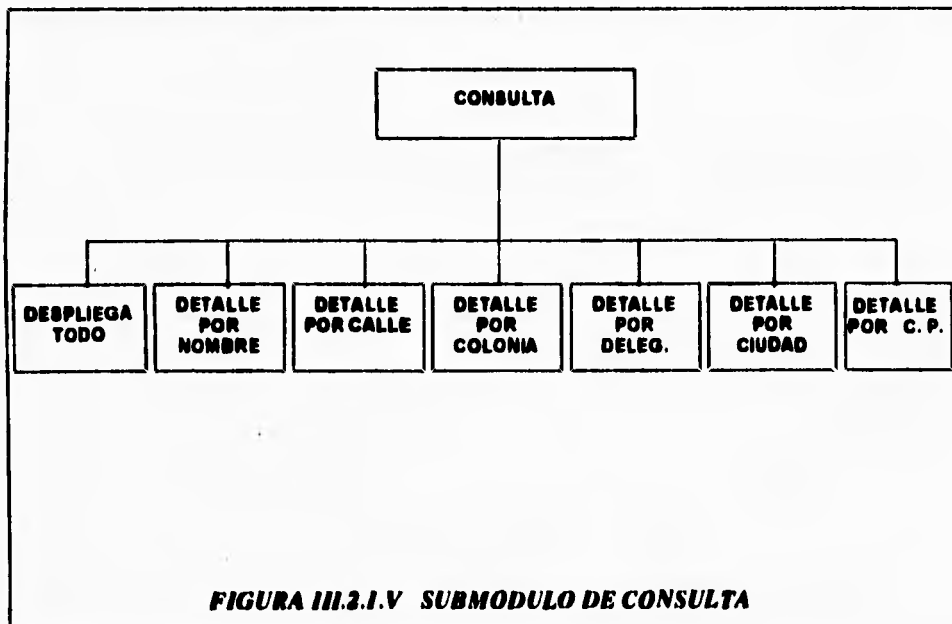
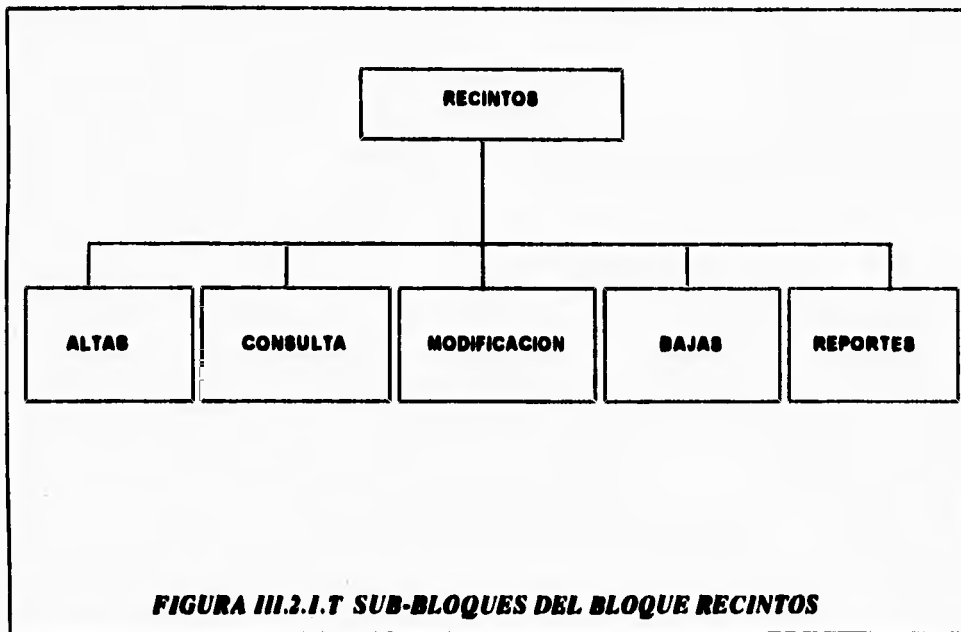


FIGURA III.2.1.U ALTA DE RECINTOS



CONSULTA DESPLEGAR TODOS LOS NOMBRES

Al seleccionar este submódulo (figura III.2.1.W), el programa desplegará todos los nombres de los recintos que se han dado de alta, se desplegará un nombre por cada renglón, al llenarse la pantalla el sistema esperará a que el usuario oprima la tecla [RETURN] para continuar desplegando nombres de recintos hasta llegar al fin de la tabla de recintos.

CONSULTA DE DETALLE POR NOMBRE

Al seleccionar este submódulo (figura III.2.1.X), el programa pedirá el nombre del recinto a consultar, validando la existencia de éste. Se desplegarán todos los datos del domicilio y teléfonos del recinto.

CONSULTA DETALLE POR CALLE

Al seleccionar este submódulo (figura III.2.1.Y), el programa pedirá un nombre de calle, el cual servirá como condición de búsqueda de todos los recintos que se encuentren ubicados en la calle indicada, se desplegarán en la pantalla los nombres de los recintos que cumplen con la condición y al lado izquierdo aparecerá el número de registro correspondiente, al llenarse la pantalla o llegar al fin de la tabla el programa pedirá a) el número de registro del recinto que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de registro asociado al registro que se desea consultar se desplegará en la pantalla toda la información referente a éste.



FIGURA III.2.1.W REPORTE DE TODOS LOS RECINTOS

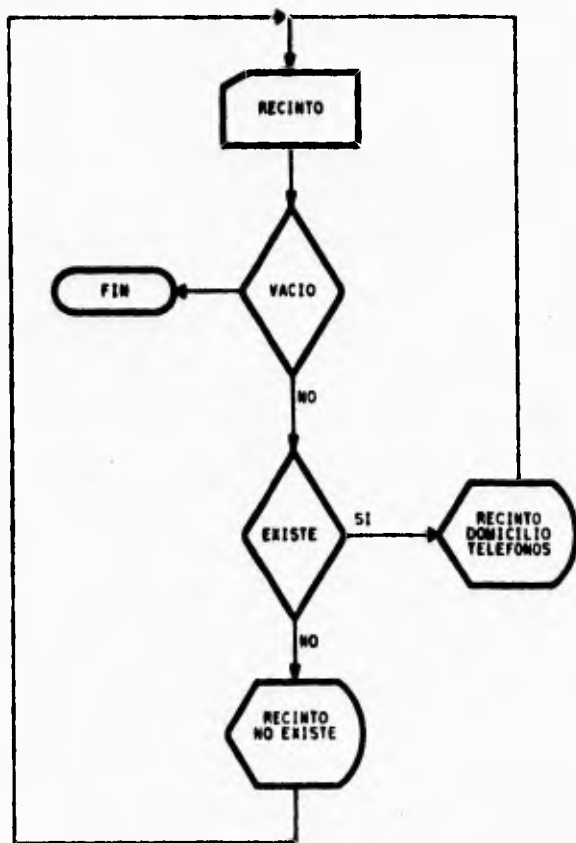
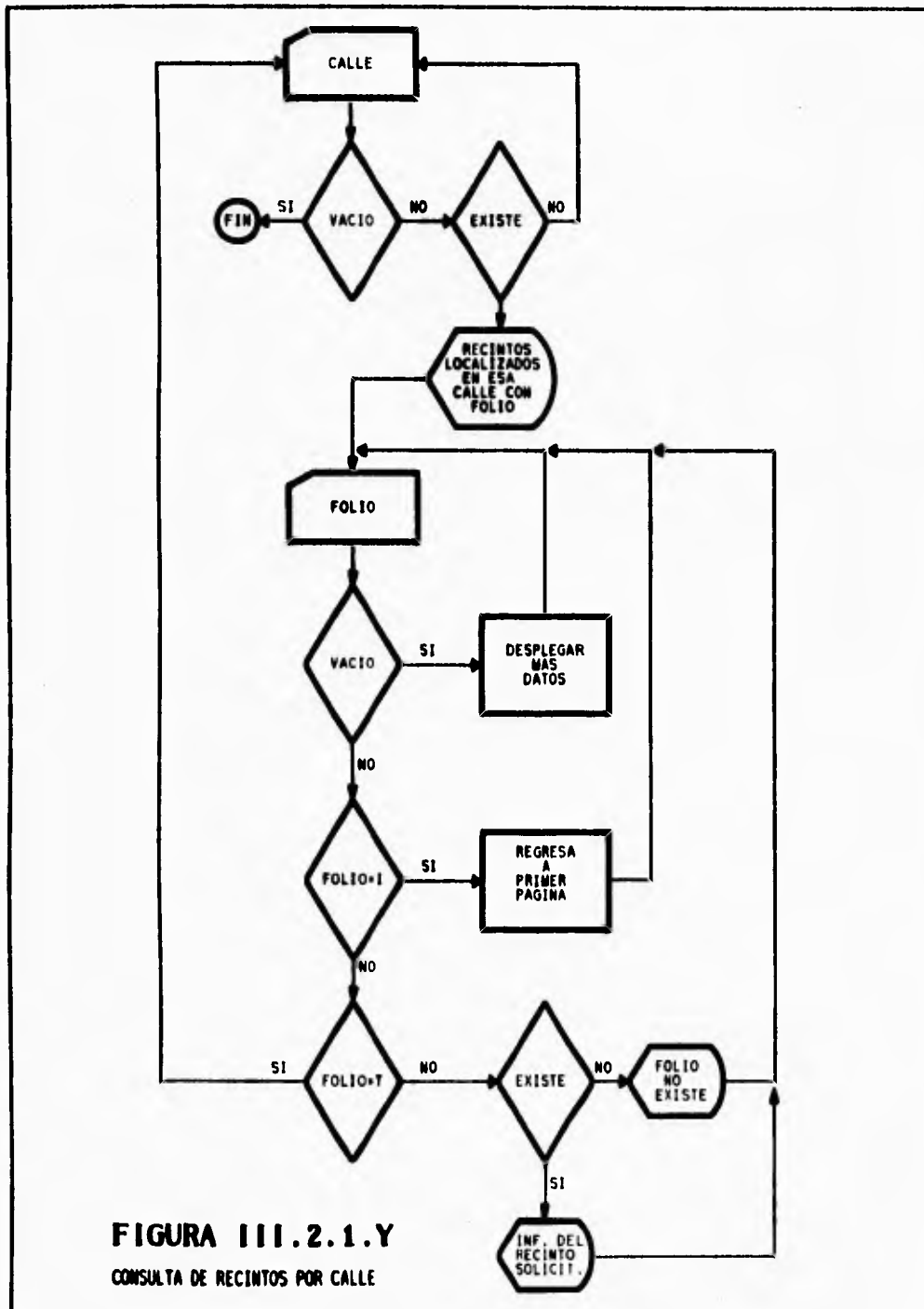


FIGURA III.2.1.X CONSULTA RECINTOS DETALLE POR NOMBRE



CONSULTA DETALLE POR COLONIA

Al seleccionar este submódulo (figura III.2.1.Z), el programa pedirá un nombre de colonia, el cual servirá como condición de búsqueda de todos los recintos que se encuentren ubicados en la colonia indicada, se desplegarán en la pantalla los nombres de los recintos que cumplen con la condición y al lado izquierdo aparecerá el número de registro correspondiente, al llenarse la pantalla o llegar al fin de la tabla el programa pedirá a) el número de folio del recinto que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado al registro que se desea consultar se desplegará en la pantalla toda la información referente a éste.

CONSULTA POR DELEGACION

Al seleccionar este submódulo (figura III.2.1.AA), el programa pedirá un nombre de delegación, el cual servirá como condición de búsqueda de todos los recintos que se encuentren ubicados en la delegación indicada, se desplegarán en la pantalla los nombres de los recintos que cumplen con la condición y al lado izquierdo aparecerá el número de folio correspondiente, al llenarse la pantalla o llegar al fin de la tabla el programa pedirá a) el número de registro del recinto que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado al registro que se desea consultar se desplegará en la pantalla toda la información referente a éste.

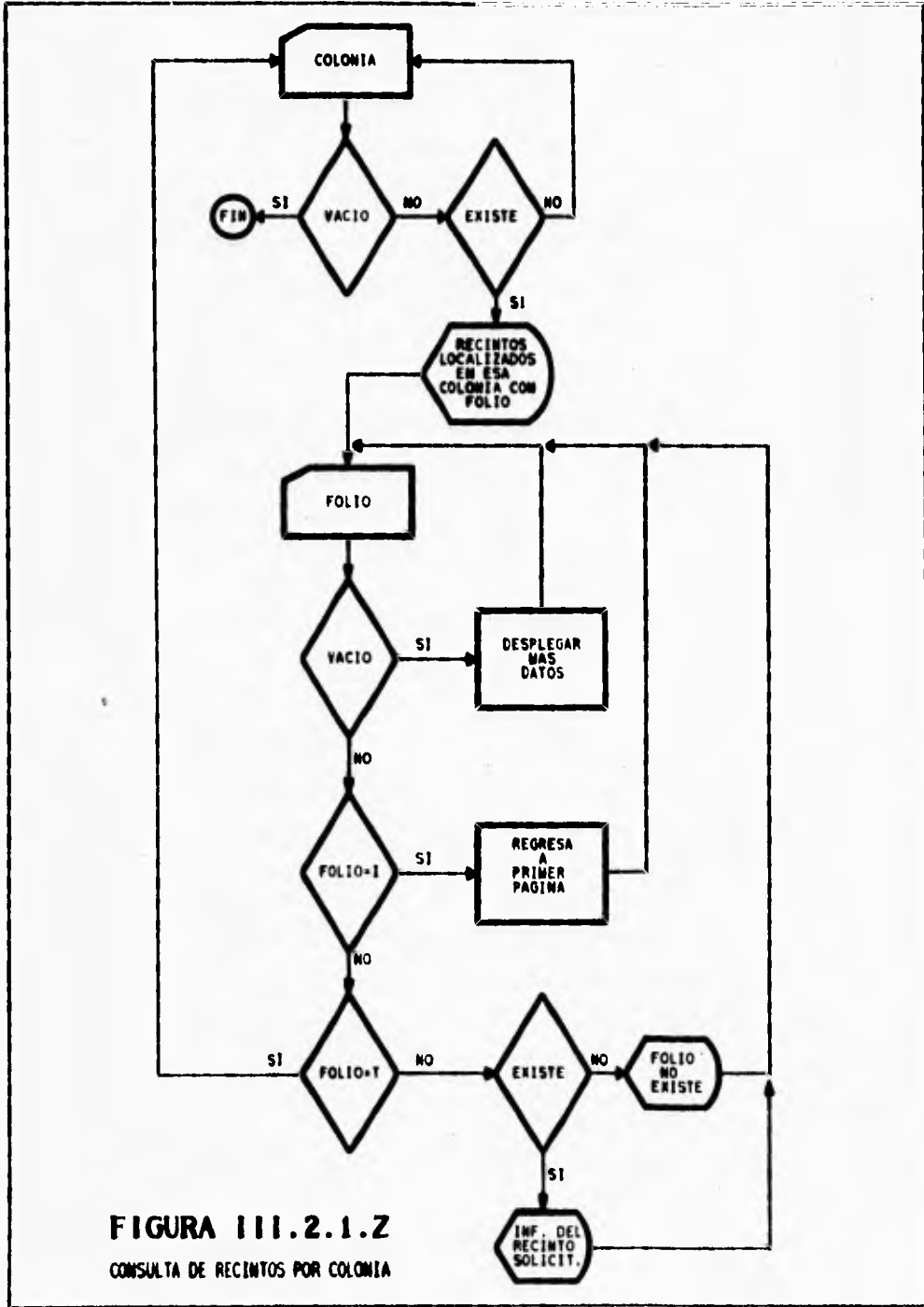


FIGURA III.2.1.Z
CONSULTA DE RECINTOS POR COLONIA

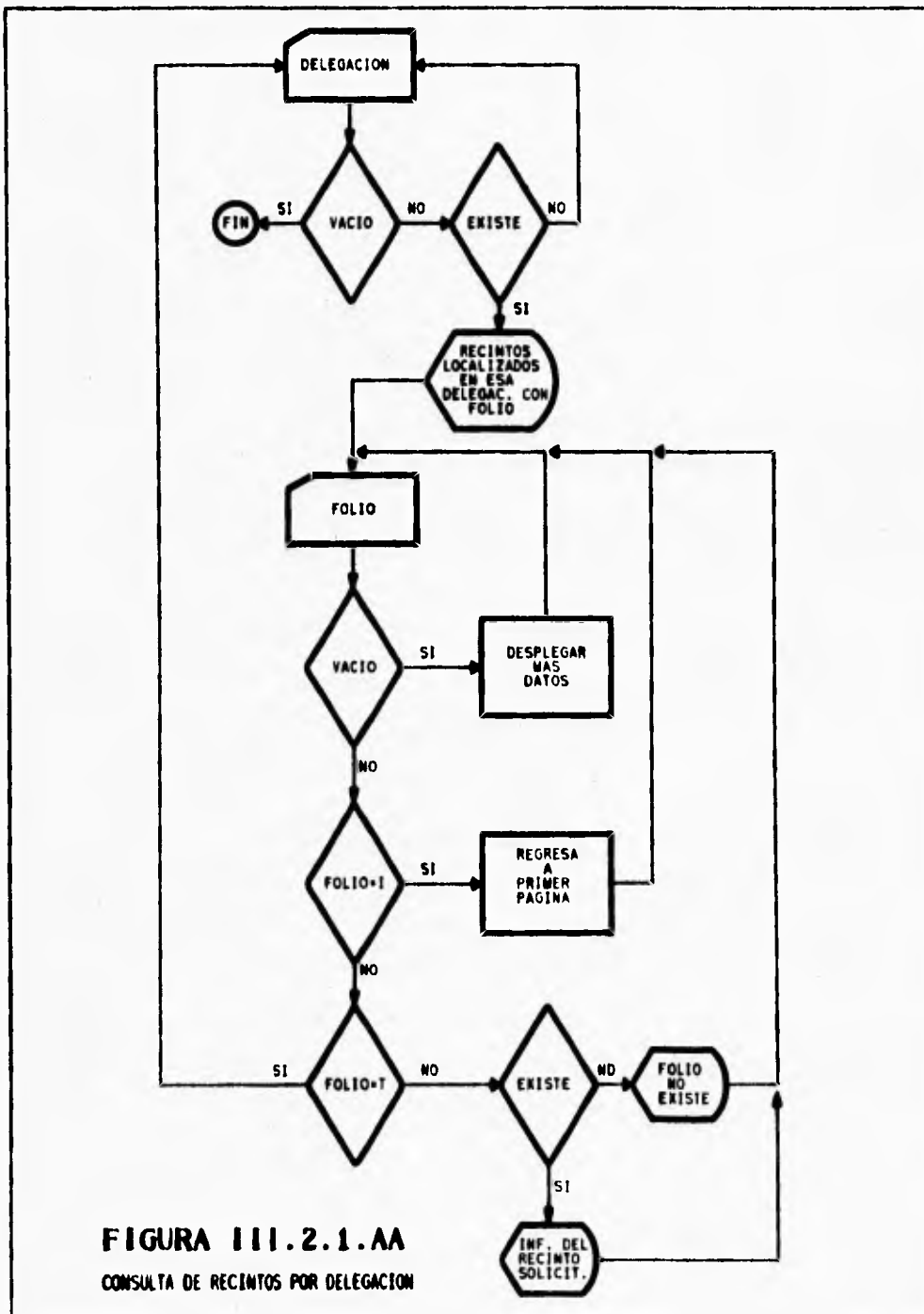


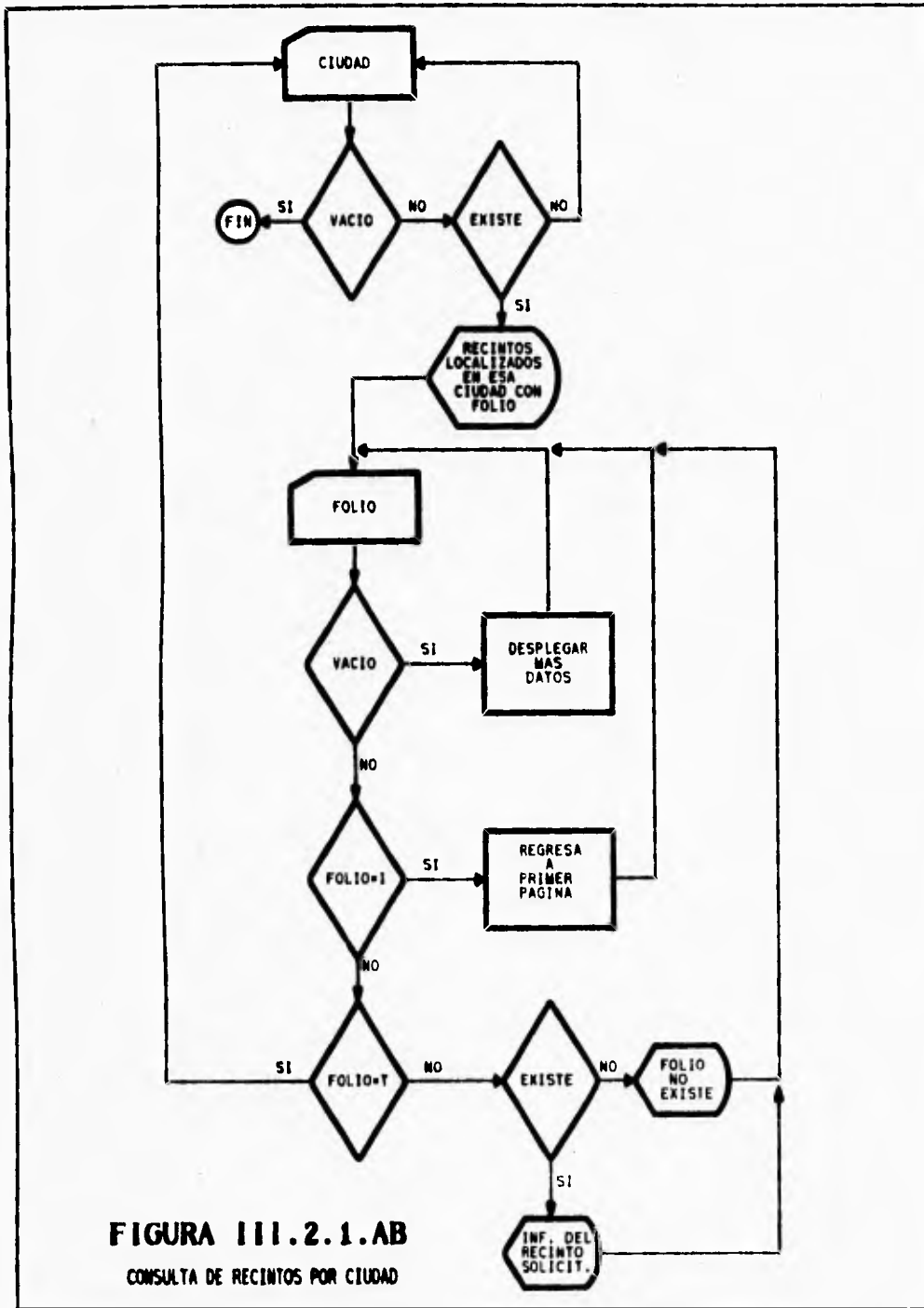
FIGURA III.2.1.AA
CONSULTA DE RECINTOS POR DELEGACION

CONSULTA POR CIUDAD

Al seleccionar este submódulo (figura III.2.1.AB), el programa pedirá un nombre de ciudad, el cual servirá como condición de búsqueda de todos los recintos que se encuentren ubicados en la ciudad indicada, se desplegarán en la pantalla los nombres de los recintos que cumplen con la condición y al lado izquierdo aparecerá el número de folio correspondiente, al llenarse la pantalla o llegar al fin de la tabla el programa pedirá a) el número de registro del recinto que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado al registro que se desea consultar se desplegará en la pantalla toda la información referente a éste.

CONSULTA POR CODIGO POSTAL.

Al seleccionar este submódulo (figura III.2.1.AC), el programa pedirá un código postal, el cual servirá como condición de búsqueda de todos los recintos que se encuentren ubicados en el código postal indicado, se desplegarán en la pantalla los nombres de los recintos que cumplen con la condición y al lado izquierdo aparecerá el número de folio correspondiente, al llenarse la pantalla o llegar al fin de la tabla el programa pedirá a) el número de folio del recinto que se desee consultar, b) oprimir la tecla [ENTER] para pasar a la siguiente página, c) oprimir la tecla [I] y la tecla [ENTER] para regresar a la primer página, d) oprimir la tecla [T] y la tecla [ENTER] para salir sin revisar las páginas subsecuentes. Cuando se proporcione el número de folio asociado al registro que se desea consultar se desplegará en la pantalla toda la información referente a éste.



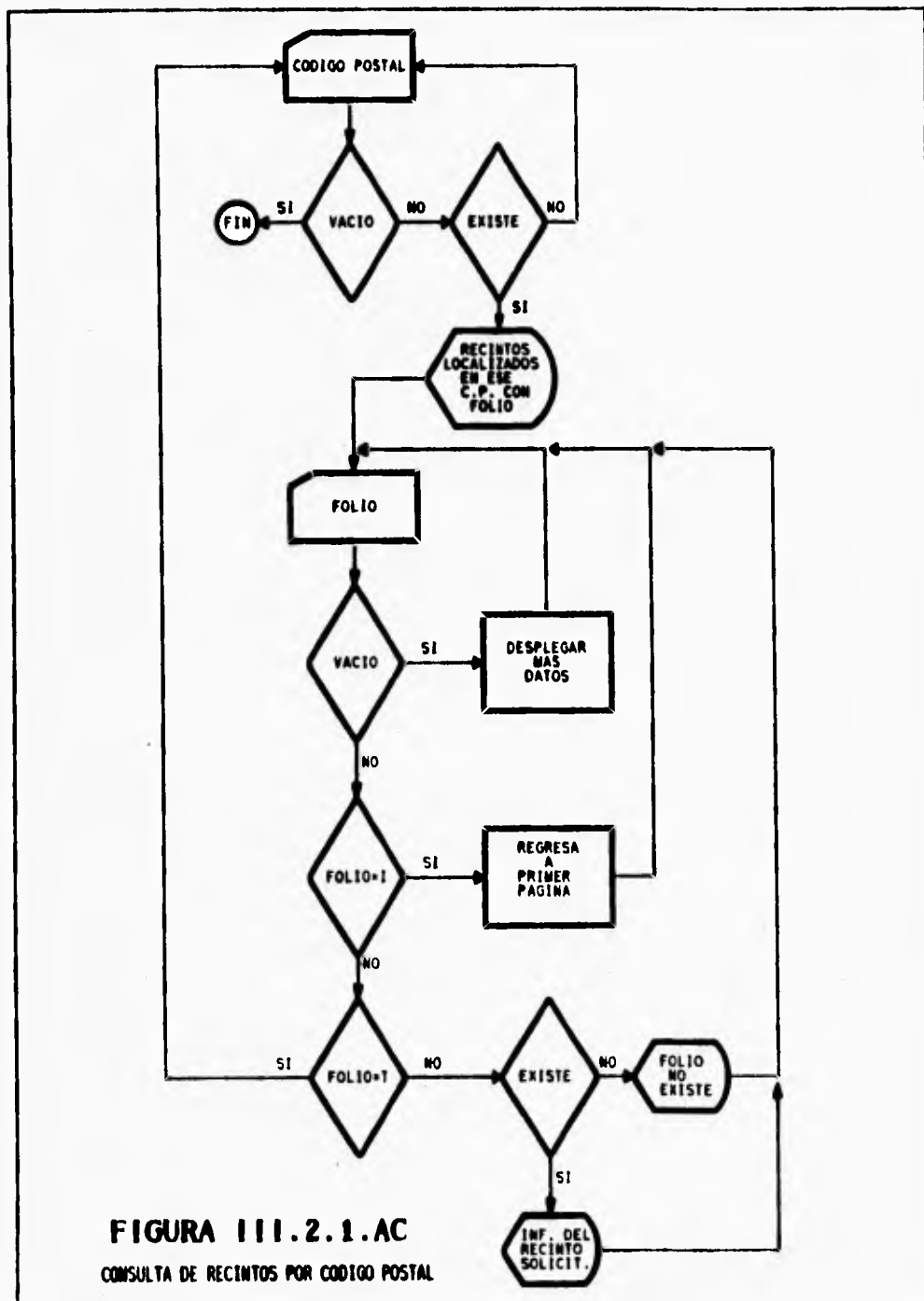


FIGURA III.2.1.AC
 CONSULTA DE RECINTOS POR CODIGO POSTAL

CAMBIOS A RECINTOS

Al seleccionar este submódulo (figura III.2.1.AD), el programa pedirá el nombre del recinto a modificar, validando la existencia de éste. Se desplegarán para modificación todos los datos del domicilio y teléfonos del recinto.

BAJAS DE RECINTOS

Al seleccionar este submódulo (figura III.2.1.AE), el programa pedirá el nombre del recinto que se desea dar de baja, validando la existencia de éste. Se desplegarán toda la información referente el recinto y el programa pedirá una confirmación para dar de baja el recinto. Si existieran actividades que se realizarán en el recinto que se desea dar de baja, esta no será permitida y se indicará al usuario cual es la razón.

REPORTE DE RECINTOS

Al seleccionar este submódulo (figura III.2.1.AF), el programa emitirá un reporte de todos los recintos que se encuentran dados de alta.

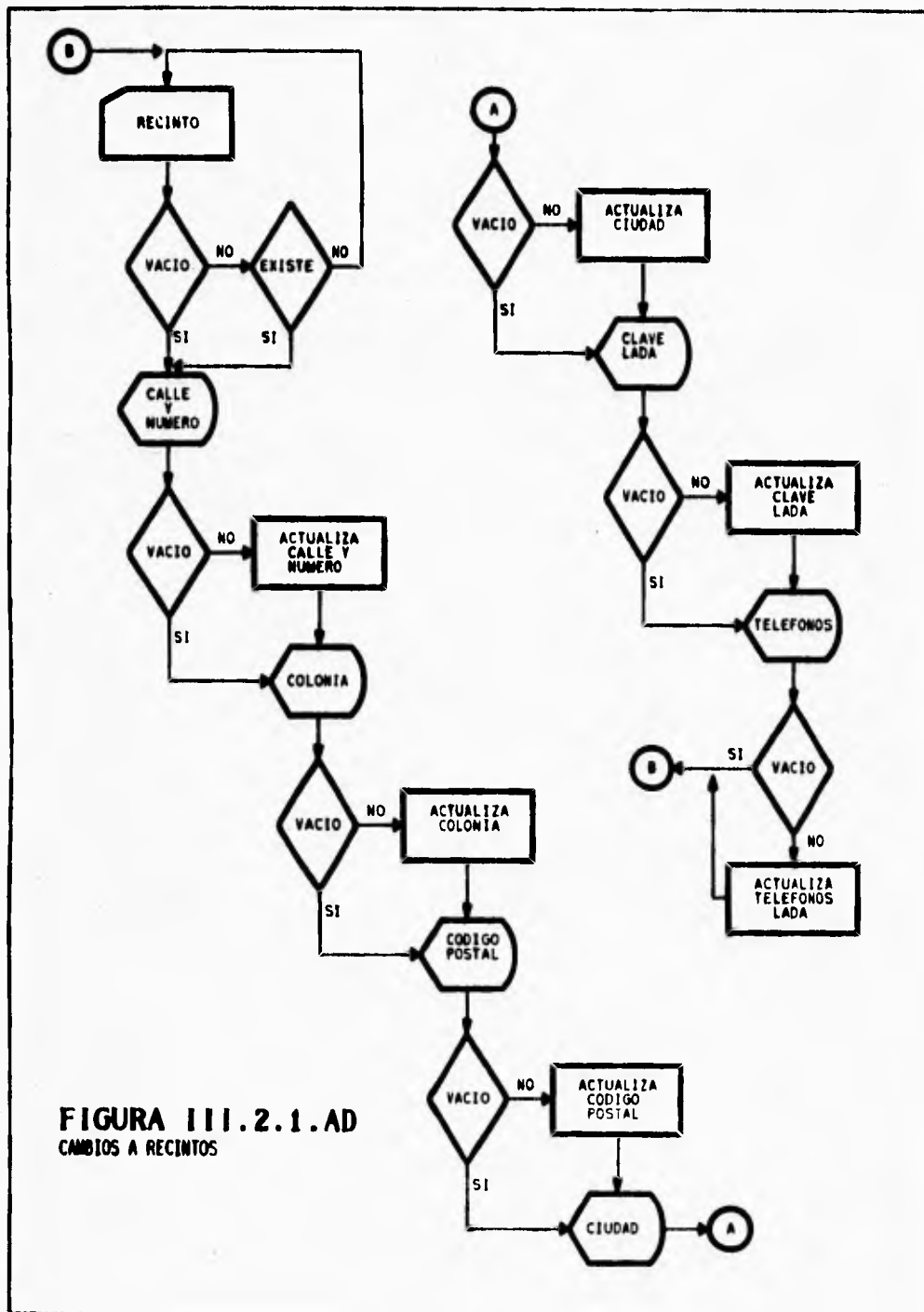


FIGURA III.2.1.AD
CAMBIOS A RECINTOS

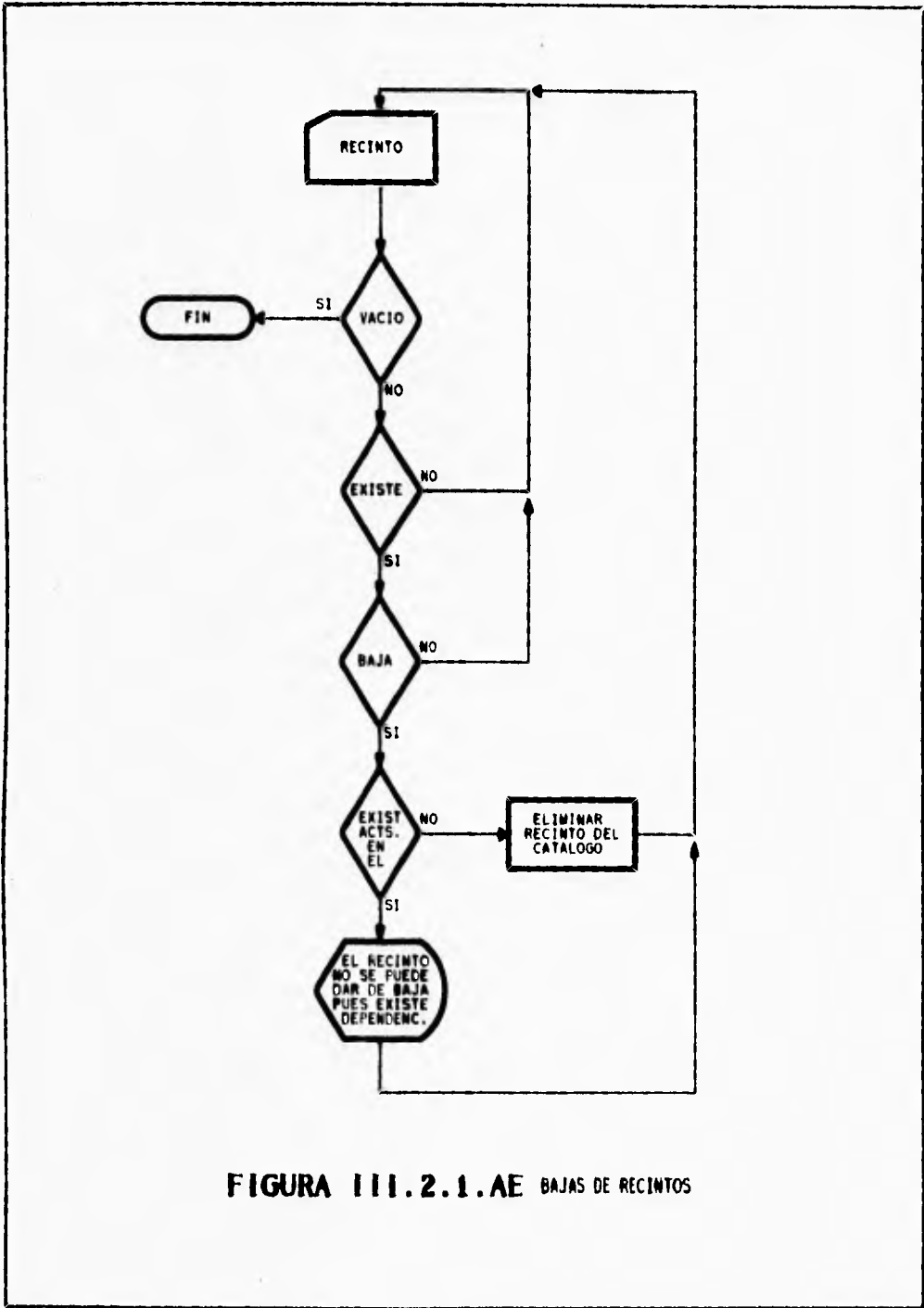


FIGURA III.2.1.AE BAJAS DE RECINTOS

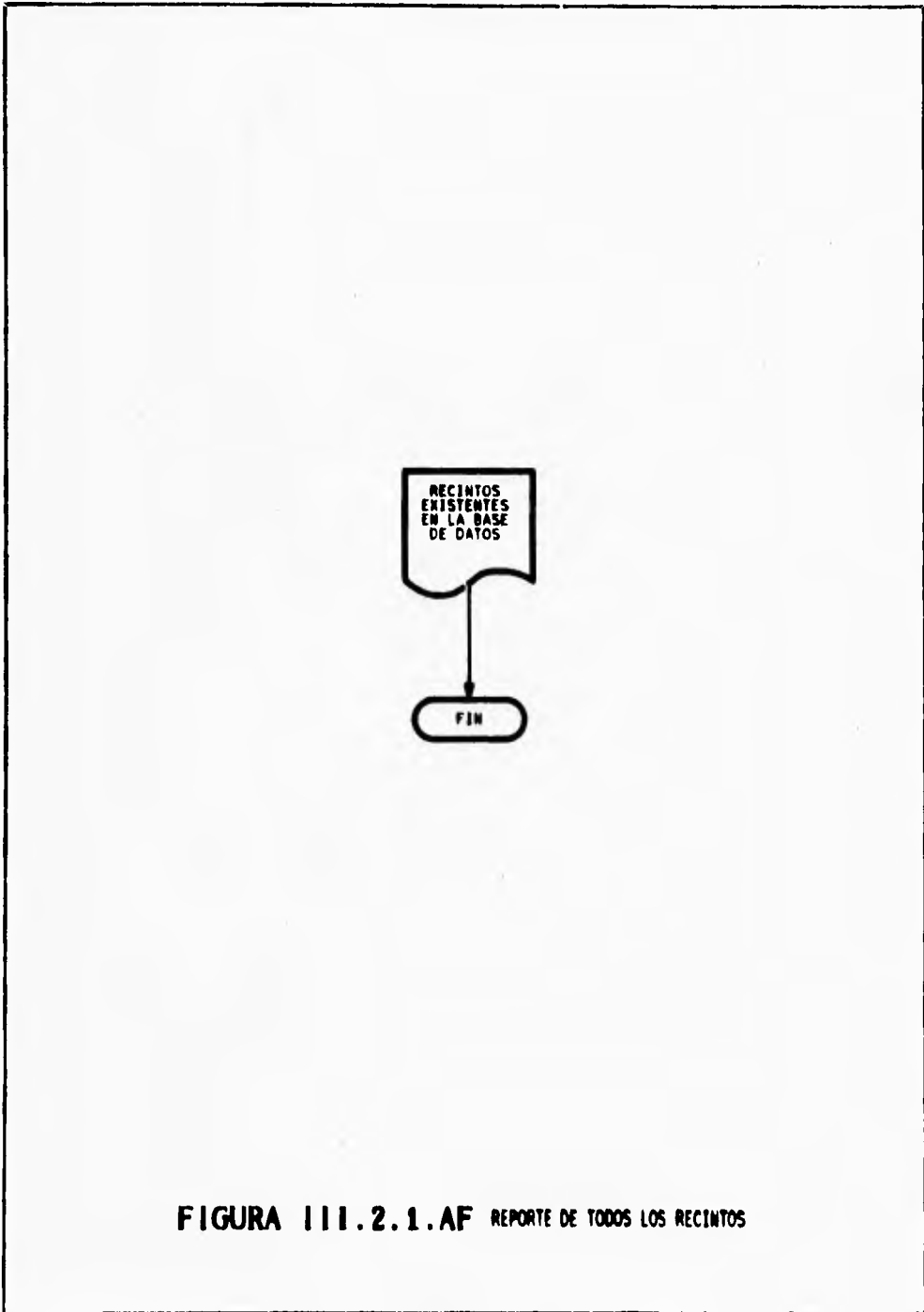


FIGURA III.2.1.AF REPORTE DE TODOS LOS RECINTOS

MODULO DE DISCIPLINAS

Este módulo está integrado por los siguientes submódulos (ver figura III.2.1.AG):

- **Altas**
- **Consulta**
- **Modificación**
- **Reporte**

ALTAS DE DISCIPLINAS

Al seleccionar este submódulo (figura III.2.AH), el programa solicitará un nombre de disciplina para integrarlo a la tabla de disciplinas, validará que no se dupliquen nombres de disciplinas.

CONSULTA DE DISCIPLINAS

Al seleccionar este submódulo (figura III.2.AI), el programa desplegará los nombres de todas las disciplinas que se encuentran en la tabla. El programa controlará la paginación de la pantalla en caso de que no se pueden desplegar todas las disciplinas en una sola pantalla.

MODIFICACION DE DISCIPLINAS

Al seleccionar este submódulo (figura III.2.AJ) el programa solicitará un nombre de disciplina para ser modificado.

REPORTE DE DISCIPLINAS

Al seleccionar este submódulo (figura III.2.1.AK), el programa emitirá un reportes de todas las disciplinas que se encuentran en la tabla.

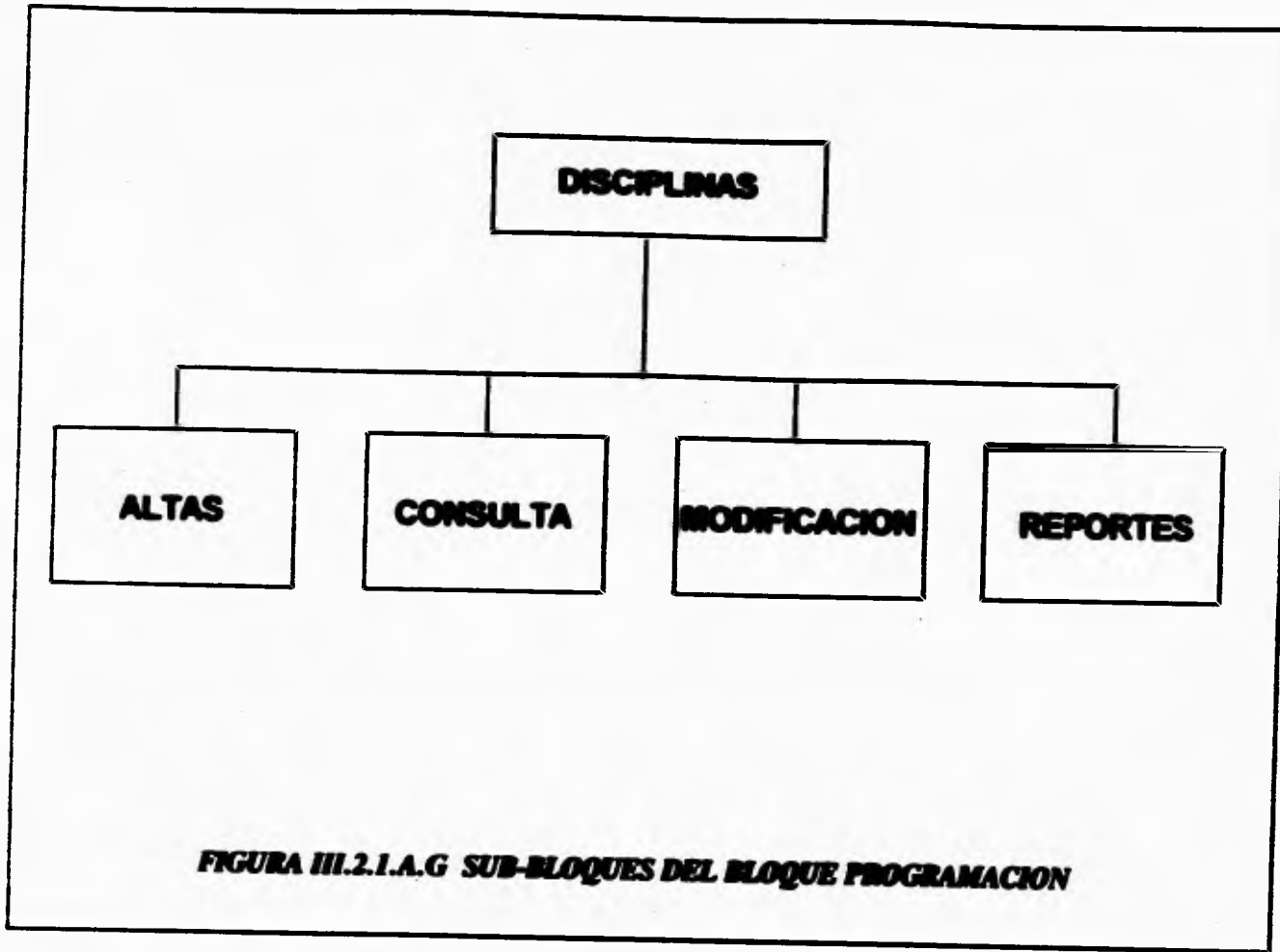


FIGURA III.2.1.A.G SUB-BLOQUES DEL BLOQUE PROGRAMACION

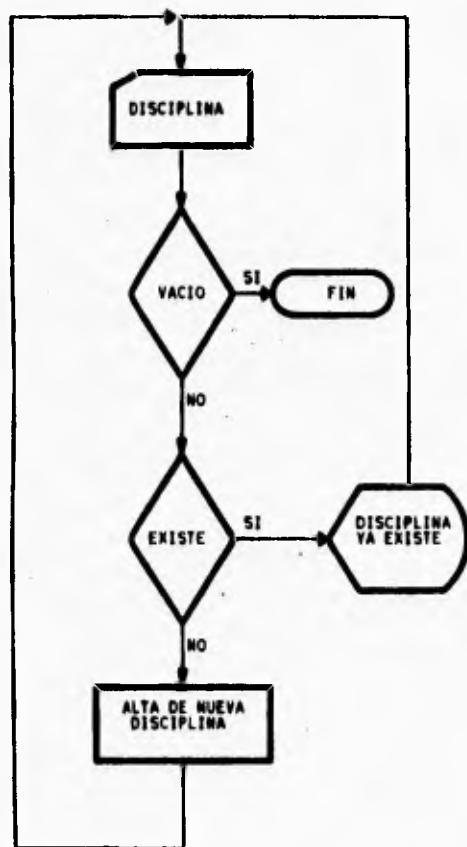


FIGURA III.2.1.AH ALTA DE DISCIPLINAS



FIGURA III.2.1.A1 REPORTE DE TODAS LAS DISCIPLINAS

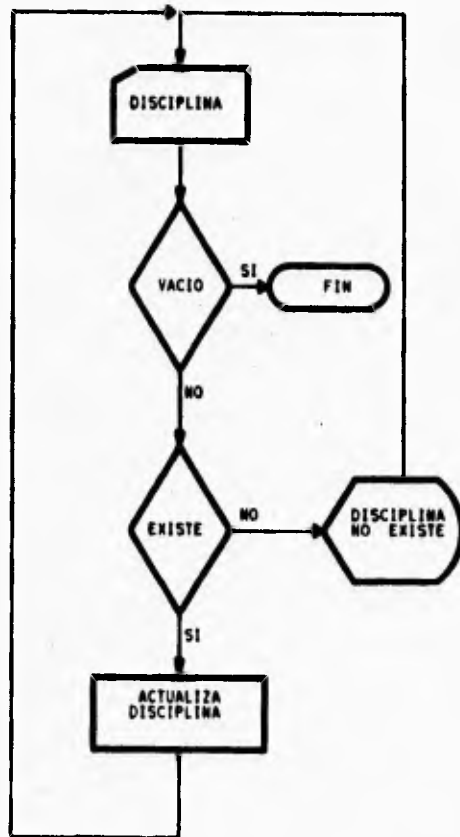


FIGURA III.2.1.AJ MODIFICACION A DISCIPLINAS



FIGURA III.2.1.AK REPORTE DE TODAS LAS DISCIPLINAS

MODULO DE REGENERACION DE INDICES

Este submódulo crea todos los índices que utiliza el sistema, en caso de que por alguna razón se llegara a dañar alguno de los archivos de índices, seleccionando esta opción se regenerarán automáticamente todos los índices.

III.2.2 DIAGRAMA DE FLUJO DE DATOS

Los Diagramas de Flujo de Datos o burbujas, son gráficas dirigidas en donde los nodos especifican las actividades de proceso y los arcos la transferencia de datos entre nodos de proceso. Como cualquier otro diagrama de flujo, éstos pueden ser utilizados en cualquier nivel de abstracción. Un diagrama de datos puede representar al flujo de datos entre estatutos individuales o entre bloques de estatutos dentro de una rutina; flujo de datos entre rutinas secuenciales, flujo de datos entre procesos concurrentes o flujo de datos entre sistemas de cómputo distribuidos, donde cada nodo representa una unidad de proceso geográficamente separada. Distinto a otros diagramas de flujo, los de datos no indican la lógica de decisión o las condiciones bajo las cuales varios nodos de proceso se activen.

A continuación se presentan los diagramas de flujo de datos de nuestros tres módulos principales :

- **MODULO DE RECINTOS (Figura III.2.2.A)**
- **MODULO DE DISCIPLINAS (Figura III.2.2.B)**
- **MODULO DE PROGRAMACION DE EVENTOS (Figura III.2.2.C)**

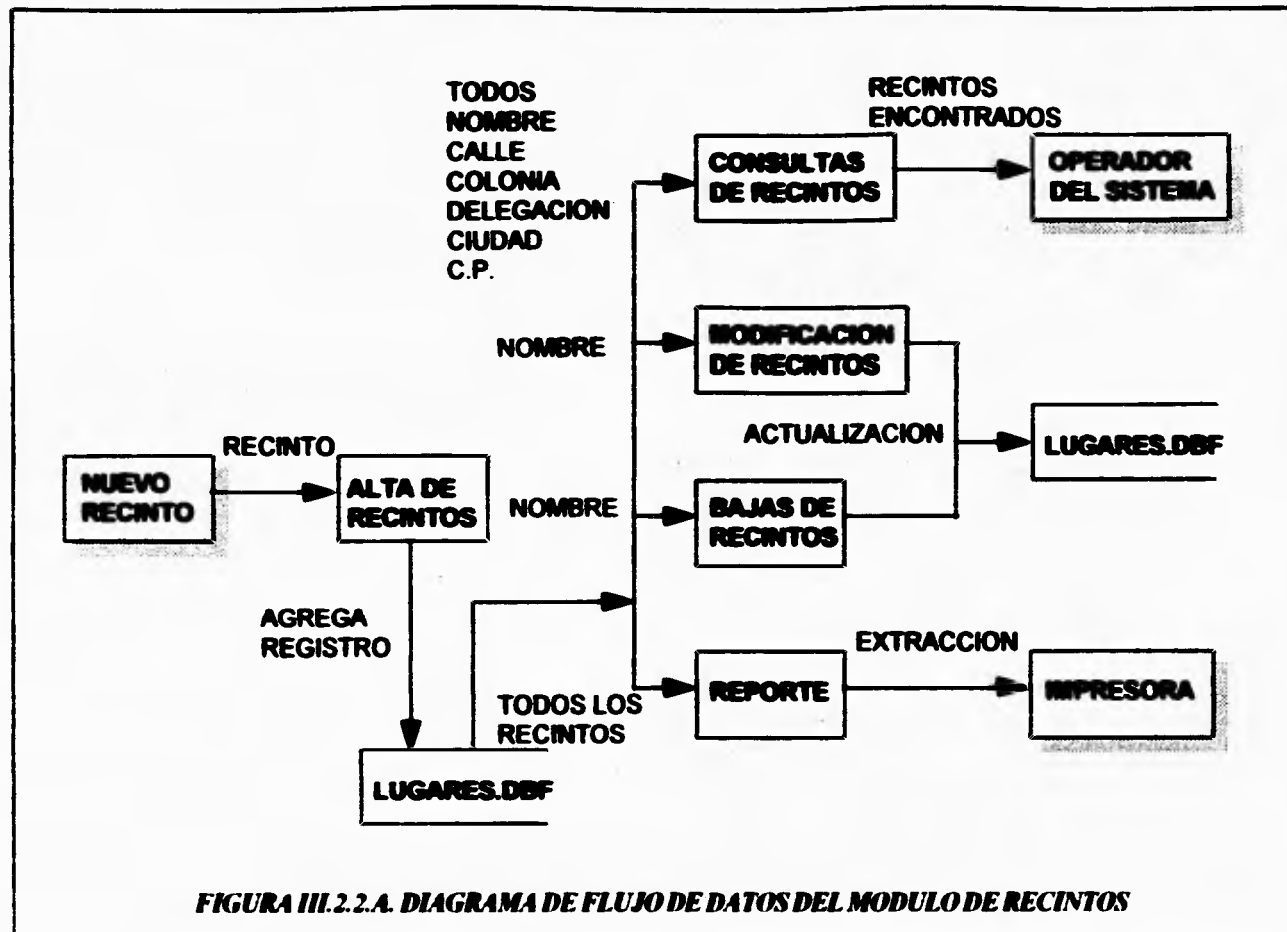
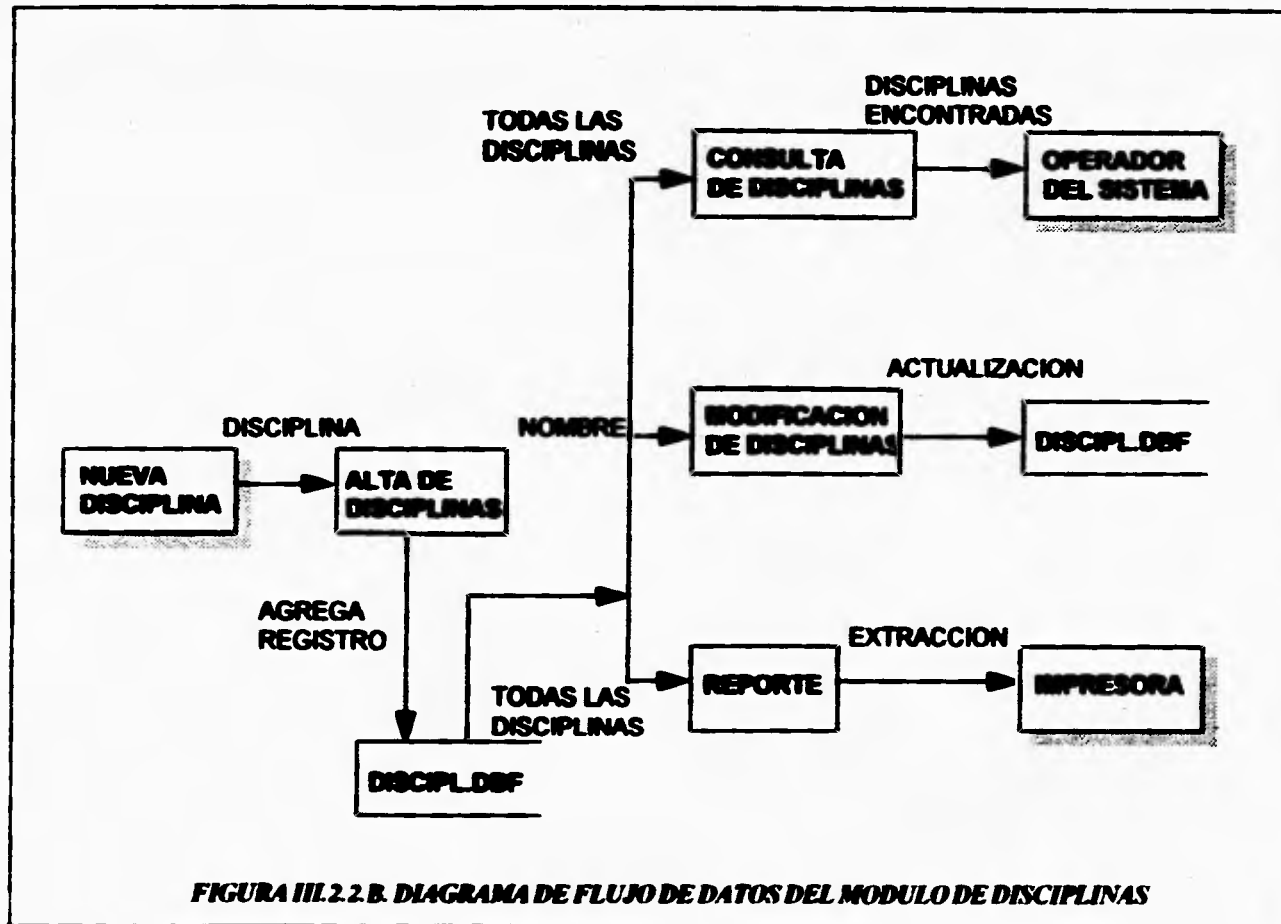
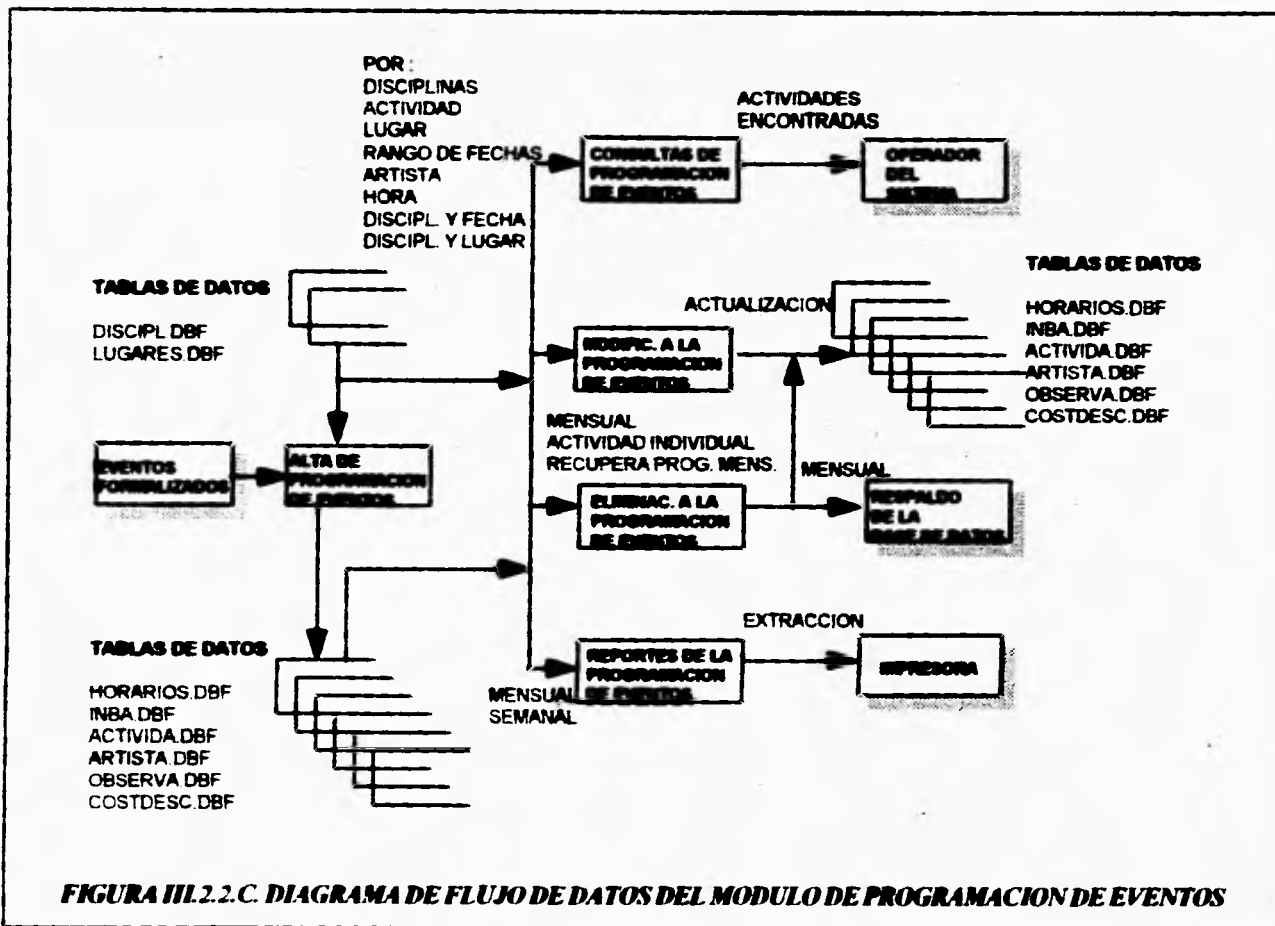


FIGURA III.2.2.A. DIAGRAMA DE FLUJO DE DATOS DEL MODULO DE RECINTOS



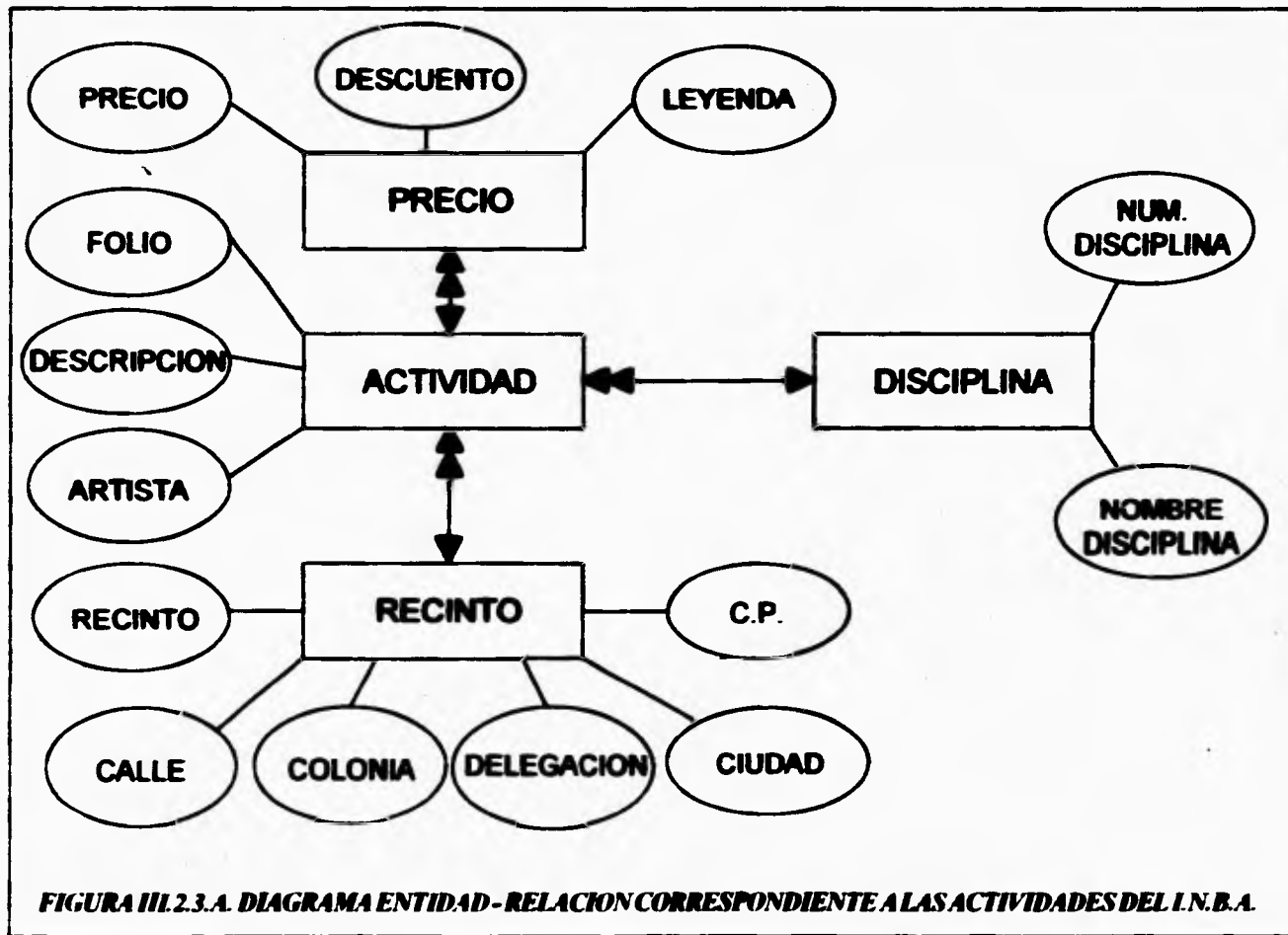


III.2.3 DIAGRAMA ENTIDAD RELACION

Toda la estructura lógica de una Base de Datos puede expresarse gráficamente mediante un Diagrama Entidad - Relación, el cual esta formado de los siguientes componentes :

- **Rectángulos : Representan Entidades.**
- **Elipses : Representan Atributos de las Entidades.**
- **Diamantes : Representan Relaciones entre Entidades.**
- **Lineas : Representan las ligas entre Atributos y Entidades, y ligas entre Entidades y sus Relaciones.**

A continuación se muestra el Diagrama de Entidad - Relación (Figura III.2.3.A.)



III.2.4 DICCIONARIO DE DATOS

El Diccionario de Datos se utiliza para definir el significado, uso, característica y otros datos relevantes de todos los data items, campos, entidades, referencias cruzadas y las relaciones que existen entre ellos. El DD es una Base de Datos que contiene datos acerca de los datos. Es importante señalar que el DD, se tiene que analizar para saber como está formada la Base de Datos en su estructura.

A continuación se detalla el DD del sistema en cuestión :

TABLAS UTILIZADAS

12 Tablas en el Sistema

ACTIVIDA.DBF

ARTISTA.DBF

COSTDESC.DBF

DISCIPL.DBF

FECHSOR.DBF

HORARIOS.DBF

HORFECHA.DBF

INBA.DBF

LISFECH.DBF

LUGARES.DBF

OBSERVA.DBF

SLISFECH.DBF

Estructura de la Tabla : ACTIVIDA.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	ACTIVIDAD	Character	240		10	249

**** Total **** 250

Esta tabla está asociada con el(los) archivo(s) de índices:

: FOLACT.NTX (FOLIO)

Usada por REPPGMEN (Procedimiento incluido en INBAPROC.PRG)

- : REPPGSEM (Procedimiento incluido en INBAPROC.PRG)
- : ELIMENS (Procedimiento incluido en INBAPROC.PRG)
- : RESTOMES (Procedimiento incluido en INBAPROC.PRG)
- : ELIMACT (Procedimiento incluido en INBAPROC.PRG)
- : CXDYF (Procedimiento incluido en INBAPROC.PRG)
- : CXDYL (Procedimiento incluido en INBAPROC.PRG)
- : CXDISC (Procedimiento incluido en INBAPROC.PRG)
- : CXACT (Procedimiento incluido en INBAPROC.PRG)
- : CXARTIS (Procedimiento incluido en INBAPROC.PRG)
- : CXLUGAR (Procedimiento incluido en INBAPROC.PRG)
- : CXFECHA (Procedimiento incluido en INBAPROC.PRG)
- : CXHORA (Procedimiento incluido en INBAPROC.PRG)
- : CAMPROG (Procedimiento incluido en INBAPROC.PRG)
- : CAMBIAR (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : ARTISTA.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	ARTISTA	Character	240		10	249

**** Total ****

250

Esta tabla está asociada con el(los) archivo(s) de índices:

: FOLARTIS.NTX (FOLIO)

Usada por INBA.PRG

: REPPGMEN (Procedimiento incluido en INBAPROC.PRG)
: REPPGSEM (Procedimiento incluido en INBAPROC.PRG)
: ELIMENS (Procedimiento incluido en INBAPROC.PRG)
: RESTOMES (Procedimiento incluido en INBAPROC.PRG)
: BORRACT (Procedimiento incluido en INBAPROC.PRG)
: CXARTIS (Procedimiento incluido en INBAPROC.PRG)
: TRAEACT (Procedimiento incluido en INBAPROC.PRG)
: CAPPROG (Procedimiento incluido en INBAPROC.PRG)
: CAMBIAR (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : COSTDESC.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	COSTO	Numeric	6	2	10	15
3	DESCCOSTO	Numeric	3		16	18
4	LEYENDA	Character	30		19	48

**** Total ****

49

Esta tabla está asociada con el(los) archivo(s) de índices:

: FOLCOST.NTX (FOLIO)

Usada por INBA.PRG

: REPPGMEN (Procedimiento incluido en INBAPROC.PRG)

: REPPGSEM (Procedimiento incluido en INBAPROC.PRG)

: ELIMENS (Procedimiento incluido en INBAPROC.PRG)

: RESTOMES (Procedimiento incluido en INBAPROC.PRG)

: BORRACT (Procedimiento incluido en INBAPROC.PRG)

: TRAEACT (Procedimiento incluido en INBAPROC.PRG)

: CAPPROG (Procedimiento incluido en INBAPROC.PRG)

: CAMBIAR (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : DISCIPL.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	NUMDISCIPL	Numeric	2		1	2
2	DISCIPLINA	Character	30		3	32

**** Total **** 33

Esta tabla está asociada con el(los) archivo(s) de índices:

- : NOMBDIS.NTX (DISCIPLINA)
- : NUMDISCI.NTX (NUMDISCIPL)

Esta tabla está asociada con el(los) reporte(s):

- : DISCIPL.FRM

Usada por INBA.PRG

- : REPPGMEN (Procedimiento incluido en INBAPROC.PRG)
- : REPPGSEM (Procedimiento incluido en INBAPROC.PRG)
- : ELIMACT (Procedimiento incluido en INBAPROC.PRG)
- : CXDYF (Procedimiento incluido en INBAPROC.PRG)
- : CXDYL (Procedimiento incluido en INBAPROC.PRG)
- : CXDISC (Procedimiento incluido en INBAPROC.PRG)
- : CXACT (Procedimiento incluido en INBAPROC.PRG)
- : CXARTIS (Procedimiento incluido en INBAPROC.PRG)
- : CXLUGAR (Procedimiento incluido en INBAPROC.PRG)
- : CXFECHA (Procedimiento incluido en INBAPROC.PRG)
- : CXHORA (Procedimiento incluido en INBAPROC.PRG)
- : CAPPROG (Procedimiento incluido en INBAPROC.PRG)

- : CAMPROG (Procedimiento incluido en INBAPROC.PRG)
- : CAPDISCI (Procedimiento incluido en INBAPROC.PRG)
- : CONDISCI (Procedimiento incluido en INBAPROC.PRG)
- : CAMDISCI (Procedimiento incluido en INBAPROC.PRG)
- : REPDISCI (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : FECHSOR.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	FECHA	Date	8		10	17
3	FECHAFIN	Date	8		18	25
4	HORA	Character	5		26	30
5	HORAFIN	Character	5		31	35
6	ESPECIAL	Character	25		36	60
7	DISCIPLINA	Numeric	2		61	62

**** Total **** **63**

Usada por REPPGMEN (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : HORARIOS.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	FECHA	Date	8		10	17
3	FECHAFIN	Date	8		18	25
4	HORA	Character	5		26	30
5	HORAFIN	Character	5		31	35
6	ESPECIAL	Character	25		36	60
7	DISCIPLINA	Numeric	2		61	62

**** Total ****

63

Esta tabla está asociada con el(los) archivo(s) de índices:

: FOLHORA.NTX (FOLIO)

: FECHA.NTX (FECHA)

: HORA.NTX (HORA)

Usada por INBA.PRG

: REPPGMEN (Procedimiento incluido en INBAPROC.PRG)

: REPPGSEM (Procedimiento incluido en INBAPROC.PRG)

: ELIMENS (Procedimiento incluido en INBAPROC.PRG)

: RESTOMES (Procedimiento incluido en INBAPROC.PRG)

: BORRACT (Procedimiento incluido en INBAPROC.PRG)

: CXDYF (Procedimiento incluido en INBAPROC.PRG)

: CXFECHA (Procedimiento incluido en INBAPROC.PRG)

: CXHORA (Procedimiento incluido en INBAPROC.PRG)

: TRAEACT (Procedimiento incluido en INBAPROC.PRG)

: CAPPROG (Procedimiento incluido en INBAPROC.PRG)

: CAMBIAR (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : HORFECHA.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	FECHA	Date	8		10	17
3	FECHAFIN	Date	8		18	25
4	HORA	Character	5		26	30
5	HORAFIN	Character	5		31	35
6	ESPECIAL	Character	25		36	60
7	DISCIPLINA	Numeric	2		61	62

**** Total **** 63

Usada por REPPGSEM (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : INBA.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	DISCIPLINA	Numeric	2		10	11
3	LUGAR	Character	38		12	49
4	HORARIO	Character	50		50	99
5	HORATAQUI	Character	50		100	149

**** Total **** 150

Esta tabla está asociada con el(los) archivo(s) de índices:

: FOLINBA.NTX (FOLIO)

: DISCIPL.NTX (DISCIPLINA)

Usada por INBA.PRG

- : REPPGMEN** (Procedimiento incluido en INBAPROC.PRG)
- : REPPGSEM** (Procedimiento incluido en INBAPROC.PRG)
- : ELIMENS** (Procedimiento incluido en INBAPROC.PRG)
- : RESTOMES** (Procedimiento incluido en INBAPROC.PRG)
- : ELIMACT** (Procedimiento incluido en INBAPROC.PRG)
- : CXDYF** (Procedimiento incluido en INBAPROC.PRG)
- : CXDYL** (Procedimiento incluido en INBAPROC.PRG)
- : CXDISC** (Procedimiento incluido en INBAPROC.PRG)
- : CXACT** (Procedimiento incluido en INBAPROC.PRG)
- : CXARTIS** (Procedimiento incluido en INBAPROC.PRG)
- : CXLUGAR** (Procedimiento incluido en INBAPROC.PRG)
- : CXFECHA** (Procedimiento incluido en INBAPROC.PRG)
- : CXHORA** (Procedimiento incluido en INBAPROC.PRG)
- : CAPPROG** (Procedimiento incluido en INBAPROC.PRG)
- : CAMPROG** (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : LISFECH.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	FECHA	Date	8		10	17
3	HORA	Character	5		18	22

**** Total ****

23

Usada por REPPGMEN (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : LUGARES.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	RECINTO	Character	40		1	40
2	DIRECCION	Character	40		41	80
3	COLONIA	Character	40		81	120
4	DELEGACIO	Character	40		121	160
5	CIUDAD	Character	40		161	200
6	CP	Character	5		201	205
7	LADA	Character	3		206	208
8	TELI	Character	9		209	217
9	TEL2	Character	9		218	226

**** Total ****

227

Esta tabla está asociada con el(los) archivo(s) de índices:

: LUGRECIN.NTX (RECINTO)

Esta tabla está asociada con el(los) reporte(s):

: LUGARES.FRM

Usada por INBA.PRG

: REPPGMEN (Procedimiento incluido en INBAPROC.PRG)
: REPPGSEM (Procedimiento incluido en INBAPROC.PRG)
: CAPPROG (Procedimiento incluido en INBAPROC.PRG)
: CAMBIAR (Procedimiento incluido en INBAPROC.PRG)
: CAMRECIN (Procedimiento incluido en INBA.PRG)
: REPRECIN (Procedimiento incluido en INBA.PRG)
: CONRCP (Procedimiento incluido en INBA.PRG)
: CONRDEL (Procedimiento incluido en INBA.PRG)

: CONRCIU (Procedimiento incluido en INBA.PRG)
 : BAJRECIN (Procedimiento incluido en INBA.PRG)
 : CONRCAL (Procedimiento incluido en INBA.PRG)
 : CONRCOL (Procedimiento incluido en INBA.PRG)
 : CAPRECIN (Procedimiento incluido en INBA.PRG)
 : CONRALL (Procedimiento incluido en INBA.PRG)
 : CONRNOM (Procedimiento incluido en INBA.PRG)

Estructura de la Tabla : OBSERVA.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	OBSERVA	Character	240		10	249

**** Total **** 250

Esta tabla está asociada con el(los) archivo(s) de índices:

: FOLOBS.NTX (FOLIO)

Usada por INBA.PRG

: REPPGMEN (Procedimiento incluido en INBAPROC.PRG)
 : REPPGSEM (Procedimiento incluido en INBAPROC.PRG)
 : ELIMENS (Procedimiento incluido en INBAPROC.PRG)
 : RESTOMES (Procedimiento incluido en INBAPROC.PRG)
 : BORRACT (Procedimiento incluido en INBAPROC.PRG)
 : TRAEACT (Procedimiento incluido en INBAPROC.PRG)
 : CAPPROG (Procedimiento incluido en INBAPROC.PRG)
 : CAMBIAR (Procedimiento incluido en INBAPROC.PRG)

Estructura de la Tabla : SLISFECH.DBF

Campo	Nombre de campo	Tipo	Longitud	Dec	Inicia	Termina
1	FOLIO	Numeric	9		1	9
2	FECHA	Date	8		10	17
3	HORA	Character	5		18	22

**** Total **** 23

Usada por REPPGMEN (Procedimiento incluido en INBAPROC.PRG)

III.2.5 NORMALIZACION DE TABLAS

De acuerdo a los datos recopilados, tenemos que la información contenida en la Base de Datos es la siguiente :

INFORMACION GENERAL(Folio, disciplina, lugar, actividad, artista, fecha_inicio, fecha_terminación, hora_inicio, hora_terminación, descripción, horario_taquilla, precio, descuento, leyanda_precio_descuento, observaciones).

Por otro lado tenemos, la información en donde se llevan a cabo las disciplinas :

RECINTO(Nombre_Recinto, dirección, colonia, delegación, ciudad, código postal, clave_lada, teléfono_1, teléfono_2).

Para garantizar que los nombres de disciplina sean únicos y no se capturen varias veces, se tendrá una tabla independiente de Disciplinas.

DISCIPLINA(Núm_disciplina, Nombre_disciplina)

Como podemos ver en la información general, para una misma actividad pueden existir varios horarios, varios precios, varios descuentos, estos atributos pueden estar repetidos, por lo tanto se tiene que normalizar la tabla a la primera forma normal y obtendríamos como resultado lo siguiente :

INFORMACION GENERAL(Folio, Núm_disciplina, Lugar, Horario_taquilla, Actividad, Artistas, Observaciones).

En el caso de la actividad, artista y observaciones, la información en cada una de ellas, puede ser tan extensa que no sea suficiente la capacidad de alojamiento en un campo (máximo 255 caracteres), por lo tanto también se tendría que separar en tablas independientes cada uno de estos campos. Como resultado la tabla de Información general se reduciría a lo siguiente:

INFORMACION GENERAL(Folio, Núm_disciplina, Lugar, Horario_taquilla).

el resto de la tablas quedarían de la siguiente forma :

ACTIVIDAD(Folio, Descripción_actividad)

ARTISTA(Folio, Nombre_artista)

OBSERVACIONES(Folio, Descripción_observaciones)

PRECIOS(Folio, Costo, Descuento, Leyenda_costo_descuento)

HORARIOS(Folio, Fecha_inicio, Fecha_terminación, Hora_inicio, Hora_terminación, Descripción)

Cabe mencionar que debido a las características de los reportes requeridos, el sistema utilizará algunas tablas de soporte, para poder realizar los ordenamientos requeridos.

III.3.1 PROGRAMACION

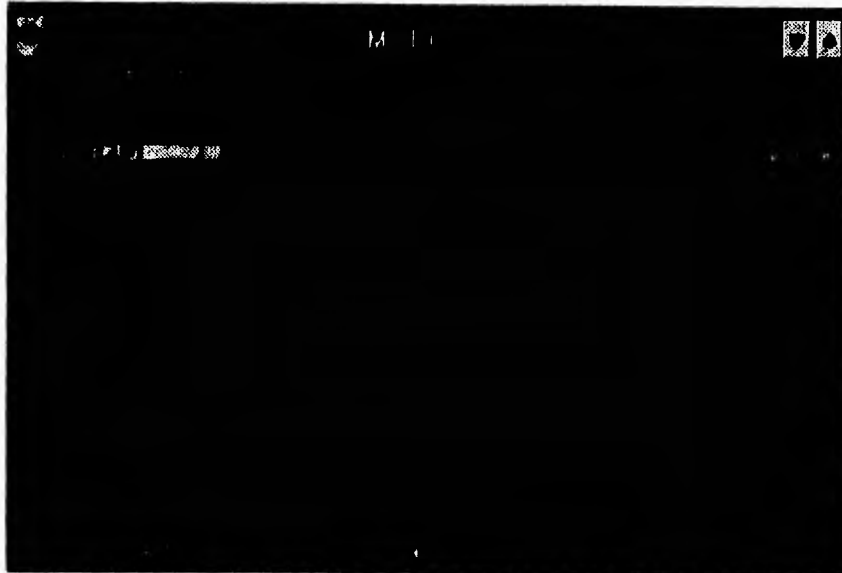
En este punto se muestra la programación elaborada del Sistema Nacional de Información del I.N.B.A. El Sistema fué realizado en Clipper, y está presentado de la siguiente forma en este documento :

- **MENU PRINCIPAL**
- **MENU PROGRAMACION**
- **MENU RECINTOS**
- **MENU DISCIPLINAS**
- **RUTINAS DE USO GENERAL**
- **RUTINAS GENERALES DEL MODULO DE PROGRAMACION**
- **RUTINAS GENERALES DEL MODULO DE RECINTOS**
- **RUTINAS DE IMPRESION**

En el caso de los menús de Programación, Recintos y Disciplinas, hay un desglose por cada uno, considerando Altas, Consultas, Bajas, etc.

También se estan anexando las pantallas de entrada a los modulos principales.

MENU PRINCIPAL



```
CLEAR
CLEAR ALL
SET TALK OFF
SET WRAP ON
SET SAFE OFF
SET STAT OFF
SET DATE BRIT
SET SCOR OFF
SET BELL OFF
SET MESSAGE TO 23
SET PROC TO inbaproc
```

```
DECLARE meses[12]
DECLARE mssg[100]
mssg[1] = CHR(25)+' '+CHR(24)+' '+CHR(27)+' '+CHR(26)+' '+Selecionar
'+CHR(17)+CHR(196)+CHR(217)+' '+Ejecutar '+[ESC] Salir'
mssg[2] = 'Para salir oprima la tecla [ESC]'
mssg[3] = 'Para terminar deje la fecha y la hora en blanco'
mssg[4] = 'No se puede dejar la fecha o la hora en blanco'
mssg[5] = 'Ingrese nombre de disciplina a la que pertenece la actividad & [ESC] para salir'
mssg[6] = 'Ingrese el nombre o descripcien de la actividad'
mssg[7] = 'Ingrese el nombre de los artistas, directores, expositores, etc.'
mssg[8] = 'Indique el nombre del recinto en el que se llevar a cabo la actividad'
mssg[9] = 'Ingrese las observaciones referentres a la actividad'
```



```

pregunta = 'Puede ver colores'
DO WHILE (resp # 'S') .AND. (resp # 'N')
  @ 23,77-13-LEN(pregunta) SAY ""+pregunta+' [S/N] ?'
  @ 23,74 GET resp PICT '!'
  READ
ENDDO
@ 23,77-13-LEN(pregunta) SAY SPACE(LEN(pregunta)+13)
IF resp = 'S'
  selcol = .T.
ELSE
  selcol = .F.
ENDIF
DO cfgcolor WITH selcol
DO encap WITH col4
DO WHILE .T.
  today = DATE()
  level1 = 1      && menivar for 1st menu
  SET COLOR TO &col4
  @ 3,20 SAY 'Á'
  @ 3,40 SAY 'Á'
  @ 3,60 SAY 'Á'
  @ 5,0 TO 5,79
  @ 4,20 SAY ' '
  @ 4,40 SAY ' '
  @ 4,60 SAY ' '
  @ 5,20 SAY 'Á'
  @ 5,40 SAY 'Á'
  @ 5,60 SAY 'Á'
  DO WHILE (level1 != 0)
    SET COLOR TO &col8
    @ 4, 0 PROMPT "PROGRAMACION      " MESSAGE '
PROGRAMACION
    @ 4,21 PROMPT "RECINTOS          " MESSAGE '
    @ 4,41 PROMPT "DISCIPLINAS      " MESSAGE '
    @ 4,61 PROMPT "INDICES (REGENERAR) MESSAGE '
INDICES
    DO indica WITH 1
    SET COLOR TO &col1
    @ 6,0 CLEAR TO 23,79
    SET COLOR TO &col8
    MENU TO level1
    IF (level1 = 0)      && esc to exit
      SET COLOR TO
      CLEAR
      CLOSE ALL
      CLEAR ALL
      QUIT
    ENDIF
    DO CASE
    CASE level1 = 1
      SAVE SCREEN TO pantalla[1]
      DO programa

```

MENU DE

MENU DE RECINTOS

MENU DE DISCIPLINAS

MANTENIMIENTO DE

```

RESTORE SCREEN FROM pantalla[1]
CASE level = 2
SAVE SCREEN TO pantalla[1]
DO recinto
RESTORE SCREEN FROM pantalla[1]
CASE level = 3
SAVE SCREEN TO pantalla[1]
DO discipl
RESTORE SCREEN FROM pantalla[1]
CASE level = 4
SAVE SCREEN TO pantalla[1]
SET COLOR TO &col4
@ 12,16 CLEAR TO 16,63
@ 12,16 TO 17,63
@ 14,18 SAY 'REGENERANDO INDICES, ESPERE POR FAVOR...'
fig = 1
ERASE folinba.ntx
DO girar WITH fig,14
ERASE discipl.ntx
DO girar WITH fig,14
ERASE folact.ntx
DO girar WITH fig,14
ERASE nombdis.ntx
DO girar WITH fig,14
ERASE numdisci.ntx
DO girar WITH fig,14
DO girar WITH fig,14
ERASE folhora.ntx
DO girar WITH fig,14
ERASE hora.ntx
DO girar WITH fig,14
ERASE fecha.ntx
DO girar WITH fig,14
ERASE folcost.ntx
DO girar WITH fig,14
ERASE folobs.ntx
DO girar WITH fig,14
ERASE activity.ntx
DO girar WITH fig,14
ERASE folartls.ntx
DO girar WITH fig,14
ERASE lugrecin.ntx
DO girar WITH fig,14
USE
USE: inba
archivo = 'INFORMACION GENERAL'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo
DO girar WITH fig,14
INDEX ON folio TO folinba
DO girar WITH fig,14
INDEX ON disciplina TO discipl
DO girar WITH fig,14
USE

```

USE discipl
archivo = 'DISCIPLINAS'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo
INDEX ON disciplina TO nombdis
DO girar WITH fig,14
INDEX ON numdiscipl TO numdisci
DO girar WITH fig,14
USE
USE horarios
archivo = 'HORARIOS'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo
INDEX ON folio TO folhora
DO girar WITH fig,14
INDEX ON fecha TO fecha
DO girar WITH fig,14
INDEX ON hora TO hora
DO girar WITH fig,14
USE
USE costdesc
archivo = 'PRECIOS'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo
INDEX ON folio TO folcost
DO girar WITH fig,14
USE
USE actividad
archivo = 'ACTIVIDADES'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo
INDEX ON folio TO folact
DO girar WITH fig,14
INDEX ON actividad TO activity
DO girar WITH fig,14
USE
USE observa
archivo = 'OBSERVACIONES'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo
INDEX ON folio TO folobs
DO girar WITH fig,14
USE
USE artista
archivo = 'ARTISTAS'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo
INDEX ON folio TO folartis
DO girar WITH fig,14
USE
USE lugares
archivo = 'RECINTOS'
@ 15,18 SAY SPACE(42)
@ 15,18 SAY 'ARCHIVO DE '+archivo


```

INDEX ON recinto TO lugrecin
DO girar WITH fig.14
RESTORE SCREEN FROM pantalla[1]
level = 4
ENDCASE
ENDDO
ENDDO

```

Procedimiento: CFGCOLOR

Llamado por: INBA.PRG

PROCEDURE cfgcolor

PARAMETERS selcol

IF selcol

```

col1 = 'w/bg, b/n+w,W' && blanco+/azul claro, azul / gris
col2 = 'gr/n+w, w+/rb,W' && amarillo/gris, blanco+ / violeta
col3 = 'w+/gr ' && blanco+ / cafe
col4 = 'w+/b, n/n+w,W' && blanco+/azul , negro/gris
col5 = 'b/n+w ' && azul+/gris
col6 = 'w+ /b,b/n+w,W' && blanco+/azul, azul / gris
col7 = 'w+/bg, gr+/b,W' && azul+/gris, amarillo/azul
col8 = 'w+/rb,N/n+W,W' && blanco+/violeta, BLANCO/NEGRO
col9 = 'w+ /b ' && blanco+ / azul,
col10 = 'w+b,n/bg,W'
col11 = 'W/BG, GR+/B,W'
col12 = 'BR/RB, BG/BG,W' && blanco+/azul claro, azul / gris
col13 = 'BG/B, G+/N+W,W' && azul claro/azul, verde/gris
col14 = 'w+/bg, w+/BG,W' && blanco+/azul claro, azul / gris

```

ELSE

```

col1 = 'W+/N,,W' && blanco+/azul claro, azul / gris
col2 = 'W+/N,,W' && amarillo/gris, blanco+ / violeta
col3 = 'W+/N,, ' && blanco+ / cafe
col4 = 'W+/N,,W' && blanco+/azul , negro/gris
col5 = 'W+/N,, ' && azul+/gris
col6 = 'W+/N,,W' && blanco+/azul, azul / gris
col7 = 'W+/N,,W' && azul+/gris, amarillo/azul
col8 = 'W+/N,,W' && blanco+/violeta, BLANCO/NEGRO
col9 = 'W+/N,, ' && blanco+ / azul,
col10 = 'W+/N,,W'
col11 = 'W+/N,,W'
col12 = 'W+/N,,W' && blanco+/azul claro, azul / gris
col13 = 'W+/N,,W' && azul claro/azul, verde/gris
col14 = 'W+/N,,W' && blanco+/azul claro, azul / gris

```

ENDIF

RETURN

Procedimiento:	ENCAP	
Llamado por:	INBA.PRG	
Llama a:	F_FECHA()	(función en INBA.PRG)

```

PROCEDURE encap
PARAMETERS COLOR
CLEAR
SET COLOR TO &color
@ 0,0 CLEAR TO 4,79
COL = 1
@ 0,COL SAY 'SISTEMA NACIONAL DE INFORMACION'
@ 1,COL SAY "Instituto Nacional de Bellas Artes"
@ 2,COL SAY 'Subdirección de Relaciones Públicas'
@ 0,58 SAY f_fecha(DATE(),4)
@ 3,0 TO 3,79
RETURN

```

Function:	F_FECHA	
Llamado por:	ENCAP	(procedimiento en INBA.PRG)
Llama a:	AFILL()	(función de Clipper)

```

FUNCTION f_fecha
PARAMETER fecha,FORMA      && fecha, formato
PRIVATE mes,text_fech
DECLARE mes{12}            && arreglo para los meses
afill(mes,'Enero',1)
afill(mes,'Febrero',2)
afill(mes,'Marzo',3)
afill(mes,'Abril',4)
afill(mes,'Mayo',5)
afill(mes,'Junio',6)
afill(mes,'Julio',7)
afill(mes,'Agosto',8)
afill(mes,'Septiembre',9)
afill(mes,'Octubre',10)
afill(mes,'Noviembre',11)
afill(mes,'Diciembre',12)
STORE SUBSTR((STR(DAY(fecha))),2,2) TO vdia && obtiene el día de la fecha
DO CASE
CASE FORMA=1              && formato dd/mm/yy
  STORE vdia TO text_fech
  STORE text_fech + '/' TO text_fech
  STORE text_fech + SUBSTR((STR(MONTH(fecha))),2,2) TO text_fech
  STORE text_fech + '/' TO text_fech
  STORE text_fech + SUBSTR((STR(YEAR(fecha))),2,4) TO text_fech
CASE FORMA=2              && formato dd/mm/yy

```

```

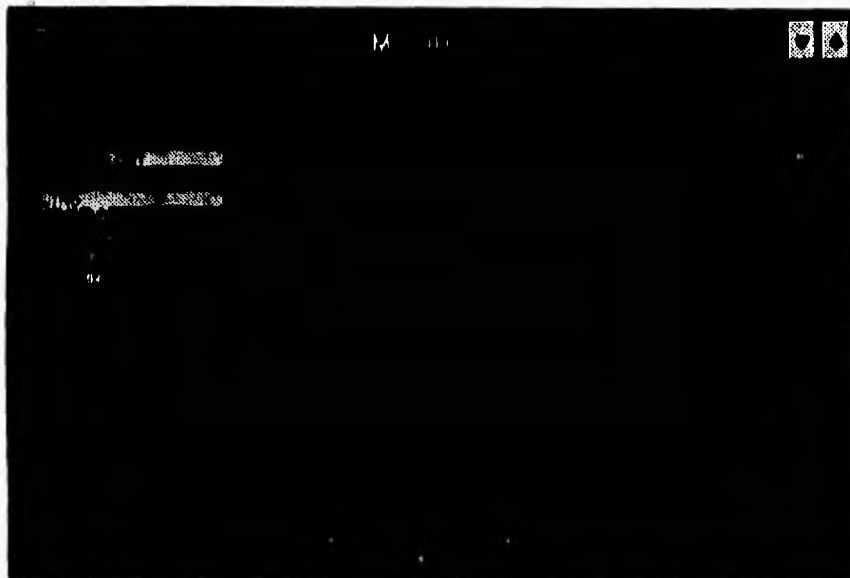
STORE vdia TO text_fech
STORE text_fech + '/' TO text_fech
STORE text_fech + SUBSTR((mes|month(fecha)),1,3) TO text_fech
STORE text_fech + '/' TO text_fech
STORE text_fech + SUBSTR((STR(YEAR(fecha))),4,2) TO text_fech

CASE FORMA=3          && formato dd de mmmmmmmun de yyyy
STORE vdia TO text_fech
STORE text_fech + ' de ' TO text_fech
STORE text_fech + mes|month(fecha)| TO text_fech
STORE text_fech + ' de ' TO text_fech
STORE text_fech + TRIM(STR(YEAR(fecha))) TO text_fech

CASE FORMA=4          && formato mmmmmmm dd de yyyy
STORE mes|month(fecha)| TO text_fech
STORE text_fech + ' ' TO text_fech
STORE text_fech + vdia TO text_fech
STORE text_fech + ' de ' TO text_fech
STORE text_fech + TRIM(STR(YEAR(fecha))) TO text_fech
ENDCASE
RETURN text_fech

```

MENU PROGRAMACION DE EVENTOS



Procedimiento:	PROGRAMA	
Llamado por:	INBA.PRG	
Llama a:	CAPPROG	(procedimiento en INBAPROC.PRG)
	: CONPROG	(procedimiento en INBA.PRG)
	: CAMPROG	(procedimiento en INBAPROC.PRG)
	: REPPROG	(procedimiento en INBA.PRG)
	: ELIMINA	(procedimiento en INBA.PRG)

```

PROCEDURE programa
level1 = 1      && memvar for 1st menu
DO WHILE (level1 != 0)
  SET COLOR TO &col4
  @ 5,20 SAY 'A'
  @ 6,20 TO 11,20
  @ 11,20 SAY 'U'
  @ 11,0 TO 11,19
  SET COLOR TO &col8
  @ 6,0 PROMPT " ALTAS      " MESSAGE '      CAPTURA DE ACTIVIDADES
  @ 7,0 PROMPT " CONSULTA  " MESSAGE '      MENU DE CONSULTA DE
ACTIVIDADES
  
```

@ 8, 0 PROMPT ' MODIFICACION	' MESSAGE '	EFFECTUAR CAMBIOS A LAS
ACTIVIDADES		
@ 9, 0 PROMPT ' REPORTES	' MESSAGE '	MENU DE IMPRESION DE
REPORTES		
@ 10, 0 PROMPT ' ELIMINACION	' MESSAGE '	MENU DE ELIMINACION DE
ACTIVIDADES		
MENU TO level1		
IF (level1 = 0)	&& esc to exit	
EXIT		
ENDIF		
DO CASE		
CASE level1 = 1		
SAVE SCREEN TO pantalla [2]		
DO camprog		
RESTORE SCREEN FROM pantalla[2]		
CASE level1 = 2		
SAVE SCREEN TO pantalla [2]		
DO conprog		
RESTORE SCREEN FROM pantalla[2]		
CASE level1 = 3		
SAVE SCREEN TO pantalla [2]		
DO camprog		
RESTORE SCREEN FROM pantalla[2]		
CASE level1 = 4		
SAVE SCREEN TO pantalla [2]		
DO repprog		
RESTORE SCREEN FROM pantalla[2]		
CASE level1 = 5		
SAVE SCREEN TO pantalla [2]		
DO elimina		
RESTORE SCREEN FROM pantalla[2]		
ENDCASE		
ENDDO		
RETURN		

Procedimiento: CAPPROG

Llamado por: PROGRAMA (procedimiento en INBA.PRG)

Llama a: INDICA (procedimiento en INBA.PRG)
: NOEXISTE (procedimiento en INBA.PRG)
: SINO (procedimiento en INBA.PRG)
: CDIS2 (procedimiento en INBAPROC.PRG)
: OPRIME (procedimiento en INBA.PRG)
: CREC2 (procedimiento en INBA.PRG)

Accesa a: DISCIPL.DBF
: HORARIOS.DBF
: INBA.DBF
: ACTIVIDAD.DBF
: ARTISTA.DBF
: LUGARES.DBF
: OBSERVA.DBF
: COSTDESC.DBF

Indices: NOMBDIS.IDX
: NUMDISC.IDX
: FOLHORA.IDX
: FECHA.IDX
: HORA.IDX
: FOLINBA.IDX
: DISCIPL.IDX
: FOLACT.IDX
: FOLARTIS.IDX
: LUGRECIN.IDX
: FOLOBS.IDX
: FOLCOST.IDX

```
PROCEDURE capprog
SET COLOR TO &col4
@ 1,0 CLEAR TO 23,79
@ 3,28 SAY "CAPTURA DE PROGRAMACION"
SELE 2
USE discipl INDEX nombdis,numdisci
REINDEX
SELE 3
USE horarios INDEX folhora,fecha,hora
REINDEX
SELE 1
USE inba INDEX folinba,discipl
REINDEX
fechav = CTOD(' / / ')
plinav = SPACE(30)
horav = SPACE(5)
horariov = SPACE(50)
DO WHILE .T.
GO BOTTOM
```

```

foliov = 0
plinav = SPACE(30)
IF .NOT. EOF()
  foliov = folio+1
ELSE
  foliov = 1
ENDIF
@ 5, 3 SAY 'Disciplina:'
DO indica WITH 5
SET COLOR TO &col4
@ 5,15 GET plinav
READ
IF LEN(RTRIM(plinav)) = 0
  EXIT
ENDIF
SELE 2
SEEK plinav
IF .NOT. FOUND()
  DO noexiste
  SET COLOR TO &col4
  @ 23, 2 SAY SPACE(77)
  resp = ''
  ask = 'DESEA DAR DE ALTA LA DISCIPLINA'
  DO sino WITH resp,ask
  SET COLOR TO &col4
  @ 23, 2 SAY SPACE(77)
  IF resp = 'N'
    SELE 1
    LOOP
  ELSE
    numbdis = 0
    SAVE SCREEN TO pantalla[4]
    @ 8, 5 CLEAR TO 15,70
    @ 7, 4 TO 13,71
    DO cdis2 WITH numbdis,plinav
    REINDEX
    SET COLOR TO &col4
    @ 7, 4 CLEAR TO 15,71
    RESTORE SCREEN FROM pantalla[4]
    @ 8,15 SAY plinav
  ENDIF
ENDIF
numbdis = numdiscipl
SELE 1
APPEND BLANK
REPLACE folio WITH foliov
REPLACE disciplina WITH numbdis
SELE 3
IF ALLTRIM(plinav) = "CURSOS Y TALLERES"
  cyt = .T.
ELSE
  cyt = .F.
ENDIF
DO WHILE .T.

```

```

horav = SPACE(5)
especialv = SPACE(25)
IF .NOT. cyt
  @ 6, 3 SAY 'Fecha:      (DD/MM/AA)'
  @ 6,32 SAY 'Hora:'
  @ 6,45 SAY 'Especial:'
  DO indica WITH 3
  SET COLOR TO &col4
  @ 6,10 GET fechav
  @ 6,38 GET horav PICT '99:99'
  @ 6,54 GET especialv
  READ
ELSE
  fech2v = fechav
  hora2v = SPACE(5)
  @ 6, 3 SAY 'Del:      AL      (DD/MM/AA)'
  @ 6,42 SAY 'De las:   a las   hrs.'
  @ 7, 3 SAY 'Especial:'
  DO indica WITH 3
  SET COLOR TO &col4
  @ 6, 8 GET fechav
  @ 6,21 GET fech2v
  @ 6,50 GET horav PICT '99:99'
  @ 6,62 GET hora2v PICT '99:99'
  @ 7,15 GET especialv
  READ
ENDIF
fechc = DTOC(fechav)
fechsin = SUBS(fechc,1,2)+SUBS(fechc,4,2)+SUBS(fechc,7,2)
IF LEN(RTRIM(SUBS(fechc,1,2))) = 0
  IF LEN(RTRIM(SUBS(horav,1,2))) = 0
    EXIT
  ELSE
    DO indica WITH 4
    SET COLOR TO &col4
    DO oprime
    LOOP
  ENDIF
ELSE
  IF LEN(RTRIM(SUBS(horav,1,2))) = 0
    DO indica WITH 4
    SET COLOR TO &col4
    DO oprime
    LOOP
  ELSE
    IF .NOT. cyt
      APPEND BLANK
      REPLACE folio WITH foliov
      REPLACE fecha WITH fechav
      REPLACE hora WITH horav
      REPI.ACE especial WITH especialv
      REPL.ACE disciplina WITH numbdis
    ELSE
      APPEND BLANK
    ENDIF
  ENDIF

```



```

REPLACE follo WITH follov
REPLACE fecha WITH fechav
REPLACE fechafin WITH fech2v
REPLACE hora WITH horav
REPLACE horafin WITH hora2v
REPLACE especial WITH especialv
REPLACE disciplina WITH numbdis
ENDIF
ENDIF
ENDIF
ENDDO
SELE 3
USE
SELE 4
USE actividad INDEX folact
REINDEX
@ 7, 3 SAY 'Actividad:'
DO indica WITH 6
SET COLOR TO &col4
DO WHILE .T.
  act1 = SPACE(60)
  act2 = SPACE(60)
  act3 = SPACE(60)
  act4 = SPACE(60)
  @ 7,15 GET act1
  @ 8,15 GET act2
  @ 9,15 GET act3
  @ 10,15 GET act4
  READ
  IF LEN(RTRIM(act1)) = 0 .AND. LEN(RTRIM(act2)) = 0 .AND. LEN(RTRIM(act3)) = 0 .AND.
LEN(RTRIM(act4)) = 0
  EXIT
ENDIF
IF LEN(RTRIM(act1)) = 0 .OR. LEN(RTRIM(act2)) = 0 .OR. LEN(RTRIM(act3)) = 0 .OR.
LEN(RTRIM(act4)) = 0
  APPEND BLANK
  REPLACE follo WITH follov
  REPLACE actividad WITH act1+act2+act3+act4
  EXIT
ENDIF
APPEND BLANK
REPLACE follo WITH follov
REPLACE actividad WITH act1+act2+act3+act4
ENDDO
SELE 4
USE
SELE 5
USE artista INDEX folartis
@ 11, 3 SAY 'Artista:'
DO indica WITH 7
SET COLOR TO &col4
DO WHILE .T.
  artis1 = SPACE(60)
  artis2 = SPACE(60)

```

```

artis3 = SPACE(60)
artis4 = SPACE(60)
@ 11,15 GET artis1
@ 12,15 GET artis2
@ 13,15 GET artis3
@ 14,15 GET artis4
READ
IF LEN(RTRIM(artis1)) = 0 .AND. LEN(RTRIM(artis2)) = 0 .AND. LEN(RTRIM(artis3)) = 0 .AND.
LEN(RTRIM(artis4)) = 0
EXIT
ENDIF
IF LEN(RTRIM(artis1)) = 0 .OR. LEN(RTRIM(artis2)) = 0 .OR. LEN(RTRIM(artis3)) = 0 .OR.
LEN(RTRIM(artis4)) = 0
APPEND BLANK
REPLACE folio WITH foliov
REPLACE artista WITH artis1+artis2+artis3+artis4
EXIT
ENDIF
APPEND BLANK
REPLACE folio WITH foliov
REPLACE artista WITH artis1+artis2+artis3+artis4
ENDDO
SELE 5
USE
SELE 6
USE lugares INDEX lugrecin
@ 15, 3 SAY 'Lugar:'
DO indica WITH 8
SET COLOR TO &col4
DO WHILE .T.
  lugarv = SPACE(38)
  @ 15,15 GET lugarv
  READ
  SEEK lugarv
  IF .NOT. FOUND()
    DO noexiste
    SET COLOR TO &col4
    @ 23, 2 SAY SPACE(77)
    resp = ''
    ask = 'DESEA DAR DE ALTA EL LUGAR'
    DO sino WITH resp.ask
    SET COLOR TO &col4
    @ 23, 2 SAY SPACE(77)
    IF resp = 'N'
      LOOP
    ELSE
      SAVE SCREEN TO pantalla[4]
      @ 5, 5 CLEAR TO 22,70
      @ 5, 5 TO 22,70
      DO crc2 WITH lugarv
      REINDEX
      SET COLOR TO &col4
      @ 5, 5 CLEAR TO 22,70
      RESTORE SCREEN FROM pantalla[4]
    ENDIF
  ENDIF
ENDWHILE

```

```

SET COLOR TO &col4
@ 15,15 SAY lugar
SELE 1
REPLACE lugar WITH lugar
EXIT
ENDIF
ELSE
SELE 1
REPLACE lugar WITH lugar
EXIT
ENDIF
ENDDO
@ 15,15 SAY lugar
SELE 1
REPLACE lugar WITH lugar
SELE 6
USE
SELE 7
USE observa INDEX folobs
@ 16, 3 SAY 'Observ:.'
DO indica WITH 9
SET COLOR TO &col4
DO WHILE .T.
  obser1 = SPACE(60)
  obser2 = SPACE(60)
  obser3 = SPACE(60)
  obser4 = SPACE(60)
  @ 16,15 GET obser1
  @ 17,15 GET obser2
  @ 18,15 GET obser3
  @ 19,15 GET obser4
  READ
  IF LEN(RTRIM(obser1)) = 0 .AND. LEN(RTRIM(obser2)) = 0 .AND. LEN(RTRIM(obser3)) = 0 .AND.
  LEN(RTRIM(obser4)) = 0
    EXIT
  ENDIF
  IF LEN(RTRIM(obser1)) = 0 .OR. LEN(RTRIM(obser2)) = 0 .OR. LEN(RTRIM(obser3)) = 0 .OR.
  LEN(RTRIM(obser4)) = 0
    APPEND BLANK
    REPLACE folio WITH foliov
    REPLACE observa WITH obser1+obser2+obser3+obser4
    EXIT
  ENDIF
  APPEND BLANK
  REPLACE folio WITH foliov
  REPLACE observa WITH obser1+obser2+obser3+obser4
ENDDO
SELE 7
USE
SELE 8
USE costdesc INDEX folcost
leyendav = SPACE(30)
preciov = 0
DO WHILE .T.

```

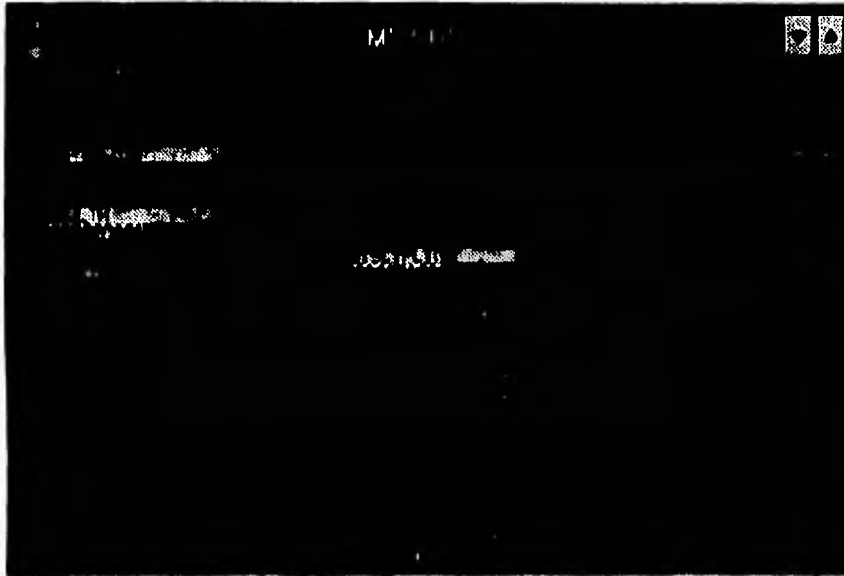
```

desctov = 0
leyendav = SPACE(30)
horav = SPACE(5)
@ 20, 3 SAY 'Precio: NS'
@ 20,25 SAY 'Desc:  % A:'
DO indica WITH 10
SET COLOR TO &col4
@ 20,15 GET preciov PICT '9,999.99'
@ 20,31 GET desctov PICT '999'
@ 20,39 GET leyendav
READ
IF preciov = 0
  IF desctov = 0
    IF LEN(RTRIM(leyendav))=0
      @ 22,15 SAY SPACE(60)
      EXIT
    ELSE
      APPEND BLANK
      REPLACE folio WITH follov
      REPLACE costo WITH preciov
      REPLACE desccosto WITH desctov
      REPLACE leyenda WITH leyendav
    ENDIF
  ELSE
    DO indica WITH 12
    SET COLOR TO &col4
    DO oprime
    LOOP
  ENDIF
ELSE
  APPEND BLANK
  REPLACE folio WITH follov
  REPLACE costo WITH preciov
  REPLACE desccosto WITH desctov
  REPLACE leyenda WITH leyendav
ENDIF
ENDDO
SELE 8
USE
hortaq = SPACE(50)
@ 21, 3 SAY 'Horario Taquilla:'
DO indica WITH 11
SET COLOR TO &col4
@ 21,21 GET hortaq
READ
SELE 1
REPLACE horataqui WITH hortaq
@ 5, 3 CLEAR TO 23,78
SELE 2
USE discipl INDEX nombdis,numdisci
REINDEX
SELE 3
USE horarios INDEX folhora,fecha,hora
REINDEX

```

```
SELE 1
USE inba INDEX folnba,disipl
REINDEX
LOOP
ENDDO
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
SELE 6
USE
SELE 7
USE
SELE 8
USE
RETURN
```

MENU CONSULTA DE LA PROGRAMACION DE EVENTOS



Procedimiento: CONPROG

Llamado por: PROGRAMA (procedimiento en INBA.PRG)

Llama a:

WINDOW	(procedimiento en INBA.PRG)
: CXDISC	(procedimiento en INBAPROC.PRG)
: CXACT	(procedimiento en INBAPROC.PRG)
: CXLUGAR	(procedimiento en INBAPROC.PRG)
: CXFECHA	(procedimiento en INBAPROC.PRG)
: CXARTIS	(procedimiento en INBAPROC.PRG)
: CXHORA	(procedimiento en INBAPROC.PRG)
: CXDYF	(procedimiento en INBAPROC.PRG)
: CXDYL	(procedimiento en INBAPROC.PRG)

```

PROCEDURE conprog
level2 = 1      @& memvar for 1st menu
DO WHILE (level2 != 0)
DO WINDOW WITH col4,8,30,17,49,'CONSULTAS'
  @ 9,32 PROMPT "DISCIPLINA " MESSAGE '      CONSULTA DE ACTIVIDADES POR
NOMBRE DE DISCIPLINA '
  @ 10,32 PROMPT "ACTIVIDAD " MESSAGE '      CONSULTA DE ACTIVIDADES POR
NOMBRE DE LA ACTIVIDAD
    
```

```

@ 11,32 PROMPT 'LUGAR ' MESSAGE ' CONSULTA DE ACTIVIDADES POR NOMBRE
DEL RECINTO EN EL QUE SE PRESENTAN '
@ 12,32 PROMPT 'RANGO DE FECHA ' MESSAGE ' CONSULTA DE ACTIVIDADES POR
RANGO DE FECHA EN QUE SE PRESENTAN '
@ 13,32 PROMPT 'ARTISTA ' MESSAGE ' CONSULTA DE ACTIVIDADES POR
NOMBRE DE ARTISTA '
@ 14,32 PROMPT 'HORA ' MESSAGE ' CONSULTA DE ACTIVIDADES POR HORA
DE PRESENTACION '
@ 15,32 PROMPT 'DISCIPL. Y FECHA' MESSAGE ' CONSULTA DE ACTIVIDADES POR
DISCIPLINA Y RANGO DE FECHA DE PRESENTACION '
@ 16,32 PROMPT 'DISCIPL. Y LUGAR' MESSAGE ' CONSULTA DE ACTIVIDADES POR
DISCIPLINA Y LUGAR DE PRESENTACION '
MENU TO level2
IF (level2 = 0) && esc to exit
EXIT
ENDIF
DO CASE
CASE level2 = 1
SAVE SCREEN TO pantalla [5]
DO cxdisc
RESTORE SCREEN FROM pantalla[5]
CASE level2 = 2
SAVE SCREEN TO pantalla [5]
DO cxact
RESTORE SCREEN FROM pantalla[5]
CASE level2 = 3
SAVE SCREEN TO pantalla [5]
DO cxlugar
RESTORE SCREEN FROM pantalla[5]
CASE level2 = 4
SAVE SCREEN TO pantalla [5]
DO cxfecha
RESTORE SCREEN FROM pantalla[5]
CASE level2 = 5
SAVE SCREEN TO pantalla [5]
DO cxartis
RESTORE SCREEN FROM pantalla[5]
CASE level2 = 6
SAVE SCREEN TO pantalla [5]
DO cxhora
RESTORE SCREEN FROM pantalla[5]
CASE level2 = 7
SAVE SCREEN TO pantalla [5]
DO cxdyf
RESTORE SCREEN FROM pantalla[5]
CASE level2 = 8
SAVE SCREEN TO pantalla [5]
DO cxdyf
RESTORE SCREEN FROM pantalla[5]
ENDCASE
ENDDO
SET COLOR TO &coll
level1 = 2
RETURN (level1)

```

Procedimiento:	CXDISC	
Llamado por:	CONPROG	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: NOEXISTE	(procedimiento en INBA.PRG)
	: PIDEFOL	(procedimiento en INBA.PRG)
	: TRAEACT	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
	: INBA.DBF	
	: ACTIVIDA.DBF	
Indices:	NUMDISC.IDX	
	: NOMBDIS.IDX	
	: FOLINBA.IDX	
	: DISCIPL.IDX	
	: FOLACT.IDX	

```

PROCEDURE cxdisc
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,20 SAY "CONSULTA DE PROGRAMACION POR DISCIPLINAS"
reng = 5
COLU = 7
DO WHILE .T.
  SET COLOR TO &col4
  @ 4,0 CLEAR TO 23,79
  ndis = SPACE(40)
  @ 5, 5 SAY "Disciplina:"
  DO indica WITH 13
  SET COLOR TO &col4
  @ 5,17 GET ndis
  READ
  ndis = ALLTRIM(ndis)
  IF LEN(ALLTRIM(ndis)) = 0
    EXIT
  ENDF
  SELE 1
  SET ORDER TO 2
  SEEK ndis
  IF .NOT. FOUND()
    DO noexiste
  LOOP
  ENDF
  numdis = numdiscipl

```



```

SELE 2
SET FILTER TO disciplina = numdis
GO BOTTOM
GO TOP
@ 7, 5 SAY 'FOLIO'
@ 7,20 SAY 'ACTIVIDAD'
rengcxd = 9
DO WHILE .NOT. EOF()
  regresa = .F.
  folnum = folio
  pointer = RECNO()
  SELE 3
  SEEK folnum
  @ rengcxd, 4 SAY folio
  @ rengcxd ,14 SAY SUBS(actividad,1,60)
  @ rengcxd+1,14 SAY SUBS(actividad,61,60)
  @ rengcxd+2,14 SAY SUBS(actividad,121,60)
  @ rengcxd+3,14 SAY SUBS(actividad,181,60)
  rengcxd = rengcxd+5
  IF rengcxd >= 24
    rengcxd = 9
    foln = 0
    DO WHILE .T.
      foln = ' '
      DO pldfol WITH foln
      SET COLOR TO &col4
      IF foln = 'I'
        GO TOP
        regresa = .T.
        EXIT
      ENDIF
      IF foln = 'T'
        SELE 1
        USE
        SELE 2
        USE
        SELE 3
        USE
        RETURN
      ENDIF
      foln = VAL(foln)
      IF foln <> 0
        SAVE SCREEN TO pantcxd
        DO traect WITH foln
        SET COLOR TO &col4
        RESTORE SCREEN FROM pantcxd
      ELSE
        EXIT
      ENDIF
    ENDDO
    @ 9, 4 CLEAR TO 23,77
  ENDIF
  SELE 2
  SET FILTER TO disciplina = numdis

```

```

IF regresa
GO TOP
LOOP
ELSE
GOTO pointer
ENDIF
SKIP
IF EOF()
foln = 0
DO WHILE .T.
foln = ' '
DO pidefol WITH foln
SET COLOR TO &col4
IF foln = ' '
GO TOP
regresa = .T.
EXIT
ENDIF
IF foln = 'T'
SELE 1
USE
SELE 2
USE
SELE 3
USE
RETURN
ENDIF
foln = VAL(foln)
IF foln <> 0
SAVE SCREEN TO pantcxd
DO traeact WITH foln
SET COLOR TO &col4
RESTORE SCREEN FROM pantcxd
ELSE
EXIT
ENDIF
ENDDO
ENDIF
ENDDO
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
level2 = 1
RETURN (level2)

```

Procedimiento:	CXACT	
Llamado por:	CONPROG	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: PIDEFOL	(procedimiento en INBA.PRG)
	: TRAEACT	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
	: INBA.DBF	
	: ACTIVIDA.DBF	
Indices:	NUMDISCI.IDX	
	: NOMBDIS.IDX	
	: FOLINBA.IDX	
	: DISCIPL.IDX	
	: FOLACT.IDX	

```

PROCEDURE exact
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,19 SAY "CONSULTA DE PROGRAMACION POR ACTIVIDAD"
reng = 5
COLU = 7
DO WHILE .T.
  GO TOP
  event = SPACE(60)
  @ 5, 5 SAY 'Actividad:'
  DO indica WITH 15
  SET COLOR TO &col4
  @ 5,17 GET event
  READ
  event = ALLTRIM(event)
  IF LEN(ALLTRIM(event)) = 0
    EXIT
  ENDFIF
  @ 7, 5 SAY 'FOLIO'
  @ 7,20 SAY 'ACTIVIDAD'
  rengcxd = 9
  DO WHILE .NOT. EOF()
    folnum = folio
    pointer = RECNO()
    SELE 3
    lastfol = folio
    IF event $ actividad
      lastfol = folio

```

```

@ rengcxd, 4 SAY folio
@ rengcxd ,14 SAY SUBS(actividad,1,60)
@ rengcxd+1,14 SAY SUBS(actividad,61,60)
@ rengcxd+2,14 SAY SUBS(actividad,121,60)
@ rengcxd+3,14 SAY SUBS(actividad,181,60)
rengcxd = rengcxd+5
IF rengcxd >= 24
  rengcxd = 9
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidefol WITH foln
    SET COLOR TO &col4
    IF foln = ' '
      GO TOP
      regresa = .T.
      EXIT
   ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
   ENDIF
    foln = VAL(foln)
    IF foln < 0
      SAVE SCREEN TO pantcxd
      DO tractact WITH foln
      SET COLOR TO &col4
      RESTORE SCREEN FROM pantcxd
    ELSE
      EXIT
   ENDIF
  ENDDO
  @ 9, 4 CLEAR TO 23,77
ENDIF
ENDIF
SELE 3
GOTO pointer
DO WHILE .T.
  SKIP
  IF folio = lastfol
    LOOP
  ELSE
    EXIT
  ENDIF
ENDDO
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = ' '

```

```

DO pidefol WITH foln
SET COLOR TO &col4
IF foln = 'I'
  GO TOP
  regresa = T.
  EXIT
ENDIF
IF foln = 'T'
  SELE 1
  USE
  SELE 2
  USE
  SELE 3
  USE
  RETURN
ENDIF
foln = VAL(foln)
IF foln <> 0
  SAVE SCREEN TO pantcxd
  DO tractact WITH foln
  SET COLOR TO &col4
  RESTORE SCREEN FROM pantcxd
ELSE
  EXIT
ENDIF
ENDDO
ENDIF
SELE 3
ENDDO
ENDDO
foln = 0
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
level2 = 2
RETURN (level2)

```

Procedimiento:	CXLUGAR	
Llamado por:	CONPROG	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: PIDEFOL	(procedimiento en INBA.PRG)
	: TRAEACT	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
	: INBA.DBF	
	: ACTIVIDA.DBF	
Indices:	NUMDISCI.IDX	
	: NOMBDIS.IDX	
	: FOLINBA.IDX	
	: DISCIPL.IDX	
	: FOLACT.IDX	

```

PROCEDURE cxlugar
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,23 SAY "CONSULTA DE PROGRAMACION POR LUGAR"
reng = 5
COLU = 7
DO WHILE .T.
  lugarx = SPACE(38)
  @ 5,5 SAY 'Lugar:'
  DO indica WITH 16
  SET COLOR TO &col4
  @ 5,17 GET lugarx
  READ
  IF LEN(RTRIM(lugarx)) = 0
    EXIT
  ENDF
  SELE 2
  SET FILTER TO lugar = lugarx
  GO BOTTOM
  GO TOP
  @ 7,5 SAY 'FOLIO'
  @ 7,20 SAY 'ACTIVIDAD'
  rengcxd = 9
  DO WHILE .NOT. EOF()
    folnum = folio
    pointer = RECNO()
    SELE 3

```

```

SEEK folnum
@ rengcx, 4 SAY folio
@ rengcx, 14 SAY SUBS(actividad,1,60)
@ rengcx+1,14 SAY SUBS(actividad,61,60)
@ rengcx+2,14 SAY SUBS(actividad,121,60)
@ rengcx+3,14 SAY SUBS(actividad,181,60)
rengcx = rengcx+5
IF rengcx >= 24
  rengcx = 9
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidedol WITH foln
    SET COLOR TO &col4
    IF foln = 'I'
      GO TOP
      regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
    ENDIF
    foln = VAL(foln)
    IF foln <> 0
      SAVE SCREEN TO pantcx
      DO treact WITH foln
      SET COLOR TO &col4
      RESTORE SCREEN FROM pantcx
    ELSE
      EXIT
    ENDIF
  ENDDO
  @ 9, 4 CLEAR TO 23,77
ENDIF
SELE 2
SET FILTER TO lugar = lugarx
GOTO pointer
SKIP
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidedol WITH foln
    SET COLOR TO &col4
    IF foln = 'I'
      GO TOP
      regresa = .T.
      EXIT

```

```
ENDIF
IF foln = 'T'
  SELE 1
  USE
  SELE 2
  USE
  SELE 3
  USE
  RETURN
ENDIF
foln = VAL(foln)
IF foln <> 0
  SAVE SCREEN TO pantcxd
  DO tract WITH foln
  SET COLOR TO &col4
  RESTORE SCREEN FROM pantcxd
ELSE
  EXIT
ENDIF
ENDDO
ENDIF
ENDDO
ENDDO
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
level2 = 3
RETURN (level2)
```


Procedimiento:	CXFECHA	
Llamado por:	CONPROG	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: OPRIME	(procedimiento en INBA.PRG)
	: PIDEFOL	(procedimiento en INBA.PRG)
	: TRAEACT	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
	: INBA.DBF	
	: ACTIVIDA.DBF	
	: HORARIOS.DBF	
Indices:	NUMDISCI.IDX	
	: NOMBDIS.IDX	
	: FOLINBA.IDX	
	: DISCIPL.IDX	
	: FOLACT.IDX	
	: FOLHORA.IDX	

```

PROCEDURE cxfecha
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,20 SAY "CONSULTA DE ACTIVIDAD POR RANGO DE FECHA"
reng = 5
COLU = 7
DO WHILE .NOT. EOF()
ndis = SPACE(40)
fecha1 = CTOD(' / / ')
fecha2 = CTOD(' / / ')
@ 5,5 SAY 'Del dia:   al dia:'
@ 6,14 SAY 'DD/MM/AA   DD/MM/AA'
DO indica WITH 17
SET COLOR TO #col4
@ 5,14 GET fecha1
@ 5,31 GET fecha2
READ
f1 = DTOC(fecha1)
f2 = DTOC(fecha2)
IF LEN(RTRIM(SUBS(f1,1,2))) = 0 .AND. LEN(RTRIM(SUBS(f2,1,2))) = 0
EXIT
ENDIF
IF LEN(RTRIM(SUBS(f1,1,2))) = 0 .OR. LEN(RTRIM(SUBS(f2,1,2))) = 0
@ 8,21 SAY 'NO SE PUEDE DEJAR UNA FECHA EN BLANCO'
DO oprime

```

```

@ 8,10 SAY SPACE(68)
LOOP
ENDIF
SELE 5
USE horarios INDEX folhora
IF fecha1 = fecha2
  SET FILTER TO fecha = fecha1
ELSE
  SET FILTER TO fecha >= fecha1 .AND. fecha <= fecha2
ENDIF
GO BOTTOM
GO TOP
@ 7, 5 SAY 'FOLIO'
@ 7,20 SAY 'ACTIVIDAD'
rengcxd = 9
DO WHILE .NOT. EOF()
  folnum = folio
  pointer = RECNO()
  SELE 3
  SEEK folnum
  @ rengcxd, 4 SAY folio
  ulfodi = folio
  @ rengcxd, 14 SAY SUBS(actividad,1,60)
  @ rengcxd+1,14 SAY SUBS(actividad,61,60)
  @ rengcxd+2,14 SAY SUBS(actividad,121,60)
  @ rengcxd+3,14 SAY SUBS(actividad,181,60)
  rengcxd = rengcxd+5
  IF rengcxd >= 24
    rengcxd = 9
    foln = 0
    DO WHILE .T.
      foln = ' '
      DO pidedfol WITH foln
      SET COLOR TO &col4
      IF foln = 'I'
        GO TOP
        regresa = .T.
        EXIT
      ENDIF
      IF foln = 'T'
        SELE 1
        USE
        SELE 2
        USE
        SELE 3
        USE
        RETURN
      ENDIF
      foln = VAL(foln)
      IF foln <> 0
        SAVE SCREEN TO pantcxd
        DO traect WITH foln
        SET COLOR TO &col4
        RESTORE SCREEN FROM pantcxd

```

```

ELSE
  EXIT
ENDIF
ENDDO
@ 9, 4 CLEAR TO 23,77
ENDIF
SELE 5
USE horarios INDEX folhora
SET FILTER TO fecha >= fecha1 .AND. fecha <= fecha2
GOTO pointer
folant = folnum
DO WHILE .NOT. EOF()
  SKIP
  IF folant = folio
    folant = folio
  LOOP
ELSE
  EXIT
ENDIF
ENDDO
ENDDO
ENDDO
foln = 0
DO WHILE .T.
  foln = 0
  DO pidefol WITH foln
  SET COLOR TO &col4
  IF foln <> 0
    SAVE SCREEN TO pantcxd
    DO traeact WITH foln
    SET COLOR TO &col4
    RESTORE SCREEN FROM pantcxd
  ELSE
    EXIT
  ENDIF
ENDDO
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
level2 = 4
RETURN (level2)

```

Procedimiento:	CXARTIS	
Llamado por:	CONPROG	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: PIDEFOL	(procedimiento en INBA.PRG)
	: TRAEACT	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
	: INBA.DBF	
	: ACTIVIDA.DBF	
	: ARTISTA.DBF	
Indices:	NUMDISCI.IDX	
	: NOMBDIS.IDX	
	: FOLINBA.IDX	
	: DISCIPL.IDX	
	: FOLACT.IDX	
	: FOLARTIS.IDX	

```

PROCEDURE cxartis
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
SELE 4
USE artista INDEX folartis
@ 3,20 SAY "CONSULTA DE PROGRAMACION POR ARTISTA"
reng = 5
COLU = 7
DO WHILE .T.
  GO TOP
  event = SPACE(60)
  @ 5, 5 SAY 'Artista:'
  DO indica WITH 20
  SET COLOR TO &col4
  @ 5,17 GET event
  READ
  event = ALLTRIM(event)
  IF LEN(ALLTRIM(event)) = 0
    EXIT
  ENDF
  @ 7, 5 SAY 'FOLIO'
  @ 7,20 SAY 'ACTIVIDAD'
  rengcxd = 9
  SELE 4
  DO WHILE .NOT. EOF()
    lastfol = folio

```

```

pointer = RECNO()
IF event $ artista
  folnum = folio
  lastfol = folio
  SELE 3
  SEEK folnum
  @ rengcx, 4 SAY folio
  @ rengcx, 14 SAY SUBS(actividad,1,60)
  @ rengcx+1, 14 SAY SUBS(actividad,61,60)
  @ rengcx+2, 14 SAY SUBS(actividad,121,60)
  @ rengcx+3, 14 SAY SUBS(actividad,181,60)
  rengcx = rengcx+5
  IF rengcx >= 24
    rengcx = 9
    foln = 0
    DO WHILE .T.
      foln = ' '
      DO pidefol WITH foln
      SET COLOR TO &col4
      IF foln = 'I'
        GO TOP
        regresa = .T.
        EXIT
      ENDF
      IF foln = 'T'
        SELE 1
        USE
        SELE 2
        USE
        SELE 3
        USE
        SELE 4
        USE
        RETURN
      ENDF
      foln = VAL(foln)
      IF foln <> 0
        SAVE SCREEN TO pantcx
        DO trackt WITH foln
        SELE 4
        USE artista INDEX folartis
        SET COLOR TO &col4
        RESTORE SCREEN FROM pantcx
      ELSE
        EXIT
      ENDF
    ENDDO
    @ 9, 4 CLEAR TO 23,77
  ENDF
  ENDF
  SELE 4
  GOTO pointer
  DO WHILE folio = lastfol
    SKIP

```

```

ENDDO
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = '
    DO pidefol WITH foln
    SET COLOR TO &col4
    IF foln = 'T'
      GO TOP
      regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      SELE 4
      USE
      RETURN
    ENDIF
    foln = VAL(foln)
    IF foln < 0
      SAVE SCREEN TO pantcxd
      DO traect WITH foln
      SELE 4
      USE artista INDEX folartis
      SET COLOR TO &col4
      RESTORE SCREEN FROM pantcxd
    ELSE
      EXIT
    ENDIF
  ENDDO
ENDIF
ENDDO
ENDDO
foln = 0
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
level2 = 5
RETURN (level2)

```

Procedimiento: CXHORA

Llamado por: CONPROG (procedimiento en INBA.PRG)

Llama a: INDICA (procedimiento en INBA.PRG)
: OPRIME (procedimiento en INBA.PRG)
: PIDEFOL (procedimiento en INBA.PRG)
: TRAEACT (procedimiento en INBAPROC.PRG)

Acceso a: DISCIPL.DBF
: INBA.DBF
: ACTIVIDA.DBF
: HORARIOS.DBF

Indices: NUMDISCI.IDX
: NOMBDIS.IDX
: FOLINBA.IDX
: DISCIPL.IDX
: FOLACT.IDX
: FOLHORA.IDX

```
PROCEDURE cxhora
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,20 SAY "CONSULTA DE ACTIVIDAD POR RANGO DE HORA"
reng = 5
COLU = 7
DO WHILE .NOT. EOF()
  ndis = SPACE(40)
  hora1 = ' : '
  hora2 = ' : '
  @ 5,5 SAY 'Entre : y las:'
  DO indica WITH 18
  SET COLOR TO &col4
  @ 5,14 GET hora1 PICT '99:99'
  @ 5,31 GET hora2 PICT '99:99'
  READ
  IF LEN(RTRIM(SUBS(hora1,1,2))) = 0 .AND. LEN(RTRIM(SUBS(hora2,1,2))) = 0
    EXIT
  ENDIF
  IF LEN(RTRIM(SUBS(hora1,1,2))) = 0 .OR. LEN(RTRIM(SUBS(hora2,1,2))) = 0
    @ 8,21 SAY 'NO SE PUEDE DEJAR UNA HORA EN BLANCO'
    DO oprime
    @ 8,10 SAY SPACE(68)
  LOOP
ENDIF
```

```

SELE 5
USE horarios INDEX folhora
IF hora1 = hora2
  SET FILTER TO hora = hora1
ELSE
  SET FILTER TO hora >= hora1 .AND. hora <= hora2
ENDIF
GO BOTTOM
GO TOP
@ 7, 5 SAY 'FOLIO'
@ 7,20 SAY 'ACTIVIDAD'
rengcxd = 9
DO WHILE .NOT. EOF()
  folnum = folio
  pointer = RECNO()
  SELE 3
  SEEK folnum
  @ rengcxd, 4 SAY folio
  ulfodi = folio
  @ rengcxd ,14 SAY SUBS(actividad,1,60)
  @ rengcxd+1,14 SAY SUBS(actividad,61,60)
  @ rengcxd+2,14 SAY SUBS(actividad,121,60)
  @ rengcxd+3,14 SAY SUBS(actividad,181,60)
  rengcxd = rengcxd+5
  IF rengcxd >= 24
    rengcxd = 9
    foln = 0
    DO WHILE .T.
      foln = ' '
      DO pidefol WITH foln
      SET COLOR TO &col4
      IF foln = 'I'
        GO TOP
        regresa = .T.
        EXIT
      ENDIF
      IF foln = 'T'
        SELE 1
        USE
        SELE 2
        USE
        SELE 3
        USE
        RETURN
      ENDIF
      foln = VAL(foln)
      IF foln <> 0
        SAVE SCREEN TO pantcxd
        DO tracact WITH foln
        SET COLOR TO &col4
        RESTORE SCREEN FROM pantcxd
      ELSE
        EXIT
      ENDIF
    ENDWHILE
  ENDIF

```



```

        ENDDO
        @. 9, 4 CLEAR TO 23,77
    ENDF
    SELE 5
    USE horarios INDEX folhora
    SET FILTER TO hora >= hora1 .AND. hora <= hora2
    GOTO pointer
    folant = foltrium
    DO WHILE .NOT. EOF()
        SKIP
        IF folant = folio
            folant = folio
        LOOP
        ELSE
            EXIT
        ENDF
    ENDDO
ENDDO
ENDDO
foln = 0
DO WHILE .T.
    foln = 0
    DO pntefol WITH foln
    SET COLOR TO &col4
    IF foln <> 0
        SAVE SCREEN TO pantcxd
        DO treact WITH foln
        SET COLOR TO &col4
        RESTORE SCREEN FROM pantcxd
    ELSE
        EXIT
    ENDF
ENDDO
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
level2 = 6
RETURN (level2)

```

Procedimiento:	CXDYF	
Llamado por:	CONPROG	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: NOEXISTE	(procedimiento en INBA.PRG)
	: OPRIME	(procedimiento en INBA.PRG)
	: PIDEFOL	(procedimiento en INBA.PRG)
	: TRAEACT	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
	: INBA.DBF	
	: ACTIVIDA.DBF	
	: HORARIOS.DBF	
Indices:	NUMDISCI.IDX	
	: NOMBDIS.IDX	
	: FOLINBA.IDX	
	: DISCIPL.IDX	
	: FOLACT.IDX	
	: FOLHORA.IDX	

```

PROCEDURE cxdyf
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdiscl,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,16 SAY "CONSULTA DE PROGRAMACION POR DISCIPLINA Y FECHA"
reng = 5
COLU = 7
DO WHILE .T.
  ndis = SPACE(40)
  @ 5,5 SAY 'Disciplina:'
  DO indica WITH 20
  SET COLOR TO &col4
  @ 5,17 GET ndis
  READ
  ndis = ALLTRIM(ndis)
  IF LEN(ALLTRIM(ndis)) = 0
    EXIT
  ENDF
  SELE 1
  SET ORDER TO 2
  SEEK ndis
  IF .NOT. FOUND()
    DO noexiste
  LOOP

```

```

ENDIF
numdis = numdiscipl
fecha1 = CTOD(' / / ')
fecha2 = CTOD(' / / ')
@ 7, 5 SAY 'Del dia:      al dia:'
@ 8,14 SAY 'DD/MM/AA      DD/MM/AA'
@ 7,14 GET fecha1
@ 7,31 GET fecha2
READ
f1 = DTOC(fecha1)
f2 = DTOC(fecha2)
IF LEN(RTRIM(SUBS(f1,1,2))) = 0 .AND. LEN(RTRIM(SUBS(f2,1,2))) = 0
  EXIT
ENDIF
IF LEN(RTRIM(SUBS(f1,1,2))) = 0 .OR. LEN(RTRIM(SUBS(f2,1,2))) = 0
  @ 10,21 SAY 'NO SE PUEDE DEJAR UNA FECHA EN BLANCO'
  DO oprime
  @ 10,10 SAY SPACE(68)
  LOOP
ENDIF
SELE 5
USE horarios INDEX folhora
IF fecha1 = fecha2
  SET FILTER TO disciplina = numdis .AND. fecha = fecha1
ELSE
  SET FILTER TO disciplina = numdis .AND. fecha >= fecha1 .AND. fecha <= fecha2
ENDIF
GO BOTTOM
GO TOP
@ 6,3 CLEAR TO 23,75
@ 7, 5 SAY 'FOLIO'
@ 7,20 SAY 'ACTIVIDAD'
rengcxd = 9
DO WHILE .NOT. EOF()
  regresa = .F.
  folnum = folio
  pointer = RECNO()
  SELE 3
  SEEK folnum
  @ rengcxd, 4 SAY folio
  @ rengcxd, 14 SAY SUBS(actividad,1,60)
  @ rengcxd+1,14 SAY SUBS(actividad,61,60)
  @ rengcxd+2,14 SAY SUBS(actividad,121,60)
  @ rengcxd+3,14 SAY SUBS(actividad,181,60)
  rengcxd = rengcxd+5
  IF rengcxd >= 24
    rengcxd = 9
    foln = 0
    DO WHILE .T.
      foln = ' '
      DO pidefol WITH foln
      SET COLOR TO &col4
      IF foln = 'I'
        GO TOP

```

```

    regresa = .T.
    EXIT
ENDIF
IF foln = 'T'
    SELE 1
    USE
    SELE 2
    USE
    SELE 3
    USE
    RETURN
ENDIF
foln = VAL(foln)
IF foln < 0
    SAVE SCREEN TO pantcxd
    DO traect WITH foln
    RESTORE SCREEN FROM pantcxd
ELSE
    EXIT
ENDIF
ENDDO
@ 9, 4 CLEAR TO 23,77
ENDIF
SELE 5
USE horarios INDEX folhora
IF fecha1 = fecha2
    SET FILTER TO disciplina = numdis .AND. fecha = fecha1
ELSE
    SET FILTER TO disciplina = numdis .AND. fecha >= fecha1 .AND. fecha <= fecha2
ENDIF
GO BOTTOM
GO TOP
IF regresa
    GO TOP
    LOOP
ELSE
    GOTO pointer
ENDIF
DO WHILE follo = folnum
    SKIP
ENDDO
IF EOF()
    foln = 0
    DO WHILE .T.
        foln = ' '
        DO pidefol WITH foln
        SET COLOR TO &col4
        IF foln = 'I'
            GO TOP
            regresa = .T.
            EXIT
        ENDIF
        IF foln = 'T'
            SELE 1

```

```
USE
SELE 2
USE
SELE 3
USE
SELE 5
USE
RETURN
ENDIF
foln = VAL(foln)
IF foln <> 0
    SAVE SCREEN TO pantcxd
    DO traect WITH foln
    RESTORE SCREEN FROM pantcxd
ELSE
    EXIT
ENDIF
ENDDO
ENDIF
ENDDO
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
level2 = 7
RETURN (level2)
```

Procedimiento: CXDYL

Llamado por: CONPROG (procedimiento en INBA.PRG)

Llama a: INDICA (procedimiento en INBA.PRG)
: NOEXISTE (procedimiento en INBA.PRG)
: PIDEFOL (procedimiento en INBA.PRG)
: TRAEACT (procedimiento en INBAPROC.PRG)

Accesa a: DISCIPL.DBF
: INBA.DBF
: ACTIVIDA.DBF

Indices: NUMDISCI.IDX
: NOMBDIS.IDX
: FOLINBA.IDX
: DISCIPL.IDX
: FOLACT.IDX

```
PROCEDURE cxdyl
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,16 SAY "CONSULTA DE PROGRAMACION POR DISCIPLINA Y LUGAR"
reng = 5
COLU = 7
DO WHILE .T.
  ndis = SPACE(40)
  @ 5, 5 SAY 'Disciplina:'
  DO indica WITH 21
  SET COLOR TO &col4
  @ 5,17 GET ndis
  READ
  ndis = ALLTRIM(ndis)
  IF LEN(ALLTRIM(ndis)) = 0
    EXIT
  ENDF
  SELE 1
  SET ORDER TO 2
  SEEK ndis
  IF .NOT. FOUND()
    DO noexiste
  LOOP
ENDF
numdis = numdiscipl
nplace = SPACE(38)
@ 7, 5 SAY 'Recinto: '
```

```

@ 7,17 GET nplace
READ
IF LEN(ALLTRIM(nplace)) = 0
  EXIT
ENDIF
SELE 2
SET FILTER TO disciplina = numdis .AND. lugar = nplace
GO BOTT
GO TOP
@ 6,3 CLEAR TO 23,75
@ 7, 5 SAY 'FOLIO'
@ 7,20 SAY 'ACTIVIDAD'
rengcxd = 9
DO WHILE .NOT. EOF()
  regresa = .F.
  folnum = folio
  pointer = RECNO()
  SELE 3
  SEEK folnum
  @ rengcxd, 4 SAY folio
  @ rengcxd, 14 SAY SUBS(actividad,1,60)
  @ rengcxd+1,14 SAY SUBS(actividad,61,60)
  @ rengcxd+2,14 SAY SUBS(actividad,121,60)
  @ rengcxd+3,14 SAY SUBS(actividad,181,60)
  rengcxd = rengcxd+5
  IF rengcxd >= 24
    rengcxd = 9
    foln = ()
    DO WHILE .T.
      foln = ' '
      DO pidedfol WITH foln
      SET COLOR TO &col4
      IF foln = ' '
        GO TOP
        regresa = .T.
        EXIT
      ENDIF
      IF foln = 'T'
        SELE 1
        USE
        SELE 2
        USE
        SELE 3
        USE
        RETURN
      ENDIF
      foln = VAL(foln)
      IF foln <> 0
        SAVE SCREEN TO pantcxd
        DO traect WITH foln
        RESTORE SCREEN FROM pantcxd
      ELSE
        EXIT
      ENDIF
    ENDWHILE
  ENDWHILE

```

```

ENDDO
@ 9, 4 CLEAR TO 23,77
ENDIF
SELE 2
USE inba INDEX fol inba, discipl
SET FILTER TO disciplina = numdis .AND. lugar = nplace
GO BOTTOM
GO TOP
IF regresa
GO TOP
LOOP
ELSE
GOTO pointer
ENDIF
SKIP
IF EOF()
foln = 0
DO WHILE .T.
foln = ' '
DO pdefol WITH foln
SET COLOR TO @col4
IF foln = 'I'
GO TOP
regresa = .T.
EXIT
ENDIF
IF foln = 'T'
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 5
USE
RETURN
ENDIF
foln = VAL(foln)
IF foln <> 0
SAVE SCREEN TO pantcxd
DO traect WITH foln
RESTORE SCREEN FROM pantcxd
ELSE
EXIT
ENDIF
ENDDO
ENDIF
ENDDO
SELE 1
USE
SELE 2
USE
SELE 3

```


USE
SELE 4
USE
SELE 5
USE
level2 = 8
RETURN (level2)

Procedimiento:	CAMPROG	
Llamado por:	PROGRAMA	(procedimiento en INBA.PRG)
Llana a:	INDICA : PIDEFOL : CAMBIAR	(procedimiento en INBA.PRG) (procedimiento en INBA.PRG) (procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF : INBA.DBF : ACTIVIDA.DBF	
Indices:	NOMBDIS.IDX : NUMDISCI.IDX : FOLINBA.IDX : DISCIPL.IDX : FOLACT.IDX	

```
PROCEDURE camprog && BUENO
SET EXACT OFF
SET COLOR TO &col4
@ 2,0 CLEAR TO 24,79
@ 3,29 SAY "CAMBIOS A PROGRAMACION"
SELE 1
USE discipl INDEX nombdis,numdisci
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
@ 3,29 SAY "CAMBIOS A PROGRAMACION"
reng = 5
COLU = 7
DO WHILE .T.
GO TOP
event = SPACE(60)
@ 5, 5 SAY 'Actividad:'
DO indica WITH 19
SET COLOR TO &col4
@ 5,17 GET event
READ
event = ALLTRIM(event)
IF LEN(ALLTRIM(event)) = 0
```

```

EXIT
ENDIF
@ 7,5 SAY 'FOLIO'
@ 7,20 SAY 'ACTIVIDAD'
rengcxd = 9
DO WHILE .NOT. EOF()
  folnum = folio
  pointer = RECNO()
  SELE 3
  IF event $ actividad
    @ rengcxd, 4 SAY folio
    @ rengcxd ,14 SAY SUBS(actividad,1,60)
    @ rengcxd+1,14 SAY SUBS(actividad,61,60)
    @ rengcxd+2,14 SAY SUBS(actividad,121,60)
    @ rengcxd+3,14 SAY SUBS(actividad,181,60)
    rengcxd = rengcxd+5
  IF rengcxd >= 24
    rengcxd = 9
    foln = 0
    DO WHILE .T.
      foln = ' '
      DO pidedfol WITH foln
      IF foln = 'I'
        GO TOP
        regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
    ENDIF
    foln = VAL(foln)
    IF foln <> 0
      SAVE SCREEN TO pantcxd
      DO cambiar WITH foln
      SET COLOR TO &col4
      RESTORE SCREEN FROM pantcxd
    ELSE
      EXIT
    ENDIF
  ENDDO
  SET COLOR TO &col4
  @ 9, 4 CLEAR TO 23,77
ENDIF
ENDIF
GOTO pointer
SKIP
DO WHILE folio = folnum
  SKIP

```

```

ENDDO
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = '
    DO pldcfol WITH foln
    IF foln = 'I'
      GO TOP
      regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
    ENDIF
    foln = VAL(foln)
    IF foln <> 0
      SAVE SCREEN TO pantcx4
      DO cambiar WITH foln
      SET COLOR TO @col4
      RESTORE SCREEN FROM pantcx4
    ELSE
      EXIT
    ENDIF
  ENDDO
ENDIF
ENDDO
ENDDO
foln = 0
SELE 1
USE
SELE 2
USE
SELE 3
USE
RETURN

```

Procedimiento: CAMBIAR

Llamado por: CAMPROG (procedimiento en INBAPROC.PRG)

Llama a: INDICA (procedimiento en INBA.PRG)
: OPRIME (procedimiento en INBA.PRG)
: NOEXISTE (procedimiento en INBA.PRG)

Acceso a: ACTIVIDA.DBF
: HORARIOS.DBF
: ARTISTA.DBF
: LUGARES.DBF
: OBSERVA.DBF
: COSTDESC.DBF

Indices: FOLACT.IDX
: FOLHORA.IDX
: FECHA.IDX
: HORA.IDX
: FOLARTIS.IDX
: LUGRECIN.IDX
: FOLOBS.IDX
: FOLCOST.IDX

```
PROCEDURE cambiar
PARAMETERS foln
SELE 3
USE
USE activida INDEX folact
SET COLOR TO &col4
@ 2,0 CLEAR TO 24,79
@ 3,29 SAY "CAMBIOS A PROGRAMACION"
@ 5,3 SAY 'Actividad:'
SEEK foln
DO indica WITH 6
SET COLOR TO &col4
folglob = folio
folant = folglob
reng = 5
DO WHILE .T.
  act1 = SUBS(actividad,1,60)
  act2 = SUBS(actividad,61,60)
  act3 = SUBS(actividad,121,60)
  act4 = SUBS(actividad,181,60)
  @ reng ,14 GET act1
  @ reng+1,14 GET act2
  @ reng+2,14 GET act3
  @ reng+3,14 GET act4
READ
REPLACE actividad WITH act1+act2+act3+act4
SKIP
```

```

IF folio <> folant
EXIT
ELSE
  reng = reng+4
  IF reng >23
    reng = 5
    @ 5,3 CLEAR TO 23,78
  ENDIF
  LOOP
ENDIF
ENDDO
reng = reng+4
IF reng >23
  reng = 5
  @ 5,3 CLEAR TO 23,78
ENDIF
SELE 2
SEEK folgjob
ndis = disciplina
SELE 1
SET ORDER TO 2
SEEK ndis
DO WHILE .T.
  plnav = disciplina
  @ reng, 3 SAY 'Disciplina:'
  DO indica WITH 27
  SET COLOR TO &col4
  @ reng,14 GET plnav
  READ
  IF LEN(ALLTRIM(plnav)) = 0
    @ reng+1,25 SAY 'Disciplina no puede quedar nulo'
    DO oprime
    LOOP
    @ reng+1,25 SAY SPACE(40)
  ENDIF
  SET ORDER TO 1
  SEEK plnav
  IF .NOT. FOUND()
    @ reng+1,18 SAY 'Deber dar de alta esa disciplina previamente'
    DO noexiste
    @ reng+1,18 SAY SPACE(50)
  SELE 2
  @ 3, 3 CLEAR TO 23,78
  RETURN
ENDIF
ndis = numdiscipl
SELE 2
REPLACE disciplina WITH ndis
EXIT
ENDDO
reng = reng + 1
IF reng >=21
  reng = 5
  @ 5,3 CLEAR TO 23,78

```

```

ENDIF
SELE 9
USE horarios INDEX folhora,fecha,hora
SEEK folgiob
finhoras = F.
IF ALLTRIM(plinav) = "CURSOS Y TALLERES"
  cyt = .T.
  fech2v = fechafin
  hora2v = horafin
ELSE
  cyt = .F.
ENDIF
DO WHILE .T.
  IF .NOT. cyt
    @ reng, 3 SAY 'Fecha:          (DD/MM/AA)'
    @ reng, 40 SAY 'Hora:'
    DO indica WITH 28
    SET COLOR TO &col4
  ENDIF
  fechav = fecha
  horav = hora
  especialv = especial
  IF finhoras
    @ reng+3, 18 SAY 'DEJE LA FECHA Y HORA EN BLANCO PARA TERMINAR'
    fechav = CTOD(' / / ')
    horav = ' : '
    fech2v = CTOD(' / / ')
    hora2v = ' : '
    especialv = SPACE(25)
  ENDIF
  IF .NOT. cyt
    @ reng, 14 GET fechav
    @ reng, 46 GET horav PICT '99:99'
    @ reng, 53 GET especialv
  READ
  ELSE
    @ reng, 3 SAY 'Del:          AL      (DD/MM/AA)'
    @ reng, 42 SAY 'De las:      a las  hrs.'
    @ reng+1, 3 SAY 'Especial:'
    DO indica WITH 29
    SET COLOR TO &col4
    @ reng+3, 18 SAY 'DEJE LA FECHA Y HORA EN BLANCO PARA TERMINAR'
    @ reng, 8 GET fechav
    @ reng, 21 GET fech2v
    @ reng, 50 GET horav PICT '99:99'
    @ reng, 62 GET hora2v PICT '99:99'
    @ reng+1, 15 GET especialv
  READ
  ENDIF
  fechc = DTOC(fechav)
  fechsln = SUBS(fechc, 1, 2)+SUBS(fechc, 4, 2)+SUBS(fechc, 7, 2)
  IF finhoras
    IF LEN(ALLTRIM(fechsln)) = 0
      IF LEN(ALLTRIM(SUBS(horav, 1, 2))) = 0

```

```

EXIT
ELSE
  @ 15,18 SAY 'NO PUEDE DEJAR LA FECHA O LA HORA EN BLANCO'
  SET INTE OFF
  nada = ''
  @ 15,64 GET nada
  READ
  SET INTE ON
  @ 15,18 SAY SPACE(50)
  LOOP
ENDIF
ELSE
  IF LEN(ALLTRIM(SUBS(horav,1,2))) = 0
    @ 15,18 SAY 'NO PUEDE DEJAR LA FECHA O LA HORA EN BLANCO'
    SET INTE OFF
    nada = ''
    @ 15,64 GET nada
    READ
    SET INTE ON
    @ 15,18 SAY SPACE(50)
    LOOP
  ENDIF
ENDIF
ENDIF
IF LEN(ALLTRIM(fechain)) = 0
  IF LEN(ALLTRIM(SUBS(horav,1,2))) = 0
    DELETE
    PACK
  ELSE
    @ 15,18 SAY 'NO PUEDE DEJAR LA FECHA O LA HORA EN BLANCO'
    SET INTE OFF
    nada = ''
    @ 15,64 GET nada
    READ
    SET INTE ON
    @ 15,18 SAY SPACE(50)
    LOOP
  ENDIF
ENDIF
ELSE
  IF LEN(ALLTRIM(SUBS(horav,1,2))) = 0
    @ 15,18 SAY 'NO PUEDE DEJAR LA FECHA O LA HORA EN BLANCO'
    SET INTE OFF
    SET COLOR TO N/N
    nada = ''
    @ 15,64 GET nada
    READ
    SET INTE ON
    @ 15,18 SAY SPACE(50)
    LOOP
  ELSE
    IF finhoras
      IF NOT cyt
        APPEND BLANK
        REPLACE folio WITH folglob
      ENDIF
    ENDIF
  ENDIF
ENDIF

```

```

REPLACE fecha WITH fechav
REPLACE hora WITH horav
REPLACE especial WITH especialv
ELSE
APPEND BLANK
REPLACE folio WITH folglob
REPLACE fecha WITH fechav
REPLACE fechafin WITH fech2v
REPLACE hora WITH horav
REPLACE horafin WITH hora2v
REPLACE especial WITH especialv
ENDIF
ELSE
IF .NOT. cyt
REPLACE fecha WITH fechav
REPLACE hora WITH horav
REPLACE especial WITH especialv
ELSE
REPLACE fecha WITH fechav
REPLACE fechafin WITH fech2v
REPLACE hora WITH horav
REPLACE horafin WITH hora2v
REPLACE especial WITH especialv
ENDIF
ENDIF
ENDIF
ENDIF
SKIP
IF EOF()
finhora = .T.
LOOP
ENDIF
IF folio <> folglob
finhora = .T.
LOOP
ENDIF
ENDDO
SELE 9
USE
reng = reng + 2
IF reng >= 21
reng = 5
@ 5,3 CLEAR TO 23,78
ENDIF
SELE 5
USE artista INDEX folartis
@ reng, 3 SAY 'Artista:'
DO indica WITH 7
SET COLOR TO &col4
SEEK folglob
nuevoreg = .F.
IF .NOT. FOUND()
nuevoreg = .T.
ENDIF

```



```

DO WHILE .T.
  IF nuevoreg
    APPEND BLANK
    REPLACE folio WITH foln
    REPLACE artista WITH SPACE(240)
    nuevoreg = .F.
  ENDF
  artis1 = SUBS(artista,1,60)
  artis2 = SUBS(artista,61,60)
  artis3 = SUBS(artista,121,60)
  artis4 = SUBS(artista,181,60)
  @ reng,14 GET artis1
  @ reng+1,14 GET artis2
  @ reng+2,14 GET artis3
  @ reng+3,14 GET artis4
  READ
  REPLACE artista WITH artis1+artis2+artis3+artis4
  SKIP
  IF folio <> folglob
    EXIT
  ENDF
  reng = reng+4
  IF reng > 23
    reng = 5
    @ 5,3 CLEAR TO 23,78
  ENDF
ENDDO
SELE 5
IF reng >= 21
  reng = 5
  @ 5,3 CLEAR TO 23,78
ENDIF
reng = reng +4
IF reng >= 21
  reng = 5
  @ 5,3 CLEAR TO 23,78
ENDIF
USE
SELE 6
USE lugares INDEX lugrecin
SELE 2
SEEK folglob
@ reng,3 SAY 'Lugar:'
DO indica WITH 8
SET COLOR TO &col4
DO WHILE .T.
  lugarv = lugar
  @ reng,14 GET lugarv
  READ
  SELE 6
  SEEK lugarv
  IF .NOT. FOUND()
    @ reng + 1,20 SAY 'Deber dar de alta previamente ese lugar'
    DO noexiste
  
```

```

SELE 2
@ 3,3 CLEAR TO 23,78
RETURN
ELSE
SELE 2
REPLACE lugar WITH lugarv
EXIT
ENDIF
ENDDO
SELE 6
USE
reng = reng + 1
IF reng >=21
reng = 5
@ 5,3 CLEAR TO 23,78
ENDIF
SELE 7
USE observa INDEX folobs
@ reng, 3 SAY 'Observa..'
DO indica WITH 9
SET COLOR TO &col4
SEEK folglob
IF .NOT. FOUND()
nuevoreg = .T.
ENDIF
DO WHILE .T.
IF nuevoreg
APPEND BLANK
REPLACE follo WITH foln
REPLACE observa WITH SPACE(240)
nuevoreg = .F.
ENDIF
obs1 = SUBS(observa,1,60)
obs2 = SUBS(observa,61,60)
obs3 = SUBS(observa,121,60)
obs4 = SUBS(observa,181,60)
@ reng ,14 GET obs1
@ reng+1,14 GET obs2
@ reng+2,14 GET obs3
@ reng+3,14 GET obs4
READ
REPLACE observa WITH obs1+obs2+obs3+obs4
SKIP
IF follo <> folglob
EXIT
ENDIF
reng = reng+4
IF reng > 23
reng = 5
@ 5,3 CLEAR TO 23,78
ENDIF
ENDDO
reng = reng+4
IF reng >=21

```

```

reng =5
@ 3,3 CLEAR TO 23,78
ENDIF
SELE 7
USE
SELE 8
USE costdesc INDEX folcost
SEEK folglob
SET INDEX TO
DO WHILE .T.
    leyendav = leyenda
    preciov = costo
    descctov = desccto
    @ reng, 3 SAY 'Precio: NS'
    @ reng,25 SAY 'Desc: % A.'
    DO indica WITH I0
    SET COLOR TO &col4
    @ 22,15 SAY 'DEJE EL PRECIO Y DESCUENTO IGUAL A CERO PARA ELIMINAR'
    @ reng,14 GET preciov PICT '9,999.99'
    @ reng,31 GET descctov PICT '999'
    @ reng,39 GET leyendav
    READ
    IF preciov = 0
        IF descctov = 0
            IF LEN(ALLTRIM(leyenda))=0
                @ 22,15 SAY SPACE(60)
                DELETE
                PACK
            ELSE
                REPLACE costo WITH preciov
                REPLACE desccto WITH descctov
                REPLACE leyenda WITH leyendav
            ENDIF
        ELSE
            @ 22,23 SAY 'NO PUEDE DEJAR PRECIO IGUAL A CERO'
            SET INTE OFF
            nada = ''
            @ 22,64 GET nada
            READ
            SET INTE ON
            @ 22,18 SAY SPACE(50)
            LOOP
        ENDIF
    ELSE
        REPLACE costo WITH preciov
        REPLACE desccto WITH descctov
        REPLACE leyenda WITH leyendav
    ENDIF
    SKIP
    IF folio <> folglob
        EXIT
    ENDIF
ENDDO
reng = reng+1

```

```

IF reng >=21
  reng =5
  @ 3,3 CLEAR TO 23,78
ENDIF
SELE 8
USE
SELE 2
SEEK folglob
hortaq = horataqui
@ reng, 3 SAY 'Horario Taquilla:'
DO indica WITH 11
SET COLOR TO &col4
@ reng,21 GET hortaq
READ
SELE 2
REPLACE horataqui WITH hortaq
@ 2, 2 CLEAR TO 23,78
RETURN

```

Procedimiento:	REPPROG	
Llamado por:	PROGRAMA	(procedimiento en INBA.PRG)
Llama a:	WINDOW	(procedimiento en INBA.PRG)
	: REPPGMEN	(procedimiento en INBAPROC.PRG)
	: REPPGSEM	(procedimiento en INBAPROC.PRG)

```

PROCEDURE repprog
SAVE SCREEN TO pantalla[19]
level1 = 1      && memvar for 1st menu
DO WHILE (level1 != 0)
  DO WINDOW WITH col4,09,28,12,51,' REPORTES '
  @ 10,30 PROMPT "PROGRAMACION MENSUAL" MESSAGE '      IMPRESION DE REPORTE
DE PROGRAMACION MENSUAL
  @ 11,30 PROMPT "PROGRAMACION SEMANAL" MESSAGE '      IMPRESION DE REPORTE
DE PROGRAMACION SEMANAL
  MENU TO level1
  IF (level1 = 0)      && esc to exit
    EXIT
  ENDIF
  DO CASE
  CASE level1 = 1
    SAVE SCREEN TO pantalla [3]
    DO reppgmen
    RESTORE SCREEN FROM pantalla[3]
  CASE level1 = 2
    SAVE SCREEN TO pantalla [3]
    DO reppgsem
    RESTORE SCREEN FROM pantalla[3]
  ENDCASE

```

```

ENDDO
RESTORE SCREEN FROM pantalla[19]
levell = 4
RETURN (levell)

```

Procedimiento:	REPPGMEN	
Llamado por:	REPPROG	(procedimiento en INBA.PRG)
Llama a:	MENUMES	(procedimiento en INBAPROC.PRG)
	: NOEXISTE	(procedimiento en INBA.PRG)
	: IMPRESOR	(procedimiento en INBA.PRG)
	: ISPRINTER()	(función de Clipper)
	: IMPRESORA()	(función en INBA.PRG)
	: GIRAR	(procedimiento en INBA.PRG)
	: DIASEM	(procedimiento en INBA.PRG)
	: HEADER	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
	: INBA.DBF	
	: ACTIVIDA.DBF	
	: ARTISTA.DBF	
	: HORARIOS.DBF	
	: FECHSOR.DBF	
	: LISFECH.DBF	
	: SLISFECH.DBF	
	: COSTDESC.DBF	
	: OBSERVA.DBF	
	: LUGARES.DBF	
Indices:	NOMBDIS.IDX	
	: NUMDISCI.IDX	
	: FOLACT.IDX	
	: FOLARTIS.IDX	
	: FOLHORA.IDX	
	: FOLOBS.IDX	
	: LUGRECIN.IDX	

```

PROCEDURE reppgmen
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
@ 3,25 SAY "REPORTE MENSUAL DE PROGRAMACION"
schizo = .T.
@ 13,15 SAY "Nombre del mes: "
npos = 1
DO menumes WITH npos,13,32,17,41
IF npos = 0
RETURN
ENDIF
@ 13,15 SAY SPACE(20)
SELE 7

```

```

USE discipl INDEX nombdis
fdis = SPACE(30)
nofill = .F.
DO WHILE .T.
  @ 12,15 SAY 'Disciplina: '
  @ 12,31 GET fdis PICT REPL('!',30)
  READ
  IF LEN(ALLTRIM(fdis)) = 0
    RETURN
  ENDIF
  SEEK fdis
  IF .NOT. FOUND()
    DO noexiste
    LOOP
  ELSE
    nd = numdiscipl
    EXIT
  ENDIF
ENDDO
USE
USE discipl INDEX numdisci
periodo = ALLTRIM(meses[NPOS])
DO impresor
IF .NOT. !sprinter()
  DO impresora
  SET COLOR TO &col4
  @ 22,0 CLEAR TO 24,79
  IF !sprinter()
    RETURN
  ENDIF
ENDIF
SET COLOR TO &col4
@ 22,0 CLEAR TO 24,79
@ 23,20 SAY 'PREPARANDO ARCHIVOS, ESPERE POR FAVOR'
@ 23,57 SAY '...'
fig = 1
DO girar WITH fig,23
SELE 1
USE inba
SELE 2
USE activida INDEX folact
SELE 3
USE artista INDEX folartis
SELE 4
USE
USE horarios INDEX folhora
SET FILTER TO MONTH(fecha) = npos .AND. disciplina = nd
DO girar WITH fig,23
GO BOTT
DO girar WITH fig,23
GO TOP
DO girar WITH fig,23
SORT TO fechsor ON follo,fecha,hora
USE

```

```

USE fechor
DO girar WITH fig,23
SELE 9
USE lisfech
ZAP
SELE 4
GO TOP
DO WHILE .NOT. EOF()
  DO girar WITH fig,23
  folv = folio
  fechv = fecha
  SELE 9
  APPEND BLANK
  REPLACE folio WITH folv
  REPLACE fecha WITH fechv
  SELE 4
  DO WHILE .T.
    IF folio = folv
      SKIP
    ELSE
      EXIT
    ENDIF
  ENDDO
ENDDO
SELE 9
SORT ON fecha TO slisfech
DO girar WITH fig,23
USE
USE slisfech
SELE 4
USE
USE horarios INDEX folhora
DO girar WITH fig,23
SELE 5
USE costdesc && INDEX FOLCOST
SELE 6
USE observa INDEX folobs
SELE 8
USE lugares INDEX lugrecin
DECLARE afecha(100)
DECLARE aactiv(100)
DECLARE aartis(100)
DECLARE alugar(100)
DECLARE aobser(100)
DECLARE adisci(100)
af = 1
aa = 1
ar = 1
al = 1
ao = 1
ad = 1
pmerpag = .T.
DO girar WITH fig,23
SET CONSOLE OFF

```

```

SET DEVICE TO PRINT
@ 0,0 SAY CHR(027)+'&#10'
@ 0,0 SAY CHR(027)+'(s#17H'
@ 0,0 SAY CHR(027)+'(10U'
SELE 9
GO TOP
DO WHILE .NOT. EOF()
  filtra = folio
  SELE 1
  SET FILTER TO folio = filtra
  GO BOTT
  GO TOP
  DO WHILE .NOT. EOF()
    SELE 1
    estadis = disciplina
    hrtaqv = horaataqui
    folglob = folio
    lugarv = lugar
    FOR i = 1 TO 100 STEP 1
      afecha[i] = ""
      aactiv[i] = ""
      aartis[i] = ""
      alugar[i] = ""
      aobser[i] = ""
    NEXT i
    af = 1
    aa = 1
    ar = 1
    al = 1
    ao = 1
    SELE 4
    SEEK folglob
    IF .NOT. FOUND()
      SELE 1
      SKIP
      LOOP
    ENDIF
    DO WHILE folio = folglob
      IF MONTH(fecha) <> npos
        SKIP
        LOOP
      ENDIF
      fechav = fecha
      horav = hora
      fechaf = fechafin
      horaf = horafin
      especialv = especial
      nombred = ''
      diames = 0
      nmes = ''
      DO diasem WITH fechav,nombred,diames,nmes
      afecha[af] = SUBS(nombred,1,3)+' '+STR(diames,2,0)+' '+horav
      af = af + 1
      IF LEN(ALLTRIM(horaf)) <> 0

```



```

nombred = ''
dianies = 0
nmes = ''
DO diasem WITH fechaf,nombred,dianies,nmes
afecha|AF| = ' A '+horaf
af = af+1
aobser|AO| = 'FECHA DE TERMINACION: '+DTCO(fechaf)
ao = ao+1
ENDIF
IF LEN(ALLTRIM(especialv)) <> 0
aobser|AO| = ALLTRIM(especialv)
ao = ao+1
ENDIF
SKIP
ENDDO
topof = af
SELE 2
SEEK folglob
DO WHILE folio = folglob
activi1 = SUBS(actividad, 1,60)
activi2 = SUBS(actividad, 61,60)
activi3 = SUBS(actividad, 121,60)
activi4 = SUBS(actividad, 181,60)
IF LEN(ALLTRIM(activi1)) <> 0
aactiv|AA| = activi1
aa = aa+1
ENDIF
IF LEN(ALLTRIM(activi2)) <> 0
aactiv|AA| = activi2
aa = aa+1
ENDIF
IF LEN(ALLTRIM(activi3)) <> 0
aactiv|AA| = activi3
aa = aa+1
ENDIF
IF LEN(ALLTRIM(activi4)) <> 0
aactiv|AA| = activi4
aa = aa+1
ENDIF
SKIP
ENDDO
SELE 3
SEEK folglob
DO WHILE folio = folglob
activi1 = SUBS(artista, 1,60)
activi2 = SUBS(artista, 61,60)
activi3 = SUBS(artista, 121,60)
activi4 = SUBS(artista, 181,60)
IF LEN(ALLTRIM(activi1)) <> 0
aactiv|AA| = activi1
aa = aa+1
ENDIF
IF LEN(ALLTRIM(activi2)) <> 0
aactiv|AA| = activi2

```

```

aa = aa+1
ENDIF
IF LEN(ALLTRIM(activi3)) <> 0
  aactiv[AA] = activi3
  aa = aa+1
ENDIF
IF LEN(ALLTRIM(activi4)) <> 0
  aactiv[AA] = activi4
  aa = aa+1
ENDIF
SKIP
ENDDO
topca = aa
alugar[AL] = lugarv
al = al+1
SELE 8
SEEK lugarv
IF LEN(ALLTRIM(direccion)) <> 0
  alugar[AL] = direccion
  al = al+1
ENDIF
IF LEN(ALLTRIM(colonia)) <> 0
  alugar[AL] = colonia
  al = al+1
ENDIF
IF LEN(ALLTRIM(ciudad)) <> 0
  alugar[AL] = ciudad
  al = al+1
ENDIF
IF LEN(ALLTRIM(cp)) <> 0
  alugar[AL] = 'C.P. '+cp
  al = al+1
ENDIF
IF LEN(ALLTRIM(lada)) <> 0
  alugar[AL] = 'CVE.LADA '+lada
  al = al+1
ENDIF
IF LEN(ALLTRIM(tel1)) <> 0
  alugar[AL] = 'TEL. '+tel1
  al = al+1
ENDIF
IF LEN(ALLTRIM(tel2)) <> 0
  alugar[AL] = ' '+tel2
  al = al+1
ENDIF
topcl = al
SELE 6
SEEK folglob
DO WHILE folio = folglob
  observ1 = SUBS(observa, 1,60)
  observ2 = SUBS(observa, 61,60)
  observ3 = SUBS(observa, 121,60)
  observ4 = SUBS(observa, 181,60)
  IF LEN(ALLTRIM(observ1)) <> 0

```

```

aobscr[AO] = observ1
ao = ao+1
ENDIF
IF LEN(ALLTRIM(observ2)) <> 0
aobscr[AO] = observ2
ao = ao+1
ENDIF
IF LEN(ALLTRIM(observ3)) <> 0
aobscr[AO] = observ3
ao = ao+1
ENDIF
IF LEN(ALLTRIM(observ4)) <> 0
aobscr[AO] = observ4
ao = ao+1
ENDIF
SKIP
ENDDO
IF LEN(ALLTRIM(hrtaqv)) <> 0
aobscr[AO] = 'HORARIO'
ao = ao+1
aobscr[AO] = hrtaqv
ao = ao+1
ENDIF
SELE 5
LOCATE FOR follo = folglob
DO WHILE follo = folglob
aox = ''
IF costo <> 0
aox = STR(costo,6,2)
aox = '$'+aox+' '
ENDIF
IF descscosto <> 0
aox = aox+STR(descscosto,3,0)+'% DESC. '
ENDIF
IF LEN(ALLTRIM(leyenda)) <> 0
aox = aox+ALLTRIM(leyenda)
ENDIF
aobscr[AO] = aox
ao = ao+1
SKIP
ENDDO
topco = ao
af = 1
aa = 1
al = 1
ao = 1
IF .NOT. pmerpag
IF estadis <> disant
tiporepo = 'MENSUAL'
rcngp = 8
SELE 1
ndis = disciplina
SELE 7
SEEK ndis

```

```

nombredis = disciplina
DO HEADER WITH tiporepo,pagina,nombredis,periodo
ENDIF
ENDIF
IF pmerpag
IF PROW() <> 0
EJECT
ENDIF
pmerpag = .F.
tiporepo = 'MENSUAL'
pagina = 0
rengp = 8
SELE 1
ndis = disciplina
SELE 7
SEEK ndis
nombredis = disciplina
DO HEADER WITH tiporepo,pagina,nombredis,periodo
ENDIF
IF topeo >= tope1 .AND. topeo >= topef .AND. topeo >= topea
itera = topeo
ENDIF
IF tope1 >= topeo .AND. tope1 >= topef .AND. tope1 >= topea
itera = tope1
ENDIF
IF topef >= topeo .AND. topef >= tope1 .AND. topef >= topea
itera = topef
ENDIF
IF topea >= tope1 .AND. topea >= topef .AND. topea >= topeo
itera = topea
ENDIF
i = 1
rengp = rengp+1
IF rengp+itera >= 45
IF rengp > 9
DO HEADER WITH tiporepo,pagina,nombredis,periodo
rengp = 8
ENDIF
ENDIF
DO WHILE i <= itera
@ rengp, 0 SAY afecha[i]
@ rengp, 13 SAY aactiv[i]
@ rengp, 74 SAY alugar[i]
@ rengp, 115 SAY aobser[i]
rengp = rengp+1
IF rengp >= 45
DO HEADER WITH tiporepo,pagina,nombredis,periodo
rengp = 8
ENDIF
i = i + 1
ENDDO
SELE 1
disant = disciplina
SKIP

```

```

ENDDO
SELE 9
SKIP
ENDDO
SET DEVICE TO SCREEN
SET CONSOLE ON
EJECT
DO girar WITH fig.23
@ 23,20 SAY SPACE(50)
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
SELE 6
USE
@ 23,20 SAY SPACE(50)
RETURN

```

Procedimiento:	REPPGSEM	
Llamado por:	REPPROG	(procedimiento en INBA.PRG)
Llama a:	OPRIME	(procedimiento en INBA.PRG)
	: IMPRESOR	(procedimiento en INBA.PRG)
	: ISPRINTER()	(función de Clipper)
	: IMPRESORA()	(función en INBA.PRG)
	: GIRAR	(procedimiento en INBA.PRG)
	: DIASEM	(procedimiento en INBA.PRG)
	: HEADER	(procedimiento en INBAPROC.PRG)
Accesa a:	INBA.DBF	
	: ACTIVIDA.DBF	
	: ARTISTA.DBF	
	: HORARIOS.DBF	
	: HORFECHA.DBF	
	: COSTDESC.DBF	
	: OBSERVA.DBF	
	: DISCIPL.DBF	
	: LUGARES.DBF	
Indices:	FOLINBA.IDX	
	: FOLACT.IDX	
	: FOLARTIS.IDX	
	: FOLCOST.IDX	
	: FOLOBS.IDX	

```
: NUMDISCI.IDX  
: LUGRECIN.IDX
```

```
PROCEDURE reppgsem  
SET COLOR TO &col4  
@ 1,0 CLEAR TO 24,79  
@ 3,25 SAY "REPORTE SEMANAL DE PROGRAMACION"  
schizo = .T.  
fechini = CTOD(' / / ')  
fechfin = CTOD(' / / ')  
pmerpag = .T.  
DO WHILE T.  
  @ 12,15 SAY 'De la semana del      al'  
  @ 13,15 SAY '          DD/MM/AA DD/MM/AA'  
  @ 12,32 GET fechini  
  @ 12,44 GET fechfin  
  READ  
  IF LEN(ALLTRIM(SUBS(DTOC(fechini),1,2))) = 0 .AND. LEN(ALLTRIM(SUBS(DTOC(fechfin),1,2))) =  
0  
  RETURN  
  ENDIF  
  IF LEN(ALLTRIM(SUBS(DTOC(fechini),1,2))) = 0 .OR. LEN(ALLTRIM(SUBS(DTOC(fechfin),1,2))) = 0  
  @ 15,17 SAY 'ES NECESARIO ESPECIFICAR EL RANGO DE LA SEMANA'  
  DO oprinc  
  @ 15,17 SAY SPACE(50)  
  LOOP  
  ENDIF  
  EXIT  
  ENDDO  
  DO impresor  
  IF .NOT. !printer()  
  DO impresora  
  SET COLOR TO &col4  
  @ 22,0 CLEAR TO 24,79  
  IF !printer()  
  RETURN  
  ENDIF  
  ENDIF  
  SET COLOR TO &col4  
  @ 22,0 CLEAR TO 24,79  
  flg = 1  
  @ 23,20 SAY 'PREPARANDO ARCHIVOS, ESPERE POR FAVOR'  
  @ 23,57 SAY '...'  
  DO girar WITH flg,23  
  SELE 1  
  USE inba INDEX folinba  
  DO girar WITH flg,23  
  SELE 2  
  USE activida INDEX folact  
  DO girar WITH flg,23  
  SELE 3  
  DO girar WITH flg,23  
  USE artista INDEX folartis
```

```

SELE 4
USE horarios && INDEX FOLHORA
SORT ON fecha,hora TO horfecha
DO girar WITH fig,23
USE
USE horfecha
SET FILTER TO fecha >= fechini .AND. fecha <= fechfin
GO BOTT
DO girar WITH fig,23
GO TOP
DO girar WITH fig,23
SELE 5
USE costdesc INDEX folcost
SELE 6
USE observa INDEX folobs
SELE 7
DO girar WITH fig,23
USE discipl INDEX numdisci
SELE 8
USE lugares INDEX lugrecin
DECLARE afecha[100]
DECLARE adisci[100]
DECLARE aactiv[100]
DECLARE alugar[100]
DECLARE aobser[100]
af = 1
ad = 1
aa = 1
al = 1
ao = 1
SELE 1
@ 23,20 SAY SPACE(50)
SELE 4
DO girar WITH fig,23
GO TOP
DO girar WITH fig,23
SET CONSOLE OFF
SET DEVICE TO PRINT
@ 0,0 SAY CHR(027)+'&I#10'
@ 0,0 SAY CHR(027)+'(i#17H'
@ 0,0 SAY CHR(027)+'(10U'
DO WHILE .NOT. EOF()
SELE 4
FOR i = 1 TO 100 STEP 1
    afecha[i] = "
    adisci[i] = "
    aactiv[i] = "
    alugar[i] = "
    aobser[i] = "
NEXT
af = 1
ad = 1
aa = 1
al = 1

```

```

ao = 1
folglob = folio
fechav = fecha
horav = hora
fechaf = fechafin
horaf = horafin
especialv = especial
nombred = ''
diames = 0
nmes = ''
DO diasem WITH fechav,nombred,diames,nmes
afecha[AF] = SUBS(nombred,1,3)+' '+STR(diames,2,0)
af = af+1
afecha[AF] = horav
af = af+1
IF LEN(ALLTRIM(horaf)) <> 0
  nombred = ''
  diames = 0
  nmes = ''
  DO diasem WITH fechaf,nombred,diames,nmes
  afecha[AF] = ' A'
  af = af+1
  afecha[AF] = horaf
  af = af+1
  aobscr[AO] = 'FECHA DE TERMINACION: '+DTC(fechaf)
  ao = ao+1
ENDIF
IF LEN(ALLTRIM(especialv)) <> 0
  aobscr[AO] = ALLTRIM(especialv)
  ao = ao+1
ENDIF
topef = af
SELE 1
SEEK folglob
numdis = disciplina
SELE 7
SEEK numdis
aactiv[AA] = disciplina
aa = aa+1
ad = 1
toped = ad
SELE 1
lugarv = lugar
alugar[AL] = lugarv
al = al+1
SELE 8
SEEK lugarv
IF LEN(ALLTRIM(direccion)) <> 0
  alugar[AL] = direccion
  al = al+1
ENDIF
IF LEN(ALLTRIM(colonia)) <> 0
  alugar[AL] = colonia
  al = al+1

```



```

ENDIF
IF LEN(ALLTRIM(ciudad)) <> 0
  alugar[AL] = ciudad
  al = al+1
ENDIF
lopecl = al
SELE 2
SEEK folglob
DO WHILE folio = folglob
  activi1 = SUBS(actividad, 1,60)
  activi2 = SUBS(actividad, 61,60)
  activi3 = SUBS(actividad, 121,60)
  activi4 = SUBS(actividad, 181,60)
  IF LEN(ALLTRIM(activi1)) <> 0
    aactiv[AA] = activi1
    aa = aa+1
  ENDIF
  IF LEN(ALLTRIM(activi2)) <> 0
    aactiv[AA] = activi2
    aa = aa+1
  ENDIF
  IF LEN(ALLTRIM(activi3)) <> 0
    aactiv[AA] = activi3
    aa = aa+1
  ENDIF
  IF LEN(ALLTRIM(activi4)) <> 0
    aactiv[AA] = activi4
    aa = aa+1
  ENDIF
SKIP
ENDDO
topca = aa
SELE 6
SEEK folglob
DO WHILE folio = folglob
  observ1 = SUBS(observa, 1,60)
  observ2 = SUBS(observa, 61,60)
  observ3 = SUBS(observa, 121,60)
  observ4 = SUBS(observa, 181,60)
  IF LEN(ALLTRIM(observ1)) <> 0
    aobser[AO] = observ1
    ao = ao+1
  ENDIF
  IF LEN(ALLTRIM(observ2)) <> 0
    aobser[AO] = observ2
    ao = ao+1
  ENDIF
  IF LEN(ALLTRIM(observ3)) <> 0
    aobser[AO] = observ3
    ao = ao+1
  ENDIF
  IF LEN(ALLTRIM(observ4)) <> 0
    aobser[AO] = observ4
    ao = ao+1
  ENDIF

```

```

    ENDIF
    SKIP
  ENDDO
  topeo = ao
  af = 1
  aa = 1
  ad = 1
  nl = 1
  ao = 1
  IF pmerpag
    pmerpag = .F.
    tiporepo = 'SEMANAL'
    rengp = 8
    pagina = 0
    periodo = 'DEL '+DTCO(fechini)+' AL '+DTCO(fechfin)
    nombredis = "
    DO HEADER WITH tiporepo,pagina,nombredis,periodo
  ENDIF
  IF topeo >= tope1 .AND. topeo >= topef .AND. topeo >= topea .AND. topeo >= toped
    itera = topeo
  ENDIF
  IF tope1 >= topeo .AND. tope1 >= topef .AND. tope1 >= topea .AND. tope1 >= toped
    itera = tope1
  ENDIF
  IF topef >= topeo .AND. topef >= tope1 .AND. topef >= topea .AND. topef >= toped
    itera = topef
  ENDIF
  IF topea >= tope1 .AND. topea >= topef .AND. topea >= topeo .AND. topea >= toped
    itera = topea
  ENDIF
  IF toped >= tope1 .AND. toped >= topef .AND. toped >= topeo .AND. toped >= topea
    itera = toped
  ENDIF
  i = 1
  rengp = rengp+1
  IF rengp+itera >= 45
    IF rengp > 9
      DO HEADER WITH tiporepo,pagina,nombredis,periodo
      rengp = 8
    ENDIF
  ENDIF
  DO WHILE i <= itera
    @ rengp, 0 SAY afecha[i]
    @ rengp, 9 SAY adisci[i]
    @ rengp, 13 SAY aactiv[i]
    @ rengp, 74 SAY alugar[i]
    @ rengp, 115 SAY aobser[i]
    rengp = rengp+1
    IF rengp >= 45
      DO HEADER WITH tiporepo,pagina,nombredis,periodo
      rengp = 8
    ENDIF
    i = i + 1
  ENDDO

```

```

SELE 1
disant = disciplina
SELE 4
SKIP
ENDDO
SET DEVICE TO SCREEN
SET CONSOLE ON
SET CONSOLE OFF
SET DEVICE TO PRINT
@ 0,0 SAY CHR(027)+'&#10'
@ 0,0 SAY CHR(027)+'(s#17H'
@ 0,0 SAY CHR(027)+'(10U'
EJECT
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
SELE 5
USE
SELE 6
USE
@ 23,20 SAY SPACE(50)
RETURN

```

Procedimiento: ELIMINA		
Llamado por:	PROGRAMA	(procedimiento en INBA.PRG)
Llama a:	WINDOW	(procedimiento en INBA.PRG)
	: ELIMENS	(procedimiento en INBAPROC.PRG)
	: ELIMACT	(procedimiento en INBAPROC.PRG)
	: RESTOMES	(procedimiento en INBAPROC.PRG)

```

PROCEDURE elimina
level1 = 1      && menvar for 1st menu
DO WHILE (level1 != 0)
  DO WINDOW WITH col4, 10,28,14,51,' ELIMINACION '
  @ 11,30 PROMPT "PROGRAMACION MENSUAL" MESSAGE '          ELIMINAR LA
PROGRAMACION DE TODO UN MES
  @ 12,30 PROMPT "ACTIVIDAD INDIVIDUAL" MESSAGE '          ELIMINAR UNA
ACTIVIDAD ESPECIFICA
  @ 13,30 PROMPT "RECUPERA PROG. MENS." MESSAGE '          RESTAURAR LA INFORMACION
DE UN MES QUE SE HABIA ELIMINADO
  MENU TO level1
  IF (level1 = 0)      && esc to exit
  EXIT

```

```

ENDIF
DO CASE
CASE level1 = 1
  SAVE SCREEN TO pantalla [3]
  DO elimens
  RESTORE SCREEN FROM pantalla[3]
CASE level1 = 2
  SAVE SCREEN TO pantalla [3]
  DO elimact
  RESTORE SCREEN FROM pantalla[3]
CASE level1 = 3
  SAVE SCREEN TO pantalla [3]
  DO restomes
  RESTORE SCREEN FROM pantalla[3]
ENDCASE
ENDDO
level1 = 5
RETURN (level1)

```

Procedimiento:	ELIMENS	
Llamado por:	ELIMINA	(procedimiento en INBA.PRG)
Llama a:	OPRIME	(procedimiento en INBA.PRG)
	: MENUMES	(procedimiento en INBAPROC.PRG)
	: SINO	(procedimiento en INBA.PRG)
	: GIRAR	(procedimiento en INBA.PRG)
Accesa a:	HORARIOS.DBF	
	: &DU	
	: INBA.DBF	
	: ACTIVIDA.DBF	
	: ARTISTA.DBF	
	: COSTDESC.DBF	
	: OBSERVA.DBF	
Indices:	FOLHORA.IDX	
	: FECHA.IDX	
	: HORA.IDX	
	: FOLINBA.IDX	
	: FOLACT.IDX	
	: FOLARTIS.IDX	
	: FOLCOST.IDX	
	: FOLOBS.IDX	

```

PROCEDURE elimens
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
texto = 'ELIMINACION DE PROGRAMACION POR MES'
@ 3,40-(LEN(texto)/2) SAY texto
mes = SPACE(10)

```

```

diskunit = ''
@ 5,12 SAY 'Para efectuar esta función se requerir un diskette para'
@ 6,12 SAY 'respaldar la información que se va a eliminar.'
@ 8,10 SAY 'En cual unidad de disco insertar el diskette: [A/B/C]'
@ 8,65 GET diskunit PICT '!' VALID(diskunit$ 'ABC')
READ
DO CASE
CASE diskunit = 'A'
  unidad = 1
CASE diskunit = 'B'
  unidad = 2
CASE diskunit = 'C'
  unidad = 3
ENDCASE
diskunit = diskunit+'.'
spcfree = DISKSPACE(unidad)
@ 10,10 SAY 'Espacio disponible en la unidad '+diskunit+'.'
@ 10,44 SAY spcfree/1024 PICT '999,999,999'
@ 10,56 SAY 'KBytes'
IF spcfree/1024 < 300
  @ 12,10 SAY 'No hay espacio suficiente, mínimo requerido 300 KBytes'
  DO oprime
  RETURN
ENDIF
@ 13,10 SAY 'Seleccione mes a eliminar.'
npos = 1
DO menumes WITH npos,13,38,17,47
IF npos = 0
  RETURN
ENDIF
@ 23,24 SAY 'PROCESANDO, ESPERE POR FAVOR...'
SELE 1
USE horarios INDEX folhora, fecha, hora
GO BOTT
j = RECNO()
DECLARE node1[j]
resp = ''
ask = 'ESTA SEGURO QUE DESEA ELIMINAR LA PROGRAMACION DE '+mcses[NPOS]
DO sino WITH resp,ask
IF resp = 'N'
  USE
  RETURN
ENDIF
@ 23,20 SAY 'PROCESANDO ARCHIVOS, ESPERE POR FAVOR...'
du = diskunit+'BKPHORA'
COPY STRUCT TO &du
SELE 9
USE &du
SELE 1
GO TOP
fig = 1
DO WHILE .NOT. EOF()
  DO girar WITH fig,23
  IF MONTH(fecha) = npos

```

```

foliov = folio
fechav = fecha
fechaf = fechafin
horav = hora
horaf = horafin
especv = especial
SELE 9
APPEND BLANK
REPLACE folio WITH foliov
REPLACE fecha WITH fechav
REPLACE fechafin WITH fechaf
REPLACE hora WITH horav
REPLACE horafin WITH horaf
REPLACE especial WITH especv
SELE 1
DELETE
ENDIF
SKIP
ENDDO
SELE 1
PACK
REINDEX
SELE 9
USE
SELE 2
USE inba INDEX folinba
du = diskunit+'BKPINBA'
COPY STRUC TO &du
SELE 9
USE &du
SELE 2
GO TOP
DO WHILE .NOT. EOF()
DO girar WITH fig,23
borrar = .F.
checa = folio
SELE 1
SEEK checa
IF .NOT. FOUND()
borrar = .T.
ENDIF
SELE 2
IF borrar
foliov = folio
disciv = disciplina
lugarv = lugar
horatv = horataqui
SELE 9
APPEND BLANK
REPLACE folio WITH foliov
REPLACE disciplina WITH disciv
REPLACE lugar WITH lugarv
REPLACE horataqui WITH horatv
SELE 2

```

```

DELETE
ENDIF
SKIP
ENDDO
SELE 9
USE
SELE 2
PACK
REINDEX
USE
USE activa INDEX folact
du = diskunit+'BKPACTI'
COPY STRUC TO &du
SELE 9
USE &du
SELE 2
DO WHILE .NOT. EOF()
DO girar WITH fig,23
borrar = .F.
checa = folio
SELE 1
SEEK checa
IF .NOT. FOUND()
borrar = .T.
ENDIF
SELE 2
IF borrar
foliov = folio
activv = actividad
SELE 9
APPEND BLANK
REPLACE folio WITH foliov
REPLACE actividad WITH activv
SELE 2
DELETE
ENDIF
SKIP
ENDDO
SELE 9
USE
SELE 2
PACK
REINDEX
USE
USE artista INDEX folartis
du = diskunit+'BKPARTI'
COPY STRUC TO &du
SELE 9
USE &du
SELE 2
DO WHILE .NOT. EOF()
DO girar WITH fig,23
borrar = .F.
checa = folio

```

```

SELE 1
SEEK checa
IF .NOT. FOUND()
  borrar = .T.
ENDIF
SELE 2
IF borrar
  foliov = folio
  artiov = artista
SELE 9
APPEND BLANK
REPLACE folio WITH foliov
REPLACE artista WITH artiov
SELE 2
DELETE
ENDIF
SKIP
ENDDO
SELE 9
USE
SELE 2
PACK
REINDEX
USE
USE condesc INDEX folcost
du = diskunit+'BKPCOST'
COPY STRUC TO &du
SELE 9
USE &du
SELE 2
DO WHILE .NOT. EOF()
  DO girar WITH fig,23
  borrar = .F.
  checa = folio
  SELE 1
  SEEK checa
  IF .NOT. FOUND()
    borrar = .T.
  ENDIF
  SELE 2
  IF borrar
    foliov = folio
    costov = costo
    descov = descosto
    leyenv = leyenda
  SELE 9
  APPEND BLANK
  REPLACE folio WITH foliov
  REPLACE costo WITH costov
  REPLACE descosto WITH descov
  REPLACE leyenda WITH leyenv
  SELE 2
  DELETE
ENDIF

```



```

SKIP
ENDDO
SELE 9
USE
SELE 2
PACK
REINDEX
USE
USE observa INDEX folobs
du = diskunit+'BKPOBSE'
COPY STRUC TO &du
SELE 9
USE &du
SELE 2
DO WHILE .NOT. EOF()
  DO girar WITH fig,23
  borrar = .F.
  checa = folio
  SELE 1
  SEEK checa
  IF .NOT. FOUND()
    borrar = .T.
  ENDIF
  SELE 2
  IF borrar
    foliov = folio
    obsevv = observa
    SELE 9
    APPEND BLANK
    REPLACE folio WITH foliov
    REPLACE observa WITH obsevv
  SELE 2
  DELETE
ENDIF
SKIP
ENDDO
SELE 9
USE
SELE 2
PACK
REINDEX
USE
SELE 1
USE
SELE 2
USE
SELE 9
USE
@: 23,20 SAY SPACE(55)
SET COLOR TO &coln
RETURN

```

Procedimiento:	ELIMACT	
Llamado por:	ELIMINA	(procedimiento en INBA.PRG)
Llama a:	PIDEFOL : BORRACT	(procedimiento en INBA.PRG) (procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF : INBA.DBF : ACTIVIDA.DBF	
Indice:	NOMBDIS.IDX : NUMDISCI.IDX : FOLINBA.IDX : DISCIPL.IDX : FOLACT.IDX	

```

PROCEDURE elimact
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
texto = 'ELIMINACION DE PROGRAMACION POR ACTIVIDAD'
@ 3,40-(LEN(texto)/2) SAY texto
SELE 1
USE discipl INDEX nombdis,numdisci
SELE 2
USE inba INDEX folinba,discipl
SELE 3
USE activida INDEX folact
reng = 5
COLU = 7
DO WHILE .T.
  GO TOP
  event = SPACE(60)
  @ 5,5 SAY 'Actividad:'
  @ 5,17 GET event
  READ
  event = ALLTRIM(event)
  IF LEN(ALLTRIM(event)) = 0
    EXIT
  ENDIF
  @ 7,5 SAY 'FOLIO'
  @ 7,20 SAY 'ACTIVIDAD'
  rengcxd = 9
  DO WHILE .NOT. EOF()
    folnum = folio
    pointer = RECNO()
    SELE 3
    lastfol = folio
    IF event $ actividad
      lastfol = folio
      @ rengcxd,4 SAY folio

```

```

@ rengcx d ,14 SAY SUBS(actividad,1,60)
@ rengcx d+1,14 SAY SUBS(actividad,61,60)
@ rengcx d+2,14 SAY SUBS(actividad,121,60)
@ rengcx d+3,14 SAY SUBS(actividad,181,60)
rengcx d = rengcx d+5
IF rengcx d >= 24
  rengcx d = 9
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidedfol WITH foln
    SET COLOR TO &col4
    IF foln = ' '
      GO TOP
      regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
    ENDIF
    foln = VAL(foln)
    IF foln <> 0
      SAVE SCREEN TO pantcx d
      DO borract WITH foln
      RESTORE SCREEN FROM pantcx d
    ELSE
      EXIT
    ENDIF
  ENDDO
  SET COLOR TO &col4
  @ 9, 4 CLEAR TO 23,77
ENDIF
ENDIF
GOTO pointer
DO WHILE .T.
  SKIP
  IF folio = lastfol
    LOOP
  ELSE
    EXIT
  ENDIF
ENDDO
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidedfol WITH foln
    SET COLOR TO &col4

```

```

IF foln = 'I'
  GO TOP
  regresa = .T.
  EXIT
ENDIF
IF foln = 'T'
  SELE 1
  USE
  SELE 2
  USE
  SELE 3
  USE
  RETURN
ENDIF
foln = VAL(foln)
IF foln < 0
  SAVE SCREEN TO pantcxd
  DO borract WITH foln
  RESTORE SCREEN FROM pantcxd
ELSE
  EXIT
ENDIF
ENDDO
ENDIF
ENDDO
foln = 0
SELE 1
USE
SELE 2
USE
SELE 3
USE
SELE 4
USE
RETURN

```

Procedimiento:	BORRACT	
Llamado por:	EI.IMACT	(procedimiento en INBAPROC.PRG)
Llama a:	SINO : GIRAR	(procedimiento en INBA.PRG) (procedimiento en INBA.PRG)
Accesa a:	HORARIOS.DBF : ARTISTA.DBF : COSTDESC.DBF : OBSERVA.DBF	
Indices:	FOLHORA.IDX : FECHA.IDX	

```
: HORA.IDX  
: FOLARTIS.IDX  
: FOLCOST.IDX  
: FOLOBS.IDX
```

```
PROCEDURE borract  
PARAMETERS foln  
resp = ''  
ask = 'ELIMINAR ESA ACTIVIDAD'  
DO sino WITH resp,ask  
IF resp = 'N'  
  RETURN  
ENDIF  
SET COLOR TO &col4  
@ 9,3 CLEAR TO 23,77  
SET COLOR TO &col4  
@ 23,24 SAY 'PROCESANDO, ESPERE POR FAVOR...'  
SELE 4  
USE horarios INDEX folhora, fecha, hora  
GO TOP  
fig = 1  
DO WHILE .NOT. EOF()  
  DO girar WITH fig,23  
  IF folio = foln  
    DELETE  
  ENDIF  
  SKIP  
ENDDO  
PACK  
REINDEX  
USE  
SELE 2  
GO TOP  
DO WHILE .NOT. EOF()  
  DO girar WITH fig,23  
  IF folio = foln  
    DELETE  
  ENDIF  
  SKIP  
ENDDO  
PACK  
REINDEX  
SELE 3  
GO TOP  
DO WHILE .NOT. EOF()  
  DO girar WITH fig,23  
  IF folio = foln  
    DELETE  
  ENDIF  
  SKIP  
ENDDO  
PACK  
REINDEX
```

SELE 4
USE
USE artista INDEX folartis
GO TOP
DO WHILE .NOT. EOF()
 DO girar WITH fig,23
 IF folio = foln
 DELETE
 ENDIF
 SKIP
ENDDO
PACK
REINDEX
SELE 4
USE
USE costdesc INDEX folcost
GO TOP
DO WHILE .NOT. EOF()
 DO girar WITH fig,23
 IF folio = foln
 DELETE
 ENDIF
 SKIP
ENDDO
PACK
REINDEX
SELE 4
USE
USE observa INDEX folobs
GO TOP
DO WHILE .NOT. EOF()
 DO girar WITH fig,23
 IF folio = foln
 DELETE
 ENDIF
 SKIP
ENDDO
PACK
REINDEX
@ 23.20 SAY SPACE(55)
@ 9.3 CLEAR TO 22,77
RETURN

Procedimiento: RESTOMES

Llamado por: ELIMINA (procedimiento en INBA.PRG)

Llama a: SINO (procedimiento en INBA.PRG)
: OPRIME (procedimiento en INBA.PRG)
: GIRAR (procedimiento en INBA.PRG)

Accesa a: INBA.DBF
: &DU
: DU.DBF
: HORARIOS.DBF
: COSTDESC.DBF
: OBSERVA.DBF
: ACTIVIDA.DBF
: ARTISTA.DBF

Indices: POLINBA.IDX
: FOLHORA.IDX
: FECHA.IDX
: HORA.IDX
: FOLCOST.IDX
: FOL OBS.IDX
: FOLACT.IDX
: FOLARTIS.IDX

PROCEDURE restomes

SET COLOR TO &col4

@ 1,0 CLEAR TO 24,79

texto = 'RECUPERACION DE PROGRAMACION MENSUAL ELIMINADA PREVIAMENTE'

@ 3,40-(LEN(texto)/2) SAY texto

mes = SPACE(10)

diskunit = ''

@ 5,12 SAY 'Para efectuar esta función deber insertar el diskette en'

@ 6,12 SAY 'el que se respaldó la información previamente.'

@ 8,10 SAY 'En cual unidad de disco inserte el diskette: [A/B/C]'

@ 8,65 GET diskunit PICT '' VALID(diskunit\$ 'ABC')

READ

DO CASE

CASE diskunit = 'A'

unidad = 1

CASE diskunit = 'B'

unidad = 2

CASE diskunit = 'C'

unidad = 3

ENDCASE

resp = ''

ask = 'EFECTUAR LA RECUPERACION DE LA PROGRAMACION MENSUAL ELIMINADA'

DO sino WITH resp,ask

IF resp = 'N'

RETURN

```

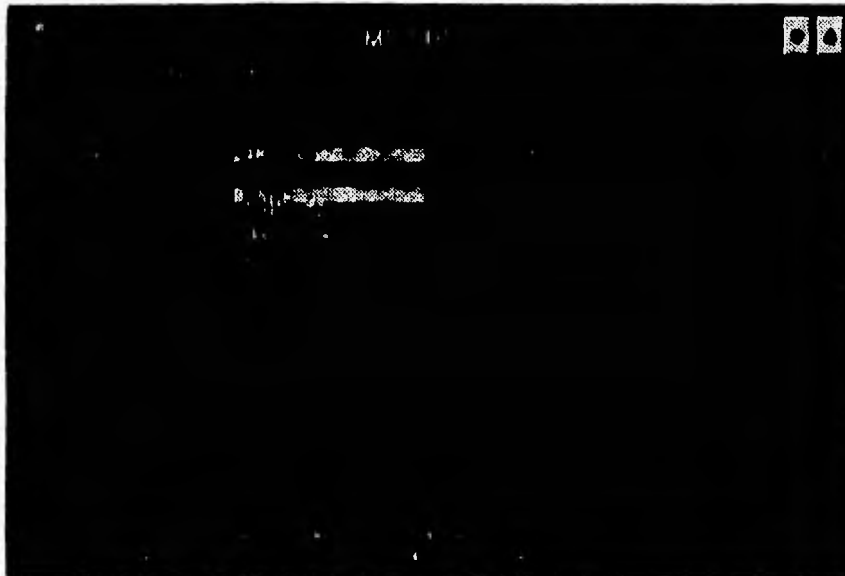
ENDIF
diskunit = diskunit+'!'
@ 23,20 SAY 'PROCESANDO ARCHIVOS, ESPERE POR FAVOR...'
du = diskunit+'BKPINBA.DBF'
IF .NOT. FILE(du)
  @ 18,14 SAY 'NO EXISTE EL ARCHIVO '+du+' EN LA UNIDAD INDICADA,'
  @ 19,14 SAY 'NO SE PUEDE EFECTUAR LA RECUPERACION.'
  @ 23,20 SAY SPACE(50)
  DO oprime
  RETURN
ENDIF
du = diskunit+'BKPHORA.DBF'
IF .NOT. FILE(du)
  @ 18,14 SAY 'NO EXISTE EL ARCHIVO '+du+' EN LA UNIDAD INDICADA,'
  @ 19,14 SAY 'NO SE PUEDE EFECTUAR LA RECUPERACION.'
  @ 23,20 SAY SPACE(50)
  DO oprime
  RETURN
ENDIF
du = diskunit+'BKPCOST.DBF'
IF .NOT. FILE(du)
  @ 18,14 SAY 'NO EXISTE EL ARCHIVO '+du+' EN LA UNIDAD INDICADA,'
  @ 19,14 SAY 'NO SE PUEDE EFECTUAR LA RECUPERACION.'
  @ 23,20 SAY SPACE(50)
  DO oprime
  RETURN
ENDIF
du = diskunit+'BKPOBSE.DBF'
IF .NOT. FILE(du)
  @ 18,14 SAY 'NO EXISTE EL ARCHIVO '+du+' EN LA UNIDAD INDICADA,'
  @ 19,14 SAY 'NO SE PUEDE EFECTUAR LA RECUPERACION.'
  @ 23,20 SAY SPACE(50)
  DO oprime
  RETURN
ENDIF
du = diskunit+'BKPACTI.DBF'
IF .NOT. FILE(du)
  @ 18,14 SAY 'NO EXISTE EL ARCHIVO '+du+' EN LA UNIDAD INDICADA,'
  @ 19,14 SAY 'NO SE PUEDE EFECTUAR LA RECUPERACION.'
  @ 23,20 SAY SPACE(50)
  DO oprime
  RETURN
ENDIF
du = diskunit+'BKPARTI.DBF'
IF .NOT. FILE(du)
  @ 18,14 SAY 'NO EXISTE EL ARCHIVO '+du+' EN LA UNIDAD INDICADA,'
  @ 19,14 SAY 'NO SE PUEDE EFECTUAR LA RECUPERACION.'
  @ 23,20 SAY SPACE(50)
  DO oprime
  RETURN
ENDIF
du = diskunit+'BKPINBA.DBF'
SELE 2
USE

```



```
USE inba INDEX folinba
fig = 1
DO girar WITH fig,23
APPEND FROM &du
REINDEX
ERASE du
du = diskunit+'BKPHORA.DBF'
SELE 2
USE
USE horarios INDEX folhora, fecha, hora
DO girar WITH fig,23
APPEND FROM &du
REINDEX
ERASE du
du = diskunit+'BKPCOST.DBF'
SELE 2
USE
USE costdesc INDEX folcost
DO girar WITH fig,23
APPEND FROM &du
REINDEX
ERASE du
du = diskunit+'BKPOBSE.DBF'
SELE 2
USE
USE observa INDEX folobs
DO girar WITH fig,23
APPEND FROM &du
REINDEX
ERASE du
du = diskunit+'BKPACTI.DBF'
SELE 2
USE
USE activida INDEX folact
DO girar WITH fig,23
APPEND FROM &du
REINDEX
ERASE du
du = diskunit+'BKPARTI.DBF'
SELE 2
USE
USE artista INDEX folartis
DO girar WITH fig,23
APPEND FROM &du
REINDEX
ERASE du
@ 23,20 SAY SPACE(50)
RETURN
```

MENU RECINTOS



Procedimiento: **RECINTO**

Llamado por: **INBA.PRG**

Llama a:	CAPRECIN	(procedimiento en INBA.PRG)
	: CONRECIN	(procedimiento en INBA.PRG)
	: CAMRECIN	(procedimiento en INBA.PRG)
	: BAJRECIN	(procedimiento en INBA.PRG)
	: REPRECIN	(procedimiento en INBA.PRG)

```
PROCEDURE recinto
SAVE SCREEN TO pantalla[20]
SET COLOR TO &col1
@ 6,0 CLEAR TO 22,79
level1 = 1      && memvar for 1st menu
DO WHILE (level1 != 0)
  SET COLOR TO &col4
  @ 5,20 TO 11,40
  @ 5,20 SAY 'A'
  @ 5,40 SAY 'A'
```

```

SET COLOR TO &col8
@ 6,21 PROMPT " ALTAS      " MESSAGE '
RECINTOS
@ 7,21 PROMPT " CONSULTA  " MESSAGE '
RECINTOS
@ 8,21 PROMPT ' MODIFICACION ' MESSAGE '
GENERALES DE RECINTOS
@ 9,21 PROMPT ' BAJAS      ' MESSAGE '

@ 10,21 PROMPT ' REPORTES  ' MESSAGE '
GENERAL DE RECINTOS
MENU TO level1
IF (level1 = 0)      && esc to exit
EXIT
ENDIF
DO CASE
CASE level1 = 1
SAVE SCREEN TO pantalla (2)
DO caprecin
RESTORE SCREEN FROM pantalla(2)
CASE level1 = 2
SAVE SCREEN TO pantalla (2)
DO comrecin
RESTORE SCREEN FROM pantalla(2)
CASE level1 = 3
SAVE SCREEN TO pantalla (2)
DO camrecin
RESTORE SCREEN FROM pantalla(2)
CASE level1 = 4
SAVE SCREEN TO pantalla (2)
DO bajrecin
RESTORE SCREEN FROM pantalla(2)
CASE level1 = 5
SAVE SCREEN TO pantalla (2)
DO rprecin
RESTORE SCREEN FROM pantalla(2)
ENDCASE
ENDDO
RESTORE SCREEN FROM pantalla(20)
level1 = 2
RETURN (level1)

```

```

CAPTURA DE DATOS GENERALES DE
MENU DE CONSULTA DE
MODIFICACION DE DATOS
BAJA DE RECINTOS
IMPRESION DEL CATALOGO

```

Procedimiento: CAPRECIN

Llamado por: RECINTO (procedimiento en INBA.PRG)

Llama a: CREC2 (procedimiento en INBA.PRG)

Accesa a: LUGARES.DBF

Indices: LUGRECIN.IDX

```
PROCEDURE caprecin
SELE 1
SET COLOR TO &col4
USE lugares INDEX lugrecin
@ 1,0 CLEAR TO 24,79
@ 3,30 SAY "CAPTURA DE RECINTOS"
lugarv = SPACE(38)
DO crec2 WITH lugarv
USE
RETURN
```

MENU CONSULTA DE RECINTOS



Procedimiento:	CONRECIN	
Llamado por:	RECINTO	(procedimiento en INBA.PRG)
Llama a:	WINDOW	(procedimiento en INBA.PRG)
	: CONRALL	(procedimiento en INBA.PRG)
	: CONRNOM	(procedimiento en INBA.PRG)
	: CONRCAL	(procedimiento en INBA.PRG)
	: CONRCOL	(procedimiento en INBA.PRG)
	: CONRDEL	(procedimiento en INBA.PRG)
	: CONRCIU	(procedimiento en INBA.PRG)
	: CONRCP	(procedimiento en INBA.PRG)

```

PROCEDURE conrecin
SAVE SCREEN TO pantalla[19]
level1 = 1      && memvar for 1st menu
DO WHILE (level1 != 0)
  DO WINDOW WITH col4,9,28,17,51,'CONSULTA RECINTO'
  @ 10,31 PROMPT "TODOS LOS NOMBRES " MESSAGE '      CONSULTA DE NOMBRES DE
RECINTOS EN EL CATALOGO
  
```

```

@ 11,31 PROMPT "DETALLE POR NOMBRE" MESSAGE '          CONSULTA DE DATOS
GENERALES DE RECINTOS POR NOMBRE
@ 12,31 PROMPT 'POR CALLE ' MESSAGE '          CONSULTA DE DATOS GENERALES DE
RECINTOS POR CALLES
@ 13,31 PROMPT 'POR COLONIA ' MESSAGE '          CONSULTA DE DATOS GENERALES DE
RECINTOS POR COLONIA
@ 14,31 PROMPT 'POR DELEGACION ' MESSAGE '          CONSULTA DE DATOS GENERALES
DE RECINTOS POR DELEGACION
@ 15,31 PROMPT 'POR CIUDAD ' MESSAGE '          CONSULTA DE DATOS GENERALES DE
RECINTOS POR CIUDAD
@ 16,31 PROMPT 'POR C.P. ' MESSAGE '          CONSULTA DE DATOS GENERALES DE
RECINTOS POR CODIGO POSTAL
MENU TO level1
IF (level1 = 0)      && esc to exit
EXIT
ENDIF
DO CASE
CASE level1 = 1
SAVE SCREEN TO pantalla[3]
DO conral
RESTORE SCREEN FROM pantalla[3]
CASE level1 = 2
SAVE SCREEN TO pantalla[3]
DO conmom
RESTORE SCREEN FROM pantalla[3]
CASE level1 = 3
SAVE SCREEN TO pantalla[3]
DO conrcal
RESTORE SCREEN FROM pantalla[3]
CASE level1 = 4
SAVE SCREEN TO pantalla[3]
DO conrcol
RESTORE SCREEN FROM pantalla[3]
CASE level1 = 5
SAVE SCREEN TO pantalla[3]
DO conrdel
RESTORE SCREEN FROM pantalla[3]
CASE level1 = 6
SAVE SCREEN TO pantalla[3]
DO conrciu
RESTORE SCREEN FROM pantalla[3]
CASE level1 = 7
SAVE SCREEN TO pantalla[3]
DO conrcp
RESTORE SCREEN FROM pantalla[3]
ENDCASE
ENDDO
RESTORE SCREEN FROM pantalla[19]
level1 = 2
RETURN (level1)

```

Procedimiento: CONRALL

Llamado por: CONRECIN (procedimiento en INBA.PRG)

Llama a: OPRIME (procedimiento en INBA.PRG)

Accesa a: LUGARES.DBF

```
PROCEDURE conrall
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE 1
USE lugares
@ 3,23 SAY "CONSULTA DE RECINTOS EN EL CATALOGO"
colu2 = .F.
reng = 5
COLU = 20
DO WHILE .NOT. EOF()
@ reng,COLU SAY recinto
SKIP
reng = reng+1
IF reng = 23
reng = 5
DO oprime
SET COLOR TO &col4
@ 5, 7 CLEAR TO 23,78
ENDIF
ENDDO
DO oprime
SET COLOR TO &col4
USE
@ 3, 5 CLEAR TO 22,78
RETURN
```

Procedimiento: CONRNOM

Llamado por: CONRECIN (procedimiento en INBA.PRG)

Llama a: INDICA (procedimiento en INBA.PRG)

: NOEXISTE (procedimiento en INBA.PRG)

: OPRIME (procedimiento en INBA.PRG)

Accesa a: LUGARES.DBF

Indices: LUGRECIN.IDX

```
PROCEDURE conrnom
SELE 1
USE lugares INDEX lugrecin
SET COLOR TO &col4
```

```

@ 1,0 CLEAR TO 24,79
@ 3,24 SAY "CONSULTA DE RECINTOS POR NOMBRE"
DO WHILE .T.
  recintov = SPACE(38)
  DO indica WITH 22
  SET COLOR TO &col4
  @ 6,10 SAY 'Nombre:'
  @ 6,21 GET recintov
  READ
  IF LEN(RTRIM(recintov)) = 0
    EXIT
  ENDIF
  SEEK recintov
  IF .NOT. FOUND()
    DO noexiste
    SET COLOR TO &col4
    @ 6,10 CLEAR TO 21,75
    LOOP
  ENDIF
  @ 8,10 SAY 'Direccion:'
  @ 10,10 SAY 'Colonia:'
  @ 12,10 SAY 'Delegación:'
  @ 14,10 SAY 'Ciudad:'
  @ 16,10 SAY 'C.P.:'
  @ 18,10 SAY 'Lada:'
  @ 20,10 SAY 'Teléfonos:'
  @ 8,21 SAY direccion
  @ 10,21 SAY colonia
  @ 12,21 SAY delegacio
  @ 14,21 SAY ciudad
  @ 16,21 SAY cp
  @ 18,21 SAY lada
  @ 20,21 SAY tell
  @ 21,21 SAY tel2
  DO oprime
  SET COLOR TO &col4
  @ 7,10 CLEAR TO 21,75
ENDDO
USE
@ 3,10 CLEAR TO 21,75
RETURN

```

Procedimiento:	CONRCAL	
Llamado por:	CONRECIN	(procedimiento en INBA.PRG)
Llama a:	PIDEFOL : TRAEREG	(procedimiento en INBA.PRG) (procedimiento en INBA.PRG)
Accesa a:	LUGARES.DBF	


```

PROCEDURE conrcal
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE 1
USE lugares
@ 3,20 SAY "CONSULTA DE RECINTOS POR CALLE"
DO WHILE .T.
  SET COLOR TO &col4
  @ 5,3 CLEAR TO 21,77
  GO TOP
  event = SPACE(40)
  primero = .F.
  @ 5,5 SAY 'Calle:'
  @ 5,17 GET event
  READ
  event = ALLTRIM(event)
  IF LEN(ALLTRIM(event)) = 0
    EXIT
  ENDIF
  @ 7,4 SAY 'REGISTRO'
  @ 7,21 SAY 'RECINTO'
  rengcxd = 9
  DO WHILE .NOT. EOF()
    IF event $ direccion
      @ rengcxd,4 SAY RECNO() PICT '999'
      @ rengcxd,14 SAY recinto
      rengcxd = rengcxd+1
      IF rengcxd >= 24
        rengcxd = 9
        foln = 0
        DO WHILE .T.
          foln = ' '
          DO pidefol WITH foln
          SET COLOR TO &col4
          IF foln = 'I'
            GO TOP
            regresa = .T.
            EXIT
          ENDIF
          IF foln = 'T'
            SELE 1
            USE
            SELE 2
            USE
            SELE 3
            USE
            RETURN
          ENDIF
          foln = VAL(foln)
          IF foln <> 0
            SAVE SCREEN TO pantcxd
            DO tracreg WITH foln
            SET COLOR TO &col4
            RESTORE SCREEN FROM pantcxd

```

```

ELSE
  EXIT
ENDIF
ENDDO
SET COLOR TO &col4
@ 9,4 CLEAR TO 23,77
ENDIF
ENDIF
SKIP
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidedfol WITH foln
    SET COLOR TO &col4
    IF foln = 'T'
      GO TOP
      regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
    ENDIF
    foln = VAL(foln)
    IF foln <> 0
      SAVE SCREEN TO pantcxd
      DO tracereg WITH foln
      SET COLOR TO &col4
      RESTORE SCREEN FROM pantcxd
    ELSE
      primero = .T.
      EXIT
    ENDIF
  ENDDO
ENDIF
IF primero
  SET COLOR TO &col4
  @ 8,0 CLEAR TO 24,79
  GO TOP
  EXIT
ENDIF
ENDDO
foln = 0
SELE 1
USE
RETURN

```

Procedimiento:	CONRCOL	
Llamado por:	CONRECIN	(procedimiento en INBA.PRG)
Llama a:	PIDEFOL : TRAEREG	(procedimiento en INBA.PRG) (procedimiento en INBA.PRG)
Accesa a:	LUGARES.DBF	

```

PROCEDURE conrcol
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE 1
USE lugares
@ 3,24 SAY "CONSULTA DE RECINTOS POR COLONIA"
DO WHILE .T.
    SET COLOR TO &col4
    @ 5,3 CLEAR TO 22,77
    GO TOP
    event = SPACE(40)
    primero = .F.
    @ 5, 5 SAY 'Colonia:'
    @ 5,17 GET event
    READ
    event = ALLTRIM(event)
    IF LEN(ALLTRIM(event)) = 0
        EXIT
    ENDIF
    @ 7, 4 SAY 'REGISTRO'
    @ 7,21 SAY 'RECINTO'
    rengcxd = 9
    DO WHILE .NOT. EOF()
        IF event $ colonia
            @ rengcxd, 4 SAY RECNO() PICT '999'
            @ rengcxd ,14 SAY recinto
            rengcxd = rengcxd+1
            IF rengcxd >= 24
                rengcxd = 9
                foln = 0
                DO WHILE .T.
                    foln = ' '
                    DO pidedfol WITH foln
                    SET COLOR TO &col4
                    IF foln = 'I'
                        GO TOP
                    regressa = .T.
                    EXIT
                ENDIF
                IF foln = 'T'
                    SELE 1
                USE
            
```

```

        SELE 2
        USE
        SELE 3
        USE
        RETURN
    ENDIF
    foln = VAL(foln)
    IF foln <> 0
        SAVE SCREEN TO pantcxd
        DO trereg WITH foln
        SET COLOR TO &col4
        RESTORE SCREEN FROM pantcxd
    ELSE
        EXIT
    ENDIF
ENDDO
SET COLOR TO &col4
@ 9,4 CLEAR TO 23,77
ENDIF
ENDIF
SKIP
IF EOF()
    foln = 0
    DO WHILE .T.
        foln = '
        DO pidefol WITH foln
        SET COLOR TO &col4
        IF foln = 'I'
            GO TOP
            regresa = .T.
        EXIT
        ENDIF
        IF foln = 'T'
            SELE 1
            USE
            SELE 2
            USE
            SELE 3
            USE
            RETURN
        ENDIF
        foln = VAL(foln)
        IF foln <> 0
            SAVE SCREEN TO pantcxd
            DO trereg WITH foln
            SET COLOR TO &col4
            RESTORE SCREEN FROM pantcxd
        ELSE
            primero = .T.
        EXIT
        ENDIF
    ENDDO
ENDIF
IF primero

```

```

SET COLOR TO &col4
@ 8,0 CLEAR TO 24,79
GO TOP
EXIT
ENDIF
ENDDO
ENDDO
foln = 0
SELE I
USE
RETURN

```

Procedimiento:	CONRDEL	
Llamado por:	CONRECIN	(procedimiento en INBA.PRG)
Llama a:	PIDEFOL : TRAEREG	(procedimiento en INBA.PRG) (procedimiento en INBA.PRG)
Accesa a:	LUGARES.DBF	

```

PROCEDURE conrdel
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE I
USE lugares
@ 3,22 SAY "CONSULTA DE RECINTOS POR DELEGACION"
DO WHILE .T.
SET COLOR TO &col4
@ 5,3 CLEAR TO 22,77
GO TOP
event = SPACE(40)
primero = .F.
@ 5,5 SAY 'Delegacion:'
@ 5,17 GET event
READ
event = ALLTRIM(event)
IF LEN(ALLTRIM(event)) = 0
EXIT
ENDIF
@ 7,4 SAY 'REGISTRO'
@ 7,21 SAY 'RECINTO'
rengcxd = 9
DO WHILE .NOT. EOF()
IF event $ delegacio
@ rengcxd,4 SAY RECNO() PICT '999'
@ rengcxd,14 SAY recinto
rengcxd = rengcxd + 1
IF rengcxd >= 24
rengcxd = 9
foln = 0

```

```

DO WHILE .T.
  foln = ' '
  DO pidedol WITH foln
  SET COLOR TO &col4
  IF foln = 'I'
    GO TOP
    regresa = .T.
    EXIT
  ENDIF
  IF foln = 'T'
    SELE 1
    USE
    SELE 2
    USE
    SELE 3
    USE
    RETURN
  ENDIF
  foln = VAL(foln)
  IF foln < 0
    SAVE SCREEN TO pantcxd
    DO traereg WITH foln
    SET COLOR TO &col4
    RESTORE SCREEN FROM pantcxd
  ELSE
    EXIT
  ENDIF
ENDDO
SET COLOR TO &col4
@ 9,4 CLEAR TO 23,77
ENDIF
ENDIF
SKIP
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidedol WITH foln
    SET COLOR TO &col4
    IF foln = 'I'
      GO TOP
      regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
    ENDIF
    foln = VAL(foln)

```

```

IF foln <> 0
  SAVE SCREEN TO pantcxd
  DO traereg WITH foln
  SET COLOR TO &col4
  RESTORE SCREEN FROM pantcxd
ELSE
  primero = .T.
  EXIT
ENDIF
ENDDO
ENDIF
IF primero
  SET COLOR TO &col4
  @ 8,0 CLEAR TO 24,79
  GO TOP
  EXIT
ENDIF
ENDDO
foln = 0
SELE 1
USE
RETURN

```

Procedimiento:	CONRCIU	
Llamado por:	CONRECIN	(procedimiento en INBA.PRG)
Llama a:	PIDEFOL : TRAEREG	(procedimiento en INBA.PRG) (procedimiento en INBA.PRG)
Accesa a:	LUGARES.DBF	

```

PROCEDURE conrciu
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE 1
USE lugares
@ 3,24 SAY "CONSULTA DE RECINTOS POR CIUDAD"
DO WHILE .T.
  SET COLOR TO &col4
  @ 5,3 CLEAR TO 22,77
  GO TOP
  event = SPACE(40)
  primero = .F.
  @ 5,5 SAY 'Ciudad:'
  @ 5,17 GET event
  READ
  event = ALLTRIM(event)
  IF LEN(ALLTRIM(event)) = 0

```

```

EXIT
ENDIF
@ 7, 4 SAY 'REGISTRO'
@ 7,21 SAY 'RECINTO'
rengcxd = 9
DO WHILE .NOT. EOF()
  IF event $ ciudad
    @ rengcxd, 4 SAY RECNO() PICT '999'
    @ rengcxd ,14 SAY recinto
    rengcxd = rengcxd+1
    IF rengcxd >= 24
      rengcxd = 9
      foln = 0
      DO WHILE .T.
        foln = ' '
        DO pidedfol WITH foln
        SET COLOR TO &col4
        IF foln = 'I'
          GO TOP
          regresa = .T.
          EXIT
        ENDIF
        IF foln = 'T'
          SELE 1
          USE
          SELE 2
          USE
          SELE 3
          USE
          RETURN
        ENDIF
        foln = VAL(foln)
        IF foln < 0
          SAVE SCREEN TO pantcxd
          DO traereg WITH foln
          SET COLOR TO &col4
          RESTORE SCREEN FROM pantcxd
        ELSE
          EXIT
        ENDIF
      ENDDO
      SET COLOR TO &col4
      @ 9, 4 CLEAR TO 23,77
    ENDIF
  ENDIF
  SKIP
  IF EOF()
    foln = 0
    DO WHILE .T.
      foln = ' '
      DO pidedfol WITH foln
      SET COLOR TO &col4
      IF foln = 'I'
        GO TOP

```



```

regresa = .T.
EXIT
ENDIF
IF foln = 'T'
  SELE 1
  USE
  SELE 2
  USE
  SELE 3
  USE
  RETURN
ENDIF
foln = VAL(foln)
IF foln <> 0
  SAVE SCREEN TO pantcxd
  DO traereg WITH foln
  SET COLOR TO &col4
  RESTORE SCREEN FROM pantcxd
ELSE
  primero = .T.
  EXIT
ENDIF
ENDDO
ENDIF
IF primero
  SET COLOR TO &col4
  @ 8,0 CLEAR TO 24,79
  GO TOP
  EXIT
ENDIF
ENDDO
ENDDO
foln = 0
SELE 1
USE
RETURN

```

Procedimiento:	CONRCP	
Llamado por:	CONRECIN	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: PIDEFOL	(procedimiento en INBA.PRG)
	: TRAEREG	(procedimiento en INBA.PRG)
Accesa a:	LUGARES.DBF	

```

PROCEDURE conrcp
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79

```

```

SELE 1
USE lugares
@ 3,25 SAY "CONSULTA DE RECINTOS POR C.P."
DO WHILE .T.
  SET COLOR TO &col4
  @ 5,3 CLEAR TO 24,77
  GO TOP
  event = SPACE(5)
  primero = .F.
  @ 5, 5 SAY 'C.P.:'
  DO indica WITH 23
  SET COLOR TO &col4
  @ 5,17 GET event
  READ
  @ 6,0 CLEAR TO 24,79
  event = ALL.TRIM(event)
  IF LEN(ALL.TRIM(event)) = 0
    EXIT
  ENDIF
  @ 7, 4 SAY 'REGISTRO'
  @ 7,21 SAY 'RECINTO'
  rengcxd = 9
  DO WHILE .NOT. EOF()
    IF event $ cp
      @ rengcxd, 4 SAY RECNO() PICT '999'
      @ rengcxd ,14 SAY recinto
      rengcxd = rengcxd+1
      IF rengcxd >= 24
        rengcxd = 9
        foln = 0
        DO WHILE .T.
          foln = ' '
          DO pidefol WITH foln
          SET COLOR TO &col4
          IF foln = 'I'
            GO TOP
            regresa = .T.
            EXIT
          ENDIF
          IF foln = 'T'
            SELE 1
            USE
            SELE 2
            USE
            SELE 3
            USE
            RETURN
          ENDIF
          foln = VAL(foln)
          IF foln <> 0
            SAVE SCREEN TO pantcxd
            DO traereg WITH foln
            SET COLOR TO &col4
            RESTORE SCREEN FROM pantcxd
        
```

```

ELSE
EXIT
ENDIF
ENDDO
SET COLOR TO &col4
@ 9,4 CLEAR TO 23,77
ENDIF
ENDIF
SKIP
IF EOF()
foln = 0
DO WHILE .T.
foln = ' '
DO pidefol WITH foln
SET COLOR TO &col4
IF foln = 'I'
GO TOP
regresa = .T.
EXIT
ENDIF
IF foln = 'T'
SELE 1
USE
SELE 2
USE
SELE 3
USE
RETURN
ENDIF
foln = VAL.(foln)
primero = .F.
IF foln <> 0
SAVE SCREEN TO pantcxd
DO traereg WITH foln
SET COLOR TO &col4
RESTORE SCREEN FROM pantcxd
ELSE
primero = .T.
EXIT
ENDIF
ENDDO
ENDIF
IF primero
SET COLOR TO &col4
@ 8,0 CLEAR TO 24,79
GO TOP
EXIT
ENDIF
ENDDO
ENDDO
foln = 0
SELE 1
USE
RETURN

```

```

ELSE
  EXIT
ENDIF
ENDDO
SET COLOR TO &col4
@ 9, 4 CLEAR TO 23,77
ENDIF
ENDIF
SKIP
IF EOF()
  foln = 0
  DO WHILE .T.
    foln = ' '
    DO pidefol WITH foln
    SET COLOR TO &col4
    IF foln = 'I'
      GO TOP
      regresa = .T.
      EXIT
    ENDIF
    IF foln = 'T'
      SELE 1
      USE
      SELE 2
      USE
      SELE 3
      USE
      RETURN
    ENDIF
    foln = VAL(foln)
    primero = .F.
    IF foln <> 0
      SAVE SCREEN TO pantcxd
      DO traereg WITH foln
      SET COLOR TO &col4
      RESTORE SCREEN FROM pantcxd
    ELSE
      primero = .T.
      EXIT
    ENDIF
  ENDDO
ENDIF
IF primero
  SET COLOR TO &col4
  @ 8,0 CLEAR TO 24,79
  GO TOP
  EXIT
ENDIF
ENDDO
ENDDO
foln = 0
SELE 1
USE
RETURN

```

Procedimiento:	CAMRECIN	
Llamado por:	RECINTO	(procedimiento en INBA.PRG)
Llama a:	INDICA	(procedimiento en INBA.PRG)
	: NOEXISTE	(procedimiento en INBA.PRG)
	: SINO	(procedimiento en INBA.PRG)
Accesa a:	LUGARES.DBF	
Indices:	LUGRECIN.IDX	

```

PROCEDURE camrecin
SELE 1
USE lugares INDEX lugrecin
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
@ 3,30 SAY "CAMBIOS A RECINTOS"
DO WHILE .T.
  recintov = SPACE(38)
  @ 7,10 SAY "Nombre:"
  DO indica WITH 24
  SET COLOR TO &col4
  @ 7,21 GET recintov
  READ
  IF LEN(RTRIM(recintov)) = 0
    EXIT
  ENDF
  SEEK recintov
  IF NOT FOUND()
    DO noexiste
    SET COLOR TO &col4
    @ 7,10 CLEAR TO 24,79
  LOOP
  ENDF
  DO indica WITH 26
  SET COLOR TO &col4
  recintov = recinto
  direccionv = direccion
  coloniav = colonia
  delegaciov = delegacio
  ciudadv = ciudad
  cpv = cp
  ladav = lada
  tel1v = tel1
  tel2v = tel2
  @ 9,10 SAY "Direccion:"
  @ 11,10 SAY "Colonia:"
  @ 13,10 SAY "Delegacion:"
  @ 15,10 SAY "Ciudad:"
  @ 17,10 SAY "C.P.:"

```

```

@ 19,10 SAY 'Lada:'
@ 21,10 SAY 'Telefonos:'
@ 7,21 GET recintov
@ 9,21 GET direccionv
@ 11,21 GET colonia v
@ 13,21 GET delegacio v
@ 15,21 GET ciudadv
@ 17,21 GET cpv PICT REPL('9',5)
@ 19,21 GET lada v PICT '999'
@ 21,21 GET tel1 v PICT '999-99-99'
@ 22,21 GET tel2 v PICT '999-99-99'
READ
@ 24,0 CLEAR TO 24,79
resp = ''
ask = 'ESTAN BIEN LOS DATOS'
DO sino WITH: resp,ask
SET COLOR TO &col4
IF resp = 'N'
  @ 7,10 CLEAR TO 24,79
  LOOP
ENDIF
REPLACE recinto WITH recintov
REPLACE direccion WITH direccionv
REPLACE colonia WITH colonia v
REPLACE delegacio WITH delegacio v
REPLACE ciudad WITH ciudadv
REPLACE cp WITH cpv
REPLACE lada WITH lada v
REPLACE tel1 WITH tel1 v
REPLACE tel2 WITH tel2 v
@ 7,10 CLEAR TO 24,79
LOOP
ENDDO
USE
@ 3,10 CLEAR TO 22,75
RETURN

```

Procedimiento: BAJRECIN

Llamado por: RECINTO (procedimiento en INBA.PRG)

Llama a: INDICA (procedimiento en INBA.PRG)
: NOEXISTE (procedimiento en INBA.PRG)
: SINO (procedimiento en INBA.PRG)
: OPRIME (procedimiento en INBA.PRG)

Accesa a: LUGARES.DBF

Indices: LUGRECIN.IDX

PROCEDURE bajrecin

```

SET COLOR TO &col4
SELE 1
USE lugares INDEX lugrecin
@ 1,0 CLEAR TO 24,79
@ 3,32 SAY "BAJA DE RECINTOS"
DO WHILE .T.
    recintov = SPACE(38)
    @ 6,10 SAY 'Nombre:'
    DO indica WITH 25
    SET COLOR TO &col4
    @ 6,21 GET recintov
    READ
    IF LEN(RTRIM(recintov)) = 0
        EXIT
    ENDIF
    SEEK recintov
    IF .NOT. FOUND()
        DO noexiste
        SET COLOR TO &col4
        @ 6,10 CLEAR TO 21,75
        LOOP
    ENDIF
    @ 24,0 CLEAR TO 24,79
    @ 8,10 SAY 'Direccion:'
    @ 10,10 SAY 'Colonia:'
    @ 12,10 SAY 'Delegacion:'
    @ 14,10 SAY 'Ciudad:'
    @ 16,10 SAY 'C.P.:'
    @ 18,10 SAY 'Lada:'
    @ 20,10 SAY 'Telefonos:'
    @ 8,21 SAY direccion
    @ 10,21 SAY colonia
    @ 12,21 SAY delegacio
    @ 14,21 SAY ciudad
    @ 16,21 SAY cp
    @ 18,21 SAY lada
    @ 20,21 SAY tel1
    @ 21,21 SAY tel2
    resp = ''
    ask = 'DESEA DARLO DE BAJA'
    DO sino WITH resp,ask
    SET COLOR TO &col4
    IF resp = 'S'
        DELETE
        PACK
        REINDEX
    ENDIF
    @ 23,2 SAY SPACE(77)
    DO oprime
    SET COLOR TO &col4
    @ 7,10 CLEAR TO 24,79
ENDDO
USE
@ 3,10 CLEAR TO 21,75

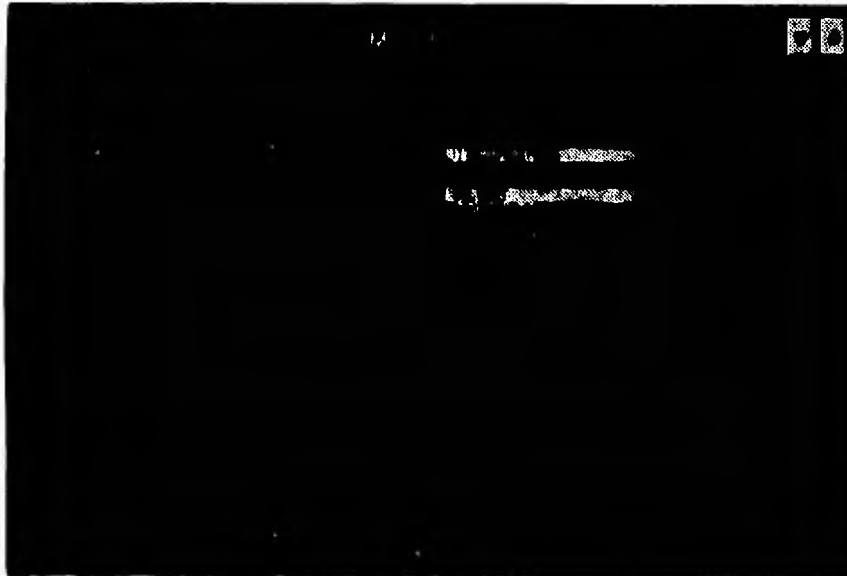
```

RETURN

Procedimiento:	REPRECIN	
Llamado por:	RECINTO	(procedimiento en INBA.PRG)
Llama a:	SINO	(procedimiento en INBA.PRG)
	: IMPRESOR	(procedimiento en INBA.PRG)
	: ISPRINTER()	(función de Clipper)
	: IMPRESORA()	(función en INBA.PRG)
Accesa a:	LUGARES.DBF	
Indices:	LUGRECIN.IDX	
Reportes:	LUGARES.FRX	

```
PROCEDURE reprecin
SELE 1
USE lugares INDEX lugrecin
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
@ 1,30 SAY "REPORTE DE RECINTOS"
resp = ''
ask = 'Desea imprimir el reporte general de lugares ?'
DO sino WITH resp,ask
SET COLOR TO &col4
IF resp = 'S'
DO impresor
IF .NOT. Isprinter()
DO impresora
IF !Isprinter()
RETURN
ENDIF
ENDIF
@ 23,20 SAY 'PREPARANDO ARCHIVOS, ESPERE POR FAVOR'
@ 23,57 SAY '...'
SET CONSOLE OFF
SET DEVICE TO PRINT
@ 0,0 SAY CHR(027)+'&1000'
@ 0,0 SAY CHR(027)+'(c#17H'
@ 0,0 SAY CHR(027)+'(10U'
REPORT FORM lugares TO PRINT
ENDIF
SET CONSOLE ON
SET DEVICE TO SCREEN
@ 3,3 CLEAR TO 23,78
RETURN
```


MENU DISCIPLINAS



Procedimiento: DISCIPL

Llamado por: INBA.PRG

Llama a: CAPDISCI (procedimiento en INBAPROC.PRG)
: CONDISCI (procedimiento en INBAPROC.PRG)
: CAMDISCI (procedimiento en INBAPROC.PRG)
: REPDISCI (procedimiento en INBAPROC.PRG)

```
PROCEDURE discipl
SAVE SCREEN TO pantalla[20]
SET COLOR TO &col1
@ 6,0 CLEAR TO 22,79
level1 = 1      && menuvar for 1st menu
DO WHILE (level1 = 0)
SET COLOR TO &col4
@ 5,40 TO 10,60
@ 5,40 SAY 'A'
@ 5,60 SAY 'A'
SET COLOR TO &col8
```

```

@ 6,41 PROMPT " ALTAS      " MESSAGE '          CAPTURA DE NOMBRES DE
DISCIPLINAS
@ 7,41 PROMPT " CONSULTA  " MESSAGE '          CONSULTA DEL CATALOGO DE
DISCIPLINAS
@ 8,41 PROMPT " MODIFICACION " MESSAGE '        MODIFICACION A LOS NOMBRES
DE DISCIPLINAS
@ 9,41 PROMPT " REPORTES   " MESSAGE '        IMPRESION DEL CATALOGO DE
DISCIPLINAS
MENU TO level1
IF (level1 = 0)      && esc to exit
EXIT
ENDIF
DO CASE
CASE level1 = 1
SAVE SCREEN TO pantalla [2]
DO capdisci
RESTORE SCREEN FROM pantalla[2]
CASE level1 = 2
SAVE SCREEN TO pantalla [2]
DO condisci
RESTORE SCREEN FROM pantalla[2]
CASE level1 = 3
SAVE SCREEN TO pantalla [2]
DO camdisci
RESTORE SCREEN FROM pantalla[2]
CASE level1 = 4
SAVE SCREEN TO pantalla [2]
DO repdisci
RESTORE SCREEN FROM pantalla[2]
ENDCASE
ENDDO
RESTORE SCREEN FROM pantalla[20]
level1 = 3
RETURN (level1)

```

Procedimiento:	CAPDISCI	
Llamado por:	DISCIPL	(procedimiento en INBA.PRG)
Llama a:	CDIS2	(procedimiento en INBAPROC.PRG)
Accesa a:	DISCIPL.DBF	
Indices:	NUMDISCI.IDX : NOMBDIS.IDX	

```

PROCEDURE capdisci
SELE I
USE discipl INDEX numdisci,nombdis
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
@ 3,29 SAY "CAPTURA DE DISCIPLINAS"

```

```

numdv = 0
plinav = SPACE(30)
DO cdis2 WITH numdv,plinav
USE
RETURN

```

Procedimiento:	CONDISCI	
Llamado por:	DISCIPL	(procedimiento en INBA.PRG)
Llama a:	OPRIME	(procedimiento en INBA.PRG)
Accesa a:	DISCIPL.DBF	
Indices:	NUMDISCI.IDX : NOMBDIS.IDX	

```

PROCEDURE condisci
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
SELE 1
USE discipl INDEX numdisci,nombdis
@ 3,28 SAY "CONSULTA DE DISCIPLINAS"
colu2 = .F.
reng = 5
COLU = 7
DO WHILE .NOT. EOF()
@ reng,COLU SAY disciplina
SKIP
reng = reng+1
IF reng = 22 .AND. .NOT. colu2
COLU = 43
reng = 5
colu2 = .T.
ENDIF
IF reng = 22 .AND. colu2
COLU = 7
reng = 5
colu2 = .F.
DO oprime
SET COLOR TO &col4
@ 5,7 CLEAR TO 23,77
ENDIF
ENDDO
DO oprime
SET COLOR TO &col4
USE
@ 3,5 CLEAR TO 22,77
RETURN

```

Procedimiento:	CAMDISCI	
Llamado por:	DISCIPL	(procedimiento en INBA.PRG)
Llama a:	NOEXISTE : SINO	(procedimiento en INBA.PRG) (procedimiento en INBA.PRG)
Accesa a:	DISCIPL.DBF	
Indices:	NUMDISCI.IDX : NOMBDIS.IDX	

```

PROCEDURE camdisci
SELE 1
USE discipl INDEX numdisci,nombdis
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
@ 3,29 SAY "CAMBIOS A DISCIPLINAS"
DO WHILE .T.
    plinav = SPACE(30)
    @ 7,10 SAY 'Nombre de la disciplina:'
    @ 7,36 GET plinav
    READ
    IF LEN(RTRIM(plinav)) = 0
        EXIT
    ENDIF
    SET ORDER TO 2
    SEEK plinav
    IF .NOT. FOUND()
        DO noexiste
        LOOP
    ENDIF
    newname = disciplina
    @ 13,16 SAY 'REALIZE LA CORRECCION Y OPRIMA '+CHR(17)+CHR(217)+' PARA CONTINUAR'
    @ 7,36 GET newname
    READ
    resp = ''
    @ 13,16 SAY SPACE(60)
    ask = 'ESTA BIEN EL DATO'
    DO sino WITH resp,ask
    SET COLOR TO &col4
    @ 23,0 CLEAR TO 24,79
    IF resp = 'N'
        @ 7, 5 CLEAR TO 22,75
        LOOP
    ENDIF
    REPLACE disciplina WITH newname
ENDIF
ENDDO
USE
@ 3, 5 CLEAR TO 22,76
RETURN

```

Procedimiento:	REPDISCI	
Llamado por:	DISCIPL	(procedimiento en INBA.PRG)
Llama a:	SINO	(procedimiento en INBA.PRG)
	: IMPRESOR	(procedimiento en INBA.PRG)
	: ISPRINTER()	(función de Clipper)
	: IMPRESORA()	(función en INBA.PRG)
Accesa a:	DISCIPL.DBF	
Indices:	NOMBDIS.IDX	
Report Forms:	DISCIPL.FRX	

```

PROCEDURE repdisci
SELE I
USE discipl INDEX nombdis
SET COLOR TO &col4
@ 1,0 CLEAR TO 24,79
@ 3,29 SAY "REPORTE DE DISCIPLINAS"
resp = ''
ask = 'Desea imprimir el reporte de disciplinas ? '
DO sino WITH resp,ask
SET COLOR TO &col4
@ 23,0 CLEAR TO 24,79
IF resp = 'S'
DO impresor
IF .NOT. !sprinter()
DO impresora
IF !sprinter()
RETURN
ENDIF
ENDIF
REPORT FORM discipl TO PRINT
ENDIF
@ 3,3 CLEAR TO 23,78
RETURN

```

RUTINAS DE USO GENERAL

Procedimiento: EXISTE

Llamado por: CDIS2 (procedimiento en INBAPROC.PRG)
: CREC2 (procedimiento en INBA.PRG)

```
PROCEDURE existe
SET COLOR TO &col14
@ 23,16 SAY 'ESE DATO YA EXISTE, OPRIMA '+CHR(17)+CHR(217)+' PARA CONTINUAR '
SET CURSOR OFF
resp = ''
@ 23,62 GET resp
READ
SET COLOR TO &col1
@ 23,16 SAY SPACE(55)
SET CURSOR ON
RETURN
```

Procedimiento: INDICA

Llamado por: INBA.PRG

: CXDYF	(procedimiento en INBAPROC.PRG)
: CXDYL	(procedimiento en INBAPROC.PRG)
: CXDISC	(procedimiento en INBAPROC.PRG)
: CXACT	(procedimiento en INBAPROC.PRG)
: CXARTIS	(procedimiento en INBAPROC.PRG)
: CXLUGAR	(procedimiento en INBAPROC.PRG)
: CXFECHA	(procedimiento en INBAPROC.PRG)
: CXHORA	(procedimiento en INBAPROC.PRG)
: CAPPROG	(procedimiento en INBAPROC.PRG)
: CAMPROG	(procedimiento en INBAPROC.PRG)
: CAMBIAR	(procedimiento en INBAPROC.PRG)
: PIDEFOL	(procedimiento en INBA.PRG)
: CAMRECIN	(procedimiento en INBA.PRG)
: CONRCP	(procedimiento en INBA.PRG)
: BAJRECIN	(procedimiento en INBA.PRG)
: CONRNOM	(procedimiento en INBA.PRG)

```
PROCEDURE Indica
PARAMETERS numsg
SET COLOR TO &col8
mensaje= SPACE((80-LEN(ALLTRIM(mssg[NUMSG])))/2)+mssg[NUMSG]
espacios = SPACE(80-LEN(SPACE((80-LEN(ALLTRIM(mssg[NUMSG])))/2)+mssg[NUMSG]))
@ 24, 0 SAY mensaje+espacios
RETURN
```

Procedimiento: NOEXISTE

Llamado por:	REPPGMEN	(procedimiento en INBAPROC.PRG)
	: CXDYF	(procedimiento en INBAPROC.PRG)
	: CXDYL	(procedimiento en INBAPROC.PRG)
	: CXDISC	(procedimiento en INBAPROC.PRG)
	: CAPPROG	(procedimiento en INBAPROC.PRG)
	: CAMBIAR	(procedimiento en INBAPROC.PRG)
	: CAMDISC	(procedimiento en INBAPROC.PRG)
	: CAMRECIN	(procedimiento en INBA.PRG)
	: BAJRECIN	(procedimiento en INBA.PRG)
	: CONRNOM	(procedimiento en INBA.PRG)

```
PROCEDURE noexiste
SET COLOR TO &col14
@ 23,16 SAY 'ESE DATO NO EXISTE, OPRIMA '+CHR(17)+CHR(217)+' PARA CONTINUAR'
resp = ''
SET CURSOR OFF
@ 23,62 GET resp
READ
SET COLOR TO &col1
@ 23,16 SAY SPACE(55)
SET CURSOR ON
RETURN
```

Procedimiento: OPRIME

Llamado por:	REPPGSEM	(procedimiento en INBAPROC.PRG)
	: ELIMENS	(procedimiento en INBAPROC.PRG)
	: RESTOMES	(procedimiento en INBAPROC.PRG)
	: CXDYF	(procedimiento en INBAPROC.PRG)
	: CXFECHA	(procedimiento en INBAPROC.PRG)
	: CXHORA	(procedimiento en INBAPROC.PRG)
	: TRAEACT	(procedimiento en INBAPROC.PRG)
	: CAPPROG	(procedimiento en INBAPROC.PRG)
	: CAMBIAR	(procedimiento en INBAPROC.PRG)
	: CONDISC	(procedimiento en INBAPROC.PRG)
	: IMPRESOR	(procedimiento en INBA.PRG)
	: TRAEREG	(procedimiento en INBA.PRG)
	: BAJRECIN	(procedimiento en INBA.PRG)
	: CONRALL	(procedimiento en INBA.PRG)
	: CONRNOM	(procedimiento en INBA.PRG)

```
PROCEDURE oprime
SET CURSOR OFF
SET COLOR TO &col14
@ 23,27 SAY 'OPRIMA '+CHR(17)+CHR(217)+' PARA CONTINUAR'
resp = ''
@ 23,52 GET resp
```

READ
SET COLOR TO &col4
@ 23,0 CLEAR TO 23,79
SET CURSOR ON
RETURN

Procedimiento: PIDEFOL

Llamado por:	ELIMACT	(procedimiento en INBAPROC.PRG)
	: CXDYF	(procedimiento en INBAPROC.PRG)
	: CXDYL	(procedimiento en INBAPROC.PRG)
	: CXDISC	(procedimiento en INBAPROC.PRG)
	: CXACT	(procedimiento en INBAPROC.PRG)
	: CXARTIS	(procedimiento en INBAPROC.PRG)
	: CXLUGAR	(procedimiento en INBAPROC.PRG)
	: CXFECHA	(procedimiento en INBAPROC.PRG)
	: CXHORA	(procedimiento en INBAPROC.PRG)
	: CAMPROG	(procedimiento en INBAPROC.PRG)
	: CONRCP	(procedimiento en INBA.PRG)
	: CONRDEL	(procedimiento en INBA.PRG)
	: CONRCIU	(procedimiento en INBA.PRG)
	: CONRCAL	(procedimiento en INBA.PRG)
	: CONRCOL	(procedimiento en INBA.PRG)

Llama a: INDICA (procedimiento en INBA.PRG)

PROCEDURE pidefol
PARAMETERS foln
DO indica WITH 14
@ 24,23 GET foln
READ
@ 24, 0 SAY SPACE(79)
RETURN

Procedimiento: SINO

Llamado por:	ELIMENS	(procedimiento en INBAPROC.PRG)
	: RESTOMES	(procedimiento en INBAPROC.PRG)
	: BORRACT	(procedimiento en INBAPROC.PRG)
	: CAPPROG	(procedimiento en INBAPROC.PRG)
	: CDIS2	(procedimiento en INBAPROC.PRG)
	: CAMDISCI	(procedimiento en INBAPROC.PRG)
	: REPDISCI	(procedimiento en INBAPROC.PRG)
	: IMPRESORA()	(función en INBA.PRG)
	: CAMRECIN	(procedimiento en INBA.PRG)
	: REPRECIN	(procedimiento en INBA.PRG)
	: BAJRECIN	(procedimiento en INBA.PRG)
	: CREC2	(procedimiento en INBA.PRG)

```
PROCEDURE sino
PARAMETERS resp,pregunta
decision=' '
DO WHILE (resp #'S') .AND. (resp #'N')
  SET COLOR TO &col2
  @ 23,77-13-LEN(pregunta) SAY ""+pregunta+" [S/N] ?"
  SET COLOR TO &despliegue
  @ 23,74 GET resp PICT " "
  READ
ENDDO
SET COLOR TO &col1
@ 23,77-13-LEN(pregunta) SAY SPACE(LEN(pregunta)+13)
RETURN
```

Procedimiento: WINDOW

Llamado por:	ELIMINA	(procedimiento en INBA.PRG)
	: CONPROG	(procedimiento en INBA.PRG)
	: REPPROG	(procedimiento en INBA.PRG)
	: CONRECIN	(procedimiento en INBA.PRG)

```
PROCEDURE WINDOW
PARAMETERS COLOR,r1,c1,r2,c2,texto
SET COLOR TO &color
@ r1,c1,r2,c2 BOX marco3
SET COLOR TO &col3
@r1,(c2-c1-LEN(texto))/2+c1 SAY texto
SET COLOR TO &color
RETURN
```

RUTINAS DE USO GENERAL DEL MODULO DE PROGRAMACION

Procedimiento: **DIASEM**

Llamado por: **REPPGMEN**
: **REPPGSEM**
: **TRAEACT**

(procedimiento en INBAPROC.PRG)

(procedimiento en INBAPROC.PRG)

(procedimiento en INBAPROC.PRG)

```
PROCEDURE diasem
PARAMETERS fechax,nombred,dia,nmes
nd = DOW(fechax)
dia = DAY(fechax)
mcs = MONTH(fechax)
DO CASE
CASE nd = 1
nombred = 'DOMINGO'
CASE nd = 2
nombred = 'LUNES'
CASE nd = 3
nombred = 'MARTES'
CASE nd = 4
nombred = 'MIERCOLES'
CASE nd = 5
nombred = 'JUEVES'
CASE nd = 6
nombred = 'VIERNES'
CASE nd = 7
nombred = 'SABADO'
ENDCASE
DO CASE
CASE nmes = 1
nmes = 'ENERO'
CASE mes = 2
nmes = 'FEBRERO'
CASE mes = 3
nmes = 'MARZO'
CASE mes = 4
nmes = 'ABRIL'
CASE mes = 5
nmes = 'MAYO'
CASE mes = 6
nmes = 'JUNIO'
CASE mes = 7
nmes = 'JULIO'
CASE mes = 8
nmes = 'AGOSTO'
CASE mes = 9
nmes = 'SEPTIEMBRE'
CASE mes = 10
nmes = 'OCTUBRE'
CASE mes = 11
```

```
nmes = 'NOVIEMBRE'  
CASE mes = 12  
nmes = 'DICIEMBRE'  
ENDCASE  
RETURN
```

Procedimiento: HEADER

Llamado por: REPPGMEN (procedimiento en INBAPROC.PRG)
: REPPGSEM (procedimiento en INBAPROC.PRG)

PROCEDURE HEADER

```
PARAMETERS tiporepo pagina,namedisci,periodo  
texto1 = 'INSTITUTO NACIONAL DE BELLAS ARTES'  
texto2 = 'SUBDIRECCION DE RELACIONES PUBLICAS'  
texto3 = 'PROGRAMACION '+tiporepo  
pagina = pagina+1  
fecha = DATE()  
@ 0,166 SAY fecha  
IF tiporepo <> 'MENSUAL'  
@ 1,164 SAY 'PAGINA'  
@ 1,171 SAY pagina PICT '999'  
ENDIF  
@ 2,86-(LEN(ALLTRIM(texto3)))/2 SAY texto3  
@ 3,86-(LEN(ALLTRIM(namedisci)))/2 SAY namedisci  
@ 4,86-(LEN(ALLTRIM(periodo)))/2 SAY periodo  
RETURN
```

Procedimiento: TRAEACT

Llamado por: CXDYF (procedimiento en INBAPROC.PRG)
: CXDYL (procedimiento en INBAPROC.PRG)
: CXDISC (procedimiento en INBAPROC.PRG)
: CXACT (procedimiento en INBAPROC.PRG)
: CXARTIS (procedimiento en INBAPROC.PRG)
: CXLUGAR (procedimiento en INBAPROC.PRG)
: CXFECHA (procedimiento en INBAPROC.PRG)
: CXHORA (procedimiento en INBAPROC.PRG)

Llama a: OPRIME (procedimiento en INBA.PRG)
: DIASEM (procedimiento en INBA.PRG)

Accesa a: HORARIOS.DBF
: ARTISTA.DBF
: OBSERVA.DBF
: COSTDESC.DBF

Indices: FOLHORA.IDX
: FOLARTIS.IDX
: FOLOBS.IDX
: FOLCOST.IDX

PROCEDURE traect

PARAMETERS foln

@ 6, 4 CLEAR TO 23,77

SELE 3

GO TOP

SEEK foln

SELE 2

SEEK foln

nd = disciplina

SELE 1

SET ORDER TO 1

SEEK nd

NOMD = disciplina

IF NOMD = 'CURSOS Y TALLERES'

cyt = .T.

ELSE

cyt = .F.

ENDIF

SET ORDER TO 2

SELE 3

reng = 6

DO WHILE folio = foln

@ 6, 5 SAY 'Actividad:'

@ reng .17 SAY SUBS(actividad,1,60)

@ reng+1,17 SAY SUBS(actividad,61,60)

@ reng+2,17 SAY SUBS(actividad,121,60)

@ reng+3,17 SAY SUBS(actividad,181,60)

```

reng = reng+4
IF reng+4 > 23
  reng = 6
  DO oprime
  @ 6, 4 CLEAR TO 23,77
ENDIF
SKIP
ENDDO
IF reng >= 19
  DO oprime
  @ 6, 4 CLEAR TO 23,77
  reng = 6
ENDIF
SELE 5
USE
USE horarios INDEX folhora
SEEK foln
renghora = reng
@ renghora, 5 SAY 'Horarios:'
renghora = renghora+1
IF .NOT. cyt
  DO WHILE folio = foln
    fechax = fecha
    nombred = ''
    diames = 0
    nmes = ''
    DO diasem WITH fechax,nombred,diames,nmes
    @ renghora,7 SAY nombred+' DE '+nmes
    @ renghora,8+LEN(nombred) SAY diames PICT '99'
    @ renghora,34 SAY hora
    @ renghora,40 SAY especial
    renghora = renghora+1
    IF renghora >= 23
      DO oprime
      renghora = reng
      @ renghora, 4 CLEAR TO 23,78
    ENDIF
  SKIP
  ENDDO
ELSE
  DO WHILE folio = foln
    fechax = fecha
    nombred = ''
    diames = 0
    nmes = ''
    DO diasem WITH fechax,nombred,diames,nmes
    @ renghora,3 SAY SUBS(nombred,1,3)+' DE '+SUBS(nmes,1,3)+' A'
    @ renghora,7 SAY diames PICT '99'
    @ renghora,35 SAY 'DE '+hora +' A'
    fechax = fechafin
    nombred = ''
    diames = 0
    nmes = ''
    DO diasem WITH fechax,nombred,diames,nmes

```

```

@ reghora,19 SAY SUBS(nombred,1,3)+' DE '+SUBS(nmes,1,3)
@ reghora,23 SAY diames PICT '99'
@ reghora,46 SAY horafin
@ reghora,53 SAY especial
reghora = reghora+1
IF reghora >=23
  DO oprime
  reghora = reng
  @ reghora, 3 CLEAR TO 23,78
ENDIF
SKIP
ENDDO
ENDIF
reng = reghora+1
DO oprime
SELE 4
USE
USE artista INDEX folartis
SEEK foln
@ 6, 3 CLEAR TO 23,78
reng = 6
rengart = reng
DO WHILE folio = foln
  @ rengart, 5 SAY 'Artista:'
  @ reng ,16 SAY SUBS(artista,1,60)
  @ reng+1,16 SAY SUBS(artista,61,60)
  @ reng+2,16 SAY SUBS(artista,121,60)
  @ reng+3,16 SAY SUBS(artista,181,60)
  reng = reng+4
  IF reng+4 > 23
    reng = 6
    DO oprime
    @ 6, 4 CLEAR TO 23,78
  ENDIF
  SKIP
ENDDO
IF reng >=19
  DO oprime
  @ 6, 4 CLEAR TO 23,78
  reng = 6
ENDIF
SELE 2
SEEK foln
@ reng, 5 SAY 'Lugar:'
@ reng,17 SAY lugar
reng = reng+1
@ reng, 5 SAY 'Horario de taquilla:'
@ reng,26 SAY horataqui
reng = reng+1
IF reng >=19
  DO oprime
  @ 6, 4 CLEAR TO 23,78
  reng = 6
ENDIF

```

```

SELE 4
USE
USE observa INDEX folobs
SEEK foln
rengob = reng
DO WHILE folio = foln
  @ rengob, 5 SAY 'Observ:.'
  @ reng ,16 SAY SUBS(observa,1,60)
  @ reng+1,16 SAY SUBS(observa,61,60)
  @ reng+2,16 SAY SUBS(observa,121,60)
  @ reng+3,16 SAY SUBS(observa,181,60)
  reng = reng+4
  IF reng+4 > 23
    reng = 6
    rengob = reng
    DO oprime
    @ 6, 4 CLEAR TO 23,78
  ENDIF
SKIP
ENDDO
IF reng >= 19
  DO oprime
  @ 6, 4 CLEAR TO 23,78
  reng = 6
ENDIF
SELE 4
USE
USE costdesc INDEX folcost
SEEK foln
rengcost = reng
DO WHILE folio = foln
  @ rengcost, 5 SAY 'Precio: NS'
  @ rengcost,16 SAY costo PICT '999.99'
  IF desccosto <> 0
    @ rengcost,24 SAY 'Descuento: %'
    @ rengcost,35 SAY descosto PICT '99'
  ENDIF
  IF LEN(TRIM(leyenda)) <> 0
    @ rengcost,40 SAY 'A:'
    @ rengcost,43 SAY leyenda
  ENDIF
  rengcost = rengcost+1
  IF rengcost >= 23
    rengcost = reng
    DO oprime
    @ 6, 4 CLEAR TO 23,77
  ENDIF
SKIP
ENDDO
DO oprime
SELE 4
USE
SELE 5
USE ; RETURN

```

RUTINAS DE USO GENERAL DEL MODULO DE RECINTOS

Procedimiento:	TRAEREG	
Llamado por:	CONRCP	(procedimiento en INBA.PRG)
	: CONRDEL	(procedimiento en INBA.PRG)
	: CONRCIU	(procedimiento en INBA.PRG)
	: CONRCAL	(procedimiento en INBA.PRG)
	: CONRCOL	(procedimiento en INBA.PRG)
Llama a:	OPRIME	(procedimiento en INBA.PRG)

```
PROCEDURE traereg  
PARAMETERS foln  
GOTO foln  
SET COLOR TO &col4  
@ 5,4 CLEAR TO 23,77  
@ 6,10 SAY 'Nombre:'  
@ 8,10 SAY 'Direccion:'  
@ 10,10 SAY 'Colonia:'  
@ 12,10 SAY 'Delegacion:'  
@ 14,10 SAY 'Ciudad:'  
@ 16,10 SAY 'C.P.:'  
@ 18,10 SAY 'Lada:'  
@ 20,10 SAY 'Telefonos:'  
@ 6,21 SAY recinto  
@ 8,21 SAY direccion  
@ 10,21 SAY colonia  
@ 12,21 SAY delegacio  
@ 14,21 SAY ciudad  
@ 16,21 SAY cp  
@ 18,21 SAY lada  
@ 20,21 SAY tel1  
@ 21,21 SAY tel2  
DO oprime  
SET COLOR TO &col4  
RETURN
```


RUTINAS DE IMPRESION

Procedimiento: IMPRESOR

Llamado por: REPPGMEN (procedimiento en INBAPROC.PRG)
: REPPGSEM (procedimiento en INBAPROC.PRG)
: REPDISCI (procedimiento en INBAPROC.PRG)
: REPRECIN (procedimiento en INBA.PRG)

Llama a: TONO (procedimiento en INBA.PRG)
: OPRIME (procedimiento en INBA.PRG)

```
PROCEDURE Impresor
DO tono
SET COLOR TO &col2
@ 22,30 SAY 'PREPARE SU IMPRESORA'
DO oprime
SET COLOR TO &col1
@ 22,30 SAY SPACE(40)
RETURN
```

Function: IMPRESORA

Llamado por: REPPGMEN (procedimiento en INBAPROC.PRG)
: REPPGSEM (procedimiento en INBAPROC.PRG)
: REPDISCI (procedimiento en INBAPROC.PRG)
: REPRECIN (procedimiento en INBA.PRG)

Llama a: ISPRINTER() (función de Clipper)
: TONO (procedimiento en INBA.PRG)
: SINO (procedimiento en INBA.PRG)

```
FUNCTION Impresora
@ 22,30 SAY SPACE(45)
DO WHILE !isprinter()
x = ''
tono()
resp = ''
ask = 'LA IMPRESORA NO ESTA LISTA, DESEA REINTENTAR'
DO sino WITH resp,ask
IF resp = 'S'
LOOP
ELSE
EXIT
ENDIF
ENDDO
RETURN isprinter()
```

Procedimiento: TONO

Llamado por: IMPRESOR (procedimiento en INBA.PRG)
: IMPRESORA() (función en INBA.PRG)

Llama a: TONE() (función de Clipper)

```
PROCEDURE tono  
tone(300,1)  
tone(499,5)  
tone(700,5)  
RETURN
```

III.3.2 PRUEBAS MODULARES.

A continuación se describen las matrices de pruebas que se elaboraron para validar el correcto funcionamiento de todos los módulos del sistema, incluyendo los menús de opciones.

PRUEBAS PARA MENUS.

La matriz que a continuación se describe y se muestra en la figura III.3.2.A, se aplicó a todos los menús del sistema, razón por la cual es muy genérica, sin embargo, se contemplan todos los puntos a validar para cada uno de los menús.

1. Información general

Deberá aparecer la información general del sistema, es decir:

- Nombre del sistema
- Fecha
- Menú de opciones
- Instrucciones de como utilizar dicho menú

Además de aparecer esta información, no debe contener errores de escritura, tales como omisión de letras o errores de ortografía; este último requerimiento, debe contemplarse para todos los demás puntos.

2. Nombre del menú

En la parte superior y en forma centrada, debe aparecer el nombre del menú en el que se encuentra posicionado el sistema.

PRUEBAS MENUS

Referencia	Campo a probar	Principal	Programación	Programación Consultas	Programación Reportes	Programación Estructuración	Redes y Comunicación	Redes y Comunicación	Disciplinas	Indicador (Puntaje)
1	Nombre del sistema	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
1	Fecha	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
1	Menú de opciones	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
1	Instrucciones de uso del menú	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
1	Ortografía	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
2	Nombre del menú	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Opciones del menú	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
4	1 hacia arriba	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
4	1 hacia abajo	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
4	hacia arriba	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
4	hacia abajo	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
4	Simulación fila circular	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
5	Teclas letra	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
6	Cursor	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
7	Esc	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
8	Descripción de selección	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple

FIGURA III.3.2A MATRIZ DE PRUEBAS PARA MENUS

3. Opciones del menú

En forma centrada en la pantalla deberá aparecer cada una de las posibles opciones del menú, y al seleccionarlás deberá ceder el control al módulo del programa correspondiente para dicha opción.

4. Teclas de flecha

Las teclas del cursor, es decir las flechas tanto de movimientos verticales como horizontales, deberán desplazar la barra de selección del menú, de la siguiente manera:

- **Hacia arriba**
- **Hacia abajo**
- **Hacia arriba**
- **Hacia abajo**
- **Cuando la barra de selección, se encuentre en la primera o en la última posición, y se oprima la flecha para llegar a la posición superior o inferior respectivamente, la barra de selección debe posicionarse en la última opción para el primer caso, y en la primera opción para el segundo caso; simulando una fila circular.**

5. Teclas de letra

Al oprimir la tecla de la primera letra de alguna de las opciones del menú, la barra de selección, deberá desplazarse a la opción que empiece con dicha letra.

6. Cursor

El cursor, es decir, la barra de selección, deberá tener el mismo tamaño en cada una de las opciones del menú.

7. Tecla ESC

Al oprimir la tecla ESC, deberá aparecer el menú anterior al actual cediendo así, el control al módulo del programa que corresponda.

8. Descripción de selección

Cuando el cursor se posicione en una opción determinada, debe aparecer en la parte inferior izquierda, una breve descripción de la selección de dicha alternativa.

PRUEBAS DE LA OPCION ALTAS DEL MODULO PROGRAMACION

Para esta opción, el sistema deberá solicitar cada uno de los campos que formarán el conjunto de información de cada una de las actividades contenidas en las bases de datos.

Podrán darse de alta más de una actividad sin necesidad de salir al menú de programación, para dicho efecto, el sistema solicitará como primer dato la disciplina a la que pertenece cada actividad, y cuando se hayan dado de alta todas las nuevas disciplinas, este campo deberá dejarse en blanco para que el sistema ceda el control al menú de programación nuevamente.

Los campos que se solicitan en esta opción son los siguientes y se ilustran de una manera más clara en la figura III.3.2.B.

1. Disciplina

Al teclear el nombre de la disciplina a la que pertenece la actividad a dar de alta, el sistema validará su existencia dentro de las bases de datos, si no existe, así lo indicará.

2. Fecha

Al teclear la fecha correspondiente a la actividad, el sistema corrobora que no sea una fecha inválida. Como pueden existir más de una fecha y una hora para cada actividad, el sistema solicita las diferentes fechas, horas y los campos de especial hasta que la fecha se teclee en blanco.

PRUEBAS PROGRAMACION - ALTAS

Referencia	Campo a probar	Estatus
1	Disciplina	si cumple
2	Fecha	si cumple
3	Hora	si cumple
4	Especial	si cumple
5	Actividad	si cumple
6	Artista	si cumple
7	Lugar	si cumple
8	Observaciones	si cumple
9	Precio	si cumple
9	Deacanto	si cumple
9	Sejeto	si cumple
10	Taquilla	si cumple

FIGURA III.3.2.B MATRIZ DE PRUEBAS PROGRAMACION ALTAS

3. Hora

El sistema verifica que la hora no sea una hora inválida.

4. Especial

Es un campo de texto libre que normalmente el usuario utilizará para indicar eventos especiales, por ejemplo si la fecha y hora corresponden al debut o clausura de la actividad.

5. Actividad

Se refiere al nombre de la actividad que se dará de alta. Este campo es un campo que proporciona grupos de cuatro líneas de 60 caracteres para teclear el texto libre deseado. Si el primer grupo de líneas es insuficiente, el sistema automáticamente, proporciona un grupo más, y así sucesivamente, hasta que una línea se teclee en blanco.

6. Artista

Es un campo en el que el usuario puede teclear texto referente al artista, y funciona de la misma manera que el campo de actividad.

7. Lugar

Al teclear el recinto en el que se llevará a cabo la actividad, el sistema validará si dicho recinto existe, si no es así, lo volverá a solicitar.

8. Observaciones

Es un campo que proporciona grupos de cuatro líneas de 60 caracteres para teclear el texto libre deseado. Si el primer grupo de líneas es insuficiente, el sistema

automáticamente, proporciona un grupo más, y así sucesivamente; hasta que una línea se se introduzca vacía.

9. Precios

Se solicitarán los diferentes precios con su correspondiente descuento, especificando las personas que son sujetas a dicho descuento. Como pueden existir más de un precio y un descuento para cada actividad, el sistema solicita los diferentes precios, descuentos y las personas sujetas a dichos descuentos, hasta que el precio sea vacío.

10. Taquilla

Es un campo en el que se pueden teclear los diferentes horarios de taquilla en los que se podrán comprar los boletos de la actividad que se está dando de alta.

PRUEBAS PARA EL MENU CONSULTAS DEL MODULO PROGRAMACION

El menú de consultas del módulo programación, cuenta con las siguientes opciones, es decir que la consulta se podrá realizar de acuerdo a diferentes parámetros:

- Consulta por disciplina
- Consulta por actividad
- Consulta por lugar
- Consulta por rango de fecha
- Consulta por artista
- Consulta por hora
- Consulta por disciplina y hora
- Consulta por disciplina y lugar

La matriz que a continuación se describe y se muestra en la figura III.3.2.C, se aplicó al menú de consultas del módulo de programación, en ella se contemplan los siguientes puntos a validar para cada una de las opciones del menú antes mencionado.

1. Condición de búsqueda

El sistema solicitará el campo de condición de búsqueda, y una vez tecleado este campo, aparecerá lo siguiente:

2. Información

Aparecerán todas las actividades pertenecientes a la condición de búsqueda tecleada, con su respectivo número de folio.

PRUEBAS PROGRAMACION CONSULTAS

Referencia	Campo a probar	Disciplina	Actividad	Lugar	Rango Fecha	Artista	Hora	Disciplina Hora	Disciplina Lugar
1	Condición de búsqueda	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
2	Información	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Consulta folio	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Folio=0	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Folio=I Inicio	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Folio=T Terminar	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
4	Disciplina	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
5	Actividad	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
6	Horarios	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
7	Lugar	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
8	Taquilla	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
9	Observaciones	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
10	Precios	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
10	Descuento	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
10	Personas	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple

FIGURA III.3.2.C MATRIZ DE PRUEBAS PARA EL MENU DE CONSULTAS

3. Opciones

Al mismo tiempo en la parte inferior de la pantalla, aparecen las posibles opciones a teclear:

- **Consulta folio:**
De acuerdo al folio se despliega la información de cada actividad
- **Folio = 0 Continuar**
Si el folio es igual al cero,
- **Folio = I Inicio**
Se despliegan los folios y actividades a partir de la primera existente en la base de datos
- **T = Terminar**
Regresa el control al menú de programación

4. Disciplina

Debe aparecer la disciplina a la que pertenece la actividad que se despliega en la pantalla, excepto cuando el campo de condición de búsqueda ha sido llenado con la palabra disciplina. En seguida se desplegará la información correspondiente a los siguientes campos:

5. Actividad

Nombre de la posible actividad conforme a la condición de búsqueda.

6. Horarios

Posibles horarios de acuerdo a la condición de búsqueda, desplegando el día de la semana, la fecha y la hora de la actividad.

7. Lugar

Recinto en el que se lleva a cabo dicha actividad.

8. Taquilla

El horario de taquilla del recinto antes mencionado.

9. Observaciones

El sistema proporcionará cuatro líneas de 60 caracteres para texto libre y si estas no son suficientes proporcionará grupos de cuatro líneas sucesivamente.

10. Precios

Se desplegarán los diferentes precios con su correspondiente descuento, especificando las personas que son sujetas a cada descuento.

PRUEBAS PARA LA OPCION MODIFICACION DEL MODULO PROGRAMACION

La matriz que a continuación se describe y que se muestra en la figura III.3.2.D, se aplicó a la opción de modificación del módulo de programación, en ella se contemplan los siguientes puntos a validar (en cada caso deberá desplegarse la información actual con posibilidad a ser modificada):

1. Actividad

Nombre de la actividad que se desea modificar, y si este es el campo a modificar, en este momento deberá teclearse la modificación.

2. Disciplina

Nombre de la disciplina a la que pertenece la actividad que se desea modificar, y si este es el campo a modificar, en este momento deberá teclearse la modificación.

3. Fecha y hora

Aparecerán todas las fechas y horas en las que se presenta la actividad, la modificación se debe hacer al momento que aparezca la fecha u hora que se desea modificar, también es factible la posibilidad de aumentar fechas u horas al llegar al final de las existentes, el sistema pasará al siguiente campo cuando se teclee la fecha y hora en blanco.

4. Artista

Aparecerá el texto libre que tecleó el usuario, y si esta se desea modificar, en este momento deberá teclearse la modificación.

PRUEBAS PROGRAMACION - MODIFICACION

Referencia	Campo a probar	Estatus
1	Actividad	si cumple
2	Disciplina	si cumple
3	Fecha	si cumple
3	Hora	si cumple
4	Artista	si cumple
5	Lugar	si cumple
6	Observaciones	si cumple
7	Precio	si cumple
7	Descuento	si cumple
7	Sujeto	si cumple
8	Taquilla	si cumple

FIGURA III.3.2.D MATRIZ DE PRUEBAS DE LA OPCION MODIFICACION

5. Lugar

Aparecerá el recinto en el que se lleva a cabo dicha actividad, y si este es el campo a modificar, en este momento se debe teclear la modificación.

6. Observaciones

Aparecerá el texto libre que tecleó el usuario, y si se desea modificar, en este momento se deberá teclear la modificación.

7. Precio

Aparecerán todos los precios, descuentos y personas sujetas a cada descuento para cada actividad, la modificación se debe hacer al momento que aparezca el precio, descuento o personas que se desean modificar, también es factible la posibilidad de aumentar precios, descuentos o personas al llegar al final de las existentes, el sistema pasará al siguiente campo cuando se tecleen los tres campos en blanco.

8. Taquilla

Aparecerán los horarios de taquilla para la compra de boletos para asistir a la actividad solicitada, si este es el campo a modificar, en este momento deberá teclearse la modificación.

PRUEBAS PARA LA OPCION REPORTES DEL MODULO PROGRAMACION

Este menú consta de dos opciones, las cuales se describen a continuación:

- **Reporte mensual**
- **Reporte semanal**

Para la opción de reporte mensual, se ha utilizado la matriz que a continuación se describe y que se muestra en la figura III.3.2.E, en ella se contemplan los siguientes puntos a validar:

1. Mes

Deberá teclarse el nombre del mes del que se requiera el reporte, el sistema corrobora que sea un mes válido.

2. Disciplina

Se requiere teclear el nombre de la disciplina en cuestión.

3. Impresora

El sistema indicará el momento en el que el usuario deberá preparar la impresora, y espera que se oprima la tecla enter para indicar que ya ha sido preparada. En caso de que se teclee enter sin preparar la impresora, el sistema enviará un mensaje indicándolo, y preguntará si se desea reintentar el envío de información al puerto de impresión.

Una vez terminada la impresión el sistema cederá el control al menú de reportes nuevamente.

PROGRAMACION REPORTES

Referencia	Campo a probar	Estatus
1	Mes	si cumple
2	Disciplina	si cumple
3	Impresora	si cumple

FIGURA III.3.2.E MATRIZ DE PRUEBAS DE LA OPCION REPORTE MENSUAL

Referencia	Campo a probar	Estatus
1	Fecha inicio	si cumple
2	Fecha fin	si cumple
3	Impresora	si cumple

FIGURA III.3.2.F MATRIZ DE PRUEBAS DE LA OPCION REPORTE POR RANGO DE FECHA

Ahora bien, para la opción de reporte semanal, se ha utilizado la matriz que se muestra en la figura III.3.2.F, en la cual se contemplan los siguientes puntos a validar:

1. Fecha de inicio

Deberá teclearse la fecha de inicio del período semanal del que se requiera el reporte, el sistema verificará que sea una fecha válida.

2. Fecha final

Deberá teclearse la fecha de fin del período semanal del que se requiera el reporte, el sistema verificará que sea una fecha válida.

3. Impresora

El sistema indicará el momento en el que el usuario deberá preparar la impresora, y espera que se oprima la tecla enter para indicar que ya ha sido preparada. En caso de que se teclee enter sin preparar la impresora, el sistema enviará un mensaje indicándolo, y preguntará si se desea reintentar el envío de información al puerto de impresión.

Una vez terminada la impresión el sistema cederá el control al menú de reportes nuevamente.

PRUEBAS PARA LA OPCION ELIMINACION DEL MODULO PROGRAMACION

Este menú consta de tres opciones, las cuales se describen a continuación:

- Programación mensual
- Actividad individual
- Recupera programación mensual

Para la opción de programación mensual, se ha utilizado la matriz que en seguida se describe y que se muestra en la figura III.3.2.G, en ella se contemplan los siguientes puntos a validar:

1. Unidad de disco

Debe indicarse la unidad de disco en la cual se hará el respaldo, el sistema verificará que dicha unidad sea válida. Una vez que el usuario indica la unidad y el disco está preparado, el sistema checa el espacio disponible en disco.

2. Mes

Se refiere al mes del cual se desea realizar el respaldo, el sistema debe verificar que sea un mes válido.

3. Confirmación

El sistema solicita que el usuario confirme su interés por respaldar ese mes.

En la opción de actividad individual, se ha utilizado la matriz que a continuación se describe y que se muestra en la figura III.3.2.H, en ella se contempla validar lo siguiente:

ELIMINACION DE PROGRAMACION

Referencia	Campo a probar	Estatus
1	Respaldo	si cumple
2	Drive	si cumple
3	Espacio	si cumple
4	Mes	si cumple
5	Confirmación	si cumple

FIGURA III.3.2.G MATRIZ DE PRUEBAS DE LA OPCION ELIMINACION MENSUAL

Referencia	Campo a probar	Estatus
1	Actividad	si cumple
2	Consulta folio	si cumple
2	Folio = 0	si cumple
2	Folio = I Inicio	si cumple
2	Folio = T Terminar	si cumple
3	Confirmación	si cumple

FIGURA III.3.2.H MATRIZ DE PRUEBAS DE LA OPCION ELIMINACION DE ACTIVIDAD

Referencia	Campo a probar	Estatus
1	Valida drive	si cumple
2	Confirmación	si cumple
3	Recuperación	si cumple

FIGURA III.3.2.I MATRIZ DE PRUEBAS DE LA OPCION RECUPERA PROGRAMACION MENSUAL

1. Actividad

Se requiere que se especifique la actividad que se desea respaldar, como ayuda aparecerán las actividades existentes en la base de datos, con su respectivo folio.

2. Opciones

Al mismo tiempo en la parte inferior de la pantalla, aparecen las posibles opciones a teclear:

- **Consulta folio:**
De acuerdo al folio se despliega la información de cada actividad
- **Folio = 0 Continuar**
Si el folio es igual al cero,
- **Folio = 1 Inicio**
Se despliegan los folios y actividades a partir de la primera existente en la base de datos
- **T = Terminar**
Regresa el control al menú de programación

3. Confirmación

El sistema solicitará la confirmación para obtener la información de la actividad tecleada.

La última opción de este menú, se refiere a la recuperación de programación mensual. El sistema solo solicitará insertar el diskette donde se haya hecho el respaldo y que se indique el drive, el sistema verificará que sea un drive válido. Por último solicitará la confirmación para realizar la recuperación (ver figura III.3.2.1).

PRUEBAS PARA LA OPCION ALTAS DEL MODULO RECINTOS

Para esta opción, el sistema deberá solicitar cada uno de los campos que formarán el conjunto de información de cada uno de los recintos contenidos en las bases de datos.

Podrán darse de alta más de un lugar sin necesidad de salir al menú de recintos, para dicho efecto, el sistema solicitará como primer dato el nombre del recinto, y cuando se hayan dado de alta todos los nuevos recintos, este campo deberá dejarse en blanco para que el sistema ceda el control al menú de recintos nuevamente.

Los campos que se solicitan en esta opción son los siguientes y se ilustran de una manera más clara en la figura III.3.2.J.

1. Nombre del recinto que se desea dar de alta en la base de datos.
2. Dirección donde se encuentra ubicado el recinto, es decir calle y número.
3. Colonia
4. Delegación o municipio en caso de estar en alguna ciudad del interior de la República Mexicana
5. Ciudad
6. Código postal

RECINTOS ALTAS

Referencia	Campo a probar	Estatus
1	Nombre	si cumple
2	Dirección	si cumple
3	Colonia	si cumple
4	Delegación	si cumple
5	Ciudad	si cumple
6	Código postal	si cumple
7	Clave lada	si cumple
8	Teléfono 1	si cumple
8	Teléfono 2	si cumple
9	Confirmación	si cumple

FIGURA III.3.2.J MATRIZ DE PRUEBAS DE LA OPCION ALTAS

7. **Clave lada**

8. **Teléfonos, existen dos campos para teclear un número telefónico respectivamente, aunque si el recinto solo cuenta con un teléfono, el segundo campo se puede dejar en blanco.**

9. **Confirmación. En este punto, el sistema solicitará la confirmación de que los datos son correctos; si lo son, el recinto será dado de alta; si no, el sistema dará opción a que sean tecleados correctamente.**

Una vez que el recinto ha sido dado de alta, se podrán dar de alta más recintos sin necesidad de salir de la opción altas de este menú, cuando se hayan tecleado los datos del último recinto que se desea dar de alta, el sistema solicitará el nombre de un nuevo recinto; para regresar al menú de recintos, el nombre deberá dejarse en blanco.

PRUEBAS PARA EL MENU CONSULTAS DEL MODULO RECINTOS

El menú de consultas del módulo recintos, cuenta con las siguientes opciones, es decir que la consulta se podrá realizar de acuerdo a diferentes parámetros:

- **Consulta de todos los nombres**
- **Consulta por nombre**
- **Consulta por calle**
- **Consulta por colonia**
- **Consulta por delegación**
- **Consulta por ciudad**
- **Consulta por código postal**

Para la primera opción del menú aparecerán todos los recintos que han sido dados de alta en la base de datos; mientras que para el resto de las opciones, se aplicó la matriz que a continuación se describe y se muestra en la figura III.3.2.K, en ella se contemplan los siguientes puntos:

1. Condición de búsqueda

El sistema solicitará el campo de condición de búsqueda, y una vez tecleado este campo, aparecerá información de recintos que cumplan con la condición.

2. Información

Aparecerán todas las actividades pertenecientes a la condición de búsqueda tecleada, con su respectivo número de folio.

3. Opciones

Al mismo tiempo en la parte inferior de la pantalla, aparecen las posibles opciones a teclear:

- **Consulta folio:**
De acuerdo al folio se despliega la información de cada actividad
- **Folio = 0 Continuar**
Si el folio es igual al cero,
- **Folio = 1 Inicio**
Se despliegan los folios y actividades a partir de la primera existente en la base de datos
- **T = Terminar**
Regresa el control al menú de programación

Al escoger un número de folio determinado, aparecerá la información referente al recinto en cuestión (dicha información se menciona a continuación).

- 4. Nombre del recinto**
- 5. Dirección**
- 6. Colonia**
- 7. Ciudad**
- 8. Código postal**
- 9. Lada**
- 10. Teléfonos**

RECINTOS CONSULTAS

Referencia	Campo a probar	Nombre	Calle	Colonia	Delegación	Ciudad	Código Postal	Disciplina a Hora	Disciplina Lugar
1	Condición de búsqueda	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
2	Información	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Consulta folio	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Folio = 0	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Folio = 1 Inicio	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
3	Folio = T Terminar	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
4	Nombre	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
5	Dirección	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
6	Colonia	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
7	Ciudad	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
8	Código postal	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
9	Lada	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
10	Teléfono 1	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple
10	Teléfono 2	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple	si cumple

FIGURA III.3.2.K MATRIZ DE PRUEBAS PARA EL MENU DE CONSULTAS

PRUEBAS PARA LA OPCION MODIFICACION DEL MODULO RECINTOS

La matriz que a continuación se describe y que se muestra en la figura III.3.2.L, se aplicó a la opción de modificación del módulo de recintos, en ella se contemplan los siguientes puntos a validar (en cada caso deberá desplegarse la información actual con posibilidad a ser modificada):

1. Nombre del recinto
2. Dirección (calle y número)
3. Colonia
4. Delegación
5. Ciudad
6. Código postal
7. Clave lada
8. Teléfonos
9. Confirmación

En seguida, el sistema solicitará la verificación de los nuevos datos, y si están correctos, dará de alta las modificaciones en la base de datos.

Una vez que los datos del recinto han sido modificados, se podrán modificar más recintos sin necesidad de salir de la opción modificación de este menú, cuando se hayan tecleado los datos del último recinto que se desea modificar, el sistema solicitará el nombre de otro recinto; para regresar al menú de recintos, el nombre deberá dejarse en blanco.

RECINTOS MODIFICACION

Referencia	Campo a probar	Estatus
1	Nombre	si cumple
2	Dirección	si cumple
3	Colonia	si cumple
4	Delegación	si cumple
5	Ciudad	si cumple
6	Código postal	si cumple
7	Clave lada	si cumple
8	Teléfono 1	si cumple
8	Teléfono 2	si cumple
9	Confirmación	si cumple

FIGURA III.3.2.L MATRIZ DE PRUEBAS DE LA OPCION MODIFICACION

PRUEBAS PARA LA OPCION BAJAS DEL MODULO RECINTOS

Al escoger esta opción, el sistema solicitará el nombre del recinto que se desea dar de baja en la base de datos. Una vez teclado el nombre del recinto, el sistema desplegará toda la información referente a dicho recinto. Por último, se solicita la confirmación para dar de baja la información. Los resultados de esta prueba, aparecen en la figura III.3.2.M.

Una vez que el recinto ha sido dado de baja, se podrán dar de baja más recintos sin necesidad de salir de la opción bajas de este menú, el sistema solicitará el nombre de otro recinto; para regresar al menú de recintos, el nombre deberá dejarse en blanco.

RECINTOS

Referencia	Campo a probar	Estatus
1	Nombre	si cumple
2	Inf. desplegada	si cumple
3	Confirmación	si cumple

FIGURA III.3.2.M MATRIZ DE PRUEBAS DE LA OPCION BAJAS

Referencia	Campo a probar	Estatus
1	Opción Impresión	si cumple
2	Impresora	si cumple

FIGURA III.3.2.N MATRIZ DE PRUEBAS DE LA OPCION REPORTES

PRUEBAS PARA LA OPCION REPORTES DEL MODULO RECINTOS

Para esta opción, se ha utilizado una pequeña matriz, la cual se muestra en la figura III.3.2.N, y se describe a continuación:

1. Opción de impresión

El sistema preguntará si realmente se desea obtener un reporte de los recintos existentes en la base de datos. Si la respuesta es no, el sistema cederá el control al menú de recintos. Si la respuesta es sí, se continúa con el siguiente punto.

2. Impresora

El sistema indicará el momento en el que el usuario deberá preparar la impresora, y espera que se oprima la tecla enter para indicar que ya ha sido preparada. En caso de que se teclee enter sin preparar la impresora, el sistema enviará un mensaje indicándolo, y preguntará si se desea reintentar el envío de información al puerto de impresión.

Una vez terminada la impresión el sistema cederá el control al menú de recintos nuevamente.

PRUEBAS PARA EL MODULO DISCIPLINAS

Este módulo cuenta con las siguientes opciones:

- **Altas**
- **Consulta**
- **Modificación**
- **Reportes**

Para cada opción se ha creado una pequeña y sencilla matriz de pruebas, las cuales se describen a continuación:

Altas

1. Nombre

Para dar de alta una disciplina bastará teclear el nombre de la misma cuando el sistema lo solicite.

2. Confirmación

El sistema solicitará la confirmación de que el dato tecleado sea correcto

Una vez que la disciplina ha sido dada de alta, se podrán dar de alta más disciplinas sin necesidad de salir de la opción altas de este menú, el sistema solicitará el nombre de otra disciplina; para regresar al menú de disciplinas, el nombre deberá dejarse en blanco (ver figura III.3.2.O).

DISCIPLINAS

Referencia	Campo a probar	Estatus
1	Nombre	si cumple
2	Confirmación	si cumple

FIGURA III.3.2.O MATRIZ DE PRUEBAS DE LA OPCION ALTAS

Referencia	Campo a probar	Estatus
1	Todas las disciplinas	si cumple

FIGURA III.3.2.P MATRIZ DE PRUEBAS DE LA OPCION CONSULTAS

Consulta

Al entrar a esta opción en seguida aparecerán todas las disciplinas existentes en la base de datos (ver figura III.3.2.P).

Modificación

1. Nombre

Será necesario teclear el nombre actual de la disciplina.

2. Modificación

Ahora debe teclearse el nombre de la disciplina correcto.

3. Confirmación

El sistema solicitará la confirmación de que el dato tecleado sea correcto.

Una vez que la disciplina ha sido modificada, se podrán modificar más disciplinas sin necesidad de salir de la opción modificación de este menú, el sistema solicitará el nombre de otra disciplina; para regresar al menú de disciplinas, el nombre deberá dejarse en blanco (ver figura III.3.2.Q).

DISCIPLINAS

Referencia	Campo a probar	Estatus
1	Nombre	si cumple
2	Modificación	si cumple
3	Confirmación	si cumple

FIGURA III.3.2.Q MATRIZ DE PRUEBAS DE LA OPCION MODIFICACION

Referencia	Campo a probar	Estatus
1	Reporte	si cumple
2	Impresora	si cumple

FIGURA III.3.2.R MATRIZ DE PRUEBAS DE LA OPCION REPORTES

Reportes

1. Reporte

Como primer paso el sistema preguntará si realmente se desea obtener un reporte de disciplinas. Si la respuesta es no, el sistema cederá el control al menú disciplinas.

2. Impresora

El sistema indicará el momento en el que el usuario deberá preparar la impresora, y espera que se oprima la tecla enter para indicar que ya ha sido preparada. En caso de que se teclee enter sin preparar la impresora, el sistema enviará un mensaje indicándolo, y preguntará si se desea reintentar el envío de información al puerto de impresión.

Una vez terminada la impresión el sistema cederá el control al menú de disciplinas nuevamente (figura III.3.2.R).

PRUEBAS PARA EL MODULO REGENERACION DE INDICES

Para probar este módulo se hizo una pequeña prueba, borrando todos los archivos que contenían los índices, esto se hizo por medio del comando delete del sistema operativo.

Los archivos que fueron borrados son los siguientes:

***.NTX**

Una vez borrados estos, se probaron diferentes partes del sistema, detectando diferentes problemas en el acceso de información; por lo tanto, se utilizó esta opción para regenerar los índices eliminados, y se comprobó que el sistema funcionara correctamente.

III.3.3 PRUEBAS INTEGRALES

Para realizar las pruebas integrales se han llevado a cabo diferentes actividades, las cuales se describen a continuación y se muestran más claramente en una matriz similar a la utilizada en las pruebas modulares para cada opción, la matriz utilizada para cada punto se menciona con un número de figura determinado.

Alta de recintos

Se han dado de alta tres diferentes recintos (figura III.3.3.A).

Alta de disciplinas

Se han dado de alta dos disciplinas, así como una disciplina llamada cursos y talleres; para esta última, el sistema deberá solicitar la fecha de inicio, fecha de fin de curso, hora de inicio y hora de terminación; mientras que para las dos primeras, solo solicitará fecha y hora de presentación (figura III.3.3.B).

Consulta de disciplinas

Al entrar a esta opción, se desplegaron en la pantalla todas las disciplinas que habían sido dadas de alta (figura III.3.3.C).

Alta de actividades

Se han dado de alta nueve actividades, de las cuales tres pertenecerán a cada una de las disciplinas. Además tendrán fechas de dos meses consecutivos (figura III.3.3.D).

III.3.3 PRUEBAS INTEGRALES

Para realizar las pruebas integrales se han llevado a cabo diferentes actividades, las cuales se describen a continuación y se muestran más claramente en una matriz similar a la utilizada en las pruebas modulares para cada opción, la matriz utilizada para cada punto se menciona con un número de figura determinado.

Alta de recintos

Se han dado de alta tres diferentes recintos (figura III.3.3.A).

Alta de disciplinas

Se han dado de alta dos disciplinas, así como una disciplina llamada cursos y talleres; para esta última, el sistema deberá solicitar la fecha de inicio, fecha de fin de curso, hora de inicio y hora de terminación; mientras que para las dos primeras, solo solicitará fecha y hora de presentación (figura III.3.3.B).

Consulta de disciplinas

Al entrar a esta opción, se desplegaron en la pantalla todas las disciplinas que habían sido dadas de alta (figura III.3.3.C).

Alta de actividades

Se han dado de alta nueve actividades, de las cuales tres pertenecerán a cada una de las disciplinas. Además tendrán fechas de dos meses consecutivos (figura III.3.3.D).

RECINTOS ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Nombre	Palacio de Bellas Artes	si cumple
2	Dirección	Av. Juárez y Eje Central	si cumple
3	Colonia	Centro	si cumple
4	Delegación	Cuauhtémoc	si cumple
5	Ciudad	México	si cumple
6	Código postal	6050	si cumple
7	Clave lada	915	si cumple
8	Teléfono 1	709-31-11	si cumple
8	Teléfono 2		si cumple
9	Confirmación	Si	si cumple

FIGURA III.3.3.A.1 MATRIZ DE PRUEBAS ALTA PALACIO DE BELLAS ARTES

RECINTOS ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Nombre	Palacio de Bellas Artes	si cumple
2	Dirección	Av. Juárez y Eje Central	si cumple
3	Colonia	Centro	si cumple
4	Delegación	Cuauhtémoc	si cumple
5	Ciudad	México	si cumple
6	Código postal	6050	si cumple
7	Clave lada	915	si cumple
8	Teléfono 1	709-31-11	si cumple
8	Teléfono 2		si cumple
9	Confirmación	Si	si cumple

FIGURA III.3.3.A.1 MATRIZ DE PRUEBAS ALTA PALACIO DE BELLAS ARTES

RECINTOS ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Nombre	Conservatorio Nacional de Música	si cumple
2	Dirección	Presidente Masarik No. 582	si cumple
3	Colonia	Polanco	si cumple
4	Delegación	Miguel Hidalgo	si cumple
5	Ciudad	Puebla	si cumple
6	Código postal	11550	si cumple
7	Clave lada		si cumple
8	Teléfono 1	280-62-21	si cumple
8	Teléfono 2	280-63-47	si cumple
9	Confirmación	Si	si cumple

FIGURA III.3.3.A.2 MATRIZ DE PRUEBAS DE ALTA CONSERVATORIO NACIONAL DE MUSICA

RECINTOS ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Nombre	Teatro el Galeón	si cumple
2	Dirección	Reforma y Campo Marte s/n	si cumple
3	Colonia	Chapultepec Polanco	si cumple
4	Delegación	Miguel Hidalgo	si cumple
5	Ciudad	México	si cumple
6	Código postal	11560	si cumple
7	Clave lada	915	si cumple
8	Teléfono 1	280-87-71	si cumple
8	Teléfono 2		si cumple
9	Confirmación	Si	si cumple

FIGURA III.3.3.A.3 MATRIZ DE PRUEBAS ALTA TEATRO EL GALEON

DISCIPLINAS

Referencia	Campo a probar	Estatus
1	Teatro	si cumple
2	Confirmación	si cumple

Referencia	Campo a probar	Estatus
1	Donza	si cumple
2	Confirmación	si cumple

Referencia	Campo a probar	Estatus
1	Cursos y talleres	si cumple
2	Confirmación	si cumple

FIGURA III.3.3.B MATRIZ DE PRUEBAS DE LA OPCION ALTAS

DISCIPLINAS CONSULTA

Referencia	Campo a probar	Estatus
I	Todas las disciplinas	si cumple

FIGURA III.3.3.C MATRIZ DE PRUEBAS DE CONSULTA

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Cursos y talleres	si cumple
2	Fecha inicio	06/02/94	si cumple
2	Fecha fin	17/02/94	si cumple
3	Hora inicio	16:00	si cumple
3	Hora fin	20:00	si cumple
4	Especial		si cumple
5	Actividad	Curso de perfeccionamiento en composición	si cumple
6	Artista	Mtro. Franco Donatoni	si cumple
7	Lugar	Conservatorio Nacional de Música	si cumple
8	Observaciones	Obj. Analizar y ejercitar con los participantes algunas técnicas y procedimientos en la composición contemporánea	si cumple
9	Precio		si cumple
9	Descuento		si cumple
9	Sujeto		si cumple
10	Taquilla		si cumple
11	Confirmación		si cumple

FIGURA III.3.3.D.1 MATRIZ DE PRUEBAS ALTA DE LA PRIMER ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Teatro	si cumple
2	Fecha 1	01/03/94	si cumple
3	Hora 1	9:00	si cumple
2	Fecha 2	01/03/94	si cumple
3	Hora 2	11:00	si cumple
4	Especial		si cumple
5	Actividad	Teatro educativo "Charivari"	si cumple
6	Artista	de: William Shakespeare dirección: Margarita Isabel	si cumple
7	Lugar	Teatro el Galeón	si cumple
8	Observaciones	Cia teatral: La Troupe Actores: Roberto Blandón, Roberto Morlet, Carlos Cabral, Arturo Allegro, Mario Iván Martínez	si cumple
9	Precio	N\$40.00	si cumple
9	Descuento	50%	si cumple
9	Sujeto	Maestros y estudiantes con credencial	si cumple
10	Taquilla	11:00 a 15:00 y 17:00 a 19:00	si cumple
11	Confirmación		si cumple

FIGURA III.3.3.D.2 MATRIZ DE PRUEBAS ALTA DE LA SEGUNDA ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Teatro	si cumple
2	Fecha 1	02/03/94	si cumple
3	Hora 1	20:30	si cumple
2	Fecha 2	03/03/94	si cumple
3	Hora 2	20:30	si cumple
2	Fecha 3	04/03/94	si cumple
3	Hora 3	19:00	si cumple
2	Fecha 4	05/03/94	si cumple
3	Hora 4	18:00	si cumple
2	Fecha 5	09/03/94	si cumple
3	Hora 5	20:30	si cumple
2	Fecha 6	10/03/94	si cumple
3	Hora 6	20:30	si cumple
2	Fecha 7	11/03/94	si cumple
3	Hora 7	19:00	si cumple
2	Fecha 8	12/03/94	si cumple
3	Hora 8	18:00	si cumple
4	Especial		si cumple
5	Actividad	Obra de teatro: Juicio Suspendido	si cumple
6	Artista	Obra y dirección: Héctor Mendoza	si cumple
7	Lugar	Teatro el Galeón	si cumple
8	Observaciones	Adolescentes y adultos. Actores: Delia Casanova, Margarita Saiz, Cynthia Klitbo, Hernán Mendoza y Jorge Marín	si cumple
9	Precio	NS\$40.00	si cumple
9	Descuento	50%	si cumple
9	Sujeto	Maestros y estudiantes con credencial	si cumple
10	Taquilla	11:00 a 15:00 y 17:00 a 19:00	si cumple
11	Confirmación		si cumple

FIGURA III.3.3.D.3 MATRIZ DE PRUEBAS ALTA DE LA TERCERA ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Teatro	si cumple
2	Fecha 1	07/02/94	si cumple
3	Hora 1	9:30	si cumple
2	Fecha 2	07/02/94	si cumple
3	Hora 2	11:00	si cumple
4	Especial		si cumple
5	Actividad	Teatro educativo "Don Quijote"	si cumple
6	Artista	versión: Salvador Novo dirección: Oscar Ledesma	si cumple
7	Lugar	Teatro el Galeón	si cumple
8	Observaciones	Actores: Charles Lake, Soraya Brito, Adriana Uribe, Jos Carpio, Manuel Armenta	si cumple
9	Precio	N\$40.00	si cumple
9	Descuento	50%	si cumple
9	Sujeto	Maestros y estudiantes con credencial	si cumple
10	Taquilla	11:00 a 15:00 y 17:00 a 19:00	si cumple
11	Confirmación		si cumple

FIGURA III.3.3.D.4 MATRIZ DE PRUEBAS ALTA DE LA CUARTA ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Donza	si cumple
2	Fecha I	15/02/94	si cumple
3	Hora I	19:30	si cumple
4	Especial		si cumple
5	Actividad	XI Concurso Interior de Coreografía Ballet Independiente	si cumple
6	Artista	fundador: Raúl Flores Canelo director: Magnolia Flores	si cumple
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Observaciones	Programa: Homo delphinus; Hormiguero; Viento; Vizcainas no. 13 4o. piso; De lo efimero; Mientras tanto; Vida; Ya es tiempo; Los días azules; Líder.	si cumple
9	Precio	Entrada libre	si cumple
9	Descuento		si cumple
9	Sujeto		si cumple
10	Taquilla		si cumple
11	Confirmación		si cumple

FIGURA III.3.3.D.5 MATRIZ DE PRUEBAS ALTA DE LA QUINTA ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Donza	si cumple
2	Fecha 1	02/02/94	si cumple
3	Hora 1	21:00	si cumple
2	Fecha 2	05/02/94	si cumple
3	Hora 2	9:30	si cumple
2	Fecha 3	08/02/94	si cumple
3	Hora 3	21:00	si cumple
2	Fecha 4	15/02/94	si cumple
3	Hora 4	21:00	si cumple
2	Fecha 5	19/02/94	si cumple
3	Hora 5	9:30	si cumple
4	Especial		si cumple
5	Actividad	Ballet Folklórico de México	si cumple
6	Artista	de: Amalia Hernández	si cumple
7	Lugar	Palacio Nacional de Bellas	si cumple
8	Observaciones	Artes	si cumple
9	Precio	N\$125.00	si cumple
9	Precio	N\$115.00	si cumple
9	Precio	N\$90.00	si cumple
9	Descuento		si cumple
9	Sujeto		si cumple
10	Taquilla	11:00 a 19:00	si cumple
11	Confirmación		si cumple

FIG. III.3.3.D.6 PRUEBAS ALTA DE LA SEXTA ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Donza	si cumple
2	Fecha 1	2/03/93	si cumple
3	Hora 1	21:00	si cumple
2	Fecha 2	5/03/93	si cumple
3	Hora 2	9:30	si cumple
2	Fecha 3	8/03/93	si cumple
3	Hora 3	21:00	si cumple
2	Fecha 4	15/03/93	si cumple
3	Hora 4	21:00	si cumple
2	Fecha 5	19/03/93	si cumple
3	Hora 5	9:30	si cumple
4	Especial		si cumple
5	Actividad	Ballet Folklórico de México	si cumple
6	Artista	de: Amalia Hernández	si cumple
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Observaciones		si cumple
9	Precio	N\$125.00	si cumple
9	Precio	N\$115.00	si cumple
9	Precio	N\$90.00	si cumple
9	Descuento		si cumple
9	Sujeto		si cumple
10	Taquilla	11:00 a 19:00	si cumple

FIG. III.3.3.D.7 PRUEBAS ALTA DE LA SEPTIMA ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Disciplina	Donza	si cumple
2	Fecha	02/02/94	si cumple
3	Hora	20:30	si cumple
4	Especial		si cumple
5	Actividad	Cuerpos en el fin del milenio 1983-1993 La danza contemporánea independiente en la década de los 80's "Afinidades electivas"	si cumple
6	Artista	dirección: Silvia Unzueta	si cumple
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Observaciones		si cumple
9	Precio	NS\$60.00	si cumple
9	Descuento	50%	si cumple
9	Sujeto	Maestros y estudiantes con credencial	si cumple
10	Taquilla	11:00 a 19:00	si cumple

FIG. III.3.3.D.8 PRUEBAS ALTA DE LA OCTAVA ACTIVIDAD

PROGRAMACION ALTAS

Referencia	Campo a	Dato cargado en el sistema	Estatus
1	Disciplina	Donza	si cumple
2	Fecha 1	16/02/94	si cumple
3	Hora 1	20:00	si cumple
2	Fecha 2	18/02/94	si cumple
3	Hora 2	17:00	si cumple
2	Fecha 3	19/02/94	si cumple
3	Hora 3	17:00	si cumple
2	Fecha 4	05/03/94	si cumple
3	Hora 4	17:00	si cumple
2	Fecha 5	07/03/94	si cumple
3	Hora 5	20:00	si cumple
2	Fecha 6	09/02/94	si cumple
3	Hora 6	20:00	si cumple
2	Fecha 7	11/02/94	si cumple
3	Hora 7	20:00	si cumple
2	Fecha 8	14/03/94	si cumple
3	Hora 8	20.00	si cumple
4	Especial		si cumple
5	Actividad	Temporada 1997 Cia. Nacional de Danza "El	si cumple
6	Artista	dirección: Fernando Alonso	si cumple
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Observaciones	Coreografía: Nina Novak Música: Piotr I.	si cumple
9	Precio	N\$80.00	si cumple
9	Precio	N\$70.00	si cumple
9	Precio	N\$60.00	si cumple
9	Precio	N\$50.00	si cumple
9	Precio	N\$30.00	si cumple
9	Descuento	50%	si cumple
9	Sujeto	maestros y estudiantes con credencial	si cumple
10	Taquilla	11:00 a 19:00	si cumple
11	Confirmación		si cumple

FIG. III.3.3.D.9 PRUEBAS ALTA DE LA NOVENA ACTIVIDAD

Modificación de disciplinas

Si existe algún error en una de las disciplinas dadas se alta, se podrá utilizar esta opción para realizar la modificación. Una vez modificada la disciplina, la modificación deberá aparecer para cada actividad perteneciente a dicha disciplina. Para realizar la prueba a esta parte del sistema, se modificará una disciplina que había sido tecleada con un error (figura III.3.3.E).

Reporte de disciplinas

Una vez realizada la modificación de la disciplina que contenía el error, se obtuvo el reporte de disciplinas, el cual también se anexa (figura III.3.3.F).

Consulta de recintos

Para el menú de recintos, existen diferentes tipos de consulta, por lo tanto se realizaron dos consultas por cada recinto, es decir una consulta por cada parámetro de búsqueda:

1. Consulta por recinto (figura III.3.3.G1)
2. Consulta por calle (figura III.3.3.G2)
3. Consulta por colonia (figura III.3.3.G3)
4. Consulta por delegación (figura III.3.3.G4)
5. Consulta por ciudad (figura III.3.3.G5)
6. Consulta por código postal (figura III.3.3.G6)

Modificación de recintos

Para probar este módulo se modificará el recinto que había sido cargado con errores (Conservatorio Nacional de Música), veáse figura III.3.3.H.

DISCIPLINAS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Nombre	Danzo	si cumple
2	Modificación	Danza	si cumple
3	Confirmación		si cumple

FIGURA III.3.3.E PRUEBAS MODIFICACION DISCIPLINA DANZA

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Reporte		si cumple
2	Impresora		si cumple

FIGURA III.3.2.F PRUEBAS REPORTE DISCIPLINAS

RECINTOS CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Palacio Nacional de Bellas Artes	si cumple
2	Información		si cumple
3	Consulta folio	1	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = I Inicio	1	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Nombre	Palacio Nacional de Bellas Artes	si cumple
5	Dirección	Av. Juárez y Eje Central	si cumple
6	Colonia	Centro	si cumple
7	Delegación	Cuauhtémoc	si cumple
8	Ciudad	México	si cumple
9	Código postal	6050	si cumple
10	Lada	915	si cumple
11	Teléfono 1	709-31-11	si cumple
11	Teléfono 2		si cumple

FIGURA III.3.3.G1 PRUEBAS MENU CONSULTAS POR NOMBRE

RECINTOS CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Reforma	si cumple
2	Información		si cumple
3	Consulta folio	3	si cumple
3	Folio = 0 Continúa	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Nombre	Teatro El Galeón	si cumple
5	Dirección	Reforma y Campo Marte s/n	si cumple
6	Colonia	Chapultepec Polanco	si cumple
7	Delegación	Miguel Hidalgo	si cumple
8	Ciudad	México	si cumple
9	Código postal	11560	si cumple
10	Lada	915	si cumple
11	Teléfono 1	280-87-71	si cumple
11	Teléfono 2		si cumple

FIGURA III.3.3.G2 PRUEBAS MENU CONSULTAS POR CALLE

RECINTOS CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Polanco	si cumple
2	Información		si cumple
3	Consulta folio	2	si cumple
3	Folio = 0 Continúa	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Nombre	Conservatorio Nacional de Música	si cumple
5	Dirección	Presidente Masarik No. 582	si cumple
6	Colonia	Polanco	si cumple
7	Delegación	Miguel Hidalgo	si cumple
8	Ciudad	Puebla	si cumple
9	Código postal	11550	si cumple
10	Lada		si cumple
11	Teléfono 1	280-62-21	si cumple
11	Teléfono 2	280-63-47	si cumple

FIGURA III.3.3.G3 PRUEBAS MENU CONSULTAS POR COLONIA

RECINTOS CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Miguel Hidalgo	si cumple
2	Información		si cumple
3	Consulta folio	3	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Nombre	Teatro El Galeón	si cumple
5	Dirección	Reforma y Campo Marte s/n	si cumple
6	Colonia	Chapultepec Polanco	si cumple
7	Delegación	Miguel Hidalgo	si cumple
8	Ciudad	México	si cumple
9	Código postal	11560	si cumple
10	Lada	915	si cumple
11	Teléfono 1	280-87-71	si cumple
11	Teléfono 2		si cumple

FIGURA III.3.3.G4 PRUEBAS MENU CONSULTAS POR DELEGACION

RECINTOS CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	México	si cumple
2	Información		si cumple
3	Consulta folio	1	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = 1 Inicio	1	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Nombre	Palacio Nacional de Bellas Artes	si cumple
5	Dirección	Av. Juárez y Eje Central	si cumple
6	Colonia	Centro	si cumple
7	Delegación	Cuauhtémoc	si cumple
8	Ciudad	México	si cumple
9	Código postal	6050	si cumple
10	Lada	915	si cumple
11	Teléfono 1	709-31-11	si cumple
11	Teléfono 2		si cumple

FIGURA III.3.3.G5 PRUEBAS MENU CONSULTAS POR CIUDAD

RECINTOS CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	11550	si cumple
2	Información		si cumple
3	Consulta folio	2	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Nombre	Conservatorio Nacional de Música	si cumple
5	Dirección	Presidente Masarik No. 582	si cumple
6	Colonia	Polanco	si cumple
7	Delegación	Miguel Hidalgo	si cumple
8	Ciudad	Puebla	si cumple
9	Código postal	11550	si cumple
10	Lada		si cumple
11	Teléfono 1	280-62-21	si cumple
11	Teléfono 2	280-63-47	si cumple

FIGURA III.3.3.G6 PRUEBAS MENU CONSULTAS POR CODIGO POSTAL

RECINTOS MODIFICACION

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Nombre	Conservatorio Nacional de Música	si cumple
2	Dirección	Presidente Masarik No. 582	si cumple
3	Colonia	Polanco	si cumple
4	Delegación	Miguel Hidalgo	si cumple
5	Ciudad	México	si cumple
6	Código postal	11550	si cumple
7	Clave lada		si cumple
8	Teléfono 1	280-62-21	si cumple
8	Teléfono 2	280-63-47	si cumple
9	Confirmación	Si	si cumple

FIGURA III.3.3.H PRUEBAS MODIFICACION RECINTOS

Consulta de actividades

La consulta de actividades se realizó de acuerdo a cada una de las condiciones de búsqueda que el sistema ofrece, es decir se realizaron ocho consultas:

1. Consulta por disciplina (figura III.3.3.11)
2. Consulta por actividad (figura III.3.3.12)
3. Consulta por lugar (figura III.3.3.13)
4. Consulta por rango de fecha (figura III.3.3.14)
5. Consulta por artista (figura III.3.3.15)
6. Consulta por hora (figura III.3.3.16)
7. Consulta por disciplina y hora (figura III.3.3.17)
8. Consulta por disciplina y lugar (figura III.3.3.18)

Modificación de actividades

Cuando se dieron de alta las actividades, la última contenía algunos errores, es decir en lugar de ser "El Cascanueces" se tecleó "El Principito", además existían otros errores en el campo de actividad; con esta opción estos errores fueron corregidos (figura III.3.3.J).

Reporte mensual de actividades

Se ha escogido el mes de febrero para obtener un reporte de las actividades que pertenecen a la disciplina teatro, el cual se anexa después de la figura correspondiente (III.3.3.K).

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Danza	si cumple
2	Información		si cumple
3	Consulta folio	8	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Danza	si cumple
5	Actividad	Cuerpos en el fin del milenio 1983-1993 La danza contemporánea independiente en la década de los 80's "Afinidades electivas"	si cumple
6	Horarios	Miércoles 2 febrero 20:30	si cumple
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Taquilla	11:00 a 19:00	si cumple
9	Observaciones		si cumple
10	Precios	NS60.00	si cumple
10	Descuento	50%	si cumple
10	Personas	Maestros y estudiantes con credencial	si cumple

FIGURA III.3.3.11 PRUEBAS MENU CONSULTAS POR DISCIPLINAS

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Curso de perfeccionamiento en composición	si cumple
2	Información		si cumple
3	Consulta folio	I	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Cursos y talleres	si cumple
5	Actividad	Curso de perfeccionamiento en composición	si cumple
6	Horarios	Dom 6 feb a jue 17 feb de 16:00 a 20:00	si cumple
7	Lugar	Conservatorio Nacional de Música	si cumple
8	Taquilla		si cumple
9	Observaciones	Obj. Analizar y ejercitar con los participantes algunas técnicas y procedimientos en la composición contemporánea	si cumple
10	Precios		si cumple
10	Descuento		si cumple
10	Personas		si cumple

FIGURA III.3.3.12 PRUEBAS MENU CONSULTAS POR ACTIVIDAD

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Palacio Nacional de Bellas Artes	si cumple
2	Información		si cumple
3	Consulta folio	5	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Danza	si cumple
5	Actividad	XI Concurso Interior de Coreografía Ballet Independiente	si cumple
6	Horarios	Martes 15 febrero 19:30	si cumple
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Taquilla		si cumple
9	Observaciones	Programa: Homo delphinus; Hormiguero; Viento; Vizcafnas no 13 4o. piso; De lo efimero; Mientras tanto; Vida; Ya es tiempo; Los días azules; Líder	si cumple
10	Precios	Entrada libre	si cumple
10	Descuento		si cumple
10	Personas		si cumple

FIGURA III.3.3.13 PRUEBAS MENU CONSULTAS POR LUGAR

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	5 Febrero al 8 de Febrero	si cumple
2	Información		si cumple
3	Consulta folio	6	si cumple
3	Folio = 0	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Donza	si cumple
5	Actividad	Ballet Folklórico de México	si cumple
6	Horarios	Sábado 5 febrero 9:30	si cumple
		Martes 8 febrero 21:00	
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Taquilla	11:00 a 19:00	si cumple
9	Observaciones		si cumple
10	Precios	NS1250.00 NS115.00 NS90.00	si cumple
10	Descuento		si cumple
10	Personas		si cumple

FIGURA III.3.3.14 PRUEBAS MENU CONSULTAS POR RANGO DE FECHA

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	dirección: Fernando Alonso	si cumple
2	Información		si cumple
3	Consulta folio	9	si cumple
3	Folio = 0 Continua	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Danza	si cumple
5	Actividad	Temporada 1994 Cia. Nacional de Danza "El Cascanueces"	si cumple
6	Horarios	Miércoles 16 febrero 20:00	si cumple
		Viernes 18 febrero 17:00	
		Sábado 19 febrero 17:00	
		Sábado 5 febrero 17:00	
		Lunes 7 marzo 20:00	
		Miércoles 9 marzo 20:00	
		Viernes 11 marzo 20:00	
		Lunes 14 marzo 20:00	
7	Lugar	Palacio Nacional de Bellas Artes	si cumple
8	Taquilla	11:00 a 19:00	si cumple
9	Observaciones		si cumple
10	Precios	N\$80.00 N\$70.00 N\$60.00 N\$50.00 N\$30.00	si cumple
10	Descuento	50%	si cumple
10	Personas	Maestros y estudiantes con credencial	si cumple

FIGURA III.3.3.15 PRUEBAS MENU CONSULTAS POR ARTISTA

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	11:00	si cumple
2	Información		si cumple
3	Consulta folio	2	si cumple
3	Folio = 0	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Teatro	si cumple
5	Actividad	Teatro educativo "Charivari"	si cumple
6	Horarios	Martes 1 marzo 11:00	si cumple
7	Lugar	Teatro El Galeón	si cumple
8	Taquilla		si cumple
9	Observaciones	Cia. teatral: La Troupe Actores: Roberto Blandón, Roberto Morlet, Carlos Cabral, Arturo Allegro, Mario Iván Martínez	si cumple
10	Precios	NS\$40.00	si cumple
10	Descuento	50%	si cumple
10	Personas	Maestros y estudiantes con credencial	si cumple

FIGURA III.3.3.16 PRUEBAS MENU CONSULTAS POR HORA

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Teatro 18:00	si cumple
2	Información		si cumple
3	Consulta folio	3	si cumple
3	Folio = 0	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Teatro	si cumple
5	Actividad	Obra de teatro: Juicio Suspendido	si cumple
6	Horarios	Sábado 12 marzo 18:00	si cumple
7	Lugar	Teatro El Galeón	si cumple
8	Taquilla	11:00 a 15:00 y 17:00 a 19:00	si cumple
9	Observaciones	Adolescentes y adultos. Actores: Delia Casanova, Margarita Sanz, Cynthia Klirbo, Hernán Mendoza y Jorge Marín	si cumple
10	Precios	NS\$40.00	si cumple
10	Descuento	50%	si cumple
10	Personas	Maestros y estudiantes con credencial	si cumple

FIGURA III.3.3.17 PRUEBAS MENU CONSULTAS POR DISCIPLINA Y HORA

PROGRAMACION CONSULTAS

Referencia	Campo a probar	Campo cargado en el sistema	Estatus
1	Condición de búsqueda	Teatro Teatro "El Galeon"	si cumple
2	Información		si cumple
3	Consulta folio	4	si cumple
3	Folio = 0	0	si cumple
3	Folio = I Inicio	I	si cumple
3	Folio = T Terminar	T	si cumple

Información desplegada por el sistema

4	Disciplina	Teatro	si cumple
5	Actividad	Teatro educativo: "Don Quijote"	si cumple
6	Horarios	Lunes 7 febrero 9:30	si cumple
		Lunes 7 febrero 11:00	
7	Lugar	Teatro El Galeón	si cumple
8	Taquilla	11:00 a 15:00 y 17:00 a 19:00	si cumple
9	Observaciones	Actores: Charles Lake, Soraya Brito, Adriana	si cumple
		Uribe, Jos Carpio, Manuel Armenta	
10	Precios	NS\$40.00	si cumple
10	Descuento	50%	si cumple
10	Personas	Maestros y estudiantes con credencial	si cumple

FIGURA III.3.3.18 PRUEBAS MENU CONSULTAS POR DISCIPLINA Y LUGAR

MODIFICACION ACTIVIDADES

Referencia	Campo a	Dato cargado en el sistema	Estatus
1	Actividad	Temporada 1997 Cia. Nacional de Danza "El Principito"	si cumple
2	Disciplina	Danza	si cumple
3	Fecha 1	16/02/94	si cumple
4	Hora 1	20:00	si cumple
3	Fecha 2	18/02/94	si cumple
4	Hora 2	17:00	si cumple
3	Fecha 3	19/02/94	si cumple
4	Hora 3	17:00	si cumple
3	Fecha 4	05/03/94	si cumple
4	Hora 4	17:00	si cumple
3	Fecha 5	07/03/94	si cumple
4	Hora 5	20:00	si cumple
3	Fecha 6	09/02/94	si cumple
4	Hora 6	20:00	si cumple
3	Fecha 7	11/02/94	si cumple
4	Hora 7	20:00	si cumple
3	Fecha 8	14/03/94	si cumple
4	Hora 8	20:00	si cumple
5	Especial		si cumple
6	Actividad	Temporada 1994 Cia. Nacional de Danza "El Cascanueces"	si cumple
7	Artista	dirección: Fernando Alonso	si cumple
8	Lugar	Palacio Nacional de Bellas Artes	si cumple
9	Observaciones	Coreografía: Nina Novak Música: Piotr I. Chaikovski Escenografía y vestuario: Eugenio Servín 16 y 18 marzo con grabación, costo N\$60.00, N\$50.00, N\$40.00 y N\$30.00	si cumple
9	Precio	N\$80.00	si cumple
9	Precio	N\$70.00	si cumple
9	Precio	N\$60.00	si cumple
9	Precio	N\$50.00	si cumple
9	Precio	N\$30.00	si cumple
9	Descuento	50%	si cumple
10	Sujeto	maestros y estudiantes con credencial	si cumple
11	Taquilla	11:00 a 19:00	si cumple
12	Confirmación		si cumple

FIG. III.3.3.J PRUEBA MODIFICACION "EL CASCANUECES"

PROGRAMACION REPORTES

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Mes	Febrero	si cumple
2	Disciplina	Teatro	si cumple
3	Impresora		si cumple

FIGURA III.3.3.K PRUEBAS REPORTE MENSUAL (FEBRERO)

Reporte de actividades por rango de fechas

Se escogió el período del 1º al 15 de marzo para obtener el reporte de actividades que se realizan durante dicha semana.

Eliminación de una actividad en específico

Se ha escogido la información de una actividad para que sea eliminada del sistema: Teatro educativo "Charivari" (figura III.3.3.M).

Eliminación de actividades de cierto mes

Para esta opción se ha elegido un mes: febrero, para que las actividades que se realizan en este período, sean eliminadas de las bases de datos del sistema (figura III.3.3.N).

Recuperación de información mensual de disco

La información que había sido cargada en diskette y dada de baja en el disco duro (mes: febrero), fue recuperada en disco, para realizar las pruebas restantes (figura III.3.3.O).

Baja de un recinto en específico

Para probar esta opción, se dió de baja en el sistema el teatro El Galeón, para esto fue necesario primero eliminar del sistema, las actividades que se realizaban en dicho recinto (Obra de teatro: Juicio Suspendido, Teatro educativo "Don Quijote"), veáse figura III.3.3.P.

Reporte general de recintos

Al entrar a esta opción se obtuvo el reporte correspondiente.

ELIMINACION DE PROGRAMACION

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Actividad	Teatro educativo "Charivari"	si cumple
2	Consulta folio	2	si cumple
2	Folio = 0 Continua	0	si cumple
2	Folio = I Inicio	I	si cumple
2	Folio = T Terminar	T	si cumple
3	Confirmación	Si	si cumple

FIGURA III.3.3.M PRUEBA ELIMINACION DE UNA ACTIVIDAD ESPECIFICA "CHARIVARI"

ELIMINACION DE PROGRAMACION

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Respaldo		si cumple
2	Drive		si cumple
3	Espacio		si cumple
4	Mes	Febrero	si cumple
5	Confirmación	Si	si cumple

FIGURA III.3.3.N PRUEBA ELIMINACION DE ACTIVIDADES DE UN MES ESPECIFICO (FEBRERO)

RECUPERACION DE PROGRAMACION

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Valida drive		si cumple
2	Confirmación	Si	si cumple
3	Recuperación		si cumple

FIGURA III.3.3.0 PRUEBA RECUPERACION DE INFORMACION MENSUAL EN DISCO

RECINTOS

Referencia	Campo a probar	Dato cargado en el sistema	Estatus
1	Nombre	Teatro educativo "Charivari"	si cumple
2	Inf. desplegada		si cumple
3	Confirmación	Si	si cumple

FIGURA III.3.3.P PRUEBA BAJA DE UN RECINTO ESPECIFICO

Regeneración de índices

Para probar esta opción, se eliminaron los archivos de índices de las diferentes tablas del sistema, después de haber borrado los archivos se intentó realizar algunas consultas obteniendo como resultado mensajes de error. Posteriormente, utilizamos esta opción y cuando el sistema indicó el fin de la regeneración se realizaron las consultas que anteriormente habían provocado los mensajes de error, obteniendo en esta ocasión resultados satisfactorios.

III.4.1 MANUAL DE MANTENIMIENTO.

III.4.1.1 DESCRIPCION DE ARCHIVOS DEL SISTEMA

En este punto se describe cada uno de los archivos del sistema, indicando nombre y extensión, tipo de archivo, y como fue generado. Ver tabla III.4.1.1.A

NOMBRE DEL ARCHIVO	TIPO DE ARCHIVO	GENERADO CON	DESCRIPCION
INBA.PRG	PROGRAMA	EDITOR MS-DOS	CODIGO FUENTE DEL PROGRAMA, PROGRAMA PRINCIPAL Y RUTINAS
INBAPROC.PRG	PROGRAMA	EDITOR MS-DOS	CODIGO FUENTE DE RUTINAS
INBA.OBJ	OBJETO	COMPILADOR CLIPPER VER 5.2	CODIGO OBJETO DE LOS PROGRAMAS
INBA.EXE	EJECUTABLE	LIGADOR PAR CLIPPER VER. 3.14B	PROGRAMA EJECUTABLE
INBA.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	DATOS GENERALES DE LA ACTIVIDAD, VER TABLA III.4.1.1.B
ACTIVIDA.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	DATOS GENERALES DE LA ACTIVIDAD, VER TABLA III.4.1.1.C
ARTISTA.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	ARTISTAS QUE SE PRESENTAN EN LA ACTIVIDAD, VER TABLA III.4.1.1.D
HORARIOS.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	HORARIOS EN LOS QUE SE PRESENTAN LAS ACTIVIDADES, VER TABLA III.4.1.1.E
COSTDESC.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	PRECIOS Y DESCUENTOS DE LAS ACTIVIDADES, VER TABLA III.4.1.1.F
OBSERVA.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	OBSERVACIONES REFERENTES A LAS ACTIVIDADES, VER TABLA III.4.1.1.G
DISCIPL.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	DISCIPLINAS EXISTENETES, VER TABLA III.4.1.1.H

LUGARES.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	DATOS DE LOS LUGARES EN DONDE SE EFECTUAN LAS ACTIVIDADES. VER TABLA III.4.1.1.I
LISFECH.DBF	BASE DE DATOS	PROGRAMA DBU (CLIPPER)	ARCHIVO DE PASO PARA IMPRESION DE REPORTES, VER TABLA III.4.1.1.J

TABLA III.4.1.1.A ARCHIVOS DEL SISTEMA

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
FOLIO	NUMERICO	9	NUMERO DE FOLIO ASIGNADO EN FORMA AUTOMATICO POR EL PROGRAMA AL DAR DE ALTA LA ACTIVIDAD EN CUESTION
DISCIPLINA	NUMERICO	2	DISCIPLINA A LA QUE PERTENECE LA ACTIVIDAD
LUGAR	CARACTER	38	NOMBRE DEL LUGAR EN EL QUE SE EFECTUA LA ACTIVIDAD
HORATAQUI	CARACTER	50	HORARIO EN EL QUE SE ENCUENTRA ABIERTA LA TAQUILLA, MUSEO, ETC.

TABLA III.4.1.1.B ESTRUCTURA DEL ARCHIVO INBA.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
FOLIO	NUMERICO	9	NUMERO DE FOLIO ASIGNADO EN FORMA AUTOMATICO POR EL PROGRAMA AL DAR DE ALTA LA ACTIVIDAD EN CUESTION
ACTIVIDAD	CARACTER	240	NOMBRE O DESCRIPCION DE LA ACTIVIDAD, SI NO SON SUFICIENTES 240 CARACTERES PARA LA DESCRIPCION, ENTONCES SE TENDRAN TANTOS REGISTROS CON EL MISMO NUMERO DE FOLIO COMO SEAN NECESARIOS.

TABLA III.4.1.1.C ESTRUCTURA DEL ARCHIVO ACTIVIDA.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
FOLIO	NUMERICO	9	NUMERO DE FOLIO ASIGNADO EN FORMA AUTOMATICO POR EL PROGRAMA AL DAR DE ALTA LA ACTIVIDAD EN CUESTION
ARTISTA	CARACTER	240	NOMBRE DE LOS ARTISTAS QUE PARTICIPAN EN LA ACTIVIDAD, SI NO SON SUFICIENTES 240 CARACTERES PARA LOS NOMBRES, ENTONCES SE TENDRAN TANTOS REGISTROS CON EL MISMO NUMERO DE FOLIO COMO SEAN NECESARIOS.

TABLA III.4.1.1.D ESTRUCTURA DEL ARCHIVO ARTISTA.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
FOLIO	NUMERICO	9	NUMERO DE FOLIO ASIGNADO EN FORMA AUTOMATICO POR EL PROGRAMA AL DAR DE ALTA LA ACTIVIDAD EN CUESTION
FECHA	FECHA	8	FECHA EN LA QUE SE INICIA LA ACTIVIDAD
FECHAFIN	FECHA	8	FECHA EN LA QUE TERMINA LA ACTIVIDAD, ESTE CAMPO SOLO SE UTILIZA CUANDO LA ACTIVIDAD PERTENECE A LA DISCIPLINA DE CURSOS Y TALLERES
HORA	CARACTER	5	HORA EN LA QUE INICIA LA ACTIVIDAD
HORAFIN	CARACTER	5	HORA EN LA QUE TERMINA LA ACTIVIDAD, ESTE CAMPO SOLO SE UTILIZA CUANDO LA ACTIVIDAD PERTENECE A LA DISCIPLINA DE CURSOS Y TALLERES
ESPECIAL	CARACTER	25	CAMPO PARA INDICAR ALGUN COMENTARIO REFERENTE A LA FECHA DE PRESENTACION
DISCIPLINA	NUMERICO	2	DISCIPLINA A LA QUE PERTENECE LA ACTIVIDAD

TABLA III.4.1.1.E ESTRUCTURA DEL ARCHIVO HORARIOS.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
FOLIO	NUMERICO	9	NUMERO DE FOLIO ASIGNADO EN FORMA AUTOMATICO POR EL PROGRAMA AL DAR DE ALTA LA ACTIVIDAD EN CUESTION
COSTO	NUMERICO	6	PRECIO DE ENTRADA A LA ACTIVIDAD
DESCCOST	NUMERICO	3	DESCUENTO DISPONIBLE
LEYENDA	CARACTER	30	LEYENDA O COMENTARIO REFERENTE AL PRECIO Y/O DESCUENTO

TABLA III.4.1.1.F ESTRUCTURA DEL ARCHIVO COSTDESC.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
FOLIO	NUMERICO	9	NUMERO DE FOLIO ASIGNADO EN FORMA AUTOMATICO POR EL PROGRAMA AL DAR DE ALTA LA ACTIVIDAD EN CUESTION
OBSERVACIO	CARACTER	240	OBSERVACIONES REFERENTES A LA ACTIVIDAD, SI NO SON SUFICIENTES 240 CARACTERES PARA LOS NOMBRES, ENTONCES SE TENDRAN TANTOS REGISTROS CON EL MISMO NUMERO DE FOLIO COMO SEAN NECESARIOS.

TABLA III.4.1.1.G ESTRUCTURA DEL ARCHIVO OBSERVA.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
NUMDISCIPL	NUMERICO	2	NUMERO DE DISCIPLINA ASIGNADO EN FORMA AUTOMATICA A LA DISCIPLINA AL MOMENTO DE DARLA DE ALTA
DISCIPLINA	CARACTER	30	NOMBRE DE LA DISCIPLINA

TABLA III.4.1.1.H ESTRUCTURA DEL ARCHIVO DISCIPL.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
RECINTO	CARACTER	40	NOMBRE DEL RECINTO
DIRECCION	CARACTER	40	CALLE Y NUMERO
COLONIA	CARACTER	40	COLONIA EN LA QUE SE ENCUENTRA
DELEGACION	CARACTER	40	DELEGACION A LA QUE PERTENECE
CIUDAD	CARACTER	40	CIUDAD EN LA QUE SE ENCUENTRA
CP	CARACTER	5	CODIGO POSTAL
LADA	CARACTER	3	CLAVE LADA CORRESPONDIENTE
TEL1	CARACTER	9	NUMERO DE TELEFONO
TEL2	CARACTER	9	NUMERO DE TELEFONO

TABLA III.4.1.1.I ESTRUCTURA DEL ARCHIVO LUGARES.DBF

NOMBRE DEL CAMPO	TIPO DE CAMPO	LONGITUD	DESCRIPCION
FOLIO	NUMERICO	9	NUMERO DE FOLIO ASIGNADO EN FORMA AUTOMATICO POR EL PROGRAMA AL DAR DE ALTA LA ACTIVIDAD EN CUESTION
FECHA	FECHA	8	FECHA EN LA QUE SE INICIA LA ACTIVIDAD
HORA	CARACTER	5	HORA EN LA QUE INICIA LA ACTIVIDAD

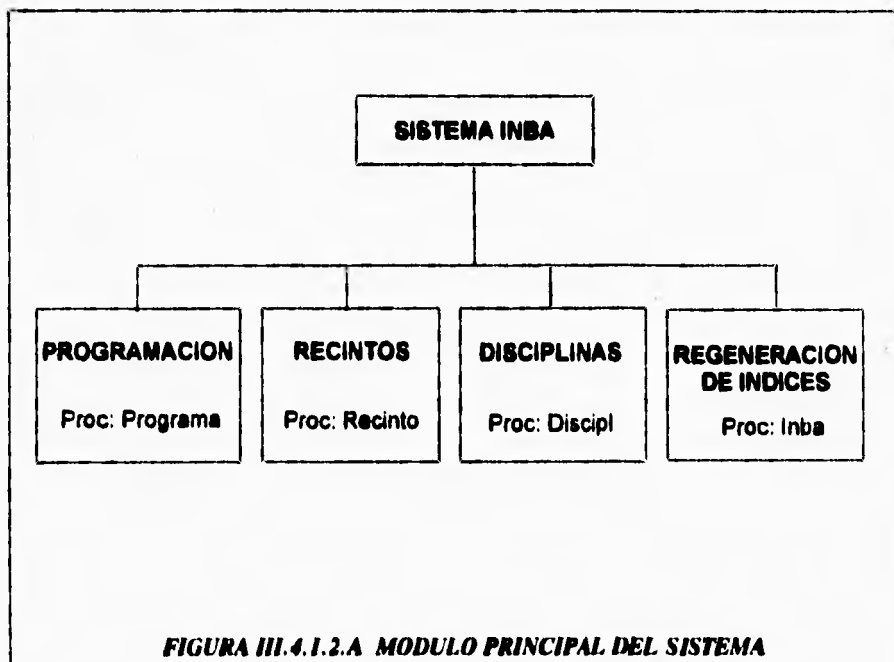
TABLA III.4.1.1.J ESTRUCTURA DEL ARCHIVO LISFECH.DBF

Esta base de datos se utiliza como archivo de paso para poder imprimir los reportes en forma cronológica, tanto por fecha como por hora, cada vez que se manda a imprimir algún archivo, primero se eliminan todos los registros.

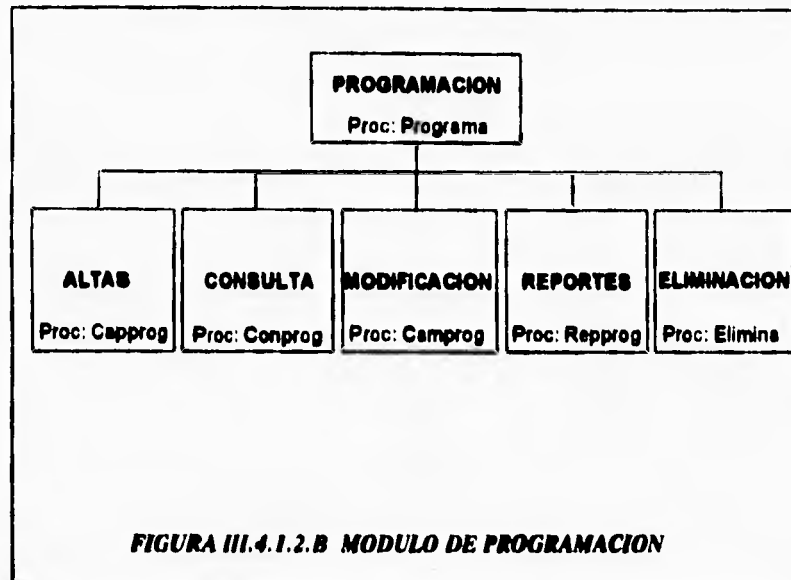
III.4.1.2 ESTRUCTURA DEL SISTEMA.

La estructura del sistema es modular, por esta razón es muy sencillo añadir opciones a cada módulo o submódulo.

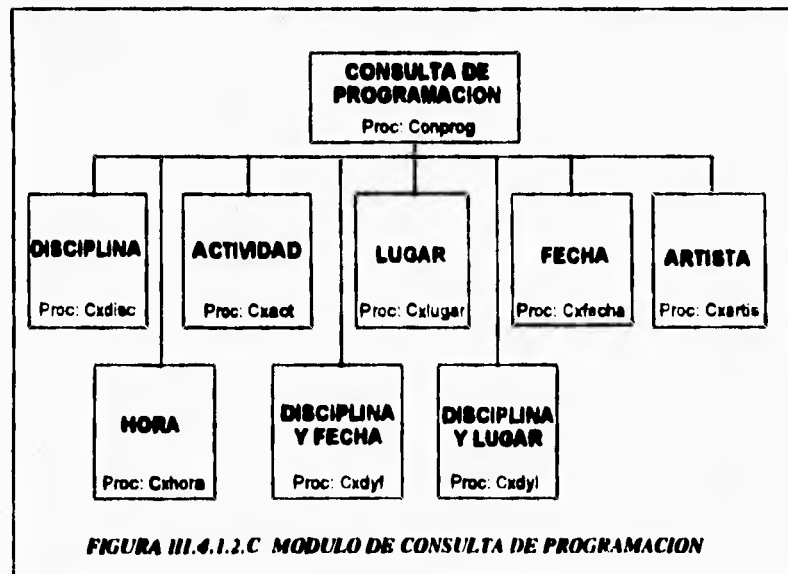
En la figura III.4.1.2.A podemos ver la estructura básica del sistema así como los nombres de los procedimientos correspondientes para cada módulo.



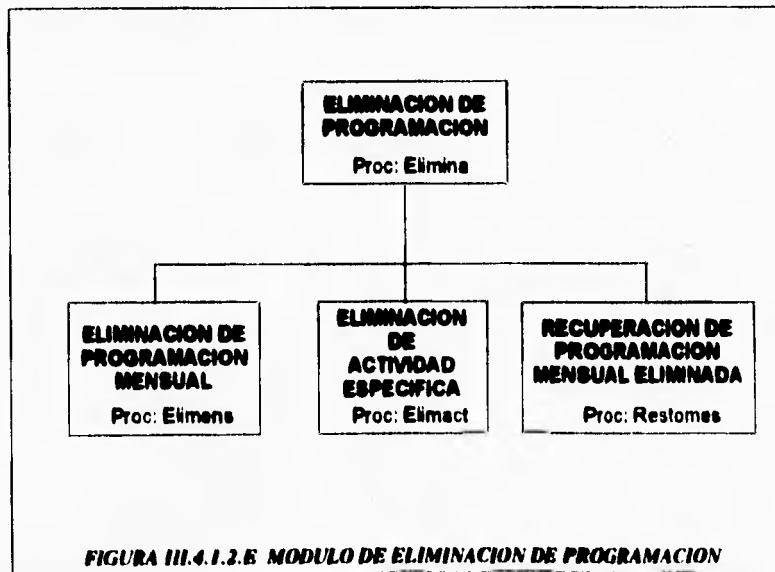
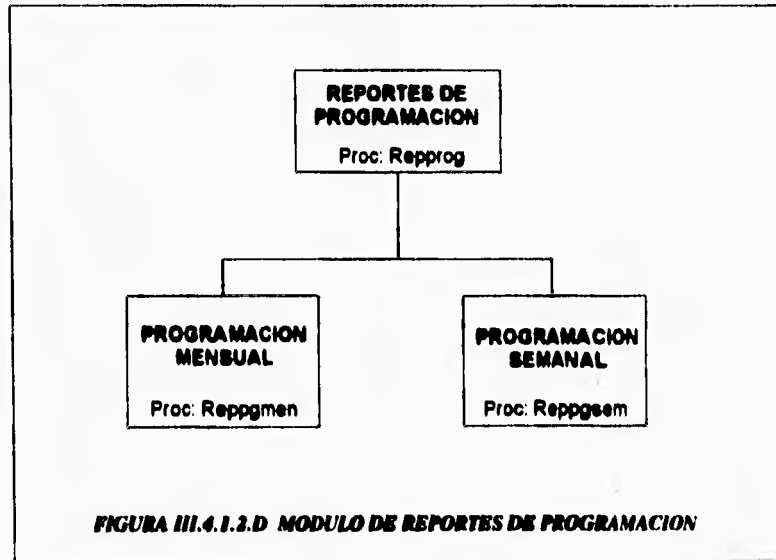
El módulo de programación se encuentra compuesto por los cinco submódulos que podemos ver en la figura III.4.1.2.B.



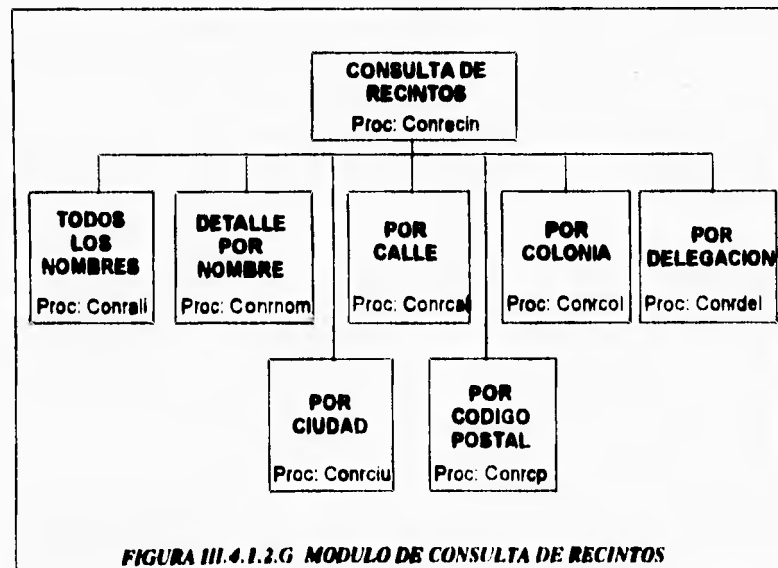
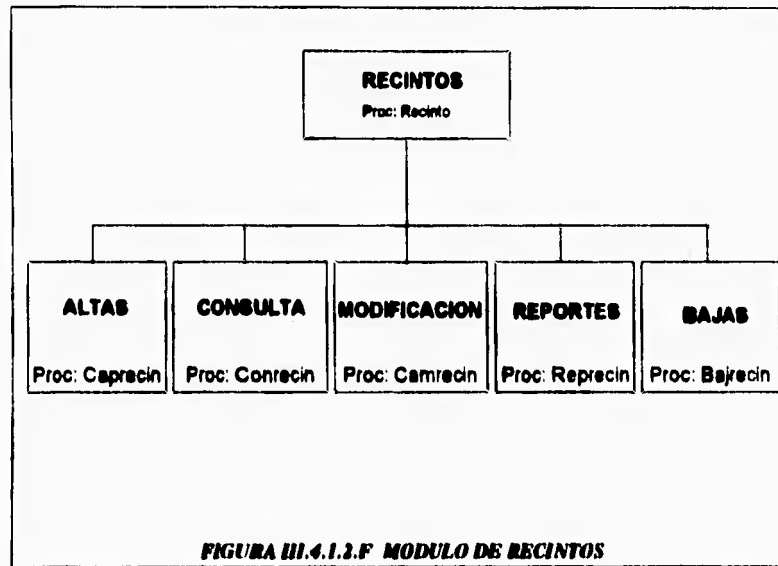
A su vez el módulo de consulta de programación se encuentra formado por los módulos que podemos ver en la figura III.4.1.2.C.



También los módulos de reportes y eliminación de programación están formados por varios módulos. Ver figuras III.4.1.2.D, III.4.1.2.E.

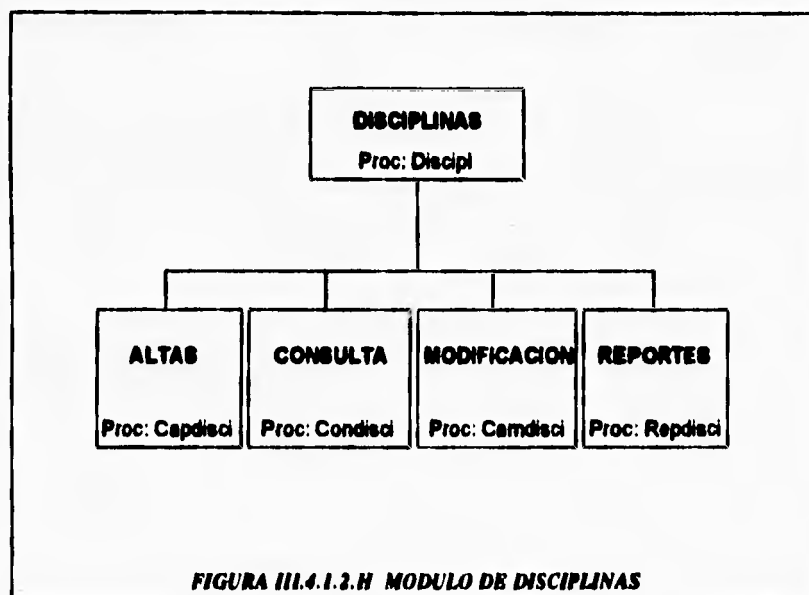


El módulo de recintos se encuentra compuesto por cinco submódulos, ver figura III.4.1.2.F.



En este caso, también el módulo de consulta de recintos está compuesto por otros submódulos. Ver figura III.4.1.2.G.

El módulo de disciplinas se encuentra compuesto por cuatro submódulos. Ver figura III.4.1.2.H.



El módulo regeneración de índices realiza una función que está integrada dentro del mismo módulo INBA (Proc. INBA).

En la figura III.4.1.2.I se observa la estructura general del sistema.

III.4.2 MANUAL DE USUARIO

III.4.2.1 INSTALACION

Este sistema se puede operar utilizando diskette o disco duro, sin embargo, se recomienda instalarlo en disco duro para tener un mejor tiempo de respuesta.

Instalación en diskette.

- 1. Copie el contenido del diskette del sistema a otro diskette, con la finalidad de tener siempre el diskette original sin modificaciones.**
- 2. Una vez copiado el diskette original, inserte el nuevo diskette en la unidad de disco A ó B, y seleccione tal unidad.**

Instalación en disco duro.

- 1. Cree un directorio llamado INBA (o asigne el nombre de su preferencia), tecleando las siguientes instrucciones:**

```
CD\  
MD INBA
```

- 2. Copie el contenido del diskette del sistema en dicho directorio, tecleando las siguientes instrucciones:**

COPY A:\C:\NBA /V

ó

COPY B:\C:\NBA /V

según la unidad de disco en la que se encuentre el disco del sistema.

COPY C:\NBA\NBA.BAT C:\ /V

III.4.2.2 OPERACION

Para entrar al sistema seleccione la unidad de disco en la que se encuentra instalado el sistema, teclee la palabra INBA y oprima la tecla [RETURN], aparecerá el menú principal, el cual tiene las siguientes opciones:

**PROGRAMACION
RECINTOS
DISCIPLINAS
REGENERAR INDICES**

Todos los menús del sistema operan de la siguiente manera; para seleccionar la opción deseada podrá utilizar las teclas de cursor para posicionar el cursor sobre la opción y al oprimir la tecla [RETURN] se activará otro menú o se entrará a la función correspondiente.

Para regresar al menú anterior oprima la tecla [ESC].

Para salir de cualquier función oprima la tecla [ESC].

Para poder operar correctamente el sistema, será necesario que después de instalar el sistema, se capturen las disciplinas y los datos de los recintos ya que cuando se capturen actividades el sistema validará que las disciplinas y recintos indicados estén dados de alta.

III.4.2.2.1 PROGRAMACION

Al seleccionar la opción de programación entraremos a un submenú desde el cual podremos seleccionar opciones para capturar, consultar, modificar y eliminar actividades así como emitir los reportes de programación mensual y semanal.

Captura.

Al entrar a la función de captura, el sistema le solicitará el nombre de la disciplina la cual pertenece la actividad, deberá ingresar el nombre de una actividad que esté dada de alta, de lo contrario el sistema le indicará que esa disciplina no existe.

el sistema pedirá la información correspondiente a la actividad que desea capturar, los campos de ACTIVIDAD, ARTISTA y OBSERVACIONES son de longitud ilimitada, es decir, puede ingresar desde un caracter hasta tantos como usted necesite, inicialmente el sistema desplegara cuatro renglones de 60 caracteres cada uno para estos campos, si requiere mayor espacio simplemente al llenar los cuatro renglones, ó parte de ellos, oprima la tecla [ENTER] y aparecerán

otros cuatro renglones y así sucesivamente hasta que deje uno o más renglones vacíos.

Los campos de FECHA y HORA son obligatorios, es decir, no puede dejar en blanco ninguno uno u otro, podrá capturar tantas fechas y horas como usted requiera para esa actividad, para indicar al sistema que ya asignó todas las fechas y horas de presentación correspondientes a la actividad deje en blanco la fecha y la hora. El campo ESPECIAL lo puede utilizar para indicar alguna anotación correspondiente a esa fecha y hora, por ejemplo, inicio de temporada, clausura, etc.

Cuando el sistema solicita el nombre del recinto en el que se llevará a cabo la actividad, usted deberá ingresar el nombre de un recinto que se encuentre dado de alta, de lo contrario el sistema le indicará que ese recinto no existe y le preguntará si desea darlo de alta. Si su respuesta es sí, aparecerá la pantalla de captura de recintos, al terminar de ingresar los datos, el sistema regresará a la pantalla inicial y continuará solicitándole el resto de la información.

Al igual que con la fecha y hora, usted podrá capturar tantos precios y descuentos como sean necesarios, dejando en blanco los campos PRECIO, DESCUENTO y LEYENDA para indicar al sistema que ya asignó todos los precios disponibles para la actividad. En el campo LEYENDA podrá indicar cualquier comentario referente al precio y/o descuento, por ejemplo, entrada libre, estudiantes y profesores con credencial (haciendo referencia al porcentaje de descuento), etc.

Al terminar de ingresar toda la información de la actividad, el sistema regresará a solicitar nuevamente el nombre de la disciplina a la cual pertenece la actividad, de

forma tal que pueda capturar tantas actividades como sea necesario sin que el sistema lo esté regresando al menú de PROGRAMACION.

Consulta.

Al seleccionar la función consulta, aparecerá un submenú de donde podrá seleccionar búsqueda por DISCIPLINA, ACTIVIDAD, LUGAR, RANGO DE FECHA, ARTISTA, HORA, DISCIPLINA Y FECHA , ó DISCIPLINA Y LUGAR, dependiendo de su selección el sistema le pedirá el campo clave por el cual desea consultar, a continuación aparecerá una lista de todas las actividades que cumplen con la solicitud de consulta, indicándose un número de folio y el nombre de la actividad (primeros 240 caracteres), al llenarse la pantalla, en la parte inferior se desplegará una leyenda que solicita el número de folio de la actividad que desea consultar, una "I" para desplegar nuevamente todas las actividades que coinciden con el nombre o fracción que se había indicado o una "T" para regresar a la pantalla en la que se solicita la condición de búsqueda.

Al ingresar el número de folio de la actividad a consultar aparecerá en la pantalla toda la información referente a esa actividad, al terminar de consultar la actividad, el sistema regresará a la pantalla que solicita el folio de actividades a consultar.

Modificación.

Al seleccionar la función modificación el sistema le pedirá el nombre de la disciplina que desea modificar, a continuación aparecerá una lista de todas las

actividades que cumplen con la solicitud de modificación, indicándose un número de folio y el nombre de la actividad (primeros 240 caracteres), al llenarse la pantalla, en la parte inferior se desplegará una leyenda que solicita el número de folio de la actividad que desea modificar, una "I" para desplegar nuevamente todas las actividades que coinciden con el nombre o fracción que se había indicado o una "T" para regresar a la pantalla en la que se solicita la condición de búsqueda.

Al ingresar el número de folio de la actividad a modificar aparecerá en la pantalla toda la información referente a esa actividad, permitiéndole editar la información utilizando las teclas de cursor, INS, DEL y Back Space. Al terminar de modificar la actividad, el sistema regresará a la pantalla que solicita el folio de actividades a modificar.

Reportes.

Al seleccionar la opción de reportes aparecerá un menú con las siguientes opciones:

Reporte mensual

Reporte semanal

Reporte mensual.

Si elige esta opción el sistema le solicitará el nombre del mes y el nombre de la disciplina de los cuales se desea el reporte de programación. Una vez proporcionados estos datos el sistema mandará a la impresora el reporte.

Reporte semanal.

Si elige esta opción el sistema le solicitará el periodo de la semana de la cual se desea el reporte de programación. Una vez proporcionados estos datos el sistema mandará a la impresora el reporte.

Eliminación.

Al seleccionar esta opción, el sistema desplegará un menú con las siguientes opciones:

Programación mensual

Actividad Individual

Recupera programación mensual

Programación mensual.

Esta opción es útil cuando se desea eliminar la programación de todo un mes el cuando se trata de un mes que ya terminó y no se requerirá de esa información.

Al seleccionar esta opción, el sistema solicitará el nombre del mes que se desea eliminar, a continuación pedirá que se inserte un diskette para respaldar la información que va a ser removida de la base de datos y pedirá que se indique en que unidad de disco se insertará el diskette. Una vez que se haya realizado el respaldo de la información el sistema eliminará toda la información correspondiente al mes indicado.

Actividad individual.

Esta opción es útil cuando se desea eliminar de la programación una actividad específica, cuando por ejemplo, se haya cancelado dicha actividad.

Al seleccionar esta opción, el sistema solicitará el nombre o fracción de nombre de la actividad que se desea eliminar, a continuación aparecerá una lista de todas las actividades que cumplen con la solicitud de eliminación, indicándose un número de folio y el nombre de la actividad (primeros 240 caracteres), al llenarse la pantalla, en la parte inferior se desplegará una leyenda que solicita el número de folio de la actividad que desea eliminar, una "I" para desplegar nuevamente todas las actividades que coinciden con el nombre o fracción que se había indicado o una "T" para regresar a la pantalla en la que se solicita la condición de búsqueda.

Al ingresar el número de folio de la actividad a eliminar aparecerá en la pantalla la información referente a esa actividad, y solicitará confirmación para eliminar esa actividad. Después de eliminar la actividad, el sistema regresará a la pantalla que solicita el folio de actividades a eliminar.

Recupera programación mensual.

Esta función se puede usar para cargar información de todo un mes que fue removida de la base de datos con la opción de eliminación mensual. El sistema le pedirá el nombre de la unidad de disco en la que se insertará el diskette del cual se recuperará la información.

III.4.2.2.2 RECINTOS

Al seleccionar la opción de programación entraremos a un submenú desde el cual podremos seleccionar opciones para capturar, consultar, modificar, eliminar, e imprimir el catálogo de recintos.

Captura.

Al entrar a la función de captura, el sistema le solicitará los datos generales del recinto, como son, calle, colonia, delegación, ciudad, código postal, clave lada y dos números telefónicos. Al terminar de proporcionar estos datos, el sistema le preguntará si los datos son correctos, si responde que no el sistema volverá a pedirle los datos, de lo contrario los grabará y regresará a pedirle nuevamente más datos para otro recinto. Para regresar al menú anterior deberá oprimir la tecla [ESC].

Consulta.

Al seleccionar esta opción el sistema desplegará un submenú desde el cual podrá seleccionar opciones para consultar todos los nombres de recintos existentes, consultar detalle por nombre, calle, colonia, delegación, ciudad, ó código postal.

En el caso de consulta de todos los nombres, al seleccionar esta opción se desplegarán en la pantalla la lista de todos los nombres de los recintos existentes, en el caso de las consultas de detalle, dependiendo de cada una, el sistema

solicitará el nombre, número o fracción de estos y desplegará en la pantalla el nombre de todos los recintos que cumplen con la condición de consulta así como un número de registro correspondiente a cada uno de ellos, al llenarse la pantalla, en la parte inferior se desplegará una leyenda que solicita el número de folio de la actividad que desea eliminar, una "I" para desplegar nuevamente todas los recintos que coinciden con el nombre o fracción que se había indicado o una "T" para regresar a la pantalla en la que se solicita la condición de búsqueda.

Modificación.

Al seleccionar esta opción el sistema solicitará el nombre del recinto a modificar, si el nombre del recinto proporcionado no existe el sistema lo indicará, de lo contrario se desplegarán los datos generales del recinto para que puedan ser editados, al terminar de editar, el sistema preguntará si los datos están correctos, si se responde que no, el sistema ignorará los cambios efectuados y solicitará nuevamente el nombre del recinto, de lo contrario se registrarán los cambios y pedirá nuevamente un nombre de recinto a editar. Para regresar al menú anterior oprima la tecla [ESC].

Bajas.

Al seleccionar esta opción el sistema solicitará el nombre del recinto que se desea dar de baja, si el nombre del recinto proporcionado no existe el sistema lo indicará, de lo contrario se desplegarán los datos generales del recinto para que puedan ser confirmados, a continuación el sistema preguntará si se da de baja, si se responde que no, el sistema ignorará la solicitud de baja y solicitará

nuevamente el nombre del recinto, de lo contrario se eliminará ese recinto del catálogo y pedirá nuevamente un nombre de recinto a dar de baja. Para regresar al menú anterior oprima la tecla [ESC].

Reportes.

Al seleccionar esta opción, el sistema le preguntará si desea imprimir el reporte, si indica que no, el sistema regresará el menú anterior de lo contrario el sistema mandará a impresora el catálogo de recintos.

III.4.2.2.3 DISCIPLINAS

Al seleccionar la opción de programación entraremos a un submenú desde el cual podremos seleccionar opciones para capturar, consultar, modificar, e imprimir el catálogo de disciplinas.

Captura.

Al entrar a la función de captura, el sistema le solicitará al nombre de la disciplina. Al terminar de proporcionar el nombre, el sistema le preguntará si el dato es correcto, si responde que no el sistema volverá a pedirle el dato, de lo contrario lo grabará y regresará a pedirle nuevamente un nombre de disciplina para darlo de alta. Para regresar al menú anterior deberá oprimir la tecla [ESC].

Consulta.

Al seleccionar esta opción el sistema desplegará en la pantalla una lista de todas las disciplinas que están dadas de alta. Para regresar al menú deberá oprimir la tecla [ENTER].

Modificación.

Al seleccionar esta opción el sistema solicitará el nombre de la disciplina a modificar, si el nombre de la disciplina proporcionada no existe el sistema lo indicará, de lo contrario se desplegará el nombre de dicha disciplina para que lo pueda editar y corregir, al terminar de editar, el sistema preguntará si el dato está correcto, si se responde que no, el sistema ignorará los cambios efectuados y solicitará nuevamente el nombre de la disciplina, de lo contrario se registrará el cambio y pedirá nuevamente un nombre de disciplina a editar. Para regresar al menú anterior oprima la tecla [ESC].

Reportes.

Al seleccionar esta opción, el sistema le preguntará si desea imprimir el reporte, si indica que no, el sistema regresará el menú anterior de lo contrario el sistema mandará a impresora el catálogo de disciplinas.

III.4.2.2.4 INDICES (REGENERA)

Al seleccionar esta opción el sistema regenerará todos los archivos de índices utilizados por el sistema. Esta opción es útil cuando por alguna razón se haya dañado alguno de los índices.

III.5 IMPLANTACION Y CAPACITACION

III.5.1 IMPLANTACION

La estrategia de implantación del sistema es la siguiente:

El metodo actual se aplicará en paralelo a los siguientes puntos.

- 1. Captura de las disciplinas**
- 2. Captura de los recintos.**
- 3. Depuración de disciplinas y recintos**
- 4. Captura de actividades del mes siguiente**
- 5. Actualización de información de actividades hasta la fecha límite de recepción de información para el reporte mensual.**
- 6. Impresión del reporte mensual**
- 7. Comparación del reporte emitido por el sistema con el reporte tradicional.**
- 8. Si los datos del reporte tradicional concuerdan con los datos del reporte del sistema, distribuir a un grupo de destinatarios internos, es decir, de la subdirección de relaciones públicas del I.N.B.A.**

9. Actualización de información de actividades hasta la fecha límite de recepción de información para el reporte semanal.

10. Impresión del reporte semanal

11. Comparación del reporte emitido por el sistema con el reporte tradicional.

12. Si los datos del reporte tradicional concuerdan con los datos del reporte del sistema, distribuir a un grupo de destinatarios internos, es decir, de la subdirección de relaciones públicas del I.N.B.A.

13. Los pasos 4 al 5 se repetirán en paralelo con la ejecución de los pasos 9 al 12. Repetir los pasos 9 al 12 durante cuatro semanas y los pasos 6 al 8 se realizarán al fin del periodo. Si dos reportes del sistema concuerdan con los dos reportes tradicionales tanto en datos como en tiempo de emisión, a partir de ese momento se podrá suspender la elaboración en paralelo del método tradicional, desarrollándose únicamente la captura en el nuevo sistema.

III.5.2 CAPACITACION

El programa de capacitación se dividirá en tres fases:

Captura

Actualización

Consulta y emisión de reportes

Captura.

Para el proceso de captura, el persona a capacitarse deberá ingresar la información de todas las disciplinas y recintos. Esta información es proporcionada por la subdirección de relaciones públicas del I.N.B.A.

Adicionalmente deberán ingresar la información de las actividades de algún mes, de preferencia la del mes siguiente a la fecha en la que se inicie la capacitación. Esta información también es proporcionada por la subdirección de relaciones públicas del I.N.B.A.

Actualización.

Para esta etapa, las personas a capacitarse deberán recibir por parte de la subdirección de relaciones públicas del I.N.B.A. las modificaciones que se presenten en el periodo anterior a la fecha límite de recepción de información para la emisión del reporte mensual, con la finalidad de actualizar la información que haya variado hasta esa fecha.

III.5.2 CAPACITACION

El programa de capacitación se dividirá en tres fases:

Captura

Actualización

Consulta y emisión de reportes

Captura.

Para el proceso de captura, el persona a capacitarse deberá ingresar la información de todas las disciplinas y recintos. Esta información es proporcionada por la subdirección de relaciones públicas del I.N.B.A.

Adicionalmente deberán ingresar la información de las actividades de algún mes, de preferencia la del mes siguiente a la fecha en la que se inicie la capacitación. Esta información también es proporcionada por la subdirección de relaciones públicas del I.N.B.A.

Actualización.

Para esta etapa, las personas a capacitarse deberán recibir por parte de la subdirección de relaciones públicas del I.N.B.A. las modificaciones que se presenten en el periodo anterior a la fecha límite de recepción de información para la emisión del reporte mensual, con la finalidad de actualizar la información que haya variado hasta esa fecha.

Consulta y emisión de reportes.

Una vez que se haya llegado a la fecha límite de recepción de información, se deberá imprimir el reporte mensual. Una vez que se tenga el reporte impreso, se deberán realizar diversas consultas de esta información, debiendo cubrir todas las opciones de los menús de consulta.

Finalmente se deberán imprimir los reportes semanales correspondientes al mes en cuestión.

Esta capacitación se deberá realizar utilizando el manual de usuario del sistema.

CONCLUSIONES

CONCLUSIONES

- **La elaboración del presente trabajo nos permitió aplicar, a un problema práctico, los conocimientos adquiridos durante nuestra formación profesional en la Facultad de Ingeniería, así como las habilidades y conocimientos adquiridos por medio de la experiencia laboral.**
- **También pudimos refrescar y actualizar nuestros conocimientos sobre la teoría de las bases de datos y sobre los manejadores de bases de datos que existen en el mercado para computadoras personales.**
- **Pudimos ver, que aún cuando ciertos manejadores ofrecen tiempos de respuesta excelentes y ayudas para el desarrollo de aplicaciones, no siempre son los más adecuados para cubrir los requerimientos del usuario.**
- **Al finalizar el proyecto, vimos la gran importancia de asignar bastante tiempo a la obtención de datos con el usuario, ya que el tener una clara definición de la problemática y de los requerimientos específicos del usuario, nos permite ahorrar tiempo en el desarrollo, ya que se eliminan posibles cambios posteriores que pueden impactar el tiempo de entrega y/o calidad del producto.**
- **Sabemos que el desarrollo de éste sistema aporta una herramienta de servicio a la Subdirección de Relaciones Públicas del Instituto Nacional de Bellas Artes, con la cual podrán ofrecer un mejor servicio a todos sus usuarios.**

5

GLOSARIO

GLOSARIO

TERMINOLOGIA DE HARDWARE

CLASES DE COMPUTADORAS

Principalmente existen dos clases de computadoras electrónicas, analógicas y digitales, que difieren en base a sus características de entrada y salida de información y a su forma de operación. También existe una tercera clase que es una mezcla de las dos anteriores, estas son las computadoras híbridas.

COMPUTADORAS ANALOGICAS

El nombre de computadoras analógicas se debe a que, en base a sus características se pueden hacer analogías o semejanzas con procesos físicos, químicos o matemáticos. En una computadora analógica los números se representan por medio de corrientes y voltajes eléctricos.

COMPUTADORAS DIGITALES

En una computadora digital, toda la información es transferida en forma discreta la cual involucra señales representadas por dos estados. Las señales binarias están creadas por la presencia o ausencia de corriente eléctrica.

HARDWARE

Término empleado para referirse a los componentes mecánicos, magnéticos, electrónicos de una computadora, es decir la parte física de un sistema de computo.

MEMORIA.

Dispositivo para almacenar información. La memoria de núcleos magnéticos tiene una gran ventaja está es su alta velocidad. Esta velocidad se debe a que no emplea sistemas mecánicos para admitir o dar información, por esta razón los núcleos magnéticos son ideales para las memorias centrales de alta velocidad que se emplean en los grandes sistemas de computación. Los núcleos magnéticos están contruidos de un material fácilmente magnetizable (ferrita) y por esto se puede saturar magnéticamente, dando una polaridad de 1 ó 0. Por lo tanto, un núcleo puede tener sólo un estado en un tiempo dado, esto es, sólo puede tener un estado de saturación, ya sea uno ó cero. La magnetización en los núcleos de ferrita puede lograrse de igual manera que cuando se magnetiza un imán de barra, esto es, enrollándole un alambre por el cual se deja pasar una cantidad de corriente. La fuerza magnetizadora dependerá del número de vueltas del alambre y de la intensidad de corriente. Las memorias de núcleos se construyen en forma de matriz. Estas memorias tienen una gran cantidad de núcleos, que son muy pequeños. Si se desea magnetizar solo un núcleo de la matriz, lo que se hace es pasar la mitad de la corriente en un alambre vertical y la otra mitad de corriente en otro alambre horizontal.

MEMORIA RAM (Random Access Memory)

Debe su nombre a la habilidad de direccionar cualquier localidad en cualquier tiempo, sin importar cual fué la última localidad accesada. El CPU obtiene de la RAM todas las instrucciones de un programa, así como los datos para llevar a cabo una tarea. Existen dos tipos de RAM, ambas pierden la información si se apaga la micro.

- Dynamic RAM(DRAM) : Consiste de capacitores (minúsculos), la computadora debe refrescar constantemente el contenido.

- **Static RAM(SRAM)** : Funciona como una serie de switches, una vez que un switch esta en 1 ó 0 ya no cambiará, no necesita refrescarse, es la más rápida y la más costosa.

ROM (Read Only Memory)

Es un tipo de memoria en la que se puede almacenar datos y programas permanentemente, usualmente es una cantidad fija en la que se guardan las instrucciones básicas para entrada y salida de datos(de ahí ROM BIOS), y el procedimiento de Boot, la ROM difiere de la memoria RAM en que la información es permanente.

EPROMS (Memoria de solo lectura programable y borrable)

Suministran al computador la información para el arranque. Pero son memorias lentas, y su información solamente puede ser modificada si antes se le expone a luz ultravioleta.

UNIDAD ARITMETICA-LOGICA

El propósito de la unidad aritmética es el de ejecutar todas las operaciones aritméticas básicas y las operaciones lógicas como son los corrimientos y comparaciones. Para lograr éste propósito, la unidad aritmética se compone de una serie de circuitos tales como: registros, sumadores, contadores y comparadores.

UNIDAD CENTRAL DE PROCESO

La unidad central de proceso es el mecanismo de trabajo primario en donde todos los datos necesarios se almacenan en un primer paso del procesamiento. Después toma una decisión acerca del tipo de cálculos que se deban efectuar sobre los datos almacenados. Al almacenamiento interno o primario; a cualquier tipo de datos que no se encuentren almacenados en esta unidad se les denominará almacenamiento externo o secundario. El tiempo promedio que una computadora requiere para localizar los datos

o para recabar información de alguna unidad de memoria, sea primaria o secundaria se le llama "Tiempo promedio de acceso". La unidad central de proceso se divide en memoria, unidad aritmética-lógica, unidad de control, registros(acumulador, registro índice, contador de programa, apuntadores de pila, timers, etc.).

UNIDAD DE CONTROL

La unidad de control es muy importante entre las computadoras digitales electrónicas, ya que de hecho, debido a ésta unidad, las computadoras son automáticas.

De las cuatro partes fundamentales en que se divide una computadora, la más importante es la Unidad de Control, ya que ésta unidad coordina las tres restantes (unidad de entrada y salida, unidad aritmética y la memoria).

La rapidez con que la computadora efectúa sus cálculos también se debe en gran parte a la Unidad de Control, ya que sigue una secuencia de operación bien definida, basada en las instrucciones almacenadas en el programa que está en memoria.

La secuencia o ciclo básico de operación es :

- Busca una instrucción almacenada en programa, y la interpreta.
- Ejecuta la instrucción. Cuando se termina la ejecución regresa y busca la siguiente instrucción en la secuencia, por medio de la repetición constante del proceso.

De ésta forma, una computadora ejecuta automáticamente el programa de instrucciones hasta terminarlo.

El trabajo fundamental de los circuitos de control es seguir éste ciclo de operación, que se debe efectuar en todas las computadoras que operen automáticamente, aunque la forma en la cual se efectúe la ejecución varía considerablemente de una computadora a otra.

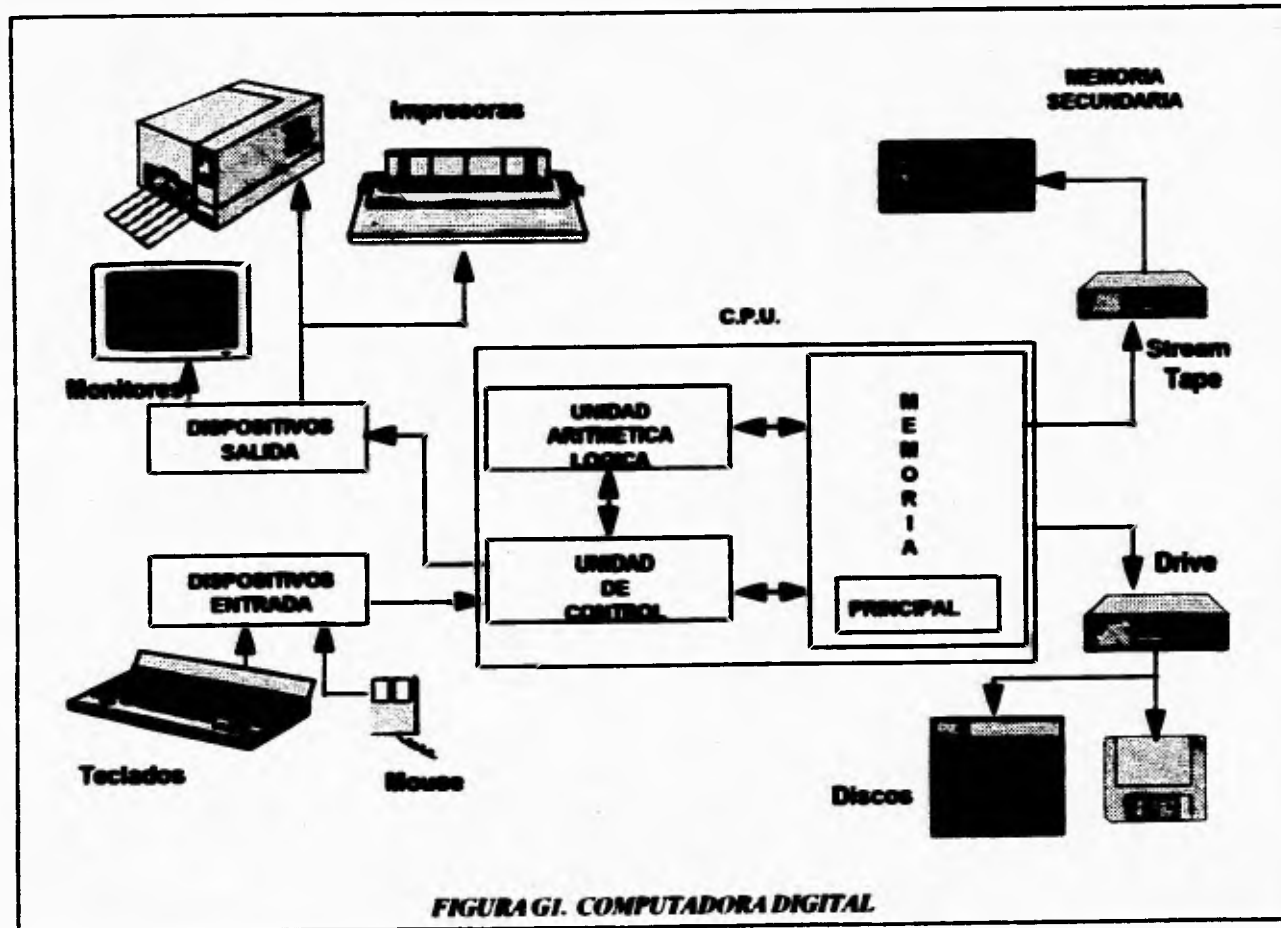
UNIDAD DE ENTRADA Y SALIDA

La función de entrada y salida de datos, como su nombre lo dice, es aquella que consiste en introducir y extraer información de un sistema electrónico de procesamiento de datos.

Esta función se lleva a cabo a través de los dispositivos de Entrada / Salida (teclado, monitor, impresora, lector óptico, discos duro, diskettes, etc.).

Para que la unidad central de proceso pueda trabajar, es necesario primero que le demos las instrucciones y los datos. Del mismo modo, para poder disponer de los datos procesados, habrá que ordenarle que ejecute la función de introducir instrucciones, de proporcionar y extraer información de la unidad central de proceso, la llevan a cabo precisamente los dispositivos de Entrada / Salida.

En la figura G1 se observa las principales características y funcionamiento de una computadora digital.



FIGURAGI. COMPUTADORA DIGITAL

TERMINOLOGIA DE DATOS

ALGORITMO

Serie de pasos necesarios para resolver un problema.

ARCHIVO (file).

Una colección de registros de un grupo de entidades con ciertos aspectos en común y organizados para algún propósito en particular, es llamado archivo, por ejemplo los datos de todos los registros de los pasajeros de un vuelo. A nivel físico un archivo se considera como una colección de registros físicos, los cuales deben residir, casi siempre de manera continua en memoria externa. El acceso correcto a los registros físicos de un archivo, es guardado en tablas almacenadas por medio de rutinas de manejo de memoria del sistema operativo, estas tablas contienen toda la información del manejo físico del registro (como el tamaño del registro), y este manejo es transparente para el usuario que esta trabajando con el archivo a nivel lógico.

ATRIBUTO (campo, ítem).

Características que identifican la entidad, o sea una unidad de información significativa de la entidad, un atributo es casi siempre llamado un campo o ítem, como ejemplo podrían ser los diferentes datos de un pasajero de un vuelo, como su nombre, número de asiento. Generalmente, un campo de un registro es un entero, real, caracter o una cadena de caracteres (figura G2).

BASES DE DATOS DISTRIBUIDAS

Es una Base de Datos no almacenada en su totalidad en un solo lugar físico, sino se distribuye a lo largo de una red de computadoras geográficamente separadas que se conectan por medio de enlaces de comunicación.

BIT.

Mínima unidad de almacenamiento.

BYTE.

La unidad mas pequeña de direccionamiento de memoria, formado por 8 bits.

CARDINALIDAD

Es el número de tuplas de una relación.

COMPILADOR

Traductor de lenguaje que convierte un programa escrito en lenguaje de alto nivel a un programa equivalente en un nivel más bajo (como el lenguaje de máquina) para ser ejecutado después.

CONCURRENCIA

Múltiples usuarios accesan al mismo tiempo la misma información.

ENTIDAD.

Una entidad es una persona, lugar, cosa, evento o concepto acerca del cual la información es grabada.

DBA(database administrator)

Uno o más individuos que manejan la Base de Datos institucional. Esta persona o grupo de personas deben de poseer las siguientes características :

- **Alto grado técnico (expertos).**
- **Habilidad para entender e interpretar los requerimientos de información de la institución a nivel de alta administración.**

Las principales responsabilidades de un DBA serán :

- **Decidir el contenido de la Base de Datos.**
- **Decidir la estructura de almacenamiento y estrategias de acceso (modelado físico).**
- **Definir las validaciones de seguridad e integridad.**
- **Definir estrategias para respaldos y recuperación.**
- **Monitorear el rendimiento de la Base de Datos y actuar en consecuencia.**
- **Manejo del Diccionario de Datos del sistema.**
- **Uso de las diferentes rutinas o utilerías de la Base de Datos, como carga de archivos externos a la Base de Datos, vaciado y almacenamiento, estadísticas, de análisis de rendimiento**

DBMS(database management system)

Es un sistema de software usado para manejar y mantener datos de una o múltiples aplicaciones al mismo tiempo para diferentes propósitos, independientemente de la clase de dispositivos de almacenamiento o métodos de acceso. Parte de las tareas de un DBMS serán :

- **mantener la consistencia de los datos.**
- **resolver los problemas de concurrencia.**
- **proveer una interfaz universal a los datos.**
- **regular el acceso a los datos.**

En general, un DBMS debe incorporar :

- Independencia de los programas de aplicación respecto a los cambios en la estructura de los datos.
- Soporte de lenguajes de programación usados en el ambiente corporativo antes de la instalación.
- Programas de utilidad para facilitar la creación, mantenimiento y reestructuración de las Bases de Datos.
- Facilidades para la reorganización de los Datos.
- Habilidad para afectar la seguridad de los datos e imponer límites de accesos a ellos.
- Capacidad de reinicio (Restart) automático en caso de falla del sistema.
- Habilidad para recuperar las operaciones en forma manual con esfuerzo mínimo.
- Facilidad del sistema para la "afinación" del DBMS.

Opcionalmente :

- Capacidad de proceso de transacciones (OLTP).
- Facilidad de consulta a la Base de Datos (Query Language).
- Generador de reportes (Report Writer).
- Diccionario de Datos.

DDL (data definition language)

Es el lenguaje que permite la descripción de los objetos de la BD (tal como los percibe el usuario).

DICCIONARIO DE DATOS

Es una librería central para definir el significado, uso, característica y otros datos relevantes de todos los data items, campos, entidades, sinónimos, referencias cruzadas y

las relaciones que existen entre ellos. Se puede decir que un DD es una Base de Datos que contiene datos acerca de los datos que contiene la Base de Datos.

DML(data manipulation language)

Es el lenguaje que apoya el manejo o procesamiento de los objetos que describe el DDL.

DOMINIO

Se define como el rango o posibles valores que puede tener un atributo.

DSL(data SubLanguage)

Es un subconjunto del lenguaje total que concierne a los objetos y a las operaciones de la Base de Datos.

INCONSISTENCIA

Se entiende por inconsistencia de datos a que a peticiones similares se obtengan diferentes salidas.

INGENIERIA DE SOFTWARE

Es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido. La IS difiere de la programación tradicional en que se utilizan técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto establecidos para el proyecto; además esta ingeniería se preocupa por aspectos administrativos que quedan fuera del dominio normal de la programación.

INTEGRAL

Es la característica que tiene una Base de Datos a diferencia de un sistema tradicional de archivos.

INTEGRIDAD

Reglas que los datos deben cumplir, estas reglas son dictadas por el mundo real.

LLAVES(primaria, candidata, foránea)

LLAVE PRIMARIA

Se le llama al atributo o atributos que identifica en forma única a cada renglón de la tabla o relación y, por tanto, se pueden usar para identificar las tuplas de esa relación. Una llave primaria no puede ser nula.

LLAVE CANDIDATA

Cuando se encuentra una relación donde hay más de una combinación de atributos que poseen la propiedad de identificación única, en este caso cualquier opción es válida para pasar a ser una llave primaria.

LLAVE FORANEA

Es un atributo (o grupo de atributos) de una relación, que es (son) llave primaria en alguna otra parte.

NORMALIZACION

Se dice que una relación está en una forma normal particular si satisface cierto conjunto específico de restricciones, estas se verán con detalle en el punto 1.2.4.

ORGANIZACIONES DE ARCHIVOS.

El primer factor el cual determina la organización de un archivo son las operaciones que se efectuaran al archivo, (solo lectura, actualización, etc), esto es dictado por la aplicación. La organización de un archivo también depende del tipo de almacenamiento externo de memoria en el cual residirá la información, o sea la organización de un archivo que esta en un dispositivo de almacenamiento secuencial (cintas magnéticas), difiere en gran forma de la organización de un archivo que se encuentra en un dispositivo que puede ser accesado de manera directa, (como discos magnéticos).

Los tipos mas comunes de organización de archivos son:

- Archivo secuencial.
- Archivo secuencial indexado.
- Archivo de acceso directo.

ARCHIVO SECUENCIAL.

En un archivo secuencial los registros son almacenados uno después de otro sobre un dispositivo de almacenamiento. Todos los dispositivos de almacenamiento externo soportan la organización de archivo secuencial. Algunos dispositivos por su naturaleza física soportan solo archivos secuenciales, por ejemplo la información almacenada en una cinta magnética tiene series de registros continuos a lo largo de toda la longitud de la cinta. El acceder un registro particular de la cinta requiere de un acceso de todos los registros previos en el archivo. Otros dispositivos los cuales sean estrictamente para archivos secuenciales son: cassettes e impresoras en línea.

Los discos magnéticos y tambores permiten ambos accesos, directo y secuencial. Un archivo secuencial esta físicamente grabado en un disco con registros secuenciales en localizaciones adjuntas sobre un mismo track, si un archivo es mas grande que el espacio disponible en un track entonces los registros son grabados en un track adjunto, esta forma de almacenamiento puede ser extendida aún a cilindros y aún a dispositivos

de almacenamiento completos donde más de un dispositivo es utilizado como una unidad común de control. Las operaciones que se pueden realizar sobre un archivo secuencial dependen del tipo de dispositivo de almacenamiento. Por ejemplo un archivo sobre una cinta magnética puede ser un archivo de entrada o un archivo de salida, pero no ambos al mismo tiempo. Un archivo almacenado sobre un disco puede ser estrictamente de lectura, estrictamente de escritura o de escritura / lectura. O sea que el último registro leído puede ser reescrito sobre el mismo archivo.

ARCHIVO SECUENCIAL INDEXADO.

La capacidad de acceso de direccionamiento a un registro se basa en una llave (o índice único), puede ser grabado si el dispositivo de almacenamiento externo usado soporta este tipo de acceso, por lo tanto el acceso directo al registro es imposible para dispositivos como cintas magnéticas y cartuchos. Los dispositivos de almacenamiento externo que soportan ambos accesos directo y secuencial, son discos magnéticos y tambores.

El archivo se crea de manera secuencial de acuerdo al orden de la llave de los registros, y se crea automáticamente por el manejador de rutinas de datos del sistema operativo del área de índices.

ARCHIVOS DE ACCESO DIRECTO.

En un archivo directo (también llamado random), una transformación o mapeo es efectuado de la llave de un registro a la dirección de la localidad de almacenamiento de ese registro en el archivo. Un mecanismo para generar estas transformaciones es llamado algoritmo hashing.

RECUPERACION

Capacidad de restaurar la integridad y consistencia de la Base de Datos después de una falla del sistema.

REDUNDANCIA

Se le llama redundancia de datos cuando los mismos datos están grabados en múltiples archivos de datos, o sea, se encuentran repetidos.

REGISTRO (record).

Es una colección de información (campos, items) acerca de una entidad particular, por ejemplo un registro puede consistir de la información de un pasajero de cierto vuelo.(figura G2)

RELACION

Dada una serie de conjuntos D_1, D_2, \dots, D_n (no necesariamente distintos), se dice que R es una relación sobre estos n conjuntos si es un conjunto de n tuplas ordenadas $\langle d_1, d_2, \dots, d_n \rangle$ tales que d_1 pertenece a D_1 , d_2 pertenece a D_2 , ..., d_n pertenece a D_n . Los conjuntos D_1, D_2, \dots, D_n son los dominios de R . El valor n es el grado de R . Es importante mencionar que es conveniente representar una R en forma de tabla.

SEGURIDAD

Protección de los datos contra accesos, modificaciones ó pérdidas, ya sea en forma intencional o no intencional.

SISTEMA OPERATIVO

Software que controla el conjunto de las operaciones efectuadas por una computadora, administrando el uso de los recursos para optimizar el rendimiento del sistema.

SISTEMA DE ARCHIVOS.

Es un conjunto de archivos, donde los registros de estos archivos tienen relación entre sí que utilizados en cierta área de aplicación generan un Sistema de Información..

SOFTWARE

Son los programas, rutinas o instrucciones que permiten a la computadora ejecutar tareas. Algunos ejemplos de "SOFTWARE" incluyen sistemas operativos, procesadores de palabras y hojas de cálculo.

SQL(STRUCTURED QUERY LANGUAGE)

Es un sublenguaje de datos que en un principio nació para un sistema relacional llamado System R. Sus operadores funcionan casi siempre en términos de conjuntos.

TRANSACCION

Una operación particular que se efectúa sobre un registro, del conjunto de registros (archivo), es llamada transacción, generalmente las operaciones que se efectúan son actualización, inserción, borrado y lectura de un registro.

TUPLA

Se le llama a cada uno de los renglones que conforman una tabla o relación. (figura G2)

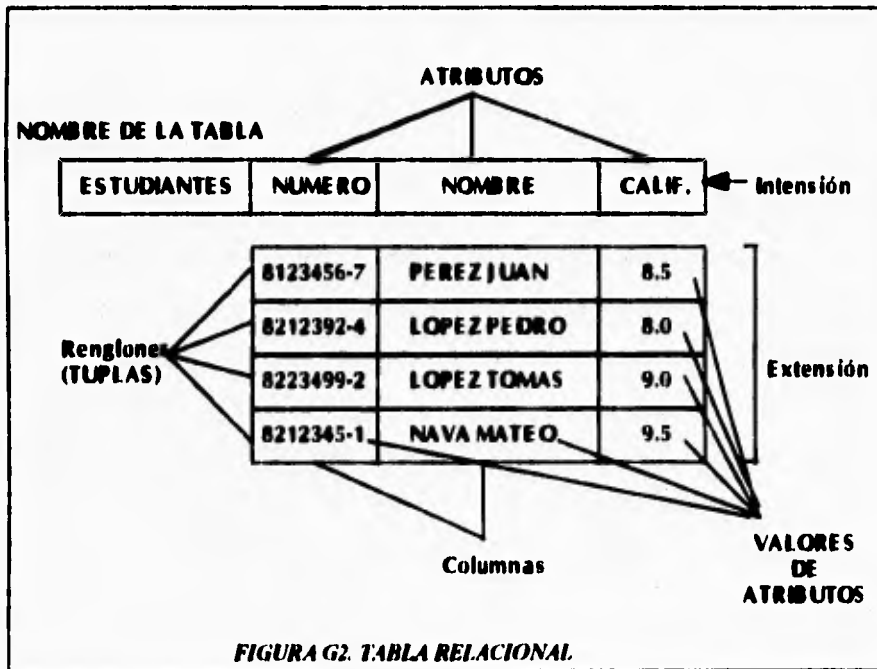


FIGURA G2. TABLA RELACIONAL

ALMACENAMIENTO FISICO

CINTAS MAGNETICAS

Un medio de almacenamiento continuo, flexible, grabable cuyo material está impregnado o recubierto con una sustancia magnética sensitiva listo para aceptar datos en forma de puntos polarizados magnéticamente.

DISPOSITIVOS DE ALMACENAMIENTO DE ACCESO DIRECTO.

Dispositivo de almacenamiento directo (DASD, direct access storage device), es aquél donde cada registro físico tiene una ubicación discreta y una dirección exclusiva. Los discos y los tambores son dispositivos de almacenamiento directo. En los dispositivos de acceso directo, los registros se almacenan de tal manera que es posible encontrar cualquiera de ellos sin mucha explotación del medio de almacenamiento. En un disco la información es transferida a través de las cabezas lectoras, cada cabeza lectora flota debajo o arriba de la superficie del disco mientras el disco gira constantemente a alta velocidad, en la siguiente figura se muestra la organización física de un disco.

A continuación se muestra la figura G3, que describe la organización física de un disco duro.

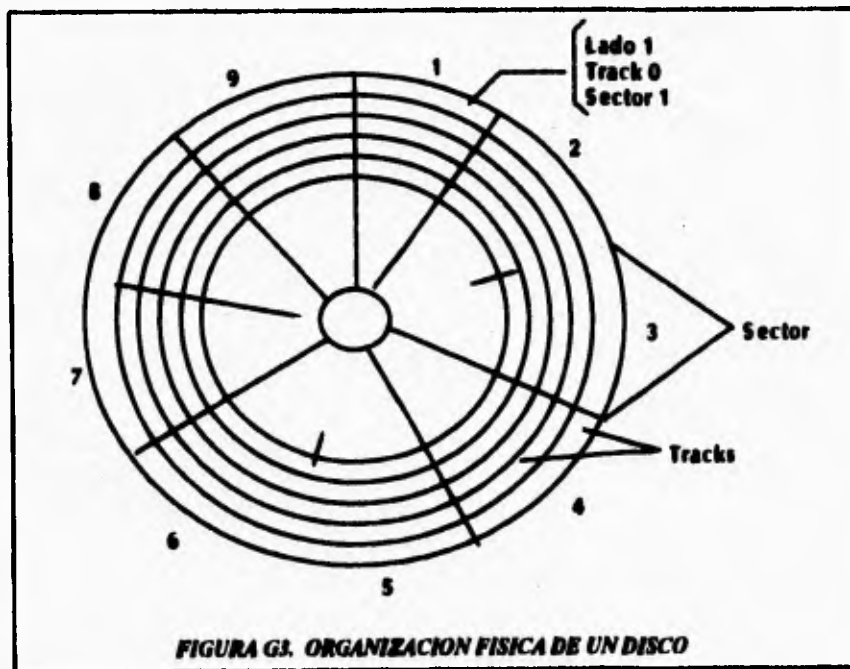


FIGURA G3. ORGANIZACION FISICA DE UN DISCO

BIBLIOGRAFIA

BIBLIOGRAFIA

I.T. Hawryszkiewicz
Database analysis and design
Second edition
Mc Millan Publishing Company

Richard F. Walters
Database principles for personal computers
Mc Millan Publishing Company

James Martin
Organización de las bases de datos
Prentice Hall Hispanoamericana, S.A.
Mayo, 1986

Richard Fairley
Ingeniería de software
Mc Graw Hill
Junio, 1987

C.J. Date
Introducción a los sistemas de bases de datos
Adison Wesley
1986

Rick Grehan & Standord Diehl
Byte
DOS databases at work
Enero, 1992

IBM Educación
DB2 Application design workshop
Abril, 1992

Henry F. Korth, Abraham Silberschatz
Database system concepts
Mc Graw Hill
1986

Jean-Paul Trenblay, Paul G. Sorenson
An introduction to data structures with applications
Mc Graw Hill
1984

Stefano Ceri, Giuseppe Pelagatti
Distributed databases principles and systems
Mc Graw Hill
1985

Niklaus, Wirth
Algoritmos y estructuras de datos
Prentice Hall
1987