

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA



**ANÁLISIS Y DISEÑO DEL SUBSISTEMA DE INFORMACIÓN
(NIVEL A.L.R.) DE LA ADMINISTRACIÓN GENERAL DE
RECAUDACIÓN.**

T E S I S

**PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

P R E S E N T A N:

**LUIS MANUEL RAMÍREZ RAMÍREZ.
AVELINA RODRÍGUEZ SÁNCHEZ.**

DIRECTOR DE TESIS:

ING. GABRIEL CASTILLO HERNANDEZ.

FALLA DE ORIGEN

MÉXICO, D.F.

NOVIEMBRE 1995



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.

Gracias Señor . . .

Gracias Señor porque día a día me das fuerzas y alimentas mi espíritu para seguir adelante.

Gracias Señor porque sé que en todo momento cuento contigo.

Gracias Señor por haberme concedido el culminar una de mis metas en esta vida.

Gracias Señor porque hoy me das la oportunidad de hacer feliz y sentirse orgullosos a dos personas muy especiales: mis padres.

Por todo lo que me has dado.

Gracias Señor . . .

Con cariño, a la familia Ramírez Ramírez.

Así como en los momentos difíciles hemos estado unidos y salido adelante. Hoy, en este día especial, quiero que juntos disfrutemos de este momento porque han sido fundamentales para conseguir esta meta.

A mi madre:

Ejemplo de humildad, dulzura y creencia.

Mujer forjadora que a diario siembras en mí la fe y el amor.

Tú que en todo momento tienes una palabra de aliento, una caricia y una oración.

A mi padre:

Ejemplo de responsabilidad, dedicación y respeto.

Tú que en todo momento me das seguridad para salir adelante, y a diario siembras en mí el deseo de ser alguien en la vida.

Con todo mi amor a Mis Padres.

*Con afecto, a mis hermanos Catalina y Eduardo Roman,
como muestra al cariño que siempre nos hemos tenido,
por su apoyo y tiempo hacia mí.*

En agradecimiento por sus palabras de aliento y consejos que me han ayudado a salir adelante.

A mi cuñado Jesús.

A mis sobrinos: Fer y Liz.

Luis Manuel Ramírez Ramírez..

Gracias Dios por darme fuerza.

*Muy en especial agradezco a Mis Padres por haberme guiado en la vida,
positiva y sensatamente, ofreciendo sus consejos y apoyándome en cada momento.
Por alentarme para continuar y haber inculcado en mí el deseo de superación.*

*A Mis Hermanos que compartieron conmigo escuela, juegos, sueños, deseos,
preocupaciones y contribuyeron de tantas formas en cada etapa de mi vida para
superarlas exitosamente.*

*Para quienes han hecho posible mi formación profesional. Por su tenacidad,
esfuerzo, vocación y tiempo prestados en una labor tan difícil y delicada como es
la enseñanza. A todos y cada uno de mis profesores mi admiración y
agradecimiento.*

Avelina Rodríguez Sánchez.

Al Ing. Gabriel Castillo Hernández agradecemos su valiosa colaboración, apoyo, esfuerzo y dedicación. Para él nuestro mayor respeto y gratitud.

Agradecemos a la Act. Martha Miranda Correa su apoyo que sin éste no hubiera sido posible la realización de este trabajo.

*A la Facultad de Ingeniería y profesores, por habernos alojado y capacitando.
Gracias.*

INDICE

CONTENIDO	PAGINA
1. Introducción	
1.1. Perspectiva del S.I.R.	1
1.2. Teoría de Base de Datos	2
1.3. Metodología	7
2. Análisis	
2.1. Análisis de la situación actual	14
2.2. Modelo del Sistema	15
2.2.1. Modelo de Datos	15
2.2.1.1. Diagrama Entidad-Relación	15
2.2.2. Modelo de Procesos	16
2.2.2.1. Diagrama de Contexto	16
2.2.2.2. Diagrama de Flujo de Datos nivel 0	16
2.2.2.3. Diagrama de Flujo de Datos de la Generación de Información de Registro (nivel 1)	17
2.2.2.4. Diagrama de Flujo de Datos de la Generación de Información de Cobranza (nivel 1)	18
2.2.2.5. Diagrama de Flujo de Datos de la Generación de Información de Control de Obligaciones (nivel 1)	18
2.2.2.6. Diagrama de Flujo de Datos de la Generación de Información de Control de Saldos (nivel 1)	18
2.2.2.7. Diagrama de Flujo de Datos de la Recepción de Información (nivel 1)	19
2.2.2.8. Diagrama de Flujo de Datos de la Solicitud de Información (nivel 1)	19
2.2.3. Modelo Tecnológico	19

CONTENIDO	PAGINA
3. Diseño	
3.1. Estructura de Datos	38
3.2. Módulos del Sistema	39
4. Construcción	
4.1. Generación de la Base de Datos	66
4.2. Menú del Subsistema de Información	70
5. Implantación	90
6. Comentarios Finales	92
Apéndice A.	93
Bibliografía	129

1. INTRODUCCIÓN

1.1 PERSPECTIVA DEL S.I.R.

La S.H.C.P. es el organismo encargado del control y manejo de las finanzas del país, una de sus funciones más importantes es la recaudación de impuestos.

La Administración de Recaudación de la Subsecretaría de Ingresos esta comprendida por:

- Administración Local de Recaudación (A.L.R.)
- Administración Regional de Recaudación (A.R.R.)
- Administración General de Recaudación (A.G.R.)

La A.L.R. es una dependencia encargada de recabar información referente a contribuyentes, ya sean personas físicas o morales para dar seguimiento de acuerdo a su situación fiscal por medio de procedimientos que son regidos por edictos presentados en el diario oficial de la Federación.

La A.R.R. es la encargada de administrar y supervisar la información que se procesa en sus A.L.R. correspondientes.

La A.G.R. se encarga de concentrar la información de los contribuyentes a nivel nacional. Por medio de su área de normatividad evalúa los resultados, y con base a éstos se realizan toma de decisiones.

El S.I.R. (Sistema Integral de Recaudación) es un sistema que integra las áreas de la Administración de Recaudación de la Subsecretaría de Ingresos de la S.H.C.P.

Las áreas que comprende son :

- Registro Federal de Contribuyentes
- Control de Obligaciones
- Cobranza
- Contabilidad
- Control de Saldos

El área de Registro Federal de Contribuyentes da de alta a contribuyentes proporcionándoles una cédula de identificación fiscal, y el código de barras a quién lo solicite.

El área de Control de Obligaciones vigila el cumplimiento de obligaciones fiscales para que los contribuyentes presenten sus declaraciones y pagos con oportunidad, enviando los resultados de sus trámites.

El área de Cobranza controla, actualiza y tramita el cobro de los adeudos fiscales. Hace efectivo los adeudos determinados por incumplimiento de las disposiciones fiscales, así como los que emiten autoridades ajenas a esta Secretaría.

El área de Contabilidad recepciona declaraciones presentadas por los contribuyentes en las A.L.R. e Instituciones bancarias para obtener el monto de las operaciones y efectúa el cierre contable de la información recibida.

El área de Control de Saldos verifica las devoluciones y compensaciones manifestadas por los contribuyentes en sus declaraciones, dando seguimiento a los saldos a favor y vigila las devoluciones que hayan sido pagadas por la Tesorería de la Federación.

1.2 TEORÍA DE BASE DE DATOS

INTRODUCCIÓN

Antes del surgimiento de las Bases de Datos, los datos eran almacenados en un sistema de archivos (no base de datos), cada aplicación contaba con sus propios archivos, ocasionando que el acceso fuera difícil, casi siempre con considerable redundancia y diversos formatos físicos.

Los programas dependían de una organización de archivos, esto impedía solventar o mantener los costos de programación. Era difícil controlar la integridad de datos, especialmente en ambientes multi-usuario, en los cuales existía escasa programación para manejar procesos concurrentes.

Los mecanismos de seguridad eran difíciles de implementar, ya que se requería uno para cada aplicación.

Era difícil manipular relaciones entre registros, ya que las diferentes organizaciones de archivos, lenguajes de programación y aplicaciones se utilizaban para diferentes áreas de aplicación.

Un archivo de datos puede tener más de un uso pero únicamente una vista de los datos almacenados.

Una base de datos puede tener más de un uso y varias vistas de los datos almacenados.

En los sistemas de bases de datos, la redundancia es reducida considerablemente al integrar archivos separados.

La capacidad de compartir es uno de los beneficios más importantes de los sistemas de bases de datos. Las aplicaciones existentes pueden hacer referencia a los mismos datos y las nuevas aplicaciones pueden hacer alusión a los datos existentes.

Existen dos mecanismos de seguridad: el control de privilegios (permisos) y el uso de vistas.

La integridad de los datos en una base de datos puede satisfacer tipos de restricciones de consistencia.

Ventajas de los sistemas de bases de datos :

- La redundancia es reducida.
- La inconsistencia es evitada.
- Los datos pueden ser compartidos.

- Las restricciones de seguridad pueden ser aplicadas.

- Se mantiene la integridad.

MODELOS DE BASES DE DATOS

Una base de datos es una colección de datos relacionados acerca de una empresa con múltiples usos. Puede tener más de un sólo uso y los múltiples usuarios pueden tener una variedad de datos almacenados.

MODELO DE RED.

En el modelo relacional, descrito posteriormente, los datos y las relaciones entre ellos se representan mediante una colección de tablas. El modelo de red difiere del modelo relacional en que los datos se representan por medio de colecciones de registros y las relaciones entre los datos mediante enlaces.

Una base de datos de red consta de una colección de registros, los cuales están conectados entre sí por medio de enlaces. Un registro en muchos aspectos es similar a una entidad en el modelo entidad-relación. Cada registro es una colección de campos (atributos), cada uno de los cuales contiene solamente el valor de un dato. Un enlace es una asociación entre dos registros exclusivamente. Así pues, un enlace puede verse como una forma restringida (binaria) de relación en el sentido del modelo E-R.

MODELO JERÁRQUICO

En el modelo de red, los datos se representan mediante colecciones de registros y las relaciones entre los datos por medio de enlaces. Esta representación también es válida para el modelo jerárquico. La única diferencia es que en ésta los registros se organizan para formar colecciones de árboles en vez de grafos arbitrarios.

Una base de datos jerárquica consiste en una colección de registros que se conectan entre sí por medio de enlaces. Los registros y el enlace son similares a los del modelo de red.

MODELO RELACIONAL

Una base de datos relacional consiste en una colección de tablas, a cada una de las cuales se asigna un nombre único. Una tabla es una colección de datos organizada en renglones y columnas. Cada tabla en una base de datos contiene diferentes tipos de datos.

Un renglón o tupla contiene todos los datos acerca de cada uno de los objetos que describe la tabla.

Una columna contiene un tipo particular de dato.

Una de las ventajas que ofrece el modelo relacional sobre los modelos jerárquico y de red, es una estructura lógica en vez de física.

NORMALIZACIÓN

La normalización consiste en separar tablas para eliminar anomalías y redundancia. Este proceso es llamado descomposición, debe ser realizado con cuidado cuando las tablas se están dividiendo para que los datos no se pierdan. Consiste de las siguientes formas normales:

Primera forma normal.

Dada una relación R, la cual contiene una llave primaria K y un atributo A, se dice que R esta en primera forma normal si y sólo si el valor de A para cualquier renglón depende del valor de K para ese renglón.

Segunda forma normal.

Dada una relación R, en primera forma normal, que contiene una llave primaria K de más de una columna con valores K1 y K2 y un atributo A se dice que R esta en la segunda forma normal si y sólo si el valor de A para cualquier renglón depende de los valores de K1 y K2. Las relaciones que tienen llaves primarias de una sola columna se considera que están siempre en segunda forma normal.

Tercera forma normal.

Dada una relación R, esta en tercera forma normal si esta en segunda forma normal y las columnas no llave no dependen entre sí.

Cuarta forma normal.

En esta forma se prohíbe la independencia de relaciones uno a muchos entre columnas llave primaria y columnas no llave.

SISTEMA MANEJADOR DE BASE DE DATOS (DBMS).

Un DBMS consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un DBMS es proporcionar un entorno que sea a la vez conveniente y eficiente para extraer y almacenar información de la base de datos.

Los sistemas de base de datos están diseñados para manejar grandes bloques de información, la organización de datos implica tanto la definición de estructuras para el almacenamiento de datos como la provisión de mecanismos para la representación de la información. Además, los sistemas de bases de datos deben mantener la seguridad de la información almacenada, pese a caídas del sistema o intentos de accesos no autorizados. Si los datos son compartidos por varios usuarios, el sistema debe evitar posibles resultados anómalos.

La seguridad es un aspecto importante en el manejo de base de datos, los datos almacenados necesitan ser protegidos de usuarios no autorizados para ser vistos, cambiados o removidos.

Los DBMS pueden correr en ambiente compartido. Los equipos de cómputo proveen seguridad a nivel de sistema operativo, típicamente se requiere una palabra secreta (password) para entrar (login). De acuerdo a los permisos de los archivos, el usuario puede tener permisos de escritura, lectura y ejecución.

Informix maneja dos herramientas de mecanismos de base de datos para el desarrollo de aplicaciones: Informix-SE (Standard Engine) e Informix-OnLine.

Informix-SE esta basado en C-ISAM, una librería de llamadas en lenguaje C que trabaja con Unix para crear y manipular archivos de base de datos.

Informix-SE trabaja automática y transparentemente, y no requiere ninguna instrucción especial en los programas. Este es un mecanismo ideal para desarrollar pequeños o medianos tamaños de aplicaciones que no requieren una máxima ejecución o un rango extensivo de integridad de datos.

Informix-OnLine es un mecanismo de proceso-transacción de una base de datos que maneja operaciones de entrada-salida directamente por lo que su ejecución es máxima. Este es diseñado para manejar los requerimientos con alta ejecución e integridad de grandes aplicaciones. Muchas aplicaciones pueden correr en uno u otro mecanismo porque pocas diferencias afectan la programación. Sin embargo, provee características que incrementan la ejecución, tipo de datos extendidos, y mejorar los aspectos administrativos del manejador de base de datos. Sus características son:

- Memoria compartida optimizada para grandes ejecuciones.
- Alto aprovechamiento y recuperación automática.
- Tipo de dato caracter de longitud variable (VARCHAR).
- Binary Large Objects (BLOBs) son colección de bytes de longitud arbitraria, como: documentos, sonido e imágenes digitalizadas.
- Búsquedas distribuidas capaces de cruzar múltiples bases de datos.

Informix-OnLine permite acceder a múltiples bases de datos en la misma computadora. Se pueden escribir aplicaciones que extraigan datos de múltiples tablas, aún si las tablas residen en diferentes bases de datos. Se puede usar los datos combinados para propósitos generales de desplegado o como entrada a un reporte.

Los sistemas basados en SQL (Structured Query Language) proveen dos mecanismos de seguridad:

- Control de privilegios (permisos), se especifica los usuarios que están permitidos para ejecutar comandos en la base de datos.
- El uso de vistas limitan el acceso a datos, permitiendo alterar la base de datos a usuarios autorizados.

Ambos mecanismos asumen que el sistema de base de datos tiene forma de saber quién lo esta accedando, para reconocer estos usuarios y verificar sus permisos.

La integridad es la consistencia, el cuidado o corrección de los datos en una base de datos. Un tipo de inconsistencia puede ser una falla del sistema, un problema del equipo o un error lógico en un programa de aplicación. Los sistemas manejadores de base de datos que protegen los datos de este tipo de inconsistencia cuentan con comandos SQL que permiten completar o cancelar una transacción. A éste proceso se le conoce como manejo de transacciones.

Existen dos tipos de Integridad:

- Integridad entidad. La llave primaria requiere que únicamente identifique cada renglón en una tabla.
- Integridad referencial. Es la consistencia entre datos que son repetidos en más de una tabla para mantener la relación entre llaves primarias y llaves foráneas. Una llave foránea es una columna que hace referencia a una llave primaria en otra tabla.

La integridad debe ser almacenada en el diccionario de datos no en programas de aplicación.

DICCIONARIO DE DATOS.

En un ambiente de base de datos, uno de los objetivos principales es el compartir datos entre múltiples usuarios, proveer datos correctos a éstos, evitar la redundancia y tener un control de uso de los datos. Para cumplir estos objetivos es necesario contar con un mecanismo de control centralizado.

El diccionario de datos es un archivo organizado de todos los elementos de datos que pertenecen al sistema, con definiciones precisas y rigurosas que permiten que el usuario y el analista del sistema tengan una misma comprensión. Con este repositorio central de información se pueden conocer las relaciones entre entidades, los elementos dato de éstas, su origen, uso y significado.

Un diccionario de datos puede ser también usado en un medio no base de datos. En un medio base de datos, la información almacenada es acerca de los datos almacenados en una base de datos, y en un medio no base de datos es acerca de los datos almacenados en archivo de datos.

Los beneficios de usar un diccionario de datos son los siguientes:

- Comunicación entre usuarios
- Controla los elementos dato de una manera simple, esto es, introduce nuevos elementos en el sistema, o cambia descripciones de elementos.
- Elimina la redundancia e inconsistencia.

Las partes básicas de un diccionario de datos son:

- Elemento dato : Un atributo describiendo una entidad. Cada elemento dato tiene un nombre único.
- Elemento Dato Grupo : Un agrupamiento de elementos dato relacionados lógicamente.

- Elemento Dato Derivado : Un elemento dato que es originado como resultado de una fórmula o un cálculo.
- Los nombres de las relaciones.
- Los nombres de las vistas definidas en las bases de datos y la definición de esas vistas.
- Las restricciones de integridad de cada relación.

I.3 METODOLOGÍA

El desarrollo de Sistemas debe ser tratado bajo una disciplina ingenieril, con el fin de desarrollar e implementar sistemas realmente eficientes y acordes con los objetivos primordiales de la organización.

La Ingeniería del 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.

Surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de elementos clave: métodos y herramientas, que facilitan al encargado controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

Sus métodos indican "cómo" construir técnicamente el software, abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requisitos del sistema y del software, diseño de estructuras de datos, codificación, prueba y mantenimiento.

Las herramientas de la ingeniería del software proveen un soporte para los métodos existentes.

La metodología a emplear es la Metodología para la Integración de Sistemas (MIS) cuyo fin es proporcionar un conjunto integrado de procedimientos, estándares y prácticas que garanticen la realización exitosa de los proyectos de sistemas a desarrollar.

Fases de la Metodología:

- Análisis
- Diseño
- Construcción
- Implantación

- Análisis.

En esta fase se identifican los requerimientos y limitaciones del sistema. Se especifica el problema por resolver y las posibles soluciones. En un concepto general significa, descomponer un problema en partes, resolver lo complejo de lo simple teniendo siempre como objetivo llegar a conocer los principios o elementos del problema.

- Diseño

Es la fase del desarrollo de sistemas que se ocupa de elaborar la arquitectura, organización y estructura de un programa, y el sistema del cual éste forma parte.

Define la organización, estructura y métodos de acceso óptimos a los datos que implique el manejo de dichos sistemas.

- Construcción

En esta etapa se desarrolla la codificación de los programas que previamente se identificaron y especificaron. Estos se elaboran con el uso de algún lenguaje de programación previamente definido; se efectúan tanto las pruebas individuales por programa como las pruebas de integración de módulos que forman el nuevo sistema.

- Implantación

Se pone en operación el nuevo sistema, realizando las pruebas de aceptación, para su instalación.

Ingeniería de Información

Es un conjunto de técnicas formales con las cuales son construídos modelos de datos y modelos de procesos en una base comprensiva de conocimientos y son usados para crear y mantener sistemas de procesamiento de datos.

Para desarrollar un Sistema de Información existen tres fases principales:

- Análisis
- Diseño
- Construcción

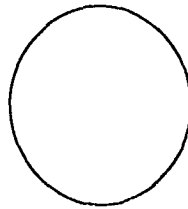
donde se desarrollan tres modelos:

- Modelo de Datos
- Modelo de Procesos
- Modelo Tecnológico

Modelo de Datos

En este modelo se pretende obtener una visión coherente de la información que mantiene una organización de manera independiente a las transformaciones que sufra por su empleo en las diversas actividades de la organización. Se utiliza el diagrama de Entidad-Relación, cuyo principal objetivo es resaltar los tipos de relaciones que se tienen entre las diferentes entidades del sistema. Utilizando la notación de Peter Chen, sus componentes son:

Atributos. Son propiedades de una entidad, cada atributo tiene asociado un dominio, éste es el conjunto de todos los posibles valores que puede tomar. Es frecuentemente llamado un elemento dato, un campo dato, un campo o una identidad dato. Su representación gráfica es la siguiente:

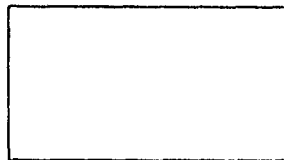


Entidad. Una entidad es un objeto que existe y es distinguible de otros objetos, puede ser concreta como una persona, lugar, cosa o abstracta como un día festivo, o un evento acerca de la información que es registrada.

Una entidad está representada por un conjunto de atributos. Cada entidad se describe por medio de un conjunto de pares (atributo, valor del dato), un par para cada atributo del conjunto de entidades.

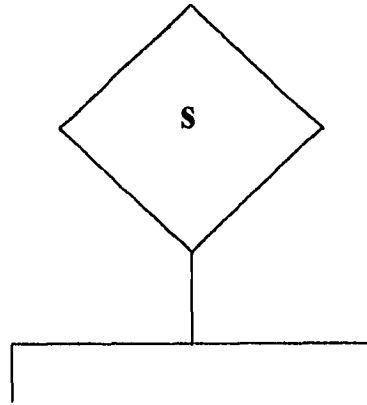
El valor del dato es la información actual contenida en cada elemento de dato. Los valores tomados por los elementos del dato pueden ser cuantitativos, cualitativos o descriptivos, dependiendo de como los elementos del dato describen una entidad.

Su representación es:



FALTA PAGINA

No.....a la 10



- Subtipo inclusivo

Una entidad padre puede ser todos o ninguno de las entidades hijos.

Relaciones Independientes. La existencia de una no depende de otra. Estas pueden clasificarse en:

- Asociación

Es una relación entre dos entidades independientes

- Generalización

Es similar al subtipo exclusivo excepto en que la entidad hijo tiene diferentes llaves primarias. La llave de la entidad padre es migrada como llave foránea a la entidad hijo.

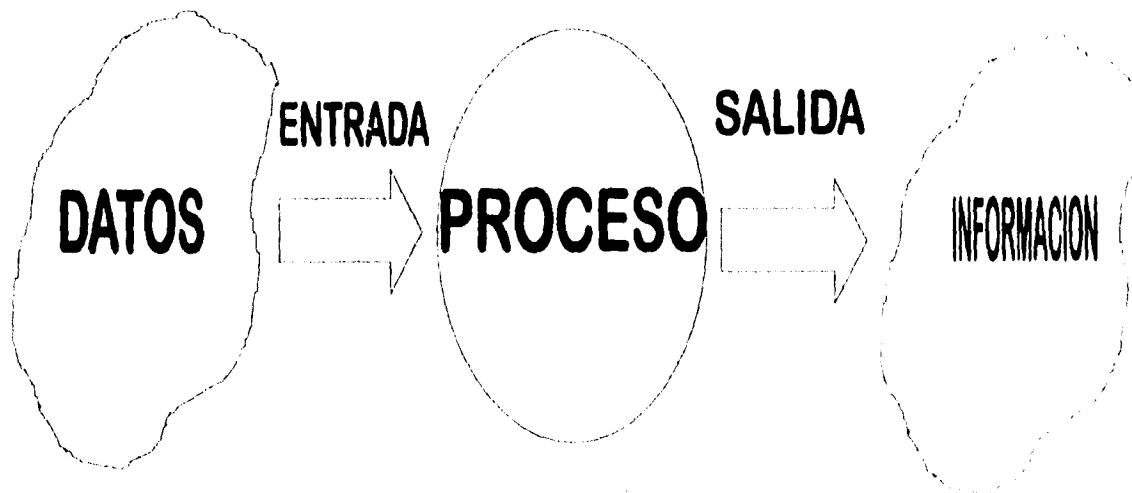
- Agregación

Representan una colección de partes de un objeto.

Modelo de Procesos

Un proceso transforma los datos de entrada en datos de salida de acuerdo a las políticas de la empresa.

El objetivo del modelo de procesos es describir la estructura o composición de los procesos, de una manera jerárquica y de esta manera comprender su funcionamiento y relación con su medio ambiente.



SON TRANSFORMACIONES DE LOS DATOS

Diagrama de Flujo de Datos (DFD).

Es una representación de un sistema, muestra procesos y el flujo de datos entre ellos. A un nivel alto se usa para mostrar los eventos de la organización y las transacciones resultantes de estos eventos. A un nivel mas bajo representan programas o módulos de programas y el flujo de datos entre ellos.

Un DFD se construye con cuatro elementos principales:

- Flujo de datos. Es el camino que sigue los datos a través de los procesos.
- Procesos. Utilizan o transforman los datos para obtener algún resultado.
- Almacenamiento de datos. Representa un archivo, se relaciona con los procesos por el flujo de datos cuya dirección indica si se consultan o se actualizan datos.
- Agentes externos. Son personas u organizaciones fuera del contexto del sistema que muestran el origen o el destino de los datos utilizados o producidos por el sistema.

Al desarrollar los DFD en un sistema muy grande, es conveniente descomponerlo en subsistemas o niveles inferiores hasta que los procesos puedan documentarse en una mini-especificación. Al nivel superior se le llama "Diagrama de Contexto" y a los niveles inferiores "funciones primitivas".

El diagrama de contexto muestra todas las entradas y salidas del sistema.

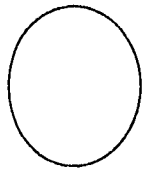
La mini-especificación, la cual describe las reglas que gobiernan la transformación de flujo de datos sin redundancia, debe ser clara, precisa y completa.

PROCESO

*FLUJO DE
DATOS*

*ALMACENAMIENTO
DE DATOS*

*ENTIDAD
EXTERNA*



Modelo Tecnológico

El modelo tecnológico crea el ambiente físico, incluyendo construcción del centro de cómputo e instalación de hardware y software.

Todo lo relacionado con la tecnología que va a soportar el sistema de información se desarrolla dentro de este modelo, normalmente se inicia en paralelo durante la realización de los Modelos de Datos y Procesos.

2. ANÁLISIS

2.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

La Recaudación es una parte importante en la Administración General de Recaudación (A.G.R.), por lo que es necesario contar con un Subsistema automatizado en donde se obtengan estadísticas confiables y oportunas para cada una de las áreas que la comprende.

Cada área cuenta con su propio Subsistema del cual se obtiene la información para realizar estadísticas manualmente, esto trae como consecuencia posibles errores de cálculo. Además, dichas estadísticas deben ser capturadas, lo que aumenta un factor más para la generación de errores, ocasionando poca confiabilidad respecto a su veracidad. El envío de esta información a las áreas centrales se realiza con retraso, siendo algunas veces ilegible (cuando es transmitida por fax) y/o maquillada para presentar resultados favorables. Todo esto ocasiona información alterada y por consiguiente toma de decisiones incorrectas.

En la actualidad la A.G.R. requiere de información que permita tomar decisiones en los niveles directivos, para aplicar las medidas preventivas y/o correctivas en la operación del S.I.R.

SOLUCIÓN PROPUESTA

Debido a la problemática existente es necesario desarrollar un sistema automatizado que genere información estadística de una manera adecuada y eficiente, que cubra todas las áreas involucradas del S.I.R. explotando su base de datos, con el fin de simplificar las tareas administrativas y evitar inconsistencia en los resultados.

Este sistema contará con las siguientes características:

- La información se generará semanalmente para cada área (excepto Contabilidad), sin riesgo de no procesarse algún período.
- Se realizarán reportes y consultas de información para cada área que comprende el Subsistema de Información.
- Contará con tablas propias en donde se almacenarán los datos generados.
- Se podrá obtener información cada vez que el área de Recaudación lo requiera.
- Utilizará los avances tecnológicos y administrativos para obtener estadísticas exactas, oportunas y un flujo de información veraz.
- Permitirá conocer comportamientos presentes y pasados, además de estimar los comportamientos futuros para una efectiva recaudación.
- El envío de información a las áreas centrales se realizará en forma electrónica y/o dispositivo de una manera eficiente.
- La depuración de la base de datos se realizará cada dos años, por lo que se podrá consultar información durante este período.
- La información de Contabilidad que se genera en las A.R.R. podrá ser cargada en base de datos.

2.2. MODELO DEL SISTEMA

2.2.1. MODELO DE DATOS

DIAGRAMAS ENTIDAD - RELACIÓN

Los diagramas E-R de las áreas que integra el S.I.R. se presentan al final de éste capítulo.

En la figura 2.1.1 se muestra el diagrama E-R del Subsistema de Cobranza :

La entidad creditos tiene una relación de uno a muchos con las entidades: tabla_01, tabla_02,..., tabla_61 y una relación de uno a uno con la entidad documentos.

En la figura 2.1.2 se muestra el diagrama E-R del Subsistema de Registro :

La entidad contrib tiene una relación de uno a muchos con las entidades bajas, tramites, inhabi y oblig, y una relación de uno a uno con las entidades consecuci y liqui.

En la figura 2.1.3 se muestra el diagrama E-R del Subsistema de Control de Obligaciones :

La entidad emision tiene una relación de uno a muchos con las entidades seguimiento, importes y multas.

En la figura 2.1.4 se muestra el diagrama E-R del Subsistema de Control de Saldos :

La entidad dev_comp tiene una relación de muchos a uno con la entidad catalogo.

A continuación se presentan los diagramas E-R del Subsistema de Información.

En la figura 2.1.5.a se muestran los diagramas E-R del Subsistema de Información Control de Saldos y Registro :

La entidad inf_informa tiene una relación de uno a muchos con las entidades inf_saldo, inf_operas, inf_estado e inf_obliga.

En la figura 2.1.5.b se muestran los diagramas E-R del Subsistema de Información Contabilidad las cuales no tienen ninguna relación.

En la figura 2.1.5.c se muestran los diagramas E-R del Subsistema de Información de Control de Obligaciones y Cobranza :

La entidad inf_vigilancia tiene una relación de uno a muchos con la entidad inf_impreso.

Las entidades inf_seguimiento e inf_cobra no tienen relación con otra tabla.

2.2.2 MODELO DE PROCESOS

Diagrama de Contexto

Los DFD de las áreas que integra el S.I.R. se presentan al final de éste capítulo.

En la figura 2.2.1 se muestra el diagrama de contexto en donde los agentes externos que interactúan con el Subsistema de Información son los Subsistemas de Registro, Cobranza, Control de Obligaciones, Control de Saldos, Usuario del Subsistema, Coordinador del Subsistema y la información proveniente de la A.R.R.

Para el Subsistema de Registro se cuenta con las siguientes tablas: baja, tramite, oblig, contrib, liqui, consec e inhabi.

Para el Subsistema de Cobranza se cuenta con las tablas: tabla_01, tabla_02, ..., tabla_61, documento y crédito.

Para el Subsistema de Control de Obligaciones se cuenta con las tablas: multa, importe, emisión y seguimiento.

Para el Subsistema de Control de Saldos se cuenta con las tablas: dev_comp y catalogo.

Para todos los Subsistemas se leen las tablas mencionadas para realizar la generación de información que requiere el Subsistema de Información.

El agente externo Usuario del Subsistema realiza una solicitud de información y el Subsistema de Información retorna el resultado de la solicitud. Asimismo el Coordinador del Subsistema realiza una solicitud de envío o depuración de información y el Subsistema retorna el resultado de solicitud de envío o depuración de información.

El Subsistema de Información recibe y carga el archivo ASCII del agente externo Información Proveniente de la A.R.R.

Diagrama de Flujo de Datos Nivel 0

En la figura 2.2.2.a se muestran los procesos de generación de información de cada uno de los Subsistemas de Información y el proceso de Recepción de Información del Subsistema de Contabilidad.

Para el proceso de generación de información del Subsistema de Cobranza se leen las tablas mencionadas anteriormente almacenando el resultado en inf_cobra.

Para el proceso de generación de información del Subsistema de Control de Saldos se leen las tablas mencionadas anteriormente almacenando el resultado en inf_saldo.

Para el proceso de generación de información del Subsistema de Registro se leen las tablas mencionadas anteriormente almacenando el resultado en inf_estado, inf_obliga e inf_operacion.

Para el proceso de recepción de información del Subsistema de Contabilidad se lee el archivo ASCII que proviene de la A.R.R. almacenando el resultado en inf_banco, inf_formulario, inf_ingreso e inf_obligacion.

En la figura 2.2.2.b se muestran los procesos Solicitud de Información y Solicitud de envío de información.

La solicitud de información puede ejecutar los procesos: Solicitar reportes o Solicitar consulta, de acuerdo a éstos se activan los procesos Explotar información de Cobranza, Registro, Contabilidad, Control de Obligaciones o Control de Saldos, obteniendo como salida el resultado de la solicitud en impresión o pantalla.

La solicitud de envío de información activa el proceso Generar información del periodo cuya salida es un archivo(s) ASCII, el cual es leído por el proceso Validar generación de archivo, que a su vez activa el proceso Transmitir archivo el cual lee el (los) archivo(s) del Subsistema que se haya(n) elegido (Registro, Cobranza, Control de Obligaciones, Control de Saldos o todos), realizando el envío por transmisión electrónica o por dispositivo (disco o cartucho).

En la figura 2.2.2.c se presenta el proceso Solicitud de la depuración de información, que activa el proceso Generar archivo ASCII del periodo requerido, el cual crea un archivo. Este es leído por los procesos: Copiar archivo a disco, obteniendo el resultado de la solicitud de la depuración de información y Validar generación de archivo ASCII el cual activa los procesos de Borrar información de Cobranza, Registro, Contabilidad, Control de Obligaciones y Control de Saldos, los cuales eliminan datos de las tablas correspondientes.

Diagrama de Flujo de Datos de la Generación de Información de Registro (Nivel 1)

En la figura 2.2.3 se muestra la generación de información de Registro, realizando los siguientes procesos :

El proceso Obtener movimientos operados lee la tabla tramite dejando el resultado en inf_operacion.

El proceso Obtener estructura de sucesión lee la tabla consecucion dejando el resultado en inf_estado.

El proceso Calcular total de liquidaciones lee las tablas liqui y oblig dejando el resultado en inf_estado.

El proceso Calcular total de bajas, inactivos y no localizados lee las tablas oblig y baja dejando el resultado en inf_estado.

El proceso Calcular total de suspensión lee las tablas baja e inhabi dejando el resultado en inf_estado.

El proceso Obtener contribuyentes activos lee las tablas contrib, oblig, baja e inhabi dejando el resultado en inf_estado.

Diagrama de Flujo de Datos de la Generación de Información de Cobranza (Nivel 1)

En la figura 2.2.4 se muestra la generación de información de Cobranza ejecutando los siguientes procesos:

Los procesos Obtener créditos no localizados y dados de baja y Generar situación de créditos leen la tabla credito dejando la salida en inf_cobra.

El proceso Obtener situación y movimientos de tramites lee la tabla documento dejando la salida en inf_cobra.

El proceso Generar movimientos operados lee las tablas tabla_01, tabla_02, ..., tabla_61 dejando la salida en inf_cobra.

Los procesos Obtener créditos por autoridad y Generar créditos dados de baja leen la tabla inf_cobra y la actualizan.

Diagrama de Flujo de Datos de la Generación de Información de Control de Obligaciones (Nivel 1)

En la figura 2.2.5 se muestran los procesos de la generación de información de Control de Obligaciones:

El proceso Seleccionar emisión lee la tabla emisión, la salida es leída por el proceso Obtener periodo de emisión, cuya salida entra en el proceso Asignar valores, el cual lee la tabla seguimiento dejando la salida en inf_vigilancia e inf_impreso.

El proceso Calcular total de multas lee la tabla multa dejando la salida en inf_seguimiento.

El proceso Calcular total de importes lee la tabla importe dejando el resultado en inf_seguimiento.

Diagrama de Flujo de Datos de la Generación de Información de Control de Saldos (Nivel 1)

En la figura 2.2.6 se muestra la generación de información de Control de Saldos realizando el proceso siguiente:

El proceso Leer claves lee las tablas catalogo y dev_comp, la salida es leída por los procesos Obtener inventario inicial y trámites recibidos, Obtener total de trámites solventados no autorizados y autorizados y Obtener total de trámites requeridos y pendientes, los cuales almacenan el resultado en inf_saldo.

Diagrama de Flujo de Datos de la Recepción de Información (Nivel 1)

En la figura 2.2.7 se presentan los procesos de la recepción de información:

La solicitud de recepción de archivo ASCII activa los procesos: Recepcionar información de disco, cartucho o transmisión electrónica, los cuales copian el archivo ASCII a disco duro. La salida de cualquiera de estos activa el proceso Validar recepción de información cuya salida activa el proceso Cargar información a base de datos, éste lee el archivo ASCII e inserta en las tablas inf_formulario, inf_obligacion, inf_banco e inf_ingreso.

Diagrama de Flujo de Datos de la Solicitud de Información (Nivel 1)

En las figuras 2.2.8.a. y 2.2.8.b. se muestran los procesos de la solicitud de información:

La solicitud de información activa los procesos Elegir información de Cobranza, Registro, Control de Obligaciones, Control de Saldos o Contabilidad.

El proceso elegir información de Cobranza activa los procesos Desplegar información de los movimientos de la cartera, por autoridades responsables, bajas de créditos, créditos estáticos, o créditos no localizados, dependiendo de la elección de alguno de estos procesos se obtiene el resultado de solicitud.

El proceso Elegir información de Registro activa los procesos Desplegar información del padrón de contribuyentes, universo de contribuyentes, resultados de la operación, solicitudes y avisos, u obligación mas representativa, dependiendo de la elección de alguno de éstos procesos se obtiene el resultado de solicitud.

El proceso Elegir información de Control de Obligaciones activa los procesos Desplegar información de vigilancia de obligaciones fiscales, requerimientos de obligaciones, o Procedencia de Requerimientos, dependiendo de la elección de alguno de estos procesos se obtiene el resultado de solicitud.

El proceso Elegir información de Control de Saldos activa el proceso Desplegar información de control de saldos cuya salida es el resultado de la solicitud.

El proceso Elegir información de Contabilidad activa los procesos Desplegar información de declaración por tipo de obligación, por tipo de formulario, por tipo de ingreso o por institución bancaria, dependiendo de la elección de alguno de estos procesos se obtiene el resultado de solicitud.

2.2.3. MODELO TECNOLÓGICO

El Subsistema de Información correrá en equipo U6000 con procesador Intel 486 de Unisys bajo sistema operativo Unix (SVR4).

Se desarrollará en lenguaje de programación Informix-4gl Versión 4.1 con interfaces en lenguaje "C" y programación en shell.

DIAGRAMA ENTIDAD-RELACION
DEL SUBSISTEMA DE COBRANZA

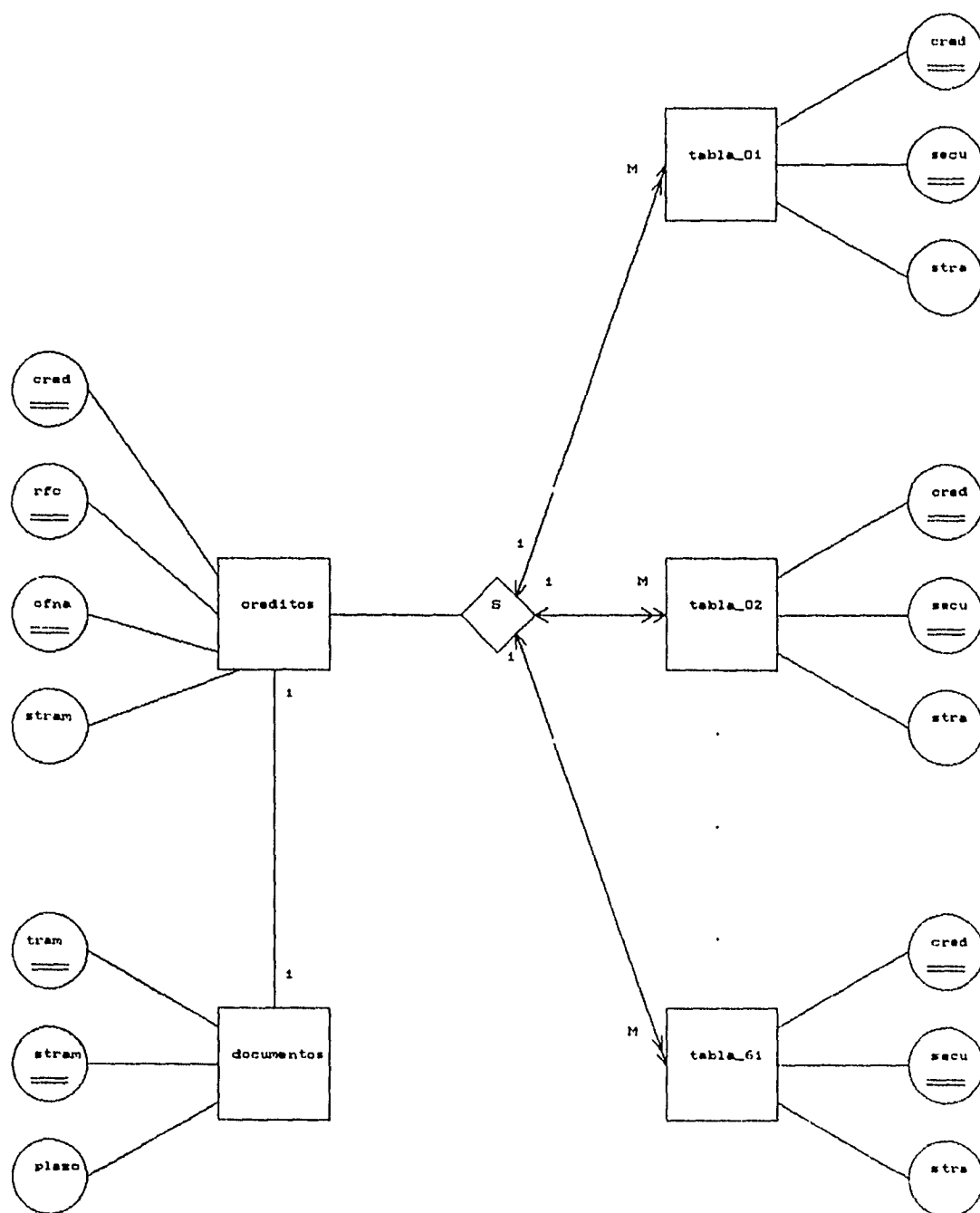


Fig. 2.1.1

DIAGRAMA ENTIDAD-RELACION
DEL SUBSISTEMA DE REGISTRO

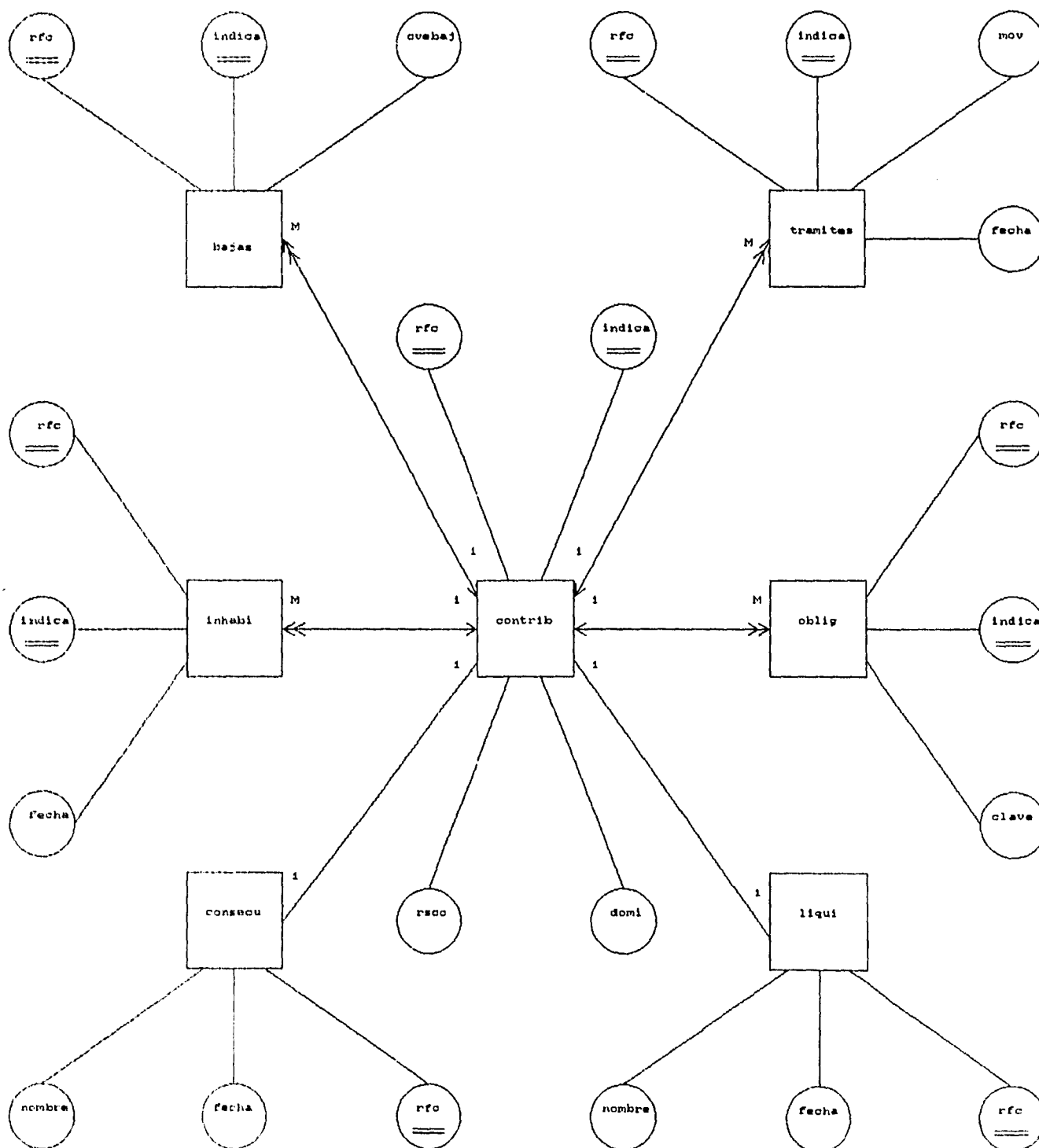


Fig. 2.1.2

DIAGRAMA ENTIDAD-RELACION DEL
SUBSISTEMA DE CONTROL DE
OBLIGACIONES

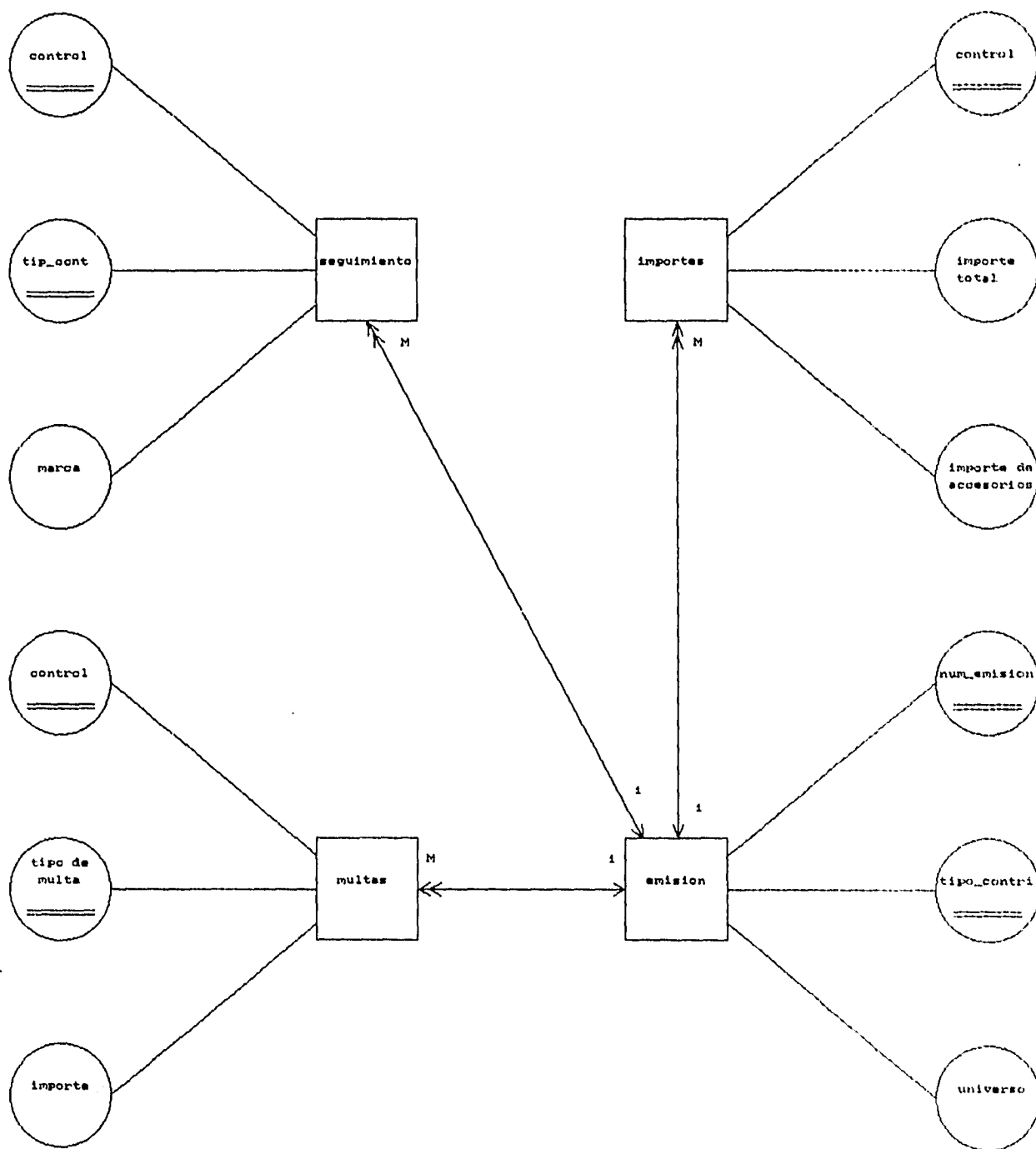


Fig. 2.1.3

DIAGRAMA ENTIDAD-RELACION DEL
SUBSISTEMA DE CONTROL DE SALDOS

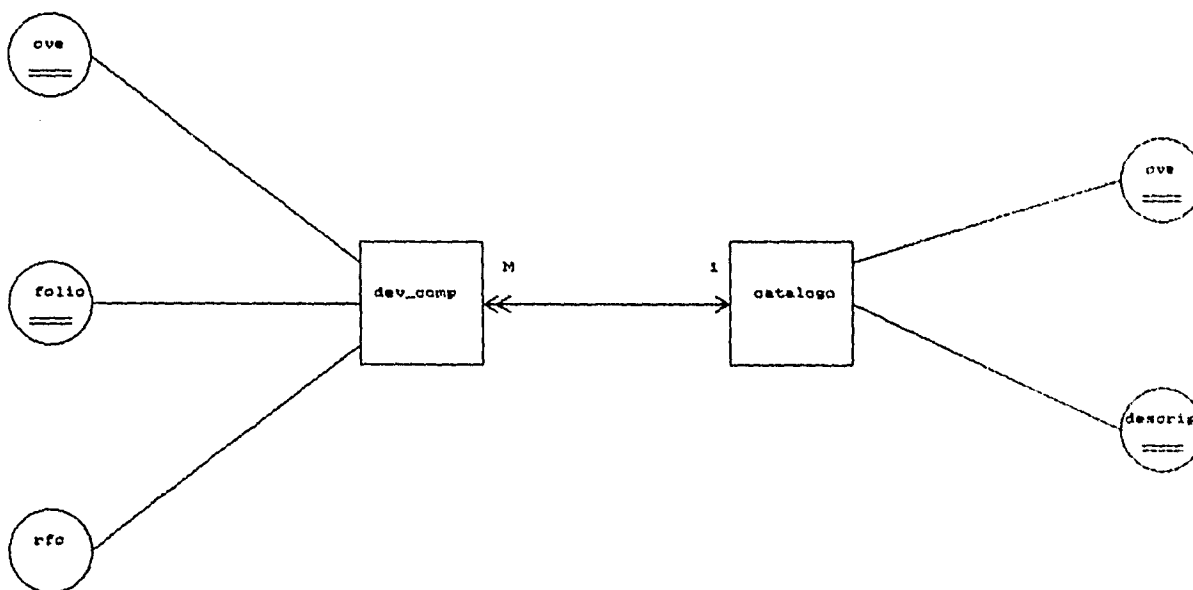


Fig. 2.1.4

DIAGRAMA ENTIDAD-RELACION DEL
SUBSISTEMA DE INFORMACION
DE CONTROL DE SALDOS Y REGISTRO

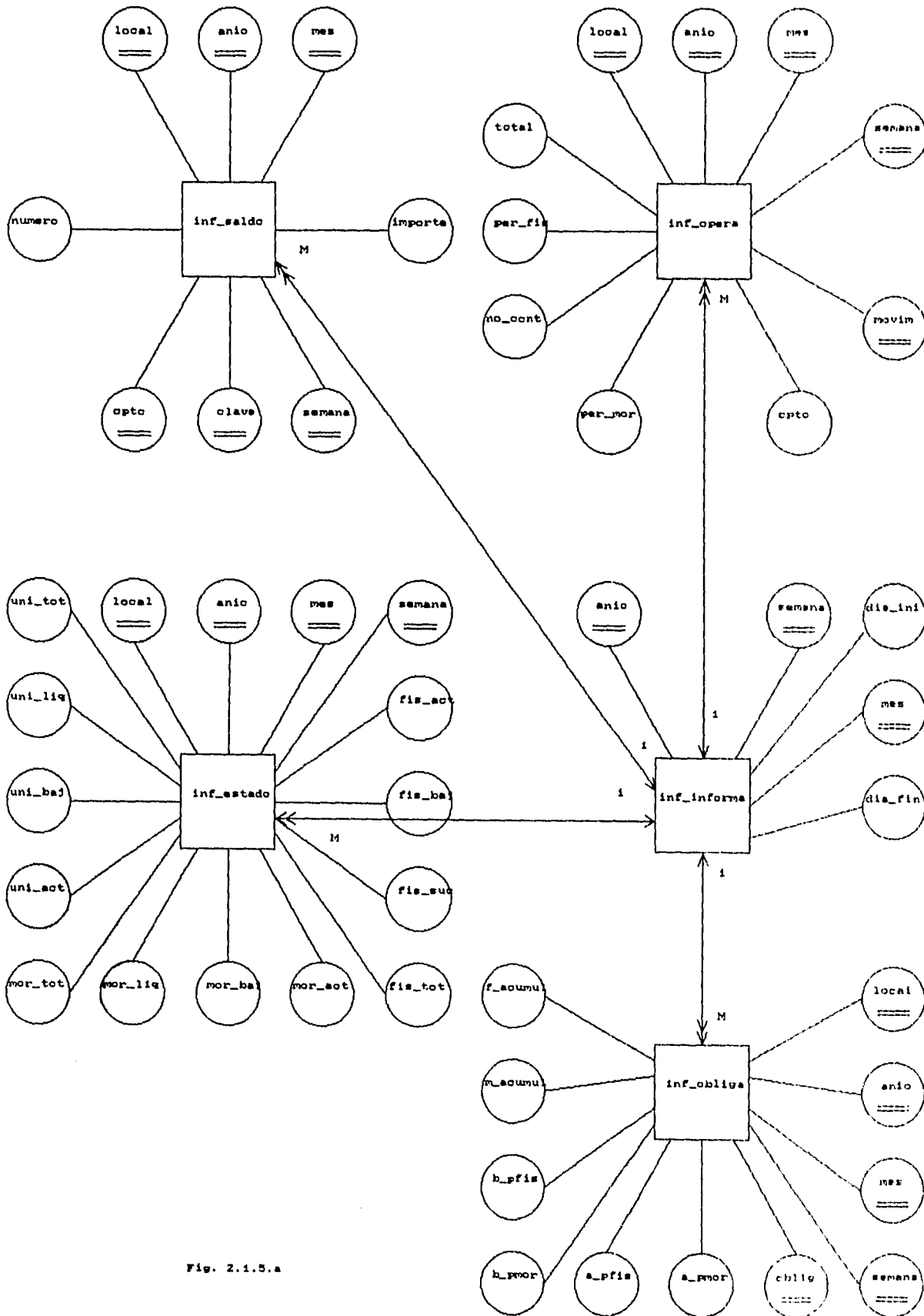


Fig. 2.1.5.a

DIAGRAMA ENTIDAD-RELACION DEL
SUBSISTEMA DE INFORMACION
DE CONTABILIDAD

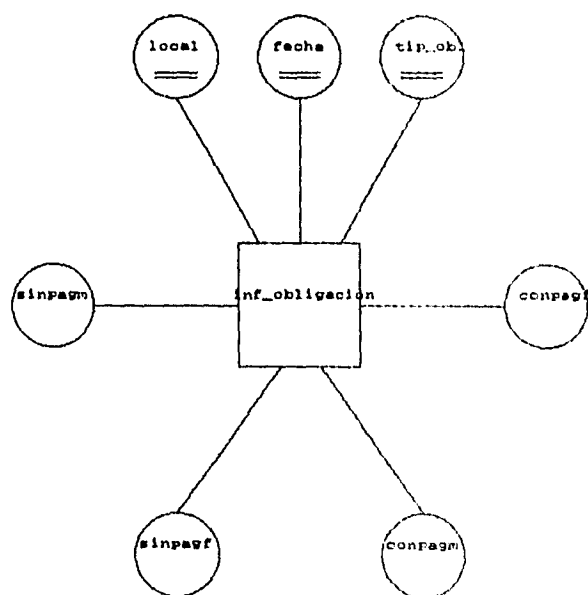
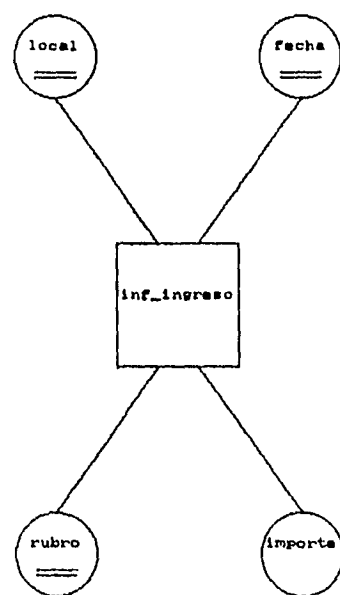
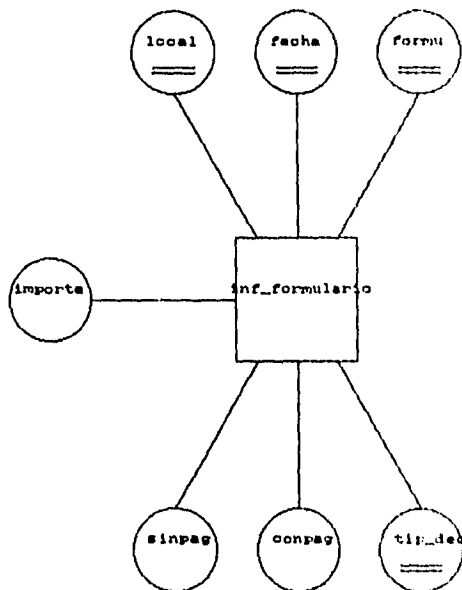
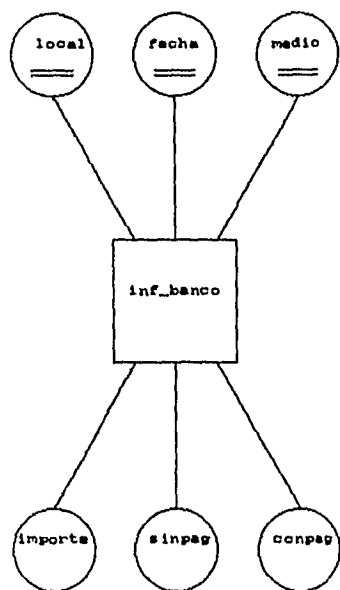


Fig. 2.1.5.b

DIAGRAMA ENTIDAD-RELACION DEL
SUBSISTEMA DE INFORMACION
DE CTRL.DE OBLIG. Y COBRANZA

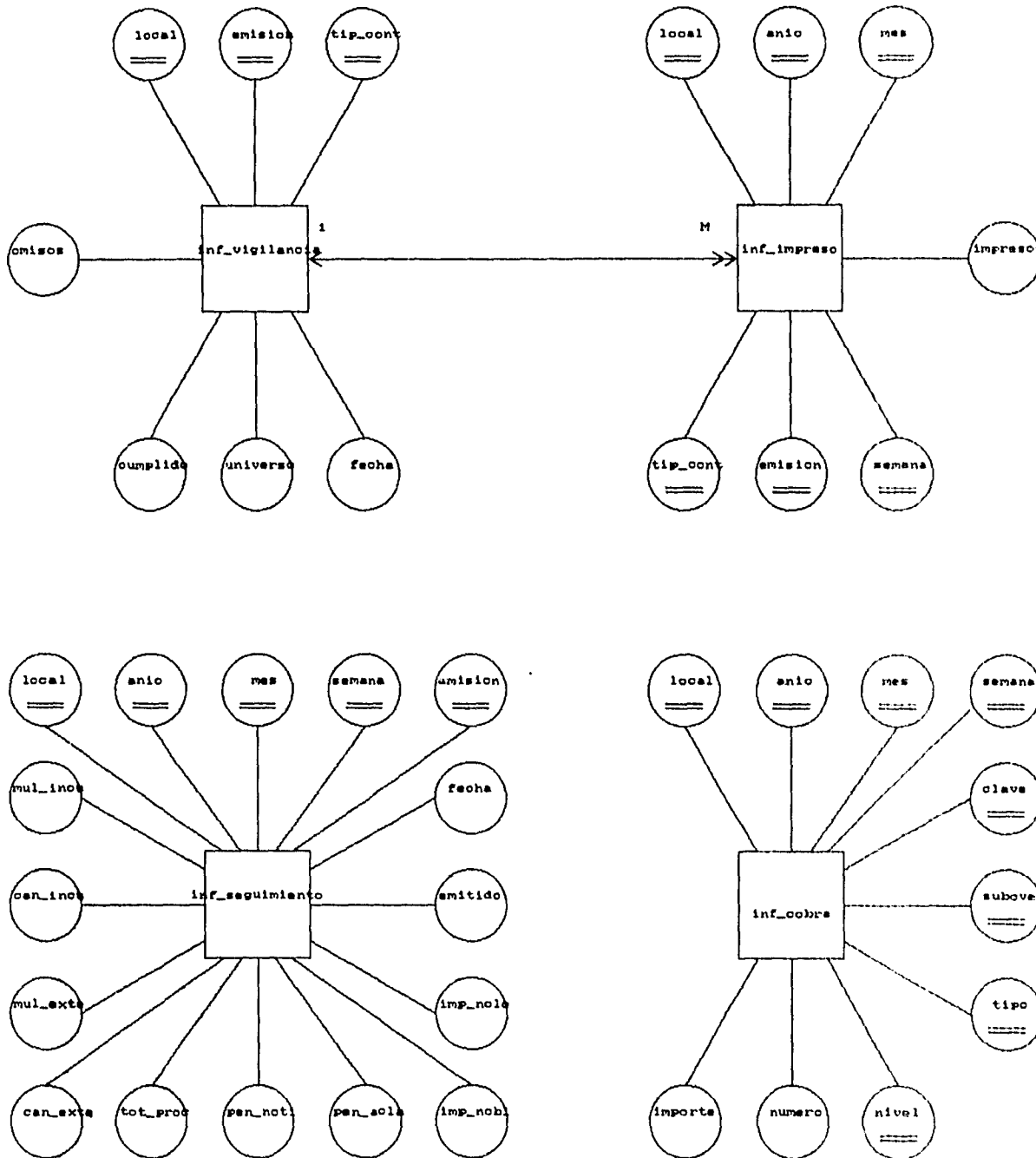


Fig. 2.1.5.c

DIAGRAMA DE CONTEXTO

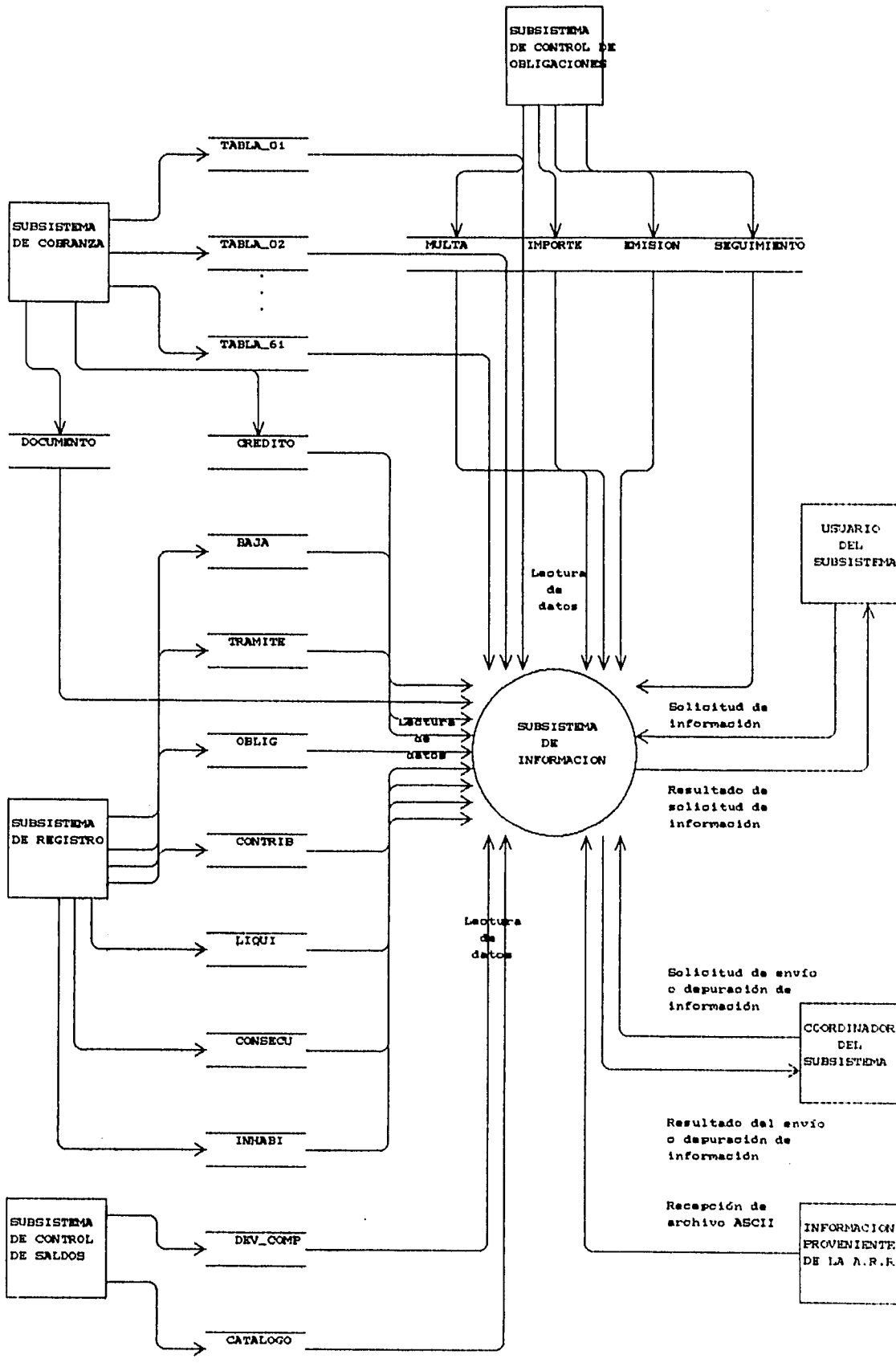


Fig. 2.2.1

DIAGRAMA DE FLUJO DE DATOS
NIVEL 0

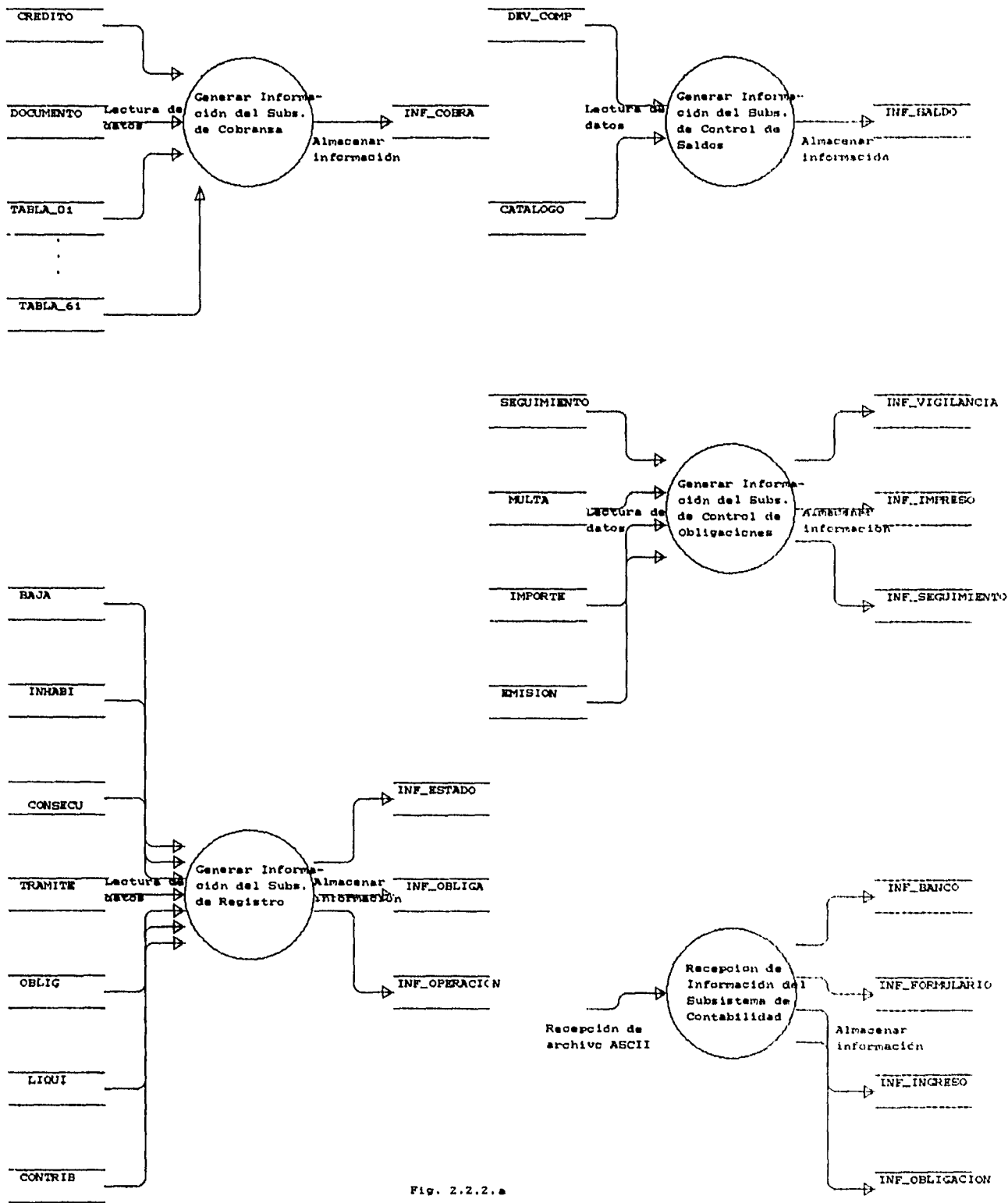


Fig. 2.2.2.a

DIAGRAMA DE FLUJO DE DATOS
NIVEL 0

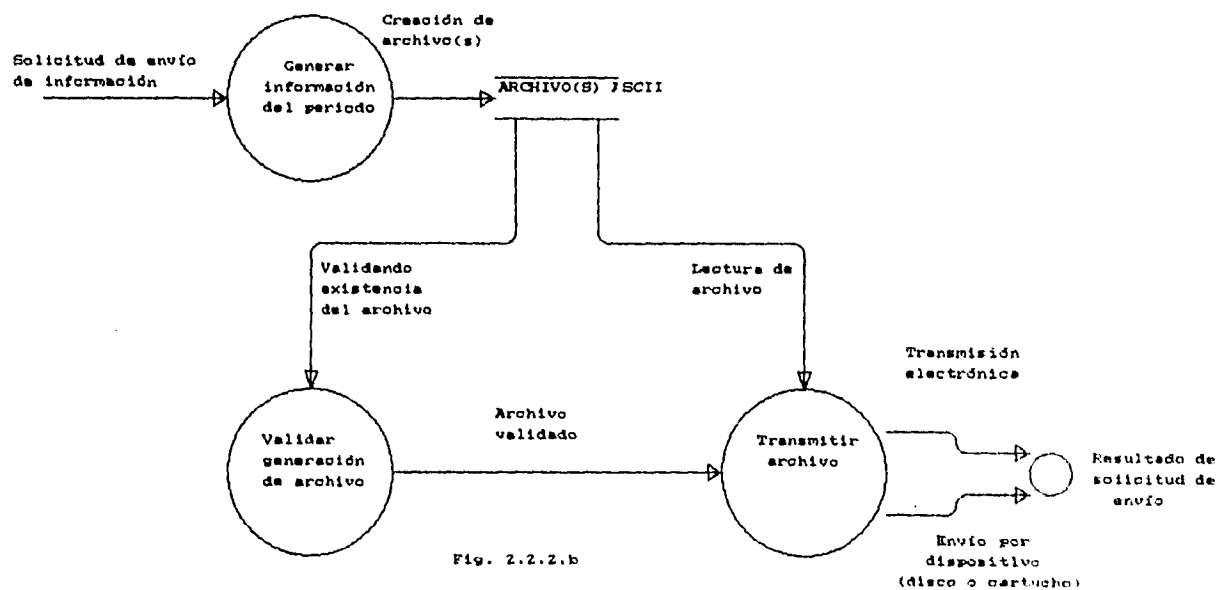
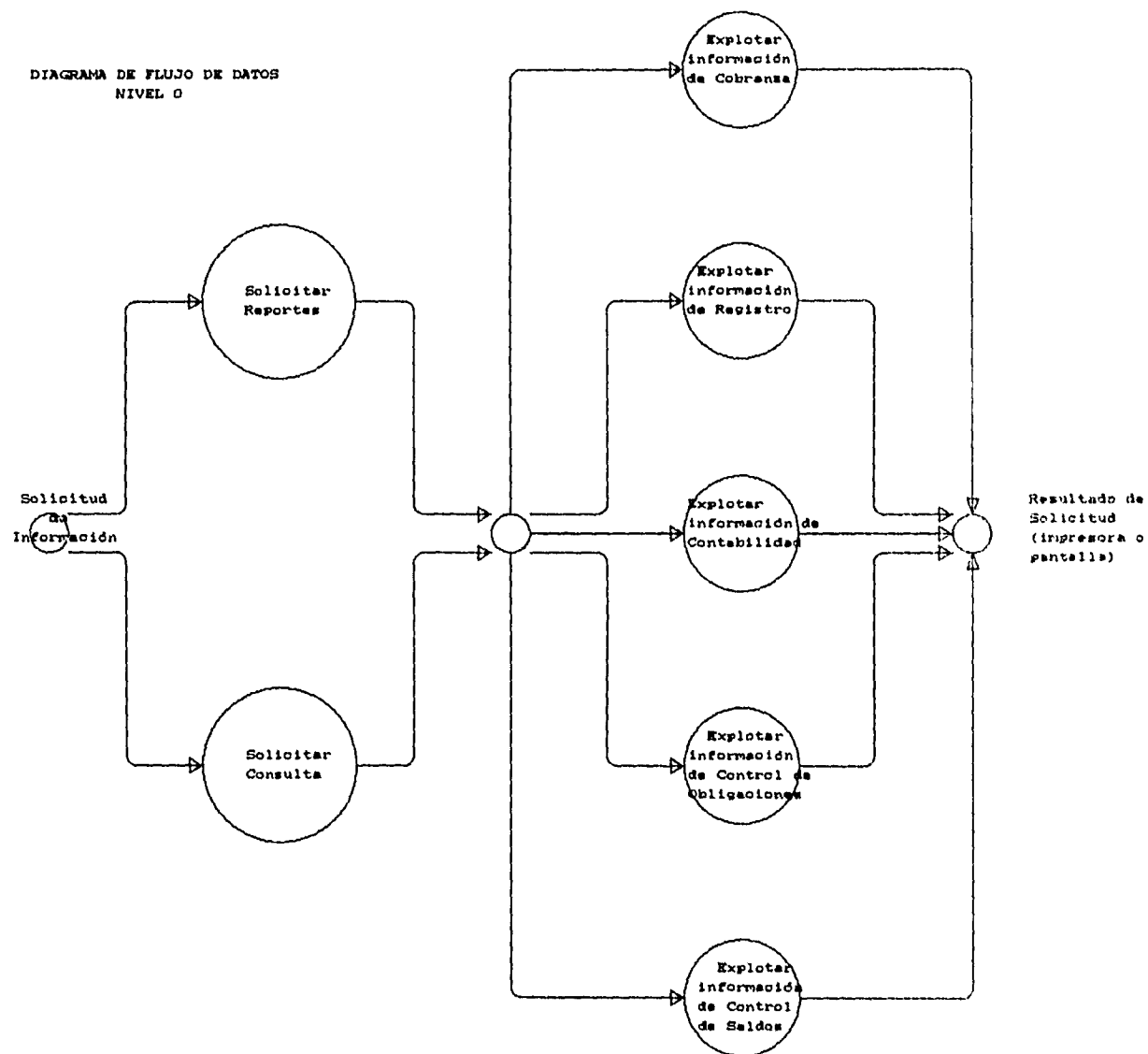


Fig. 2.2.2.b

DIAGRAMA DE FLUJO DE DATOS
DE LA DEPURACION DE
INFORMACION (NIVEL 0)

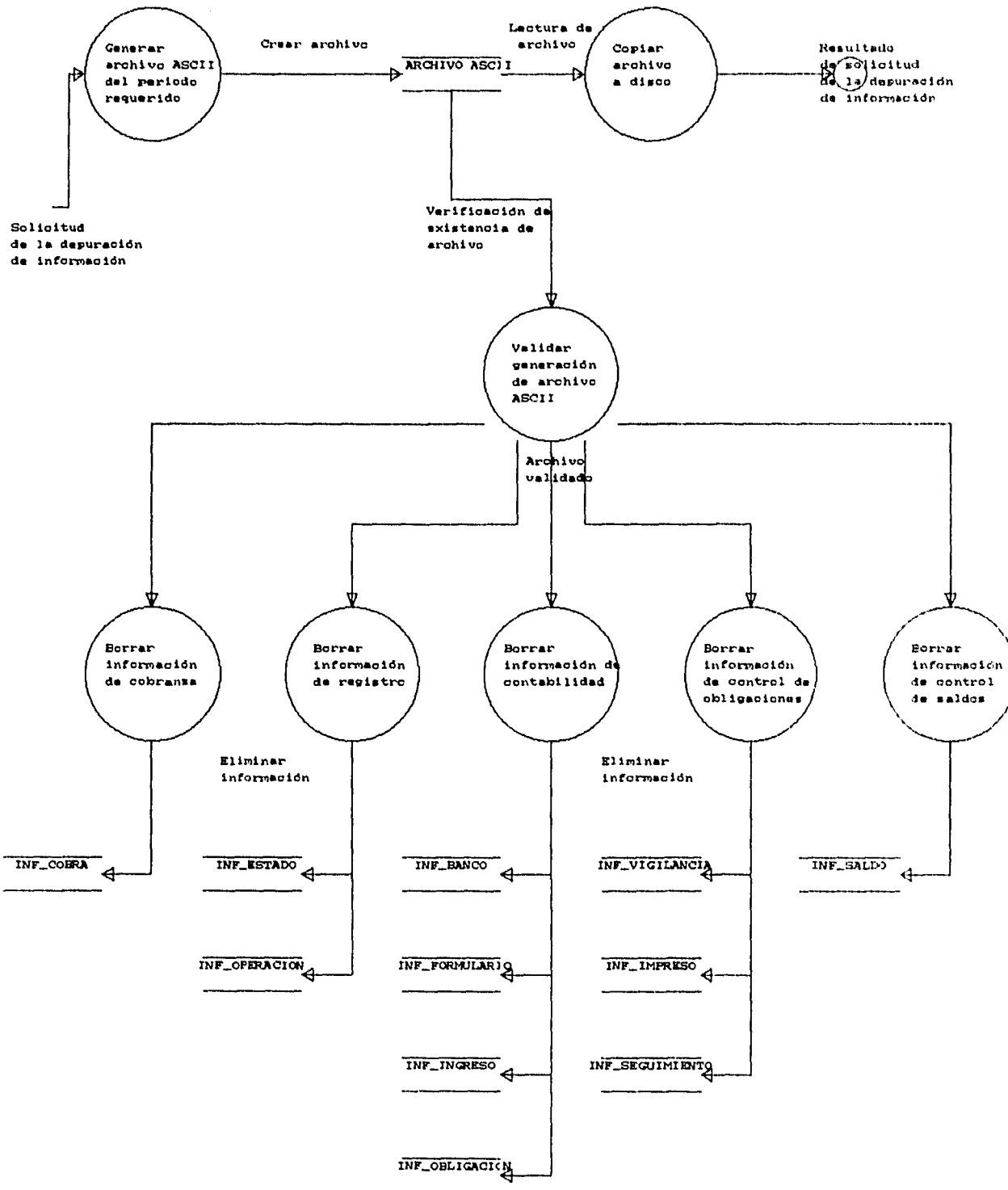


Fig. 2.2.2.a

DIAGRAMA DE FLUJO DE DATOS DE
LA GENERACION DE INFORMACION
DE REGISTRO (NIVEL 1)

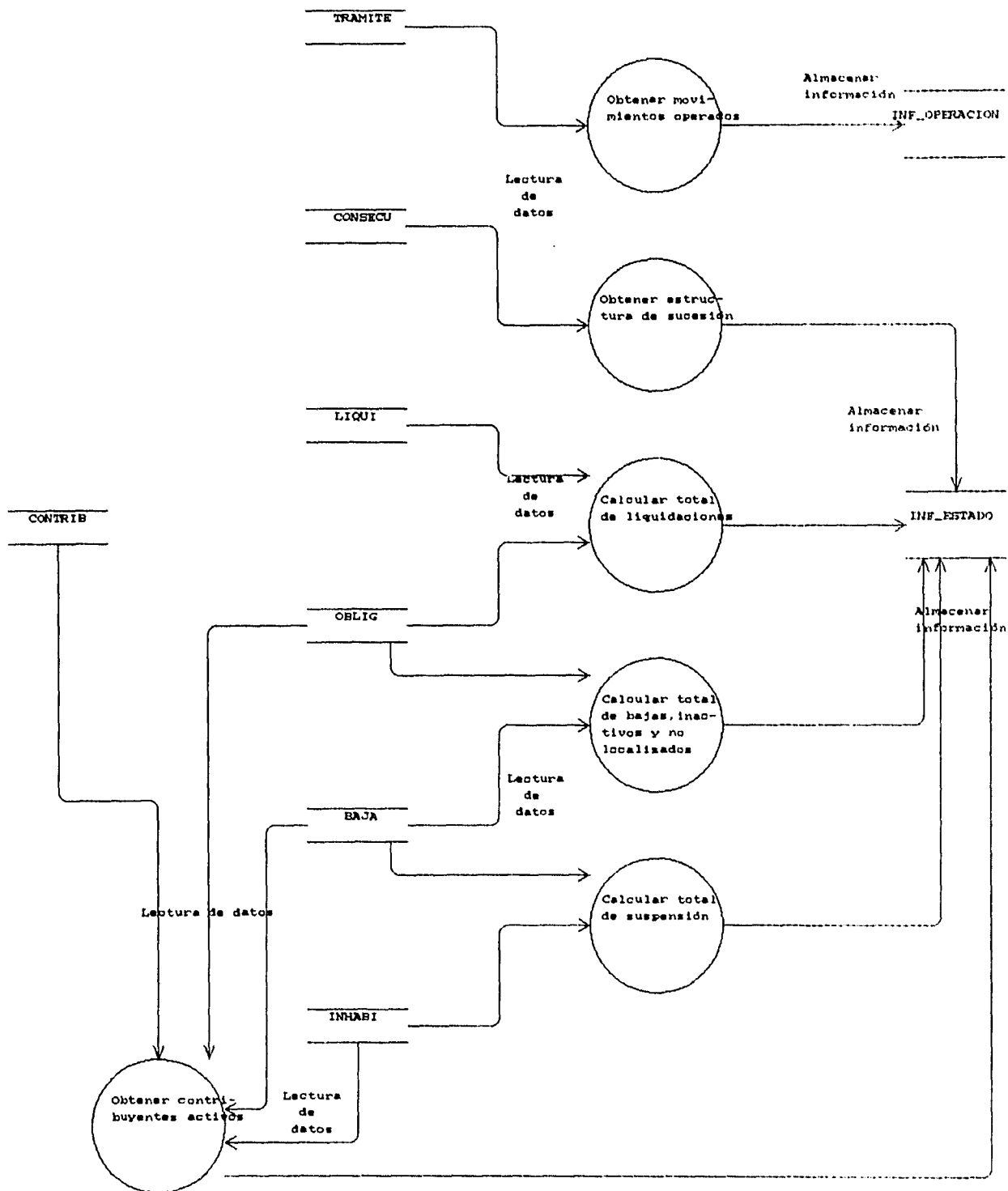


Fig. 2.2.3

DIAGRAMA DE FLUJO DE DATOS
DE LA GENERACION DE INFORMACION
DE COBRANZA (NIVEL 1)

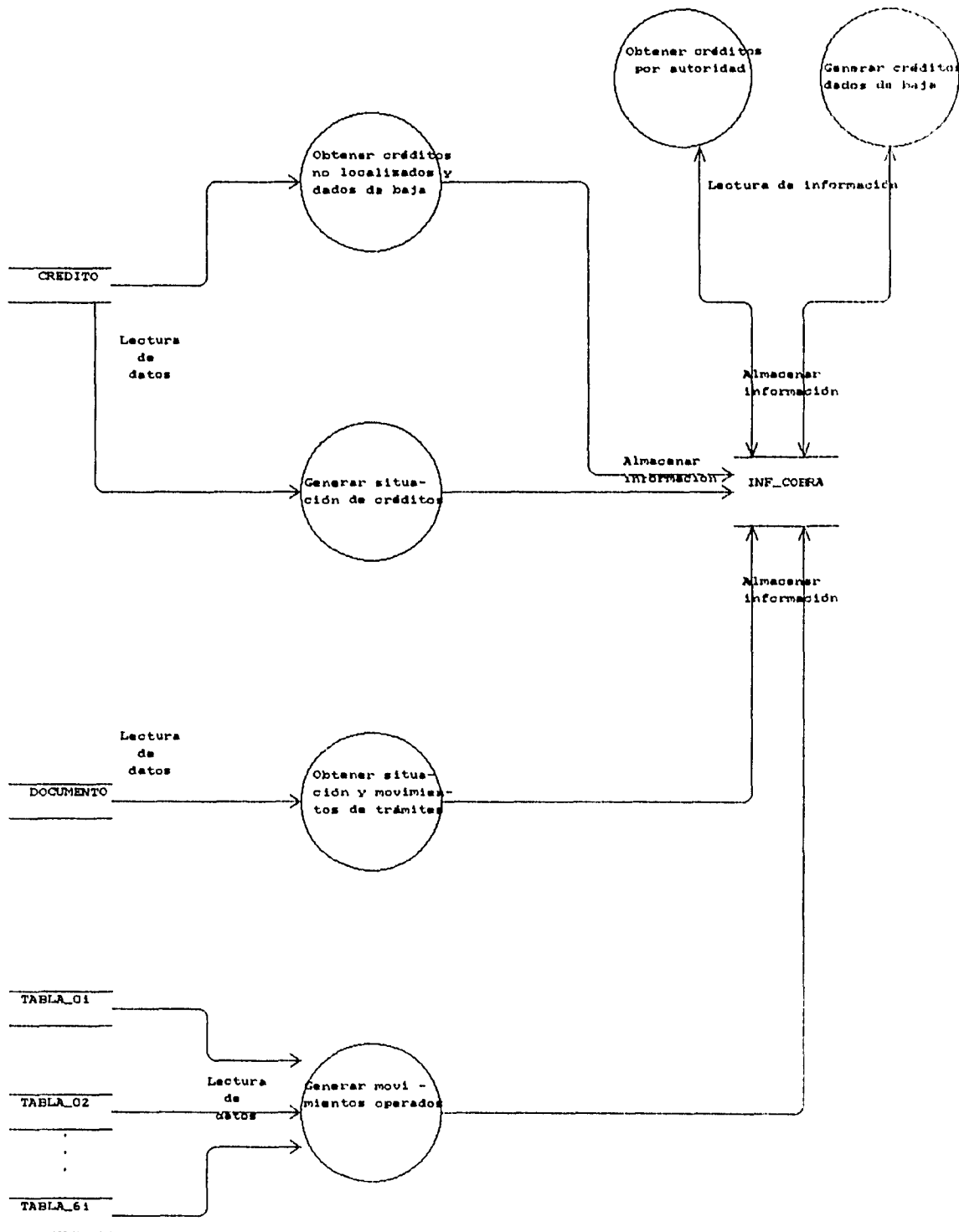


Fig. 2.2.4

DIAGRAMA DE FLUJO DE DATOS DE
LA GENERACION DE INFORMACION
DE CONTROL DE OBLIGACIONES
(NIVEL 1)

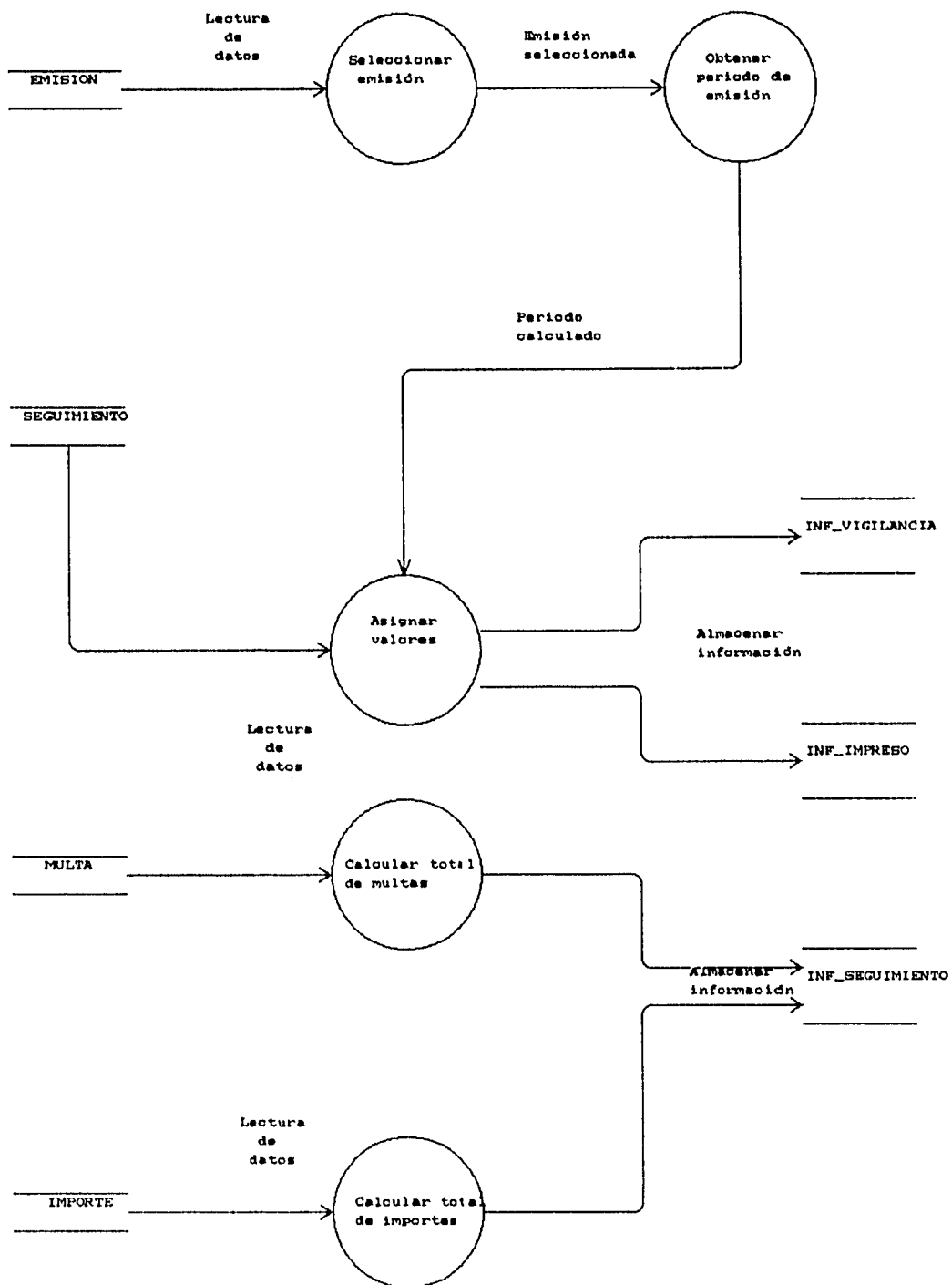


Fig. 2.2.5

DIAGRAMA DE FLUJO DE DATOS DE
LA GENERACION DE INFORMACION
DE CONTROL DE SALDOS (NIVEL 1)

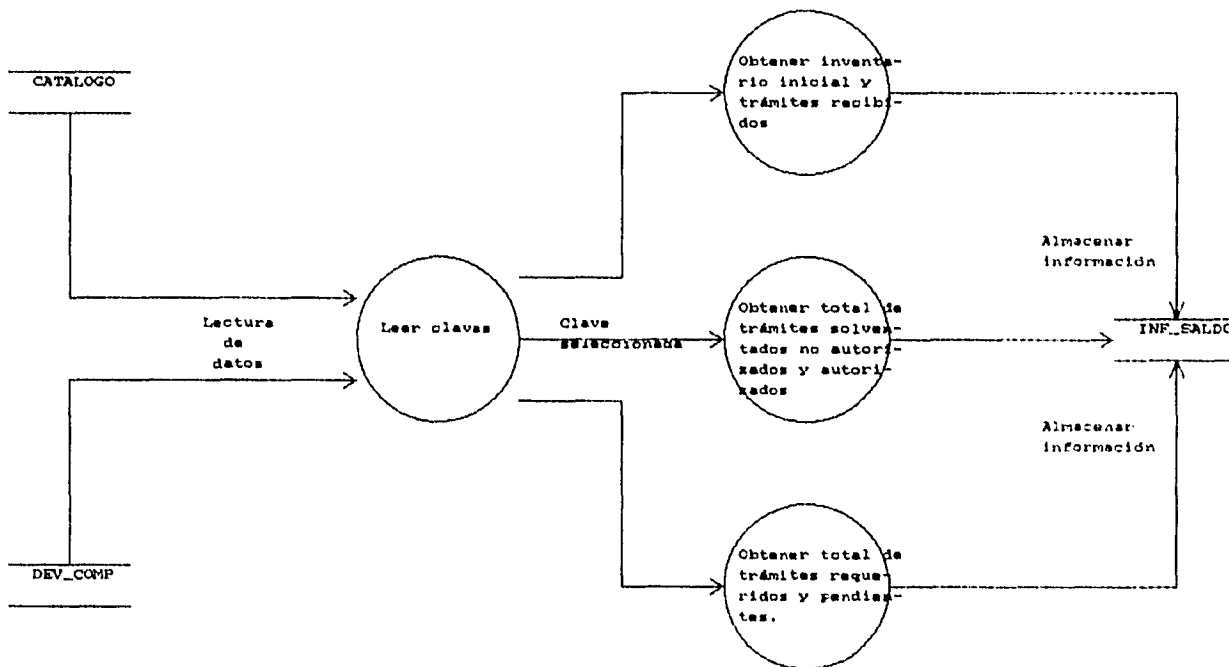


Fig. 2.2.6

DIAGRAMA DE FLUJO DE DATOS
DE LA RECEPCION DE INFORMACION
(NIVEL 1)

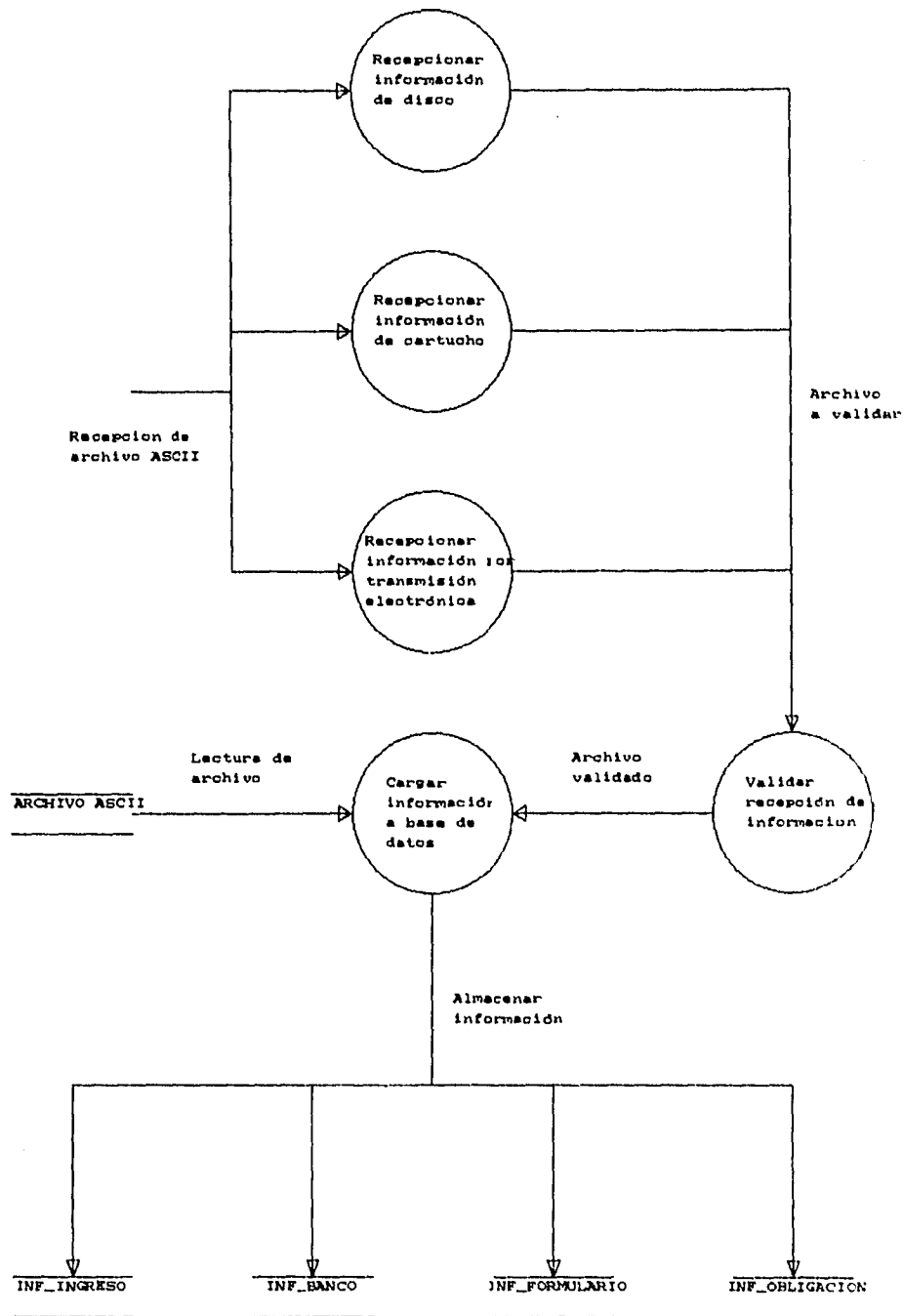


Fig. 2.2.7

DIAGRAMA DE FLUJO DE DATOS
DE SOLICITUD DE INFORMACION
DE COBRANZA Y REGISTRO
(NIVEL 1)

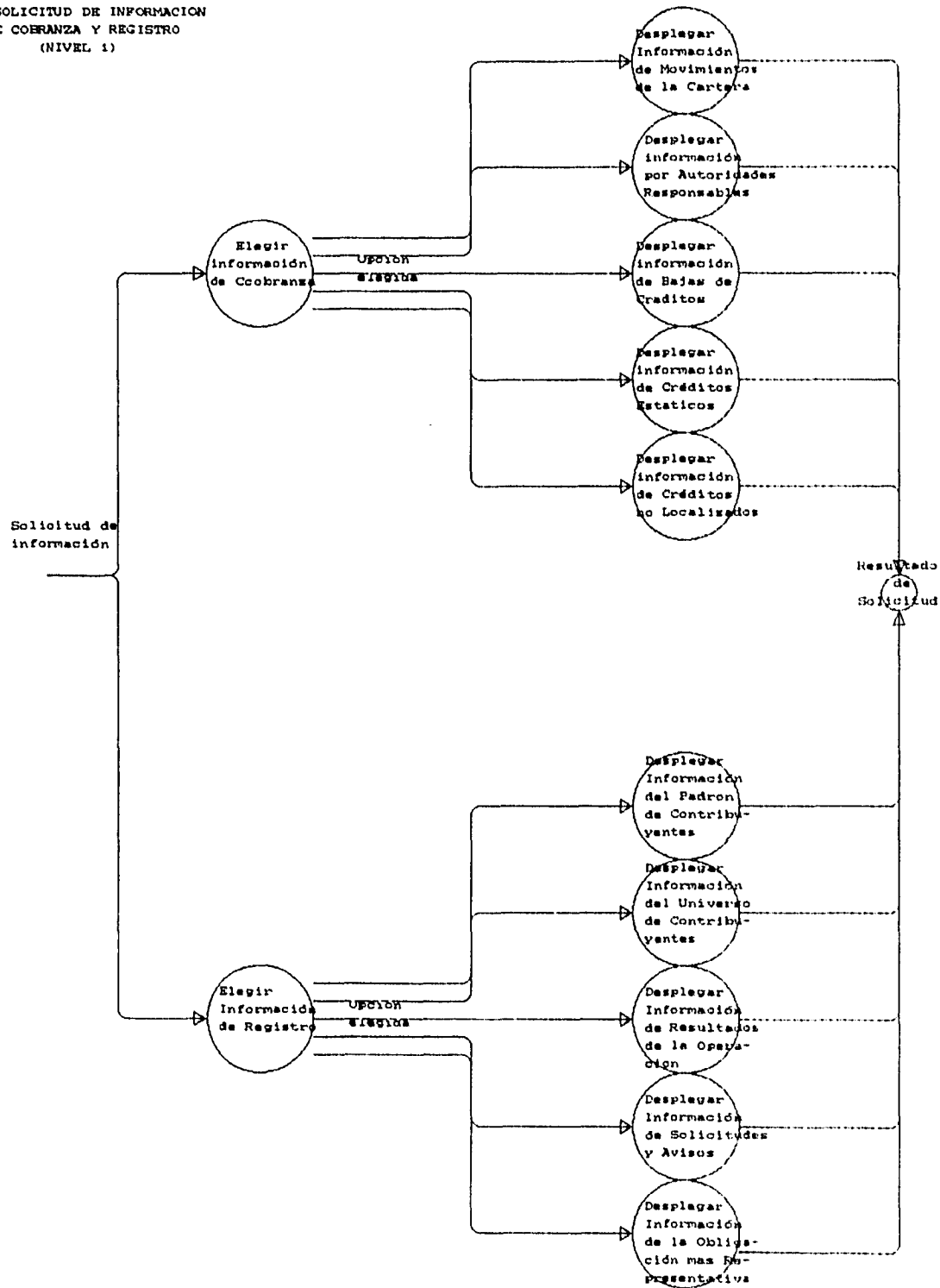


Fig. 2.2.8.a

DIAGRAMA DE FLUJO DE DATOS DE
SOLICITUD DE INFORMACION DE CON-
TROL DE OBLIGACIONES, CONTROL DE
SALDOS Y CONTABILIDAD (NIVEL 1)

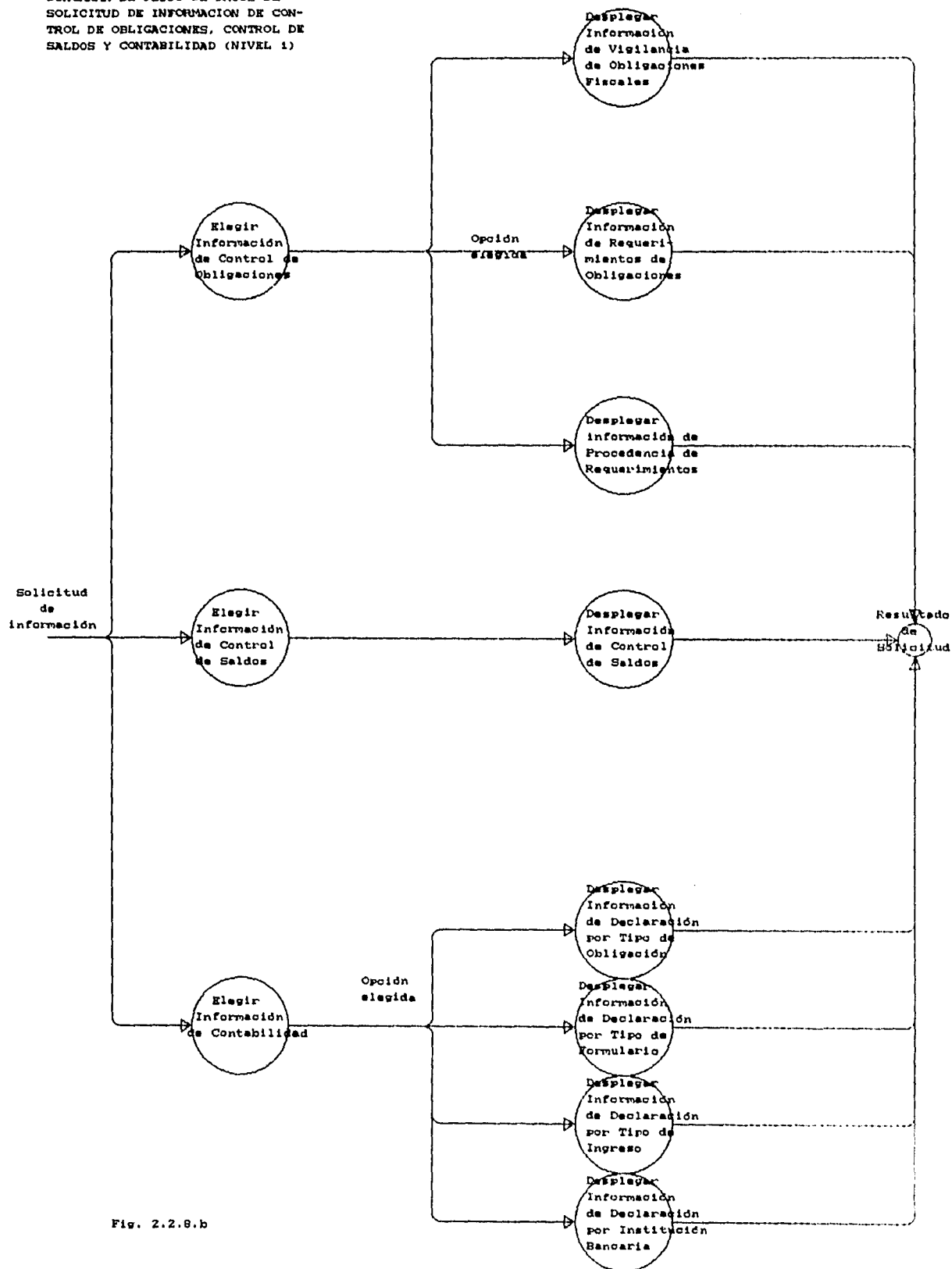


Fig. 2.2.8.b

3. DISEÑO

3.1. ESTRUCTURA DE DATOS

Con base al modelo E-R y determinados los atributos de cada una de las entidades, así como sus llaves primarias se obtiene la estructura de datos del Subsistema de Información:

INF_ESTADO

local	anio	mes	semana	fis_act	fis_baj	fis_suc	fis_tot
PK	PK	PK	PK				

mor_act	mor_baj	mor_liq	mor_tot	uni_act	uni_baj	uni_liq	uni_tot
---------	---------	---------	---------	---------	---------	---------	---------

INF_OPERA

local	anio	mes	semana	movim	cpto	per_mor	no_cont	per_fis	total
PK	PK	PK	PK						

INF_OBLIGA

local	anio	mes	semana	oblig	a_pmor	a_pfis	b_pmor	b_pfis	m_acum	f_acum
PK	PK	PK	PK	PK						

INF_SALDOS

local	anio	mes	semana	clave	cpto	numero	importe
PK	PK	PK	PK	PK	PK		

INF_COBRA

local	anio	mes	semana	clave	subcve	tipo	nivel	numero	importe
PK	PK	PK	PK	PK	PK	PK	PK		

INF_VIGILANCIA

local	emision	tip_cont	fecha	universo	cumplido	omisos
PK	PK	PK				

INF_IMPRESO

local	anio	mes	semana	emision	tip_cont	impresos
PK	PK	PK	PK	PK	PK	

INF_SEGUIMIENTO

local	anio	mes	semana	emision	fecha	emitido	imp_nolo	imp_nobl
PK	PK	PK	PK	PK				

pen_acla	pen_noti	tot_proc	can_exte	mul_exte	can_incu	mul_incu
----------	----------	----------	----------	----------	----------	----------

INF_BANCO

local	fecha	medio	conpag	sinpag	importe
PK	PK	PK			

INF_FORMULARIO

local	fecha	formu	tip_dec	conpag	sinpag	importe
PK	PK	PK	PK			

INF_INGRESO

local	fecha	rubro	importe
PK	PK	PK	

INF_OBLIGACION

local	fecha	tip_obl	conpagf	conpagm	sinpagf	sinpagm
PK	PK	PK				

INF_INFORMA

anio	mes	semana	dia_ini	dia_fin
PK	PK	PK		

3.2 MÓDULOS DEL SISTEMA

En la figura 3.1 se muestra el menú principal del Subsistema de Información :

Los módulos principales con los que cuenta son:

- Generación
- Reportes
- Consulta
- Envío de Información
- Recepción de Información
- Depuración de Información

En la figura 3.2 se muestran los módulos de la generación de información: Registro, Control de Obligaciones, Cobranza y Control de Saldos.

En la figura 3.3 se muestran las opciones del módulo de reportes: Registro, Control de Obligaciones, Cobranza, Contabilidad, y Control de Saldos.

En la figura 3.4 se presentan los reportes que comprende el módulo de Registro: Solicitudes y Avisos, Estructura del Padrón de Contribuyentes, Universo de Contribuyentes Activos, Resultados de la Operación, Obligaciones más Representativas y Todos.

En la figura 3.5.a se presentan las opciones del módulo de Estructura del Padrón de Contribuyentes: Total, Personas Físicas, Personas Morales Matrices, Personas Morales Sucursales y Todos.

Las opciones del módulo de Resultados de la Operación se presentan en la figura 3.5.b: Movimientos del Contribuyente, Mecanismos de Corrección y Todos.

Para el módulo de Obligación más Representativas se tienen las siguientes opciones: Personas Morales, Personas Físicas y Todos. Se muestran en la figura 3.5.c.

Las opciones del módulo de Reportes de Control de Obligaciones se presentan en la figura 3.6: Vigilancia, Requerimientos de Obligaciones y Procedencia de Requerimientos.

Para los módulos de Vigilancia, Requerimientos de Obligaciones y Procedencia de Requerimientos, las opciones con que cuentan son: Emisión, Año de Emisión y Todas las Emisiones, esto se muestra en las figuras 3.7.a, 3.7.b y 3.7.c.

En la figura 3.8 se muestran las opciones que comprende el módulo de Reportes de Cobranza: Créditos Fiscales, Pago en Parcialidades y Todos.

Las opciones del módulo de Créditos Fiscales se hacen referencia en la figura 3.9.a: Movimientos de la Cartera de Créditos Fiscales, Cartera de Créditos por Claves de Trámite, Movimientos de la Cartera por claves de trámite, Créditos por Autoridad Responsable, Créditos dados de Baja, Créditos Estáticos y Todos.

El módulo de Pago en Parcialidades se presenta en la figura 3.9.b: Totales, Millonarios, No millonarios y Todos.

Los reportes del módulo de Contabilidad son: Declaraciones por Tipo de Obligación, Declaraciones por tipo de Formulario, Declaraciones por tipo de Ingresos, Declaraciones por Institución Bancaria y Todos, los cuales se presentan en la figura 3.10.

La opción del módulo de Reportes de Control de Saldos es Comportamiento de Control de Saldos que se muestra en la figura 3.11.

Nota: En la opción de Todos se emiten todos los reportes del módulo correspondiente.

En la figura 3.12 se muestran las opciones del módulo de Consulta: Registro, Control de Obligaciones, Cobranza, Contabilidad, y Control de Saldos.

Las opciones del módulo de Consulta de Registro se presentan en la figura 3.13: Solicitudes y Avisos, Estructura del Padrón de Contribuyentes, Universo de Contribuyentes Activos, Resultados de la Operación y Obligación más Representativa.

Las opciones que comprende el módulo de Estructura del Padrón de Contribuyentes se muestran en la figura 3.14.a: Total, Personas Físicas, Personas Morales Matrices y Personas Morales Sucursales.

En la figura 3.14.b se presentan las opciones del módulo de Resultados de la Operación: Movimientos del Contribuyente y Mecanismos de Corrección.

El módulo de Obligación más Representativa cuenta con las opciones: Personas Morales y Personas Físicas, esto se muestra en la figura 3.14.c.

El módulo Consulta de Control de Obligaciones consta de las opciones: Vigilancia, Requerimientos de Obligaciones y Procedencia de Requerimientos, el cual se muestra en la figura 3.15.

En las figuras 3.16.a, 3.16.b y 3.16.c se presentan los módulos Vigilancia, Requerimientos de Obligaciones y Procedencia de Requerimientos cuyas opciones son: Emisión, Año de Emisión y Todas las Emisiones.

El módulo Consulta de Cobranza se muestra en la figura 3.17, contando con las opciones Créditos Fiscales y Pago en Parcialidades.

Las opciones del módulo de Créditos Fiscales: Movimientos de la Cartera de Créditos Fiscales, Cartera de Créditos por Claves de Trámite, Movimientos de la Cartera por claves de trámite, Créditos por Autoridad Responsable, Créditos dados de Baja y Créditos Estáticos, se presentan en la figura 3.18.a.

En la figura 3.18.b se presentan las opciones del módulo de Pago en Parcialidades, las cuales son: Totales, Millonarios y No millonarios.

Las opciones del módulo Consulta de Contabilidad: Declaraciones por tipo de Obligación, Declaraciones por tipo de Formulario, Declaraciones por tipo de Ingreso y Declaraciones por Institución Bancaria se presentan en la figura 3.19.

El módulo Consulta de Control de Saldos cuenta con la opción Comportamiento de Control de Saldos que se presenta en la figura 3.20.

En la figura 3.21 se muestran los módulos que comprende el envío de información: Todos, Registro, Control de Obligaciones, Cobranza y Control de Saldos.

En la opción de Todos se puede realizar el envío de información de todas las áreas que comprende el Subsistema (excepto Contabilidad).

En la figura 3.22.a se muestran las opciones del módulo de Envío de Información: Generación de archivo(s) ASCII y Forma de Envío.

En la figura 3.22.b se muestran los módulos de la Forma de Envío: Discos Flexibles, Cartucho y Transmisión Electrónica.

El módulo de Recepción de la Información cuenta con las opciones: Forma de Recepción y Carga de Información, esto se presenta en la figura 3.23.a.

Las opciones que comprende el módulo Forma de Recepción se muestran en la figura 3.23.b y son: Discos Flexibles, Cartucho y Transmisión Electrónica.

El módulo Depuración de Información cuenta con las opciones Respaldo de Información y Eliminación de Información de la Base de Datos, mostrado en la figura 3.24.a.

En la figura 3.24.b se presenta el módulo Respaldo de Información cuyas opciones son: Generación de Archivo ASCII y Copiado a Disco Flexible.

Las opciones que comprende el módulo Copiado a Disco Flexible son: Alta Densidad y Doble Densidad mostradas en la figura 3.24.c.

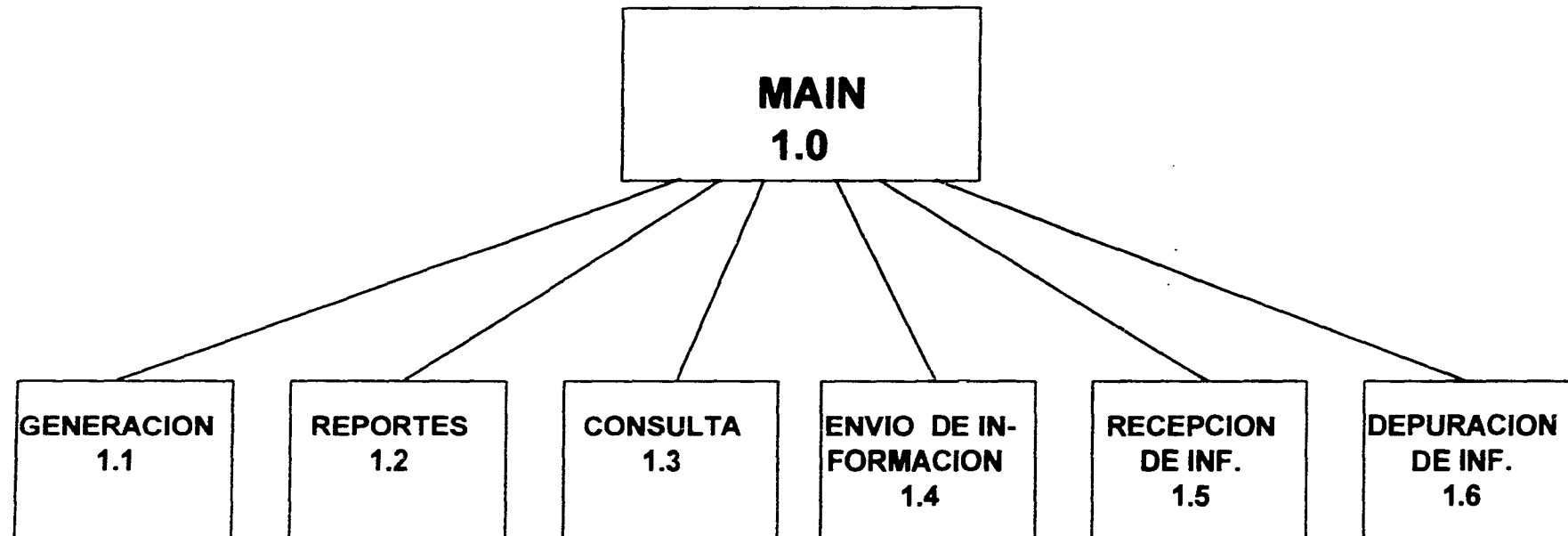


Figura 3.1

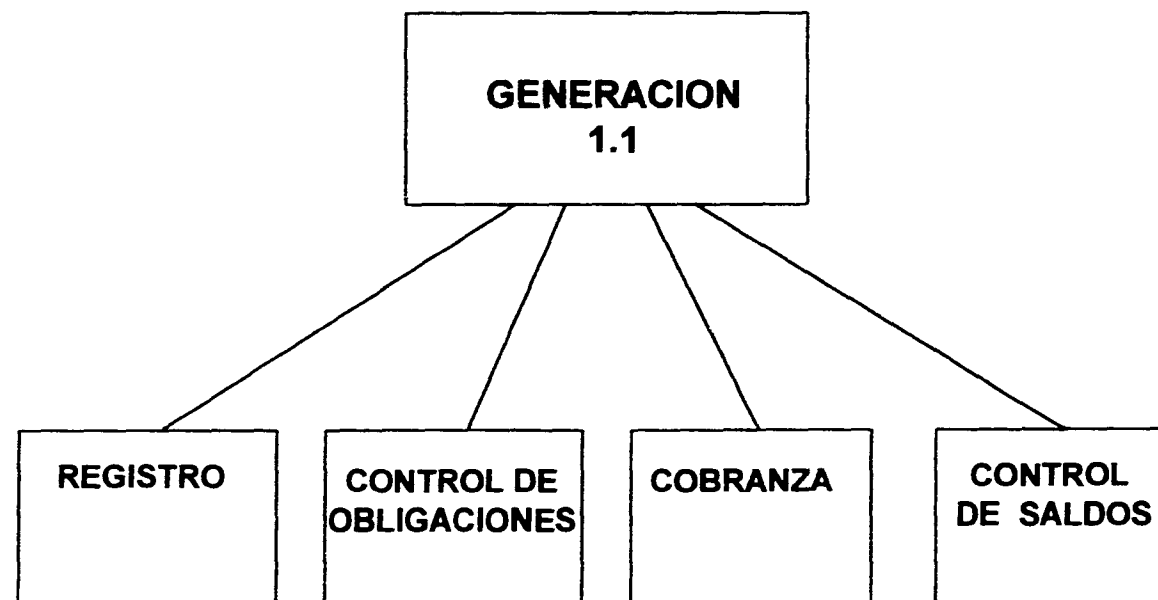


Figura 3.2

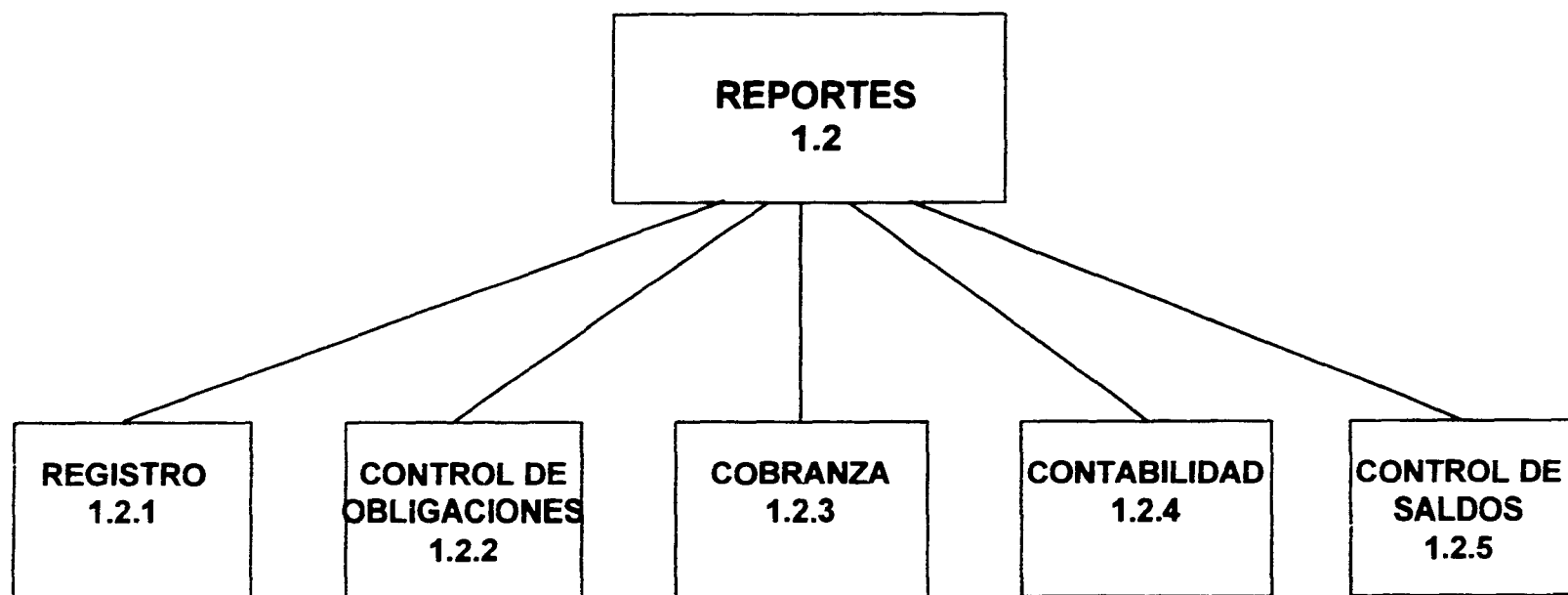


Figura 3.3



Figura 3.4

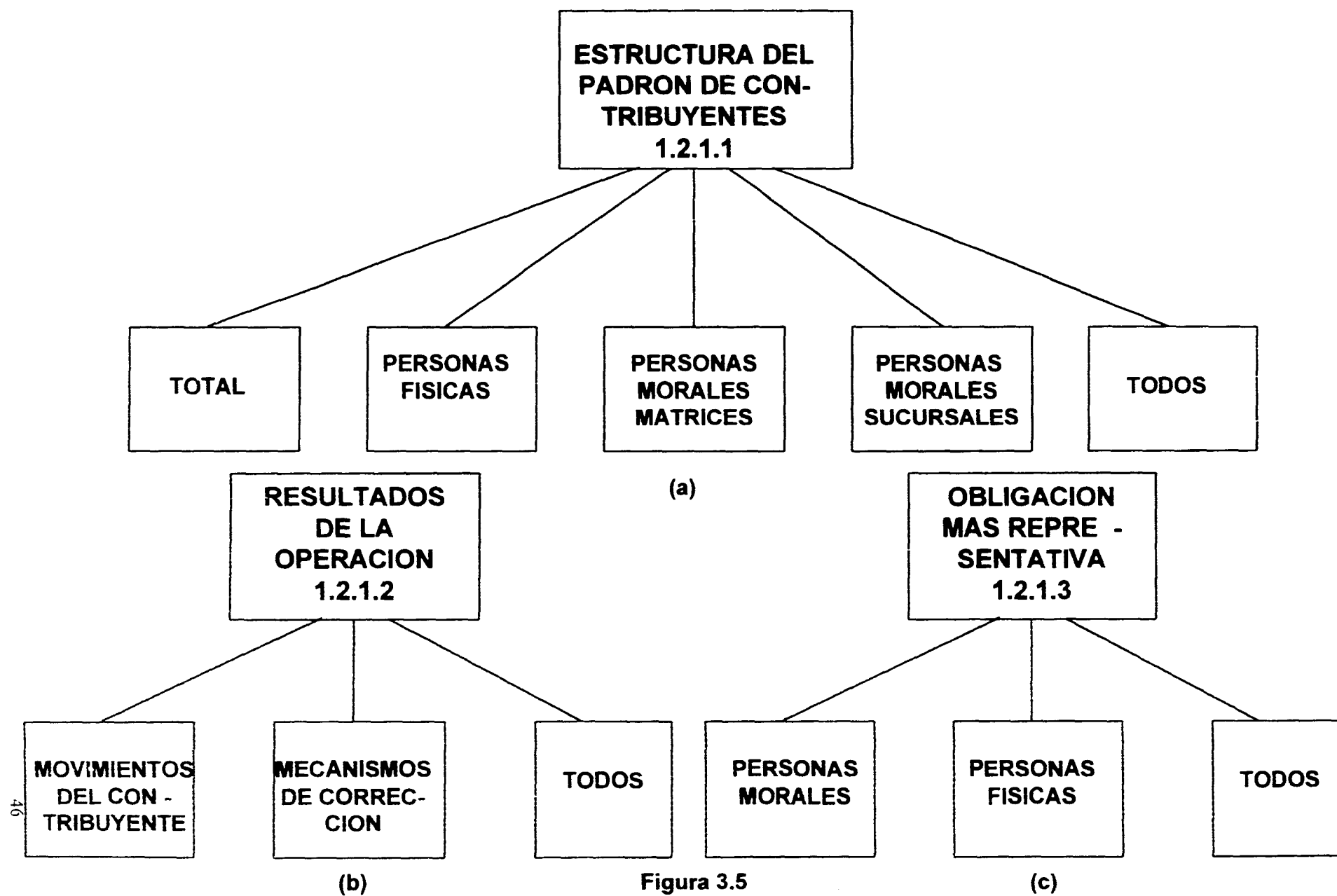
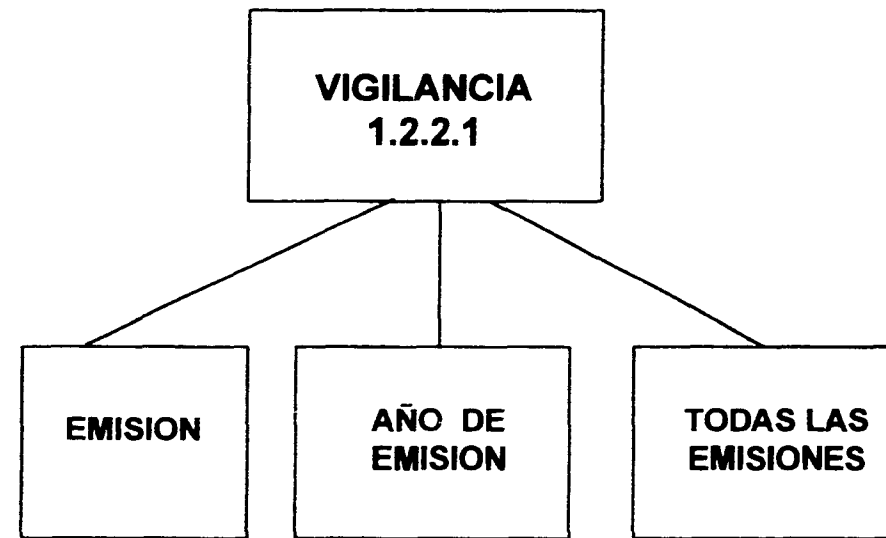


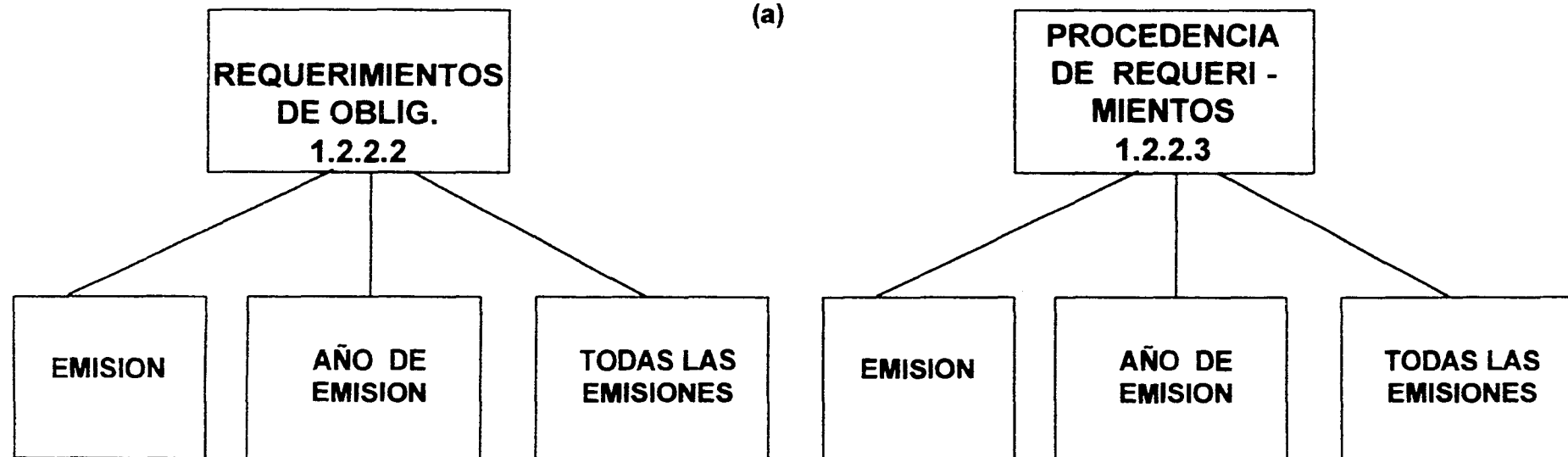
Figura 3.5



Figura 3.6



(a)



(b)

Figura 3.7

(c)

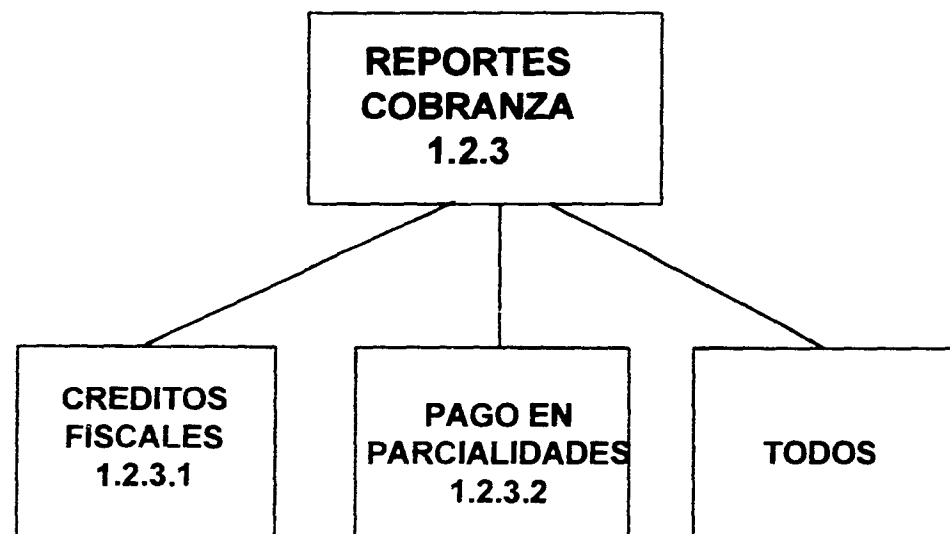
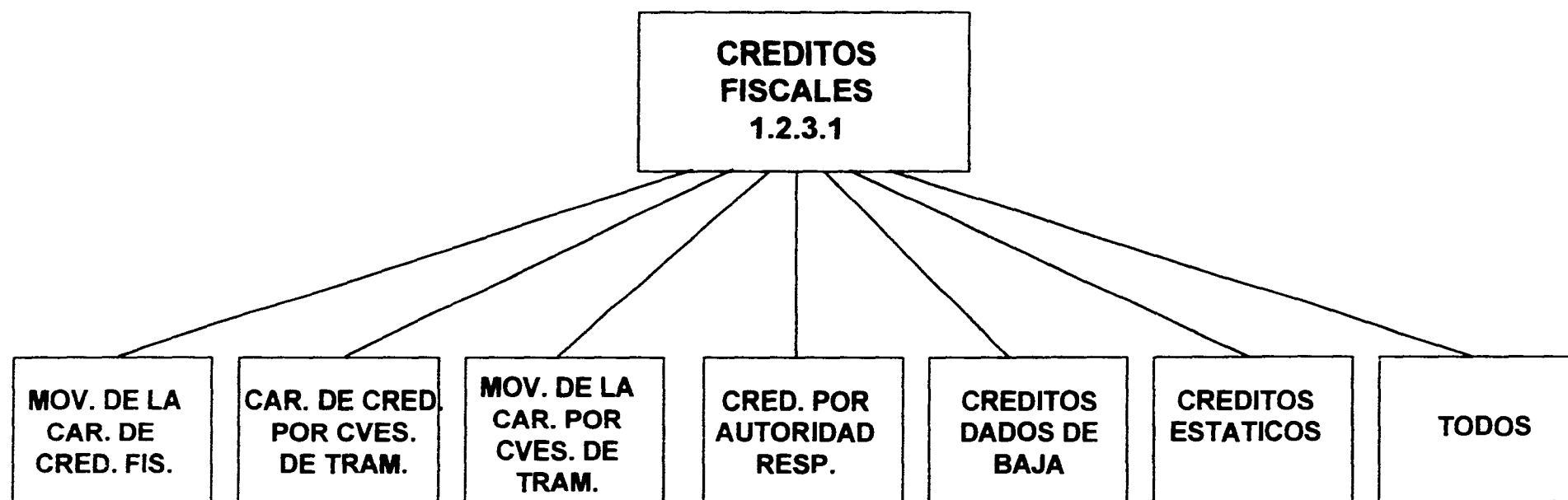
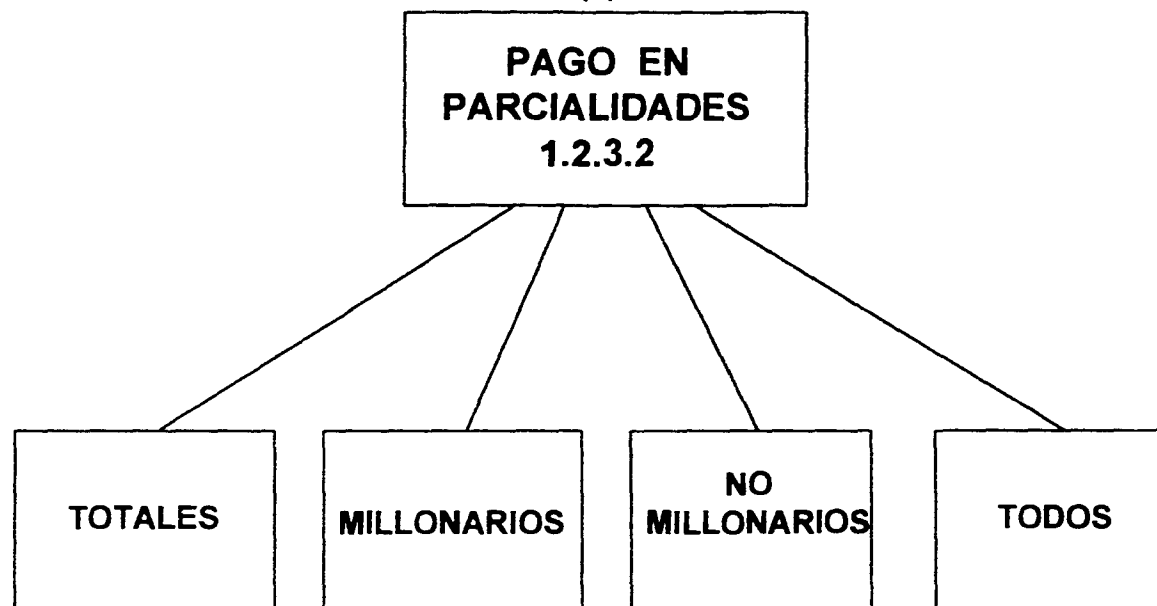


Figura 3.8



(a)



(b)

Figura 3.9



Figura 3.10

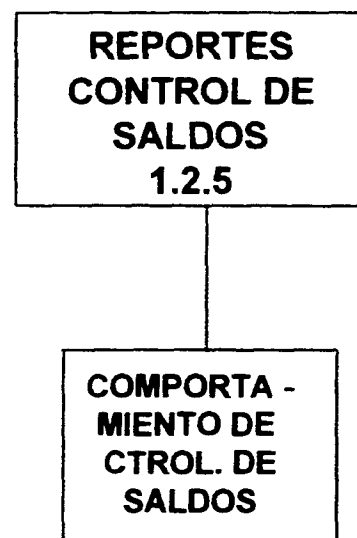


Figura 3.11

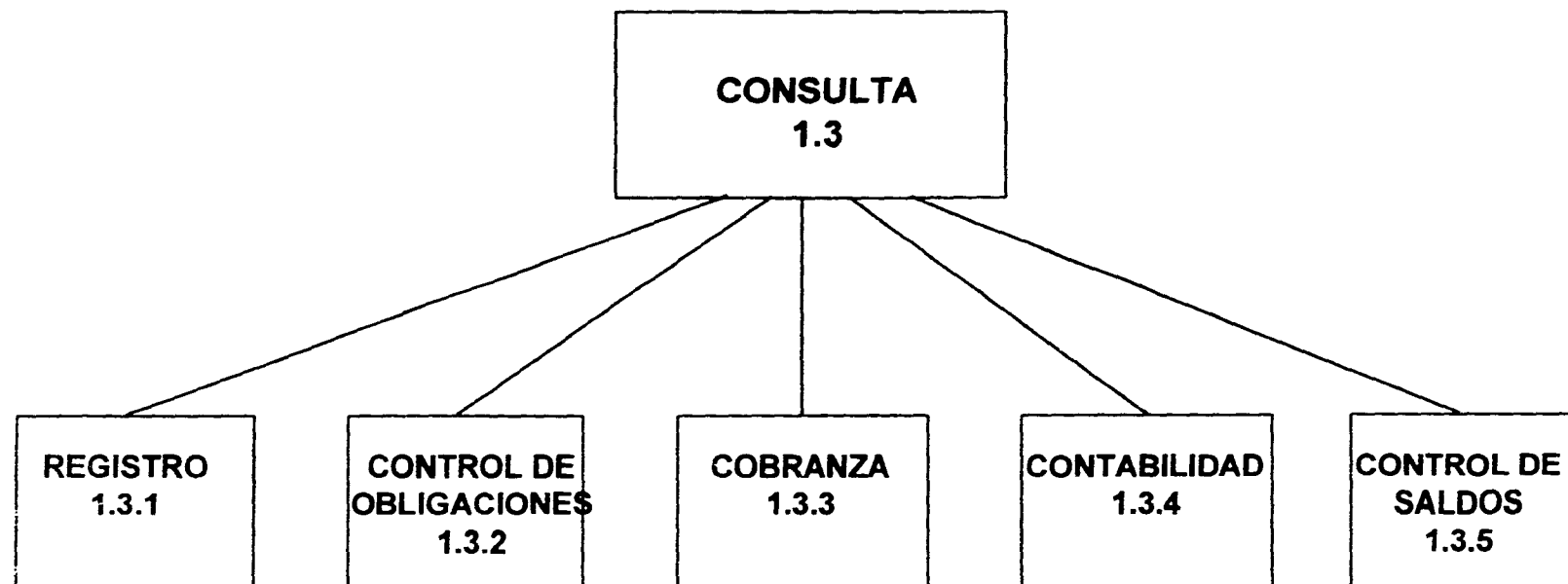


Figura 3.12

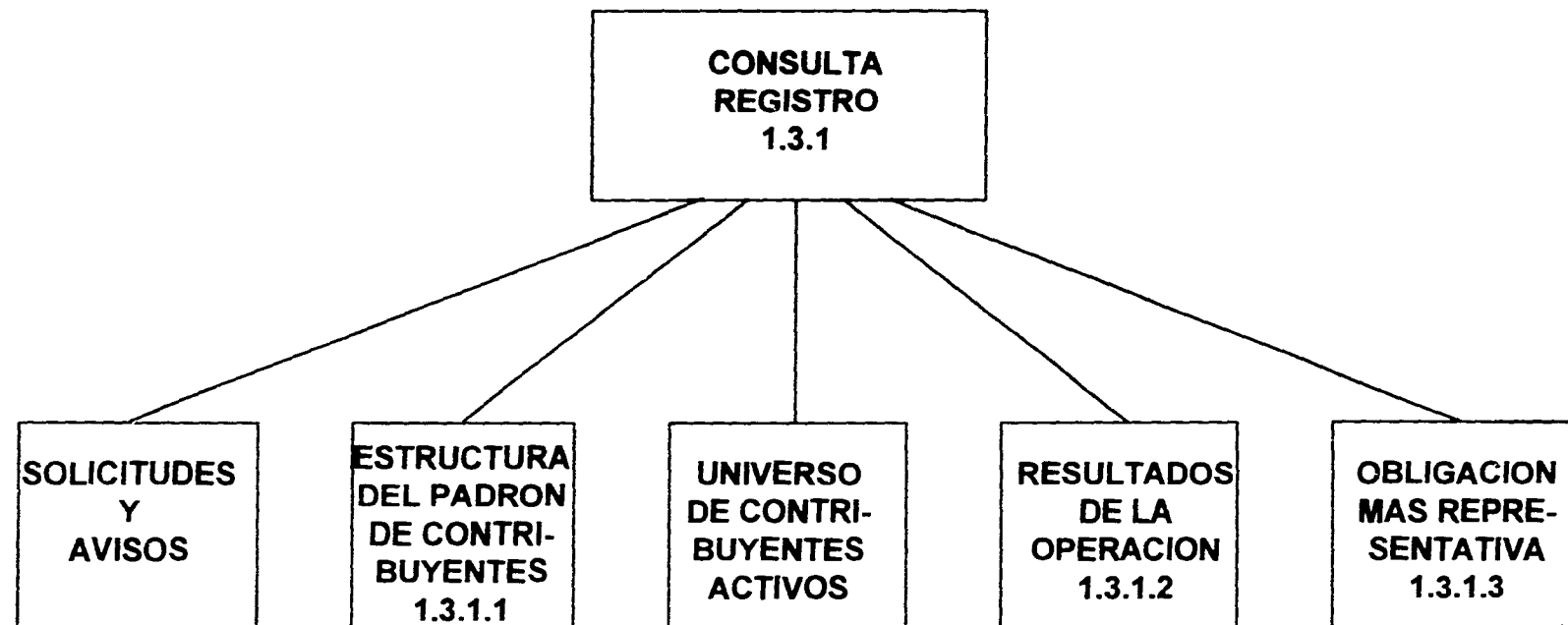


Figura 3.13

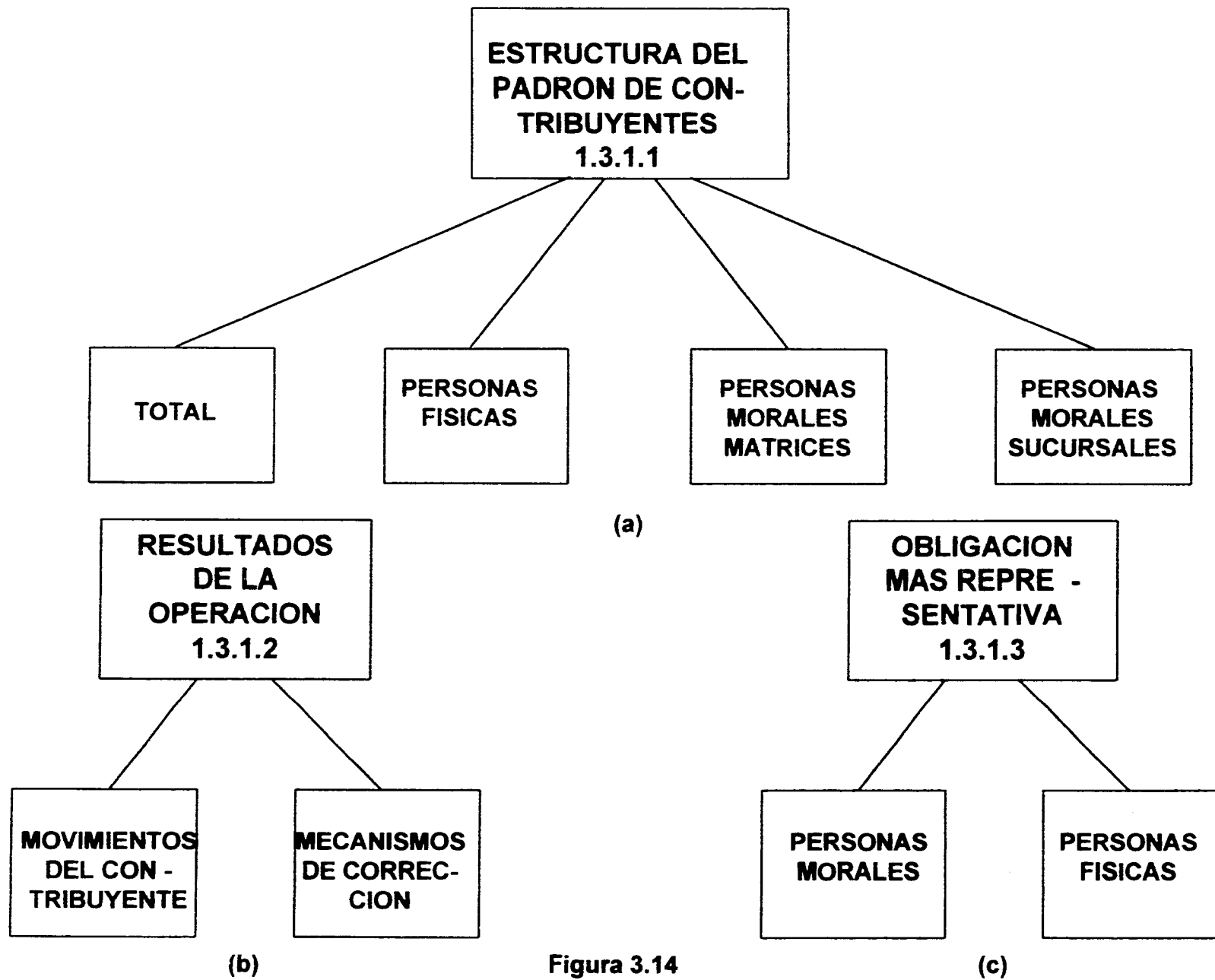


Figura 3.14

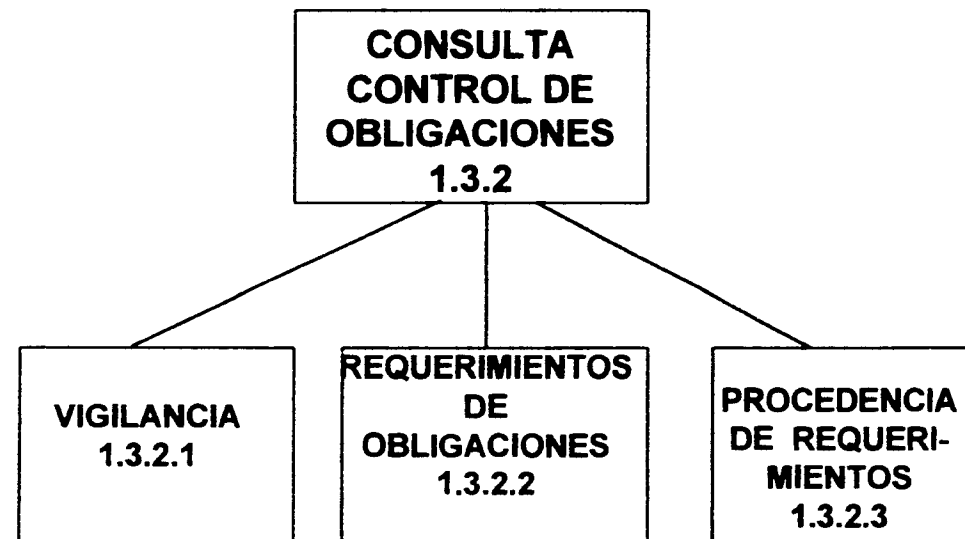
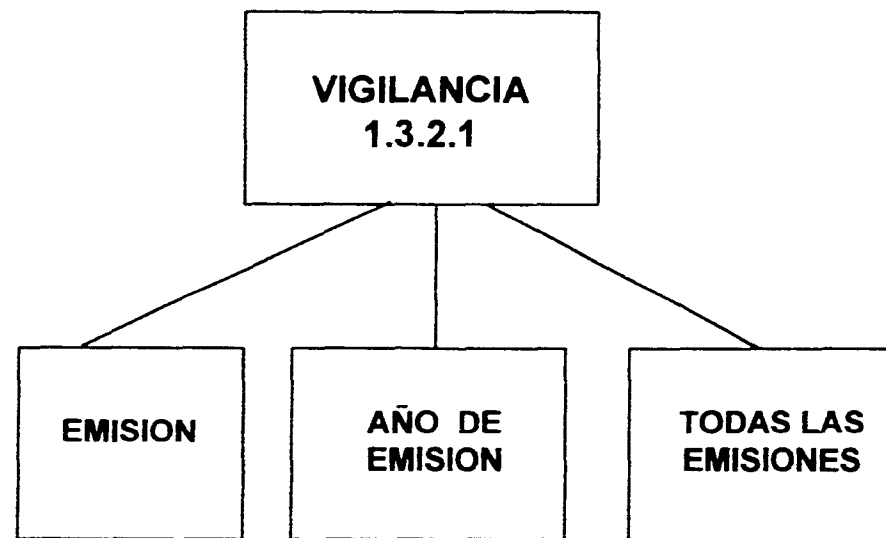
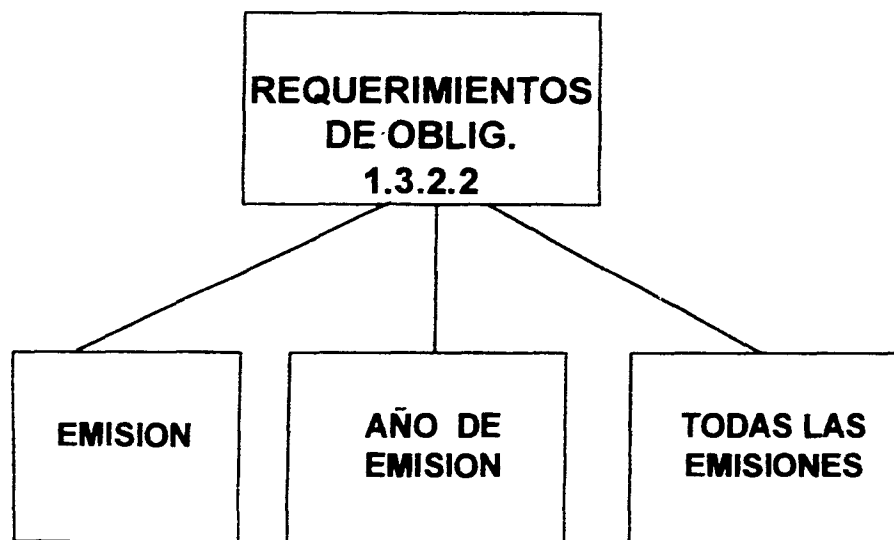


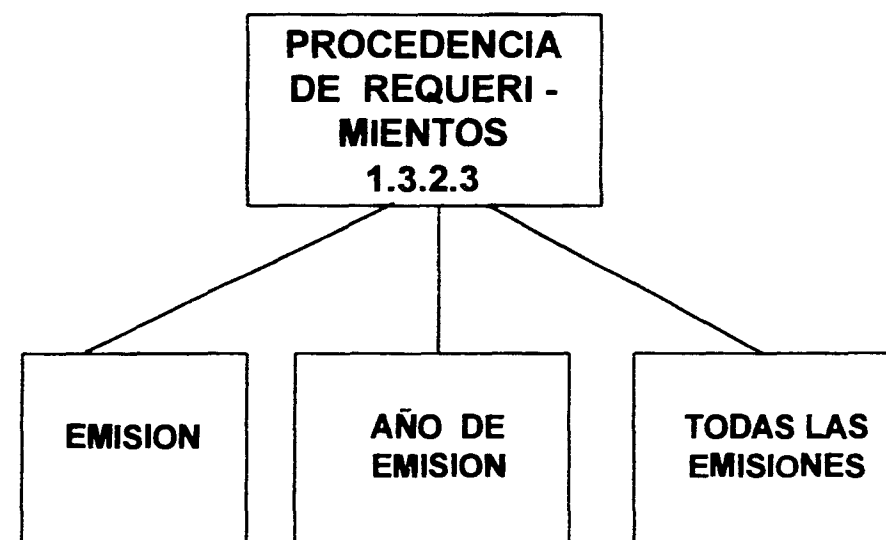
Figura 3.15



(a)



(b)



(c)

Figura 3.16

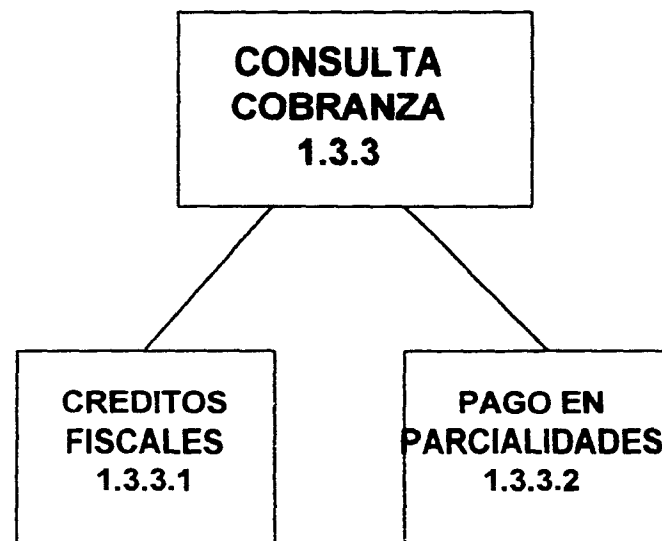
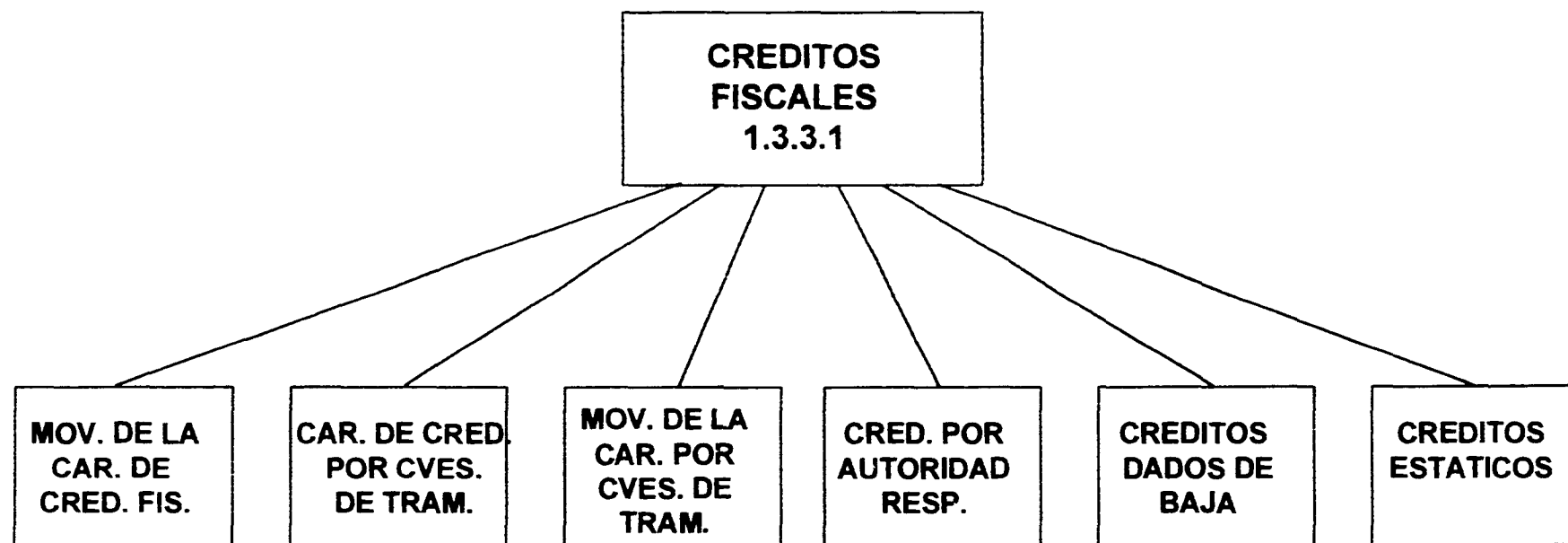
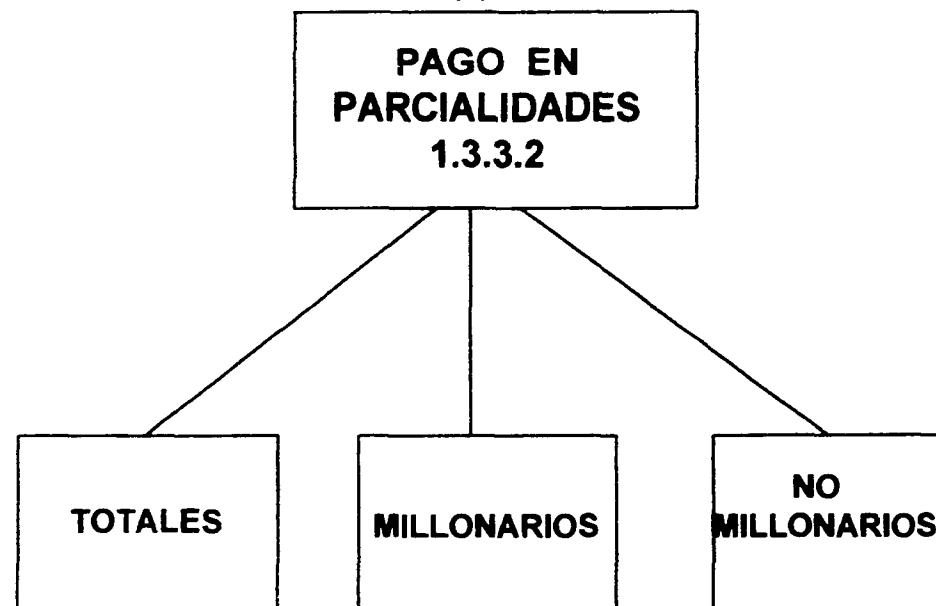


Figura 3.17



(a)



(b)

Figura 3.18

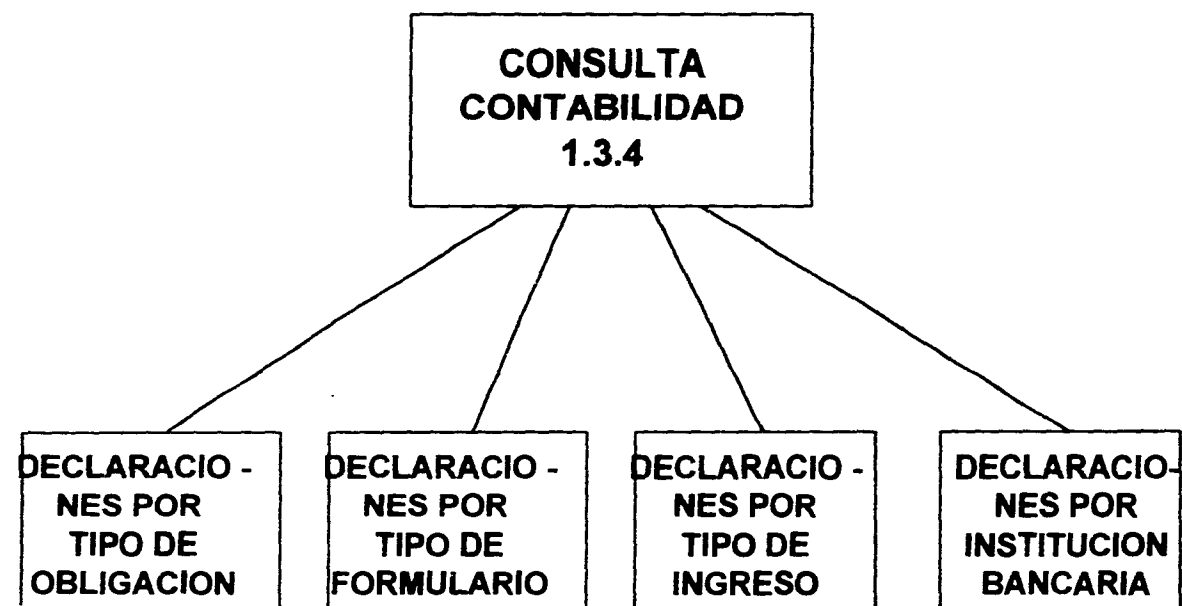


Figura 3.19

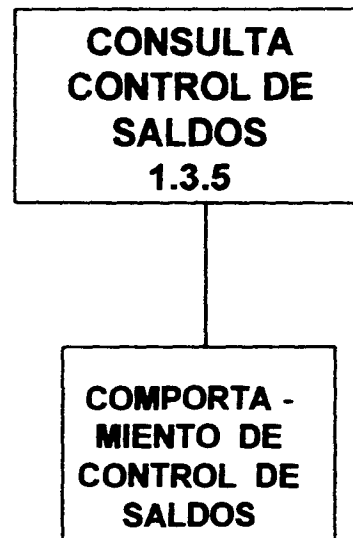


Figura 3.20

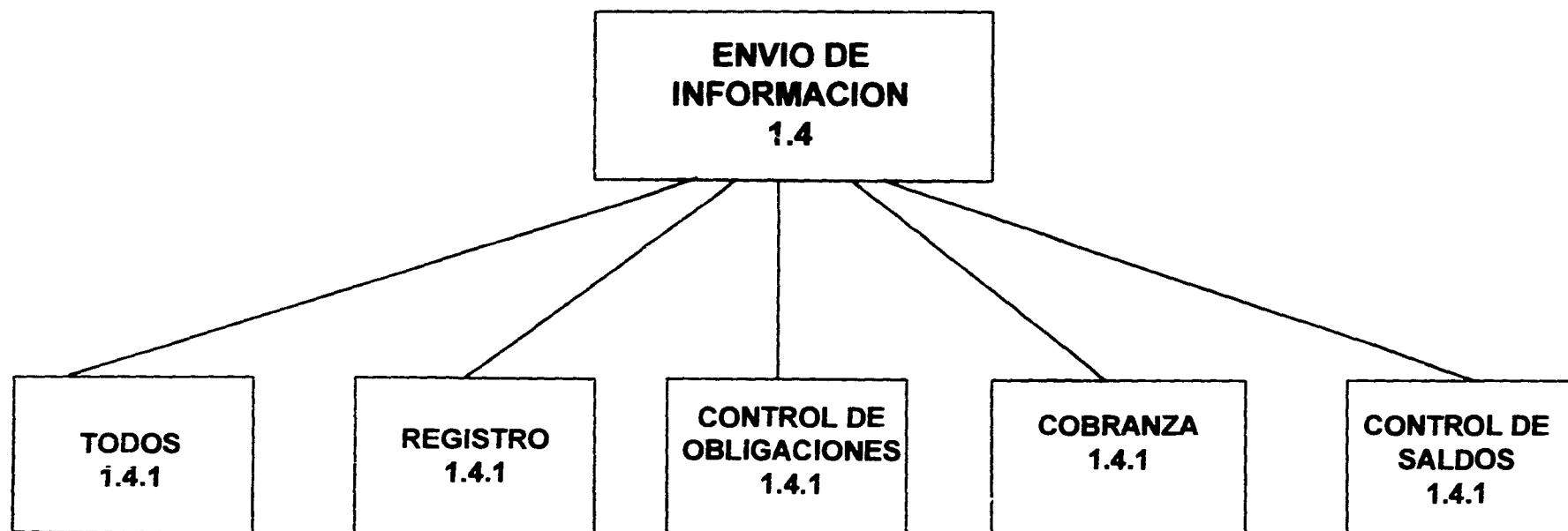


Figura 3.21

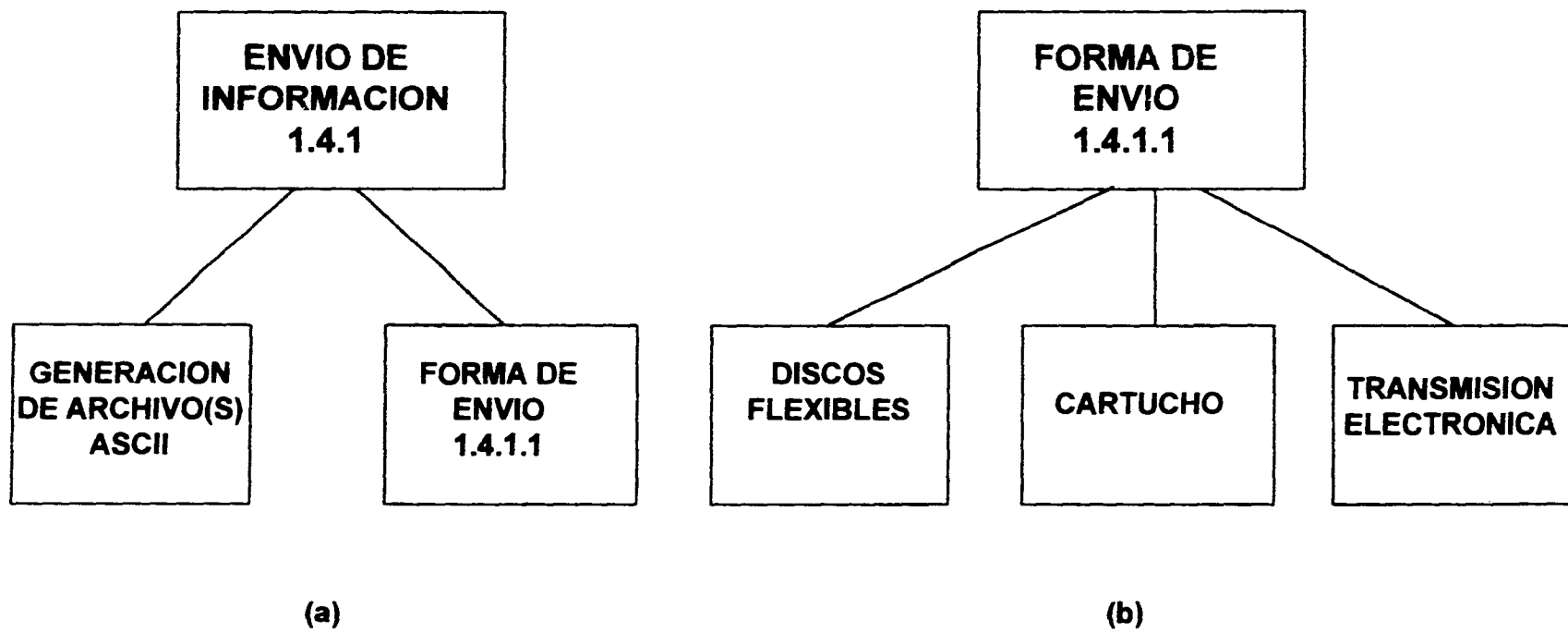
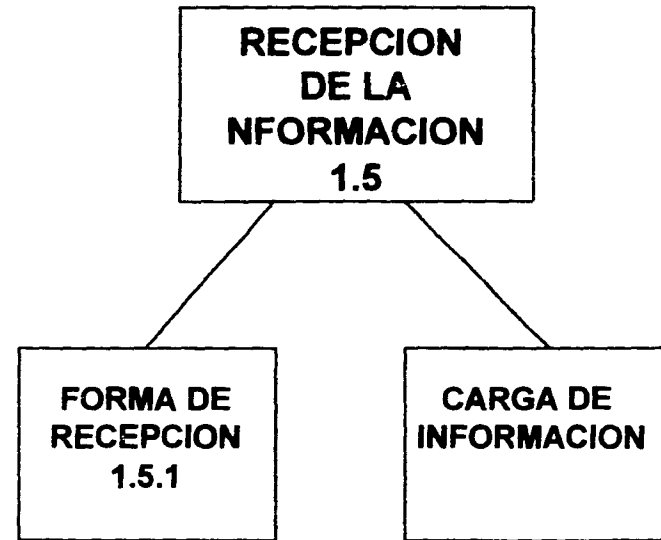
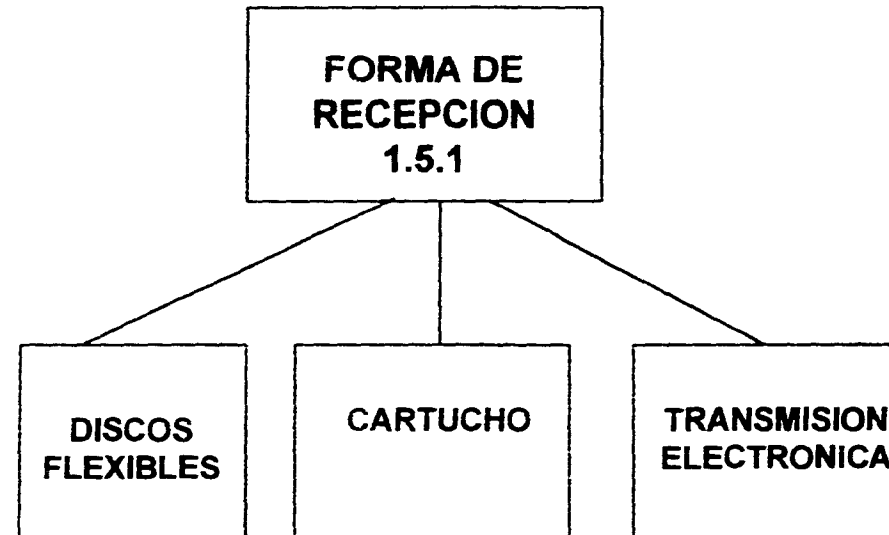


Figura 3.22



(a)



(b)

Figura 3.23

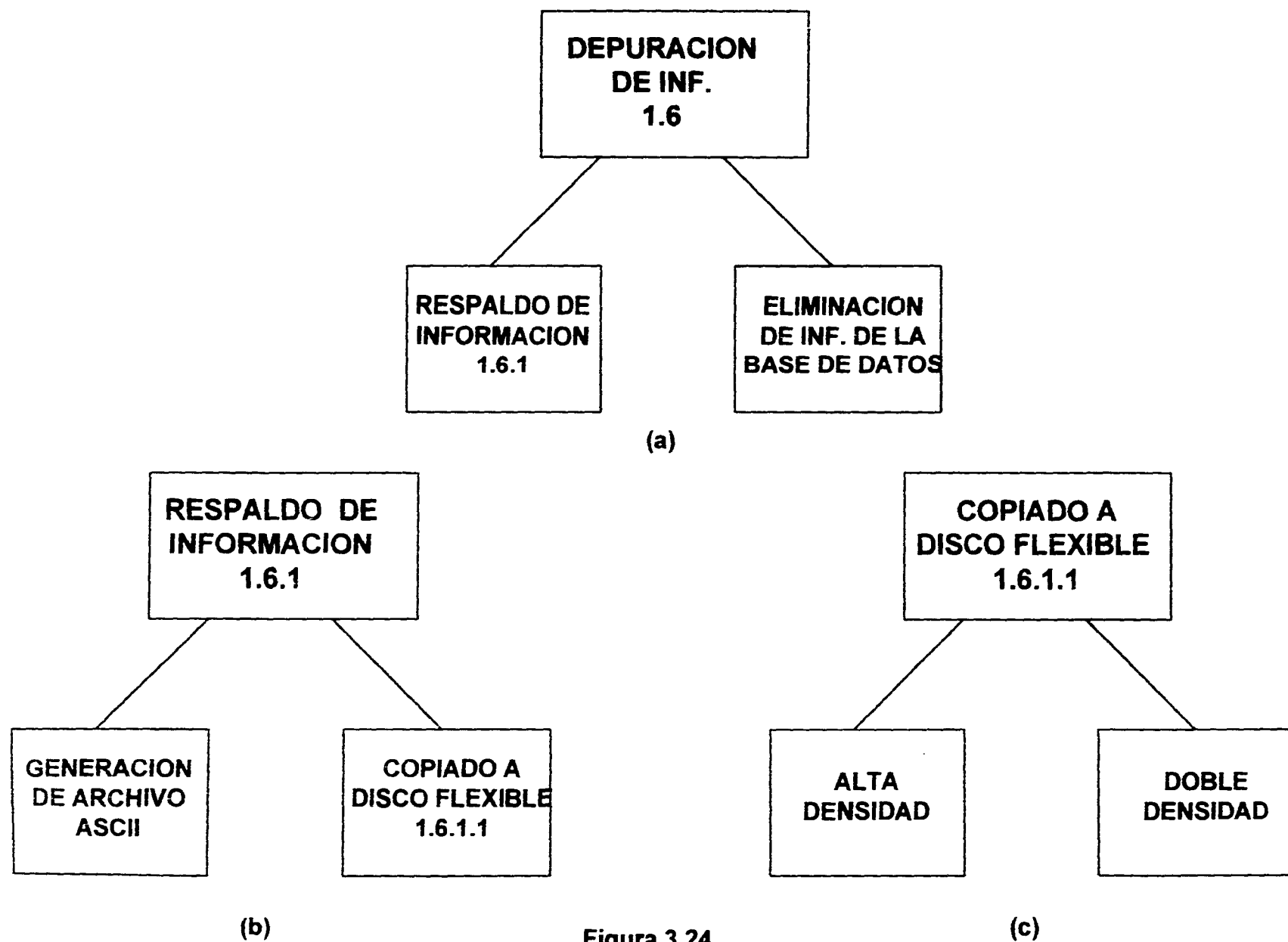


Figura 3.24

4 CONSTRUCCIÓN

4.1 GENERACIÓN DE LA BASE DE DATOS

El diseño realizado en la estructura de datos dio como resultado la construcción de las tablas del rmación, evitando la redundancia y/o duplicidad al crear índices únicos sobre los campos llave enSubsistema de Info cada una de las tablas.

A continuación se presenta la estructura de las tablas:

```
CREATE TABLE inf_estado
```

```
(  
  local      SMALLINT,  
  anio       SMALLINT,  
  mes        SMALLINT,  
  semana     SMALLINT,  
  fis_act    INTEGER,  
  fis_baj    INTEGER,  
  fis_suc    INTEGER,  
  fis_tot    INTEGER,  
  mor_act    INTEGER,  
  mor_baj    INTEGER,  
  mor_liq    INTEGER,  
  mor_tot    INTEGER,  
  uni_act    INTEGER,  
  uni_baj    INTEGER,  
  uni_liq    INTEGER,  
  uni_tot    INTEGER  
)
```

```
CREATE UNIQUE INDEX ix_estado on inf_estado (local,anio,mes,semana);
```

```
CREATE TABLE inf_opera
```

```
(  
  local      SMALLINT,  
  anio       SMALLINT,  
  mes        SMALLINT,  
  semana     SMALLINT,  
  movim      CHAR(02),  
  cpto       CHAR(01),  
  per_mor    INTEGER,  
  no_cont    INTEGER,  
  per_fis    INTEGER,  
  total      INTEGER  
)
```

```
CREATE UNIQUE INDEX ix_opera on inf_opera (local,anio,mes,semana);
```

CREATE TABLE inf_obliga

```
(
  local      SMALLINT,
  anio       SMALLINT,
  mes        SMALLINT,
  semana     SMALLINT,
  oblig      CHAR(3),
  a_pmor     INTEGER,
  a_pfis     INTEGER,
  b_pmor     INTEGER,
  b_pfis     INTEGER,
  m_acum     INTEGER,
  f_acum     INTEGER
)
```

CREATE UNIQUE INDEX ix_obliga on inf_obliga (local,anio,mes,semana,oblig);

CREATE TABLE inf_saldos

```
(
  local      SMALLINT,
  anio       SMALLINT,
  mes        SMALLINT,
  semana     SMALLINT,
  clave      CHAR(01),
  cpto       CHAR(02),
  numero     INTEGER,
  importe    MONEY(18,2)
)
```

CREATE UNIQUE INDEX ix_saldos on inf_saldos (local,anio,mes,semana,clave,cpto);

CREATE TABLE inf_cobra

```
(
  local      SMALLINT,
  anio       SMALLINT,
  mes        SMALLINT,
  clave      CHAR(02),
  subcve     CHAR(01),
  tipo       CHAR(01),
  nivel      CHAR(01),
  numero     INTEGER,
  importe    MONEY(14,2)
)
```

CREATE UNIQUE INDEX ix_cobra on inf_cobra (local,anio,mes,clave,subcve,tipo,nivel);

```
CREATE TABLE inf_vigilancia
```

```
(  
  local      SMALLINT,  
  emision    SMALLINT,  
  tip_cont   CHAR(02),  
  fecha      DATE,  
  universo   INTEGER,  
  cumplido   INTEGER,  
  omisos     INTEGER  
)
```

```
CREATE UNIQUE INDEX ix_vigila on inf_vigilancia (local,emision,tip_cont);
```

```
CREATE TABLE inf_impreso
```

```
(  
  local      SMALLINT,  
  anio       SMALLINT,  
  mes        SMALLINT,  
  semana     SMALLINT,  
  emision    SMALLINT,  
  tip_cont   CHAR(02),  
  impresos   INTEGER  
)
```

```
CREATE UNIQUE INDEX ix_impreso on inf_impreso (local,anio,mes,semana,tip_cont);
```

```
CREATE TABLE inf_seguimiento
```

```
(  
  local      SMALLINT,  
  anio       SMALLINT,  
  mes        SMALLINT,  
  semana     SMALLINT,  
  emision    SMALLINT,  
  fecha      DATE,  
  emitido    INTEGER,  
  imp_nolo   INTEGER,  
  imp_nobl   INTEGER,  
  pen_acia   INTEGER,  
  pen_noti   INTEGER,  
  tot_proc   INTEGER,  
  can_exte   INTEGER,  
  mul_exte   MONEY(20,2),  
  can_incu   INTEGER,  
  mul_incu   MONEY(20,2)  
)
```

```
CREATE UNIQUE INDEX ix_segui on inf_seguimiento (local,anio,mes,semana,emision);
```

```
CREATE TABLE inf_banco
```

```
(  
  local      SMALLINT,  
  fecha      DATE,  
  medio      SMALLINT,  
  conpag     INTEGER,  
  sinpag     INTEGER,  
  importe    MONEY(12,0)  
)
```

```
CREATE UNIQUE INDEX ix_banco ON inf_banco (local,fecha,medio);
```

```
CREATE TABLE inf_formulario
```

```
(  
  local      SMALLINT,  
  fecha      DATE,  
  formu      CHAR(08),  
  tip_dec    CHAR(02),  
  conpag     INTEGER,  
  sinpag     INTEGER,  
  importe    MONEY(12,0)  
)
```

```
CREATE UNIQUE INDEX ix_formu ON inf_formulario (local,fecha,formu,tip_dec);
```

```
CREATE TABLE inf_ingreso
```

```
(  
  local      SMALLINT,  
  fecha      DATE,  
  rubro      SMALLINT,  
  importe    MONEY(12,0)  
)
```

```
CREATE UNIQUE INDEX ix_ingreso ON inf_ingreso (local,fecha,rubro);
```

```
CREATE TABLE inf_obligacion
```

```
(  
  local      SMALLINT,  
  fecha      DATE,  
  tip_obl    CHAR(02),  
  conpagf    INTEGER,  
  conpagn    INTEGER,  
  sinpagf    INTEGER,  
  sinpagn    INTEGER  
)
```

```
CREATE UNIQUE INDEX ix_obliga ON inf_obligacion (local,fecha,tip_obl);
```

```
CREATE TABLE inf_informa
(
  anio      SMALLINT,
  mes       SMALLINT,
  semana    SMALLINT,
  dia_ini   DATE,
  dia_fin   DATE
)
```

```
CREATE UNIQUE INDEX ix_informa ON inf_informa (anio,mes,semana);
```

El crecimiento mensual de cada una de las tablas es el siguiente:

Tabla	No. de registros por mes
inf_estado	5
inf_opera	400
inf_obliga	800
inf_saldos	1000
inf_cobra	6150
inf_vigilancia	100
inf_impreso	100
inf_seguimiento	50
inf_banco	27
inf_formulario	22
inf_ingreso	15
inf_obligacion	7
inf_informa (catálogo)	sin crecimiento

4.2. MENÚ DEL SUBSISTEMA DE INFORMACIÓN

Con base en la elaboración de los módulos del Subsistema realizados en el punto 3.2 se logró la creación del menú del Subsistema de Información que a continuación se presenta.

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :		FECHA :
TERMINAL :		EST. CAP. :
VER.	MENU PRINCIPAL	USUARIO :

INFORMACION

Generacion
Reportes
Consulta
Envio de Informacion
Recepcion de Informacion
Depuracion de Informacion
Salir del Subsistema

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	GENERACION	USUARIO :

Registro Federal de Contribuyentes
Control de Obligaciones
Cobranza
Control de Saldos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA
TERMINAL :	REPORTES	EST. CAP. :
VER.		USUARIO :

Registro Federal de Contribuyentes
Control de Obligaciones
Cobranza
Contabilidad
Control de Saldos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :	REPORTES	EST. CAP. :
VER.		USUARIO :

REGISTRO FEDERAL DE CONTRIBUYENTES

Solicitudes y Avisos
Estructura del Padron de Contribuyentes
Universo de Contribuyentes Activos
Resultados de la Operacion
Obligaciones mas Representativas
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

R. F. C.
ESTRUCTURA DEL PADRON DE CONTRIBUYENTES

Total
Personas Fisicas
Personas Morales Matrices
Personas Morales Sucursales
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

R. F. C.
RESULTADOS DE LA OPERACION

Movimientos del Contribuyente
Mecanismos de Correccion
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

R. F. C.

OBLIGACION MAS REPRESENTATIVA

Personas Morales
Personas Fisicas
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

CONTROL DE OBLIGACIONES

Vigilancia
Requerimientos de Obligaciones
Procedencia de Requerimientos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

CONTROL DE OBLIGACIONES
VIGILANCIA

Emission
Año de Emission
Todas las Emisiones
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

CONTROL DE OBLIGACIONES
REQUERIMIENTOS DE OBLIGACIONES

Emission
Año de Emission
Todas las Emisiones
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

CONTROL DE OBLIGACIONES
PROCEDENCIA DE REQUERIMIENTOS

Emision
Año de Emision
Todas las Emisiones
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

COBRANZA

Creditos Fiscales
Pago en Parcialidades
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

CONTROL DE OBLIGACIONES
PROCEDENCIA DE REQUERIMIENTOS

Emission
Año de Emission
Todas las Emisiones
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

COBRANZA

Creditos Fiscales
Pago en Parcialidades
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :	REPORTES	EST. CAP. :
VER.		USUARIO :

COBRANZA

CREDITOS FISCALES

Mov. de la Car. de Cred. Fiscales
Car. de Cred. por Cves. de Tramites
Mov. de la Car. por Cves. de Tramites
Cred. por Autoridad Responsable
Creditos Dados de Baja
Creditos Estaticos
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :	REPORTES	EST. CAP. :
VER.		USUARIO :

COBRANZA

PAGO EN PARCIALIDADES

Totales
Millonarios
No Millonarios
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

CONTABILIDAD

Declaraciones por Tipo de Obligacion
Declaraciones por Tipo de Formulario
Declaraciones por Tipo de Ingreso
Declaraciones por Institucion Bancaria
Todos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	REPORTES	USUARIO :

CONTROL DE SALDOS

Comportamiento de Control de Saldos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO
:		

Registro Federal de Contribuyentes
Control de Obligaciones
Cobranza
Contabilidad
Control de Saldos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

REGISTRO FEDERAL DE CONTRIBUYENTES

Solicitudes y Avisos
Estructura del Padron de Contribuyentes
Universo de Contribuyentes Activos
Resultados de la Operacion
Obligaciones mas Representativas
Salida

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

R. F. C.
ESTRUCTURA DEL PADRON DE CONTRIBUYENTES

Total
Personas Fisicas
Personas Morales Matrices
Personas Morales Sucursales
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

R. F. C.
RESULTADOS DE LA OPERACION

Movimientos del Contribuyente
Mecanismos de Correccion
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

R. F. C.

OBLIGACION MAS REPRESENTATIVA

Personas Morales
Personas Fisicas
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

CONTROL DE OBLIGACIONES

Vigilancia
Requerimientos de Obligaciones
Procedencia de Requerimientos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

CONTROL DE OBLIGACIONES
VIGILANCIA

Emission
Año de Emission
Todas las Emisiones
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

CONTROL DE OBLIGACIONES
REQUERIMIENTOS DE OBLIGACIONES

Emission
Año de Emission
Todas las Emisiones
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

CONTROL DE OBLIGACIONES
PROCEDENCIA DE REQUERIMIENTOS

Emission
Año de Emission
Todas las Emisiones
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

COBRANZA

Creditos Fiscales
Pago en Parcialidades
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

COBRANZA

CREDITOS FISCALES

Mov. de la Car. de Cred. Fiscales
Car. de Cred. por Cves. de Tramites
Mov. de la Car. por Cves. de Tramites
Cred. por Autoridad Responsable
Creditos Dados de Baja
Creditos Estaticos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

COBRANZA

PAGO EN PARCIALIDADES

Totales
Millonarios
No Millonarios
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

CONTABILIDAD

Declaraciones por Tipo de Obligacion
Declaraciones por Tipo de Formulario
Declaraciones por Tipo de Ingreso
Declaraciones por Institucion Bancaria
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	CONSULTA	USUARIO :

CONTROL DE SALDOS

Comportamiento de Control de Saldos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	ENVIO DE INFORMACION	USUARIO :

Todos
Registro Federal de Contribuyentes
Control de Obligaciones
Cobranza
Control de Saldos
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	ENVIO DE INFORMACION	USUARIO :

Generacion de Archivo(s) ASCII
Forma de Envio
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	ENVIO DE INFORMACION	USUARIO :

FORMA DE ENVIO

Discos Flexibles
Cartucho
Transmision Electronica
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	RECEPCION DE INFORMACION	USUARIO :

Forma de Recepcion
Carga de Informacion
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	RECEPCION DE INFORMACION	USUARIO :

FORMA DE RECEPCION

Discos Flexibles
Cartucho
Transmision Electronica
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	DEPURACION DE INFORMACION	USUARIO :

Respaldo de Informacion
Eliminacion de Informacion de la B.D.
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	DEPURACION DE INFORMACION	USUARIO :

RESPALDO DE INFORMACION

Generacion de Archivo ASCII
Copiado a Disco Flexible
Salida

SECRETARIA DE HACIENDA Y CREDITO PUBLICO
SUBSECRETARIA DE INGRESO
SISTEMA INTEGRAL DE RECAUDACION

A.L.R. :	INFORMACION	FECHA :
TERMINAL :		EST. CAP. :
VER.	DEPURACION DE INFORMACION	USUARIO :

COPIADO A DISCO FLEXIBLE

Alta Densidad
Doble Densidad
Salida

5. IMPLANTACIÓN

El Subsistema de Información se implantó en tres fases.

En la primer fase se liberó Cobranza y Registro Federal de Contribuyentes, realizándose antes de esto pruebas piloto en la A.L.R. del Centro del D.F., donde se detectó que el proceso de generación de información es de varias horas debido al gran volumen de información, tomándose en cuenta para mejorar este proceso.

Durante la segunda fase se liberó el Subsistema de Información de Control de Saldos y Contabilidad, la cual se transmitió electrónicamente a las A.R.R., que a su vez la enviaron a sus A.L.R. correspondientes.

La Información de Contabilidad se genera en las A.R.R. y en esta fase se liberó el módulo de Recepción de Información en las A.L.R.

No se realizó capacitación, por lo que el personal operativo ignoraba cuando realizar el proceso de generación de información de Control de Saldos.

Finalmente, para la tercer fase se implementó Control de Obligaciones y modificaciones requeridas a los subsistemas ya implementados.

Para la primer y tercer fase la implantación consistió de la siguiente manera:

Se instaló el Subsistema en un equipo de pruebas en las A.R.R. el día anterior a la capacitación y brindamos apoyo técnico durante ésta.

Durante la instalación se tuvieron los siguientes problemas :

- El medio ambiente del equipo era incorrecto.
- Se contaba con impresoras que no soportaban la impresión de más de 200 caracteres por línea, ocasionando reportes incompletos.
- Se detectó que algunos archivos del sistema operativo y del Informix tenían permisos incorrectos ocasionando que la aplicación abortara, mandando mensajes de error en donde se intuía que el disco donde estaba instalado el Informix estaba dañado.
- Configuración incorrecta de terminales e impresoras.

La capacitación la realizó el personal administrativo de la A.G.R. a las áreas involucradas de las A.L.R. y A.R.R.

El paquete de instalación del Subsistema se envió por mensajería a las A.L.R. incluyendo el manual de instalación.

Una vez instalado el Subsistema en las A.L.R. surgieron los siguientes problemas :

- El proceso de generación de información de Cobranza, se realizaba fuera de tiempo, causando estadísticas incorrectas.
- Dudas sobre el envío de información a la A.R.R.

El Subsistema de Información se desarrolló aproximadamente en once meses, consta de dieciocho programas ejecutables, cada uno conteniendo de uno a ocho programas fuentes, tres programas en shell y treinta y tres formas.

Los programas fuentes varían en líneas de código, pueden llegar a tener hasta mil líneas.

6. COMENTARIOS FINALES

El uso de la tecnología es esencial en cualquier actividad. Poco a poco se han ido automatizando tareas que eran tediosas y monótonas, como el realizar informes estadísticos manuales.

La problemática que existía en la S.H.C.P. de contar con informes estadísticos confiables y oportunos sobre el comportamiento de la recaudación, nos llevó a realizar un análisis para definir las características con que debía contar el Subsistema de Información.

En el diseño se obtuvo el modelo de datos y el modelo de procesos. Con el primero se logró obtener la estructura de la base de datos sin caer en ambigüedades, con el segundo controlamos el flujo de datos para obtener un sistema equilibrado.

La construcción del Subsistema se realizó con base a los modelos anteriores, manteniendo una programación estructurada y evitando redundancia y/o duplicidad de la información en la base de datos.

Con base a los productos que arroja el Subsistema de Información se han podido conocer los problemas más importantes de las áreas del S.I.R. para que el personal administrativo tome las medidas necesarias para corregirlas.

Este Subsistema es el primero en dar resultados estadísticos, por lo que está sujeto a crecer y poder dar información aún más detallada acerca de la recaudación.

Debido a la utilidad que representa, se desarrollará a nivel regional y central. A partir de éstos se obtendrán informes que se presentarán a niveles directivos para tener un mejor control sobre la recaudación obtenida.

Esta tesis nos deja una gran satisfacción al haber logrado los objetivos propuestos y haber obtenido resultados adecuados, los cuales son importantes dentro de la S.H.C.P. Obtuvimos nuevas experiencias conociendo un poco más sobre el proceso de la recaudación, así como sugerencias del personal que diariamente trabaja para ello.

Debido a la confidencialidad de la información en la S.H.C.P., sólo se presenta parte de la estructura de las tablas del Subsistema de Información, así como también se han efectuado cambios en los nombres. Sin embargo, la esencia del análisis y diseño no varía.

APÉNDICE A

INFORMIX-SQL

HISTORIA DE SQL.

El modelo de datos relacional se publicó en 1970 por E. F. Codd, un empleado de IBM. La historia de SQL va ligada con la historia del proyecto System R de IBM. El propósito de este proyecto consistió en desarrollar un sistema manejador de bases de datos relacional experimental.

Sequel fué seleccionado como lenguaje de base de datos para el sistema R, posteriormente se renombró como SQL.

El sistema R se realizó en tres fases :

En la primer fase *-fase cero-*, realizada en 1974 y 1975, solamente una parte de SQL se implementó. Por ejemplo el *join* no fué implementado, y la versión trabajó sólo para sistemas monousuario.

En 1976 se inició la *fase uno*. Los programas de la fase anterior se hicieron a un lado y se comenzó de nuevo. Esta fase comprendió el sistema total. Implicó, entre otras cosas, la inclusión en sistemas multiusuarios y el *join*.

En la *fase dos*, el sistema R fué evaluado durante 1978 y 1979 instalándose en varios lugares.

SQL consta de tres componentes:

- Lenguaje de definición de datos (**DDL**). Define la estructura de la base de datos.
- Lenguaje de manipulación de datos (**DML**). Recupera, actualiza, inserta y borra los datos.
- Lenguaje de control de datos (**DCL**). Controla el acceso a la base de datos.

En 1983 la International Standards Organization (ISO) comenzó a desarrollar un standard para lenguajes de bases de datos relacionales basados en SQL.

Versiones de SQL han sido implementadas en todas las plataformas de hardware desde PCs a mainframes. Corren bajo cualquier sistema operativo: MS/DOS, OS/2, Unix, VAX/VMS, VM/CMS y MVS.

Algunos productos líderes son : Professional Oracle de Oracle Corp, XDB-SQL de XDB Systems, Ingress de Relational Technologies e Informix de SQL/4GL.

INFORMIX-SQL

Es un sistema manejador de base de datos, consiste de módulos o programas útiles diseñados para ejecutar tareas de manejo de datos. Un buen sistema manejador de base de datos puede reducir la cantidad de tiempo requerido para organizar, almacenar y extraer información. Puede sumarizar, agrupar y formatear la información de diversas formas que en otros casos sería tedioso y propenso a errores.

SQL es un lenguaje de búsqueda interactiva que puede ser usado en base de datos. Con SQL se puede:

- Crear y borrar una base de datos
- Crear y borrar tablas e índices
- Diseñar pantallas
- Introducir y borrar datos
- Desplegar información acerca de una o más tablas
- Renombrar tablas y columnas
- Checar y reparar tablas
- Otorgar, cancelar permisos a una base de datos y dar privilegios a tablas
- Producir reportes formateados
- Realizar búsquedas desde pantallas
- Cargar datos desde archivos

Variables de Medio Ambiente

Las variables de medio ambiente afectan el ambiente de la computadora en la cual Informix-SQL opera. Algunas de éstas que usa Informix-SQL son las siguientes:

DBDATE. Especifica el formato que se desea usar para el valor de la fecha. El valor por default es MDY4/ donde M representa el mes, D representa el día, Y4 representa un año de cuatro dígitos, y el separador es una diagonal (mm/dd/yyyy).

DBDELIMITER. Especifica el campo delimitador usado en carga y descarga de archivos. La barra vertical (|) es el default.

DBEDIT. Indica el nombre del editor que se desea utilizar para modificar archivos, reportes y comandos desde Informix-SQL. El default es vi.

DBPRINT. Especifica el comando de impresión de la computadora. El default es lp.

DBTEMP. Es el directorio en el cual Informix-SQL almacena archivos temporales. El default es /tmp.

INFORMIXDIR. Especifica la localización de los programas de Informix-SQL. Estos archivos son usualmente almacenados en directorios de /usr/informix.

DECLARACIONES SQL

Seis diferentes tipos de declaraciones SQL son utilizadas con Informix-4GL:

- Definición de datos
- Manipulación de datos
- Manejo de cursor
- Manejo dinámico
- Acceso de datos
- Integridad de datos

SQL es un lenguaje de búsqueda interactiva que puede ser usado en base de datos. Con SQL se puede:

- Crear y borrar una base de datos
- Crear y borrar tablas e índices
- Diseñar pantallas
- Introducir y borrar datos
- Desplegar información acerca de una o más tablas
- Renombrar tablas y columnas
- Checar y reparar tablas
- Otorgar, cancelar permisos a una base de datos y dar privilegios a tablas
- Producir reportes formateados
- Realizar búsquedas desde pantallas
- Cargar datos desde archivos

Variables de Medio Ambiente

Las variables de medio ambiente afectan el ambiente de la computadora en la cual Informix-SQL opera. Algunas de éstas que usa Informix-SQL son las siguientes:

DBDATE. Especifica el formato que se desea usar para el valor de la fecha. El valor por default es MDY4/ donde M representa el mes, D representa el día, Y4 representa un año de cuatro dígitos, y el separador es una diagonal (mm/dd/yyyy).

DBDELIMITER. Especifica el campo delimitador usado en carga y descarga de archivos. La barra vertical (|) es el default.

DBEDIT. Indica el nombre del editor que se desea utilizar para modificar archivos, reportes y comandos desde Informix-SQL. El default es vi.

DBPRINT. Especifica el comando de impresión de la computadora. El default es lp.

DBTEMP. Es el directorio en el cual Informix-SQL almacena archivos temporales. El default es /tmp.

INFORMIXDIR. Especifica la localización de los programas de Informix-SQL. Estos archivos son usualmente almacenados en directorios de /usr/informix.

DECLARACIONES SQL

Seis diferentes tipos de declaraciones SQL son utilizadas con Informix-4GL:

- Definición de datos
- Manipulación de datos
- Manejo de cursor
- Manejo dinámico
- Acceso de datos
- Integridad de datos

FALTA PAGINA

No.....a la 96

BEFORE	Es una palabra reservada opcional para indicar donde se coloca <i>nombre nueva columna</i> en la lista de columnas. El default es al final de la lista.
<i>nombre columna original</i>	Es el nombre de una columna existente.
DROP	Se usa para borrar una columna.
MODIFY	Cambia el tipo de dato de una columna existente.

CLOSE DATABASE

Cierra una base de datos.

Sintaxis:

CLOSE DATABASE *nombre de la base de datos*

Explicación:

CLOSE DATABASE	Son palabras reservadas para cerrar una base de datos..
<i>nombre de la base de datos</i>	Es el nombre de una base de datos.

CREATE DATABASE

Crea una base de datos.

Sintaxis:

CREATE DATABASE *nombre de la base de datos*

Explicación:

CREATE DATABASE	Son palabras reservadas para crear una base de datos.
<i>nombre de la base de datos</i>	Es el nombre que se asigna a la base de datos.

CREATE INDEX

Crea un índice para una o más columnas en una tabla.

Sintaxis:

CREATE [UNIQUE] [CLUSTER] INDEX *nombre del índice*
ON *nombre de la tabla (nombre de la columna [ASC | DESC] [, ...])*

Explicación:

CREATE INDEX	Son palabras reservadas para crear un índice a una tabla.
UNIQUE	Previene entradas duplicadas en una o más columnas.
CLUSTER	Es una palabra opcional que ordena físicamente la tabla de acuerdo al índice.
<i>nombre del índice</i>	Es el identificador SQL al que se le asigna el índice.
ON	Es una palabra reservada.
<i>nombre de la tabla</i>	Es el nombre de la tabla que contiene la(s) columna(s) que se desea indexar.
<i>nombre de la columna</i>	Es el nombre de una columna a ser indexada. Para crear un índice con varias columnas, estas deben ir separadas por comas. Todas las columnas deben ser de la misma tabla.

ASC
DESC

Especifica un índice que mantiene un orden ascendente.
Especifica un índice que mantiene un orden descendente.

CREATE TABLE

Crea una nueva tabla en la base de datos actual.

Sintaxis:

```
CREATE [ TEMP ] TABLE nombre de la tabla  
( nombre de columna tipo de dato  
[ NOT NULL ] [...])
```

Explicación:

CREATE TABLE	Son palabras reservadas para crear una tabla.
TEMP	Es una palabra opcional para crear una tabla temporal.
<i>nombre de la tabla</i>	Es el identificador SQL que se asigna a la tabla.
<i>nombre de la columna</i>	Es el identificador SQL que se asigna a cada columna.
<i>tipo de dato</i>	Especifica el tipo de dato para cada columna.
NOT NULL	Son palabras reservadas opcionales para prevenir la entrada de valores nulos.

Los tipos de datos son los siguientes :

CHAR (n)	Es una cadena de caracteres de longitud n. Donde: $1 \leq n \leq 32,511$.
SMALLINT	Es un número de rango -32,767 a 32,767.
INTEGER	Es un número de rango -2,147,483,647 a 2,147,483,647.
FLOAT[(n)]	Es un número de punto flotante que corresponde a el tipo de dato "double" en lenguaje "C". <i>n</i> especifica la precisión del tipo de dato FLOAT, debe estar entre 1 y 14.
DECIMAL[(m[,n])]	Es un número decimal de punto flotante con un total de <i>m</i> (≤ 32) dígitos significativos y <i>n</i> ($\leq m$) dígitos a la derecha del punto decimal.
MONEY[(m[,n])]	Es un número tipo DECIMAL, desplegado con el símbolo \$.
SERIAL[(n)]	Es un entero secuencial asignado automáticamente por 4GL. Se puede asignar un valor inicial <i>n</i> . El default es 1.
DATE	Es una fecha introducida como una cadena de caracteres con un formato determinado.
DATETIME	Es un momento en el tiempo que puede incluir el año, mes, día, hora, minuto, segundo, y una fracción de segundo.

DATABASE

Selecciona una base de datos.

Sintaxis:

```
DATABASE nombre de la base de datos
```

Explicación:

DATABASE	Es una palabra reservada.
<i>nombre de la base de datos</i>	Es el nombre de una base de datos.

DROP DATABASE
Borra una base de datos.

Sintaxis:
DROP DATABASE *nombre de la base de datos*

Explicación:

DROP DATABASE	Son palabras reservadas para eliminar una base de datos.
<i>nombre de la base de datos</i>	Es el nombre de una base de datos.

DROP INDEX
Borra un índice.

Sintaxis:
DROP INDEX *nombre del índice*

Explicación:

DROP INDEX	Son palabras reservadas para eliminar un índice.
<i>nombre del índice</i>	Es el nombre del índice que se desea borrar.

DROP TABLE
Borra una tabla, con sus índices asociados y datos.

Sintaxis:
DROP TABLE *nombre de la tabla*

Explicación:

DROP TABLE	Son palabras reservadas para eliminar una tabla.
<i>nombre de la tabla</i>	Es el nombre de la tabla que se desea borrar.

RENAME COLUMN
Cambia el nombre de una columna.

Sintaxis:
RENAME COLUMN *tabla.nombre columna original TO nombre columna nueva*

Explicación:

RENAME COLUMN	Son palabras reservadas para renombrar una columna.
<i>tabla</i>	Es el nombre de la tabla que contiene <i>nombre columna original</i> .
<i>nombre columna original</i>	Es el nombre de la columna a ser renombrada.
TO	Es una palabra reservada.
<i>nombre columna nueva</i>	Es el nombre nuevo a ser asignado a la columna.
	No se puede duplicar el nombre de una columna en la tabla.

RENAME TABLE

Cambia el nombre de una tabla.

Sintaxis

RENAME TABLE *nombre original* TO *nombre nuevo*

Explicación:

RENAME TABLE	Son palabras reservadas para renombrar una tabla.
<i>nombre original</i>	Es el nombre actual de la tabla a ser renombrada.
TO	Es una palabra reservada.
<i>nombre nuevo</i>	Es el nuevo nombre que se desea asignar a la tabla.

Manipulación de datos.

Este tipo de declaraciones son las más utilizadas y son :

DELETE

Borra uno o más renglones de una tabla de acuerdo a la condición

Sintaxis:

DELETE FROM *nombre de la tabla*
{ WHERE { *condición* | CURRENT OF *nombre del cursor* } }

Explicación:

DELETE FROM	Son palabras reservadas para borrar renglones de una tabla.
<i>nombre de la tabla</i>	Es el nombre de la tabla de la cual se desea borrar renglones
WHERE	Es una palabra reservada.
<i>condición</i>	Es una condición de la cláusula WHERE. Informix-4gl borra todos los renglones que satisfacen ésta condición.
CURRENT OF	Son palabras reservadas que indican la posición actual del <i>nombre del cursor</i> .
<i>nombre del cursor</i>	Es el identificador SQL declarado previamente.

INSERT

Inserta uno o más renglones nuevos en una tabla existente.

Sintaxis:

INSERT INTO *nombre de la tabla* { (*lista de columnas*) }
{ VALUES (*lista de valores*) | declaración SELECT }

Explicación:

INSERT INTO	Son palabras reservadas para insertar.
<i>nombre de la tabla</i>	Es el nombre de la tabla a la cual se agregan los renglones.
<i>lista de columnas</i>	Es una lista de los nombres de las columnas en la cual se insertan datos.
VALUES	Es una palabra reservada.
<i>lista de valores</i>	Son los valores a insertar en las columnas especificadas. Deben ir separadas por comas.
<i>declaración SELECT</i>	Es una declaración válida SELECT

LOAD

Inserta renglones en una tabla existente desde un archivo ASCII.

Sintaxis:

```
LOAD FROM "archivo ascii" [ DELIMITER "character" ]  
{ INSERT INTO nombre de la tabla [ ( nombre de la columna {,...}) ] }
```

Explicación:

LOAD FROM	Son palabras reservadas para cargar el archivo a la tabla.
<i>"archivo ascii"</i>	Es una variable o constante <i>character</i> que contiene la ruta del archivo ASCII. Debe ir entre comillas.
DELIMITER	Es una palabra opcional para indicar que el siguiente <i>character</i> aparecerá al final de cada campo de dato en el archivo.
<i>"character"</i>	Es un carácter sencillo que sirve como delimitador entre campos. Debe aparecer entre comillas.
INSERT INTO	Son palabras reservadas para indicar donde van a ser almacenados los datos.
<i>nombre de la tabla</i>	Es el identificador de una tabla existente.
<i>nombre de la columna</i>	Es el nombre de una columna (en paréntesis) de <i>nombre de la tabla</i> .

SELECT

Especifica los datos que se desean extraer de una o más tablas en una base de datos.

Sintaxis:

```
SELECT cláusula [ INTO cláusula ] FROM cláusula  
[ WHERE cláusula ]  
[ GROUP BY cláusula ]  
[ HAVING cláusula ]  
[ ORDER BY cláusula ]  
[ INTO TEMP cláusula ]
```

Las cláusulas tienen la siguiente sintaxis :

```
SELECT [ ALL | DISTINCT | UNIQUE ] lista de selección  
INTO lista de variables  
FROM { nombre de la tabla [ alias de la tabla ] |  
OUTER nombre de la tabla [ alias de la tabla ] }  
WHERE condición
```

GROUP BY *lista de columnas*
HAVING *condición*
ORDER BY *nombre de columna* [ASC | DESC] [, ...]
INTO TEMP *nombre de la tabla*

Explicación:

SELECT	Es una palabra reservada.
ALL	Selecciona todos los renglones que satisfacen la cláusula WHERE, sin eliminar duplicados.
DISTINCT	Elimina renglones duplicados del resultado de la búsqueda.
UNIQUE	Es sinónimo de DISTINCT.
<i>lista de selección</i>	Es una lista de nombres de columnas y/o expresiones separadas por comas.
INTO	Es una palabra reservada, que indica que <i>lista de variables</i> almacenará los datos extraídos..
<i>lista de variables</i>	Es una lista de las variables que corresponden con el tipo de datos de las columnas o expresiones en la <i>lista de selección</i> .
FROM	Es una palabra reservada.
OUTER	Es una palabra opcional para indicar que se extraerán los registros que cumplan con la condición del WHERE y los registros que no cumplan con la condición de la tabla dominante. Esta es <i>nombre de la tabla</i> después del OUTER.
<i>nombre de la tabla</i>	Es el nombre o sinónimo de una tabla o vista en la cual se buscan los datos.
<i>alias de la tabla</i>	Es un alias opcional para el <i>nombre de la tabla</i> .
WHERE	Es una palabra reservada.
<i>condición</i>	Es una colección de una o más condiciones de búsqueda, opcionalmente conectados por los operadores lógicos AND, OR o NOT.
GROUP BY	Son palabras reservadas para agrupar.
<i>lista de columnas</i>	Es una lista de una o más columnas, separadas por comas, que determinan el grupo.
HAVING	Es una palabra reservada para aplicar una o más condiciones calificadas a grupos. Se utiliza para realizar condiciones sobre los valores de las columnas del GROUP BY.
ORDER BY	Son palabras reservadas.
<i>nombre de columna</i>	Es el nombre de una(s) columna(s) con las cuales se desea ordenar el resultado de la búsqueda.
ASC	Especifica que el resultado debe ser en orden ascendente.
DESC	Especifica que el resultado de ser en orden descendente.
INTO TEMP	Son palabras reservadas.
<i>nombre de la tabla</i>	Es el identificador SQL que se asigna a la tabla temporal.

UNLOAD

Escribe los datos de una tabla a un archivo ASCII.

Sintaxis:

UNLOAD TO "*archivo ascii*" [DELIMITER "*caracter*"] *declaración SELECT*.

Explicación:

UNLOAD TO <i>archivo ascii</i>	Son palabras reservadas. Es una variable o constante caracter que contiene la ruta del archivo en el cual se almacenarán los datos.
DELIMITER	Es una palabra opcional para indicar que el siguiente <i>caracter</i> aparecerá al final de cada campo de dato en el archivo.
<i>caracter</i>	Es un caracter sencillo que sirve como delimitador entre campos. Debe aparecer entre comillas.
<i>declaración SELECT</i>	Extrae los datos de una tabla a ser escritos en un archivo.

UPDATE

Modifica los valores en una o más columnas de uno o más renglones en una tabla.

Sintaxis:

```
UPDATE nombre de la tabla SET { nombre de la columna = expr. [, ... ] |  
{ ( lista de columnas ) | [ nombre de la tabla.]* } = { ( lista de expr. )  
| nombre del registro.* } }  
[ WHERE { condición | CURRENT OF nombre del cursor } ]
```

Explicación:

UPDATE	Es una palabra reservada.
<i>nombre de la tabla</i>	Es el nombre de la tabla que contiene los renglones a actualizar
SET	Es una palabra reservada.
<i>nombre de la columna</i>	Es el nombre de una columna a actualizar. Con asterisco se especifican todas las columnas de la tabla.
<i>expr.</i>	Es cualquier combinación de nombres de columnas, constantes, variables, operadores aritméticos, o una subbúsqueda que regresa un renglon de valor.
<i>lista de columnas</i>	Es una lista de los nombres de las columnas a ser actualizadas.
<i>lista de expr.</i>	Es una lista de expresiones que representan valores correspondientes a las columnas en <i>lista de columnas</i> o las columnas representadas por el asterisco.
<i>nombre del registro</i>	Es el nombre de una variable de tipo RECORD. El asterisco especifica todas las variables del registro.
WHERE	Es una palabra reservada.
<i>condición</i>	Compara los valores en una columna con los valores en otra columna, variables o constantes.
CURRENT OF	Son palabras reservadas que indican la posición actual del <i>nombre del cursor</i> .
<i>nombre del cursor</i>	Es un identificador SQL previamente declarado.

Manejo de cursor.

Informix-4GL proporciona dos tipos funcionales de cursor:

- Un cursor SELECT, el cual debe ser usado para manejar una declaración SELECT que retorna más de un renglón.

- Un cursor INSERT, el cual puede ser usado para insertar renglones en la base de datos.

Cursor SELECT.

Cuando más de un renglón es regresado de una declaración SELECT, es necesario tener un apuntador para distinguir un renglón de otro. Este apuntador es llamado *cursor*.

El conjunto de renglones regresados por una declaración SELECT es llamado conjunto activo.

Un cursor puede estar en dos estados: abierto o cerrado. Cuando un cursor SELECT está en estado abierto, está asociado con el conjunto activo y puede apuntar al renglón actual, entre dos renglones, antes del primer renglón o después del último renglón. Cuando está en estado cerrado, el cursor no está asociado con un conjunto activo, aunque permanezca asociado con la declaración SELECT.

Se debe utilizar la declaración DECLARE para nombrar un cursor y asociarlo con una declaración SELECT. En el DECLARE se especifica el tipo de cursor que se desea usar:

- Un cursor regular (o no SCROLL) permite extraer renglones del conjunto activo en orden consecutivo. Con este tipo de declaración se puede borrar o actualizar el renglón actual.
- Un cursor SCROLL permite extraer renglones del conjunto activo en orden aleatorio.
- Un cursor regular o SCROLL puede ser especificado con un WITH HOLD para no ser cerrado al final de una transacción.

Una vez que un cursor ha sido declarado para una declaración SELECT, puede ser usado en declaraciones FOREACH, OPEN, FETCH y CLOSE para extraer y procesar los renglones especificados.

CLOSE

Coloca el apuntador en estado cerrado y libera el conjunto activo.

Sintaxis:

CLOSE *nombre del cursor*

Explicación:

CLOSE	Es una palabra reservada.
<i>nombre del cursor</i>	Es el nombre de un cursor declarado por SELECT o INSERT.

FETCH

Mueve el cursor a un nuevo renglón del conjunto activo y extrae su valor.

Sintaxis:

FETCH [NEXT | { PREVIOUS | PRIOR } | FIRST | LAST | CURRENT |
 RELATIVE *m* | ABSOLUTE *n*] *nombre del cursor*
 [INTO *lista de variables*]

Explicación:

FETCH	Es una palabra reservada.
NEXT	Es una palabra reservada indicando el siguiente renglón en la lista activa. Es el default.
PREVIOUS	Es una palabra reservada indicando el renglón anterior en la lista activa.
PRIOR	Es una palabra reservada sinónimo de PREVIOUS.
FIRST	Es una palabra reservada indicando el primer renglón en la lista activa.
LAST	Es una palabra reservada indicando el último renglón en la lista activa.
CURRENT	Indica el renglón actual de la lista activa.
RELATIVE <i>m</i>	Indica el <i>m-ésimo</i> renglón relativo a la posición actual del cursor en la lista activa. <i>m</i> puede ser un entero o una variable de programa (positivo o negativo).
ABSOLUTE <i>n</i>	Indica el <i>n-ésimo</i> renglón en la lista activa. <i>n</i> puede ser un entero o una variable de programa (positivo o negativo).
<i>nombre del cursor</i>	Es un identificador especificado previamente en una declaración DECLARE. Debe estar abierto el cursor.
INTO	Es una palabra reservada opcional.
<i>lista de variables</i>	Es una lista de variables de programa 4GL que contienen los valores de las columnas del renglón apuntado por <i>nombre del cursor</i> .

FOREACH

Ejecuta una serie de declaraciones para cada renglón extraído de una búsqueda.

Sintaxis:

```
FOREACH nombre del cursor [ INTO lista de variables ]  
  declaración  
  ...  
  [ CONTINUE FOREACH ]  
  ...  
  [ EXIT FOREACH ]  
  ...  
END FOREACH
```

Explicación:

FOREACH	Es una palabra reservada.
<i>nombre del cursor</i>	Es el nombre de un apuntador previamente declarado.
INTO	Es una palabra opcional.
<i>lista de variables</i>	Es una lista de una o más variables de programa separadas por comas.
CONTINUE FOREACH	Es una declaración opcional.
EXIT FOREACH	Es una declaración opcional.
END FOREACH	Es una palabra reservada.

OPEN

Pone el cursor en un estado abierto con respecto a la declaración SELECT.

Sintaxis:

OPEN *nombre del cursor*

Explicación:

OPEN

Es una palabra reservada.

nombre del cursor

Es el nombre del apuntador previamente declarado.

Cursor INSERT.

Se puede asociar un cursor con una declaración INSERT. Permite insertar los datos eficientemente a una base de datos almacenando en memoria (buffer) y escribiendo a disco sólo cuando el buffer esta lleno. Las siguientes declaraciones permiten declarar y manipular un cursor INSERT: DECLARE (descrito después), OPEN y CLOSE (ya descritos), PUT y FLUSH.

FLUSH

Informix-4GL forza a insertar los renglones en el buffer en la base de datos sin cerrar el cursor.

Sintaxis:

FLUSH *nombre del cursor*

Explicación:

FLUSH

Es una palabra reservada.

nombre del cursor

Es el nombre de un cursor declarado por un INSERT.

PUT

Almacena un renglón en el buffer para posteriormente insertarlo en la tabla.

Sintaxis:

PUT *nombre del cursor*

Explicación:

PUT

Es una palabra reservada.

nombre del cursor

Es el nombre de un cursor declarado por un INSERT.

Manejo dinámico.

Si el usuario del programa especifica algún parámetro referente a una declaración entonces se utiliza este tipo de manejo. Asimismo, se utilizan cuando se intenta trabajar con diferentes bases de datos cuya estructura puede variar. Hay cuatro declaraciones de manejo dinámico:

DECLARE

Asigna un nombre de cursor a una declaración SELECT o INSERT. Un cursor es requerido para una declaración SELECT que selecciona más de un renglón.

Sintaxis:

```
DECLARE nombre del cursor [ SCROLL ] CURSOR [ WITH HOLD ] FOR  
{ declaración SELECT [ FOR UPDATE [ OF lista de columnas ] ] |  
declaración INSERT | identificador de declaración }
```

Explicación:

DECLARE	Es una palabra reservada.
<i>nombre del cursor</i>	Es un identificador Informix-4GL.
SCROLL	Es una palabra opcional que puede ser usada solo con una declaración SELECT o un <i>identificador de declaración</i> de una declaración SELECT que se preparó. Se utiliza para manipular el apuntador.
CURSOR FOR	Son palabras reservadas para indicar que el <i>nombre del cursor</i> sea un apuntador.
WITH HOLD	Son palabras opcionales para impedir que el cursor sea cerrado cuando la transacción finaliza.
<i>declaración SELECT</i>	Es una declaración SELECT.
FOR UPDATE	Son palabras reservadas que son requeridas si el cursor será usado para modificar renglones.
OF	Es una palabra opcional.
<i>lista de columnas</i>	Es una lista de nombres de columnas de tablas en la cláusula FROM de <i>declaración SELECT</i> .
<i>declaración INSERT</i>	Es una declaración INSERT.
<i>identificador de declaración</i>	Es el identificador de una declaración INSERT o SELECT que previamente se preparó.

EXECUTE

Ejecuta una declaración especificada previamente por un PREPARE.

Sintaxis:

```
EXECUTE identificador de declaración [ USING lista de entrada ]
```

Explicación:

EXECUTE	Es una palabra reservada.
<i>identificador de declaración</i>	Es una declaración SQL que se nombró previamente en un PREPARE.
USING	Es una palabra opcional para indicar los valores de <i>lista de entrada</i> .
<i>lista de entrada</i>	Es una lista de variables de programa a ser sustituidas por valores en la declaración indicada por <i>identificador de declaración</i> .

FREE

El manejador de la base de datos libera los recursos que son requeridos por una declaración PREPARE.

Sintaxis:

FREE { *identificador de declaración* | *nombre de cursor* }

Explicación:

FREE	Es una palabra reservada.
<i>identificador de declaración</i>	Es el nombre de una declaración que ha sido preparada.
<i>nombre de cursor</i>	Es el nombre de un cursor cuya declaración DECLARE incluye las palabras reservadas SELECT o INSERT.

PREPARE

Se utiliza para preprocesar una declaración SQL para ejecutarla posteriormente.

Sintaxis:

PREPARE *identificador de declaración* FROM *cadena esperada*

Explicación:

PREPARE	Es una palabra reservada.
<i>identificador de declaración</i>	Es un identificador SQL para una declaración.
FROM	Es una palabra reservada.
<i>cadena esperada</i>	Es una cadena constante encerrada en comillas o una variable de programa tipo CHAR, debe contener una declaración SQL.

Acceso de Datos.

Un usuario tiene acceso a la base de datos y a columnas específicas de una tabla únicamente cuando el Administrador de la Base de Datos o el propietario de la tabla le da privilegios.

SQL proporciona las siguientes declaraciones que afecta el acceso de datos :

GRANT

Concede privilegios de acceso a la base de datos o a tablas a usuarios específicos.

Sintaxis:

GRANT *privilegio a tabla* ON *nombre de la tabla* TO { PUBLIC | *lista de usuarios* }
[WITH GRANT OPTION] [AS *otorgante*]

GRANT *privilegio a b.d.* TO { PUBLIC | *lista de usuarios* }

Explicación:

GRANT	Es una palabra reservada.
<i>privilegio a tabla</i>	Es uno o más de los siguientes tipos de acceso a nivel tabla:
ALTER	Agrega o borra columnas o modifica tipo de dato de columnas.
DELETE	Borra renglones.

	INDEX	Crea índices.
	INSERT	Inserta renglones.
	SELECT [(cols)]	Extrae datos de columnas específicas.
	UPDATE [(cols)]	Cambia valores en columnas específicas.
	ALL [PRIVILEGES]	Todos los tipos de acceso. La palabra reservada PRIVILEGES es opcional.
ON		Es una palabra reservada.
<i>nombre de la tabla</i>		Es el nombre de la tabla a la cual se le da privilegios de acceso.
TO		Es una palabra reservada.
PUBLIC		Es una palabra reservada que da privilegios de acceso para todos los usuarios.
<i>lista de usuarios</i>		Es una lista de los nombres de usuarios a los que se les da privilegios de acceso. Deben ir separados por comas.
WITH GRANT OPTION AS		Son palabras reservadas opcionales para dar privilegios a nivel tabla a otros usuarios
<i>otorgante</i>		Es el nombre del usuario que da privilegios.
<i>privilegio a b.d.</i>		Es uno de los siguientes tipos de acceso a nivel base de datos:
	CONNECT	Permite el acceso a las tablas de la base de datos sin permisos para crear tablas permanentes e índices.
	RESOURCE	Permite el acceso a las tablas de la base de datos con permisos para crear tablas permanentes e índices.
	DBA	Da todos los privilegios de administrador de base de datos.

LOCK TABLE

Permite el acceso a la tabla a el usuario actual únicamente o solamente la lectura de la tabla a otros usuarios.

Sintaxis:

LOCK TABLE *nombre de la tabla* IN { SHARE | EXCLUSIVE } MODE

Explicación:

LOCK TABLE	Son palabras reservadas.
<i>nombre de la tabla</i>	Es el nombre de la tabla que se desea bloquear.
IN	Es una palabra reservada.
SHARE	Da acceso de sólo lectura a otros usuarios a la tabla.
EXCLUSIVE	Impide que otros usuarios tengan algún acceso a la tabla.
MODE	Es una palabra reservada que indica el modo en que será bloqueada la tabla.

REVOKE

Remueve privilegios de acceso a cualquier usuario para una tabla o base de datos.

Sintaxis:

```
REVOKE { privilegio a tabla ON nombre de la tabla | privilegio a b.d. }  
FROM { PUBLIC | lista de usuarios }
```

Explicación:

REVOKE	Es una palabra reservada.
<i>privilegio a tabla</i>	Es uno o más de los siguientes privilegios de acceso a nivel tabla: ALTER Agrega o borra columnas o modifica tipo de datos de columnas. DELETE Borra renglones. INDEX Crea índices. INSERT Inserta renglones. SELECT Extrae datos de columnas específicas. UPDATE Cambia valores en columnas específicas. ALL [PRIVILEGES] Todos los tipos de acceso mencionados. La palabra reservada PRIVILEGES es opcional.
ON	Es una palabra reservada.
<i>nombre de la tabla</i>	Es el nombre de la tabla para la cual se eliminan los privilegios de acceso.
<i>privilegio a b.d.</i>	Es uno de los siguientes tipos de acceso nivel base de datos: CONNECT Permite el acceso a las tablas de la base de datos sin permisos para crear tablas permanentes e índices. RESOURCE Permite el acceso a las tablas de la base de datos con permisos para crear tablas permanentes e índices. DBA Da todos los privilegios de administrador de base de datos.
FROM	Es una palabra reservada.
PUBLIC	Elimina privilegios de acceso a todos los usuarios.
<i>lista de usuarios</i>	Es una lista de los nombres de los usuarios cuyo privilegio será revocado.

SET LOCK MODE

Determina si la siguiente llamada de SQL espera para bloquear un renglón.

Sintaxis:

```
SET LOCK MODE TO [ NOT ] WAIT
```

Explicación:

SET LOCK MODE	Son palabras reservadas.
TO	Es palabra reservada.
NOT	Es una palabra reservada opcional que indica a Informix-4GL que retorne un error si una declaración intenta alterar o eliminar un renglón que ha sido bloqueado por otro proceso.

WAIT

Es una palabra reservada que indica a Informix-4GL que espere para alterar o borrar un renglón hasta que este quede desbloqueado por otro proceso.

UNLOCK TABLE

Restaura el acceso a la tabla previamente bloqueada.

Sintaxis:

UNLOCK TABLE *nombre de la tabla*

Explicación:

UNLOCK TABLE
nombre de la tabla

Son palabras reservadas.
Es el nombre de la tabla que se desea desbloquear.

Integridad de Datos.

Informix-SQL proporciona integridad de datos para llevar a cabo una transacción. Una transacción es una serie de operaciones (declaraciones SQL) sobre una base de datos que se desea completar enteramente o no. Las siguientes declaraciones son usadas para el control de transacciones:

BEGIN WORK

Marca el inicio de una transacción.

Sintaxis:

BEGIN WORK

Explicación:

BEGIN WORK Son palabras reservadas para iniciar una transacción.

COMMIT WORK

Marca el final de una transacción autorizando todas las modificaciones hechas en la base de datos durante ésta.

Sintaxis:

COMMIT WORK

Explicación:

COMMIT WORK Son palabras reservadas que indican el final de la transacción.

ROLLBACK WORK

Deshace todas las modificaciones hecha a la base de datos durante la transacción.

Sintaxis:

ROLLBACK WORK

Explicación:

ROLLBACK WORK Son palabras reservadas.

Declaraciones empleadas en el programa de aplicación.

CALL

Invoca una función.

Sintaxis:

CALL *función* ([*lista de argumentos*]) [RETURNING *lista de variables*]

Explicación:

CALL	Es una palabra reservada requerida.
<i>función</i>	Es el nombre de la función a invocar.
<i>lista de argumentos</i>	Es una lista de cero o más expresiones, separadas por comas y encerradas en paréntesis que serán pasadas a la función. Los paréntesis son requeridos aunque no existan argumentos.
RETURNING	Es una palabra reservada opcional para especificar las variables que la función retornará.
<i>lista de variables</i>	Es una lista de una o más variables de programa, separadas por comas.

CASE

Selecciona una secuencia de declaraciones, dependiendo del valor actual de una expresión.

Sintaxis:

```
CASE { (expresión) }  
  WHEN { expresión | expresión booleana }  
    declaración  
    ...  
  [ EXIT CASE ]  
  ...  
  WHEN { expresión | expresión booleana }  
    declaración  
    ...  
  [ EXIT CASE ]  
  ...  
...
```

```

[ OTHERWISE ]
    declaración
    ...
[ EXIT CASE ]
    ...
END CASE

```

Explicación:

CASE	Es una palabra reservada requerida.
<i>expresión</i>	Es un valor que retorna un INTEGER, SMALLINT, DECIMAL, o CHAR(1).
WHEN	Es una palabra reservada requerida.
<i>expresión booleana</i>	Es una expresión verdadera o falsa.
<i>declaración</i>	Es una declaración Informix-4GL.
EXIT CASE	Es una declaración opcional que causa la salida del CASE.
OTHERWISE	Es una palabra reservada opcional que indica una secuencia de declaraciones a ser ejecutadas si ninguna de las cláusulas WHEN es ejecutada.
END CASE	Es una palabra reservada requerida que termina la declaración CASE.

CLEAR

Limpia la pantalla y todos los campos en una forma o un conjunto de campos.

Sintaxis:

```
CLEAR { SCREEN | FORM | lista de campos }
```

Explicación:

CLEAR	Es una palabra reservada requerida.
SCREEN	Es una palabra reservada para limpiar la pantalla.
FORM	Es una palabra reservada para limpiar los valores en todos los campos de una forma.
<i>lista de campos</i>	Es una lista de uno o más nombres de campos a ser limpiados.

DEFINE

Define las variables del programa asignándoles un espacio de memoria

Sintaxis:

```

DEFINE lista de variables { tipo de dato
    | LIKE nombre de tabla.columna
    | RECORD { LIKE nombre de tabla.* | lista de variables [, ... ]
    END RECORD } [, ... ]

```

Explicación:

DEFINE	Es una palabra reservada requerida.
<i>lista de variables</i>	Es uno o más identificadores de variables del programa.
<i>tipo de dato</i>	Es uno de los tipos de datos que maneja Informix-4GL.
LIKE	Es una palabra reservada para especificar el tipo de dato indirectamente.
<i>nombre de tabla.columna</i>	Es el identificador completo para una columna en la base de datos actual. La declaración DATABASE debe ir antes que DEFINE para usar el <i>tipo de dato</i> indirectamente.
RECORD	Es un tipo de dato que describe un conjunto de variables de diferentes tipos de datos.
<i>nombre de tabla</i>	Es el nombre de una tabla de la base de datos.
END RECORD	Son palabras reservadas que siguen al último elemento del registro del programa.

DISPLAY

Despliega valores de datos en la pantalla.

Sintaxis:

```
DISPLAY { BY NAME lista de variables |  
          lista de variables [ TO { lista de campos | registro de la forma { [ n ] } ] } {,... }  
          | AT renglon, columna } }  
[ ATTRIBUTE (lista de atributos) ]
```

Explicación:

DISPLAY	Es una palabra reservada requerida.
BY NAME	Son palabras reservadas que indican a Informix-4GL que compare los nombres de variables con nombres de campos de pantalla.
<i>lista de variables</i>	Es una lista requerida de una o más variables de programa y/o constantes, separadas por comas.
TO	Es una palabra reservada para especificar que se desplegará la <i>lista de variables</i> en los campos de la forma o en un arreglo de la forma.
<i>lista de campos</i>	Es una lista de uno o más nombres de campos de la forma.
<i>registro de la forma</i>	Es el identificador de una colección de nombres de campos definidos en una forma como un SCREEN RECORD.
[<i>n</i>]	Es un entero, encerrado en corchetes, para especificar el renglón de un arreglo de la forma donde <i>lista de variables</i> será desplegada.
AT	Es una palabra reservada para especificar las coordenadas donde se desplegarán los datos.
<i>renglón</i>	Es una variable o constante entera, que indica el renglón de la pantalla en donde se desplegarán los datos.
<i>columna</i>	Es una variable o constante entera, que indica una columna de la pantalla en donde se desplegarán los datos.
ATTRIBUTE	Es una palabra reservada para especificar los atributos a desplegar.

(lista de atributos)

Es una lista (en paréntesis) de uno o más atributos a desplegar, separados por comas. Los más usuales son REVERSE, BLINK y UNDERLINE.

DISPLAY FORM

Despliega una forma de pantalla precompilada.

Sintaxis:

DISPLAY FORM *nombre de la forma*

Explicación:

DISPLAY FORM

Son palabras reservadas requeridas que indican el desplegado de una forma.

nombre de la forma

Es un identificador de Informix-4GL que ha sido asociado con una forma de pantalla en una declaración OPEN FORM.

ERROR

Despliega un mensaje de error en la línea de error (por default la última línea de la pantalla).

Sintaxis:

ERROR *lista a desplegar*

Explicación:

ERROR

Es una palabra reservada o requerida.

lista a desplegar

Es una lista de una o más variables de programa y/o cadena de caracter (encerradas en comillas), separadas por comas.

EXIT PROGRAM

Indica la terminación del programa.

Sintaxis:

EXIT PROGRAM [*expresión entera*]

Explicación:

EXIT PROGRAM

Son palabras opcionales.

expresión entera

Es un número entero que regresa el código de retorno al sistema operativo.

GLOBALS

Define una o mas variables que seran usadas globalmente o se indica el archivo donde se encuentran definidas las variables globales.

Sintaxis:

```
GLOBALS { "nombre del archivo" |  
    DEFINE-declaración  
    ...  
END GLOBALS }
```

Explicación:

GLOBALS	Es una palabra reservada requerida.
"nombre del archivo"	Es la ruta y nombre del archivo donde se encuentran definidas las variables globales. Debe ir entre comillas.
DEFINE-declaración	Es la declaración DEFINE para las variables globales.
END GLOBALS	Son palabras reservadas requeridas que terminan la declaración GLOBALS donde las variables son definidas.

INITIALIZE

Inicializa una variable de programa.

Sintaxis:

```
INITIALIZE lista de variables { LIKE lista de columnas | TO NULL }
```

Explicación:

INITIALIZE	Es una palabra reservada requerida.
lista de variables	Es una lista de una o más variables, separadas por comas.
LIKE	Es una palabra reservada opcional que indica como se ini- cializa <i>lista de variables</i> .
lista de columnas	Es una lista de nombres de columnas, precedidas por nom- bre de la tabla y separadas por comas.
TO NULL	Son palabras reservadas opcionales para asignar valores nulos.

INPUT

Asigna valores a variables de programa que el usuario introduce desde una forma de pantalla.

Sintaxis:

```
INPUT { BY NAME lista de variables [ WITHOUT DEFAULTS ] |  
    lista de variables [ WITHOUT DEFAULTS ]  
    FROM { lista de campos | registro de pantalla ( [ n ] . * ) [ , ... ] }  
        [ { BEFORE FIELD sublista de campo  
            | AFTER { FIELD sublista de campo | INPUT }  
            | ON KEY ( lista de llaves ) }  
            declaración  
        ...  
        [ NEXT FIELD nombre de campo ]  
        ...  
        [ EXIT INPUT ]  
    ...  
END INPUT }
```

Explicación:

INPUT	Es una palabra reservada. Permite al usuario cambiar los datos desplegados en pantalla.
BY NAME	Son palabras reservadas que indican a Informix-4GL que compare los nombres de variables con nombres de campos en la forma.
<i>lista de variables</i>	Es una lista de variables de programa a desplegar.
WITHOUT	Son palabras reservadas para desplegar en pantalla el valor actual de <i>lista de variables</i> .
DEFAULTS	
FROM	Es una palabra reservada para especificar los campos de la forma, cuyos valores serán asignados a <i>lista de variables</i> .
<i>lista de campos</i>	Es una lista de uno o más nombres de campos de la forma.
<i>registro de pantalla</i>	Es el identificador de una colección de nombres de campos definidos en la forma como un SCREEN RECORD.
[<i>n</i>]	Es un entero, encerrado en corchetes, para especificar el renglón de un arreglo de la forma.
BEFORE FIELD	Son palabras reservadas para transferir el control a una declaración 4GL cuando el cursor entra en el campo <i>sublista de campo</i> .
<i>sublista de campo</i>	Es una lista de uno o más campos referenciados explícitamente o implícitamente en la declaración INPUT.
AFTER FIELD	Son palabras reservadas para transferir el control a una declaración 4GL cuando el cursor sale de un campo en la <i>sublista de campo</i> .
AFTER INPUT	Son palabras reservadas para transferir el control a una declaración 4GL cuando el usuario ha finalizado de introducir los datos.
ON KEY	Son palabras reservadas para especificar las teclas.
(<i>lista de llaves</i>)	Es una lista de teclas designada(s), generalmente teclas de función o de control. Si una de éstas es presionada durante el INPUT, una declaración en 4GL es ejecutada.
<i>declaración</i>	Es una declaración 4GL. Esta es ejecutada durante un INPUT, si las cláusulas BEFORE FIELD, AFTER FIELD u ON KEY son satisfechas. Es ejecutada después de un INPUT si una cláusula AFTER INPUT es usada.
NEXT FIELD	Son palabras reservadas que indican a Informix-4GL mover el cursor inmediatamente al campo especificado.
<i>nombre de campo</i>	Identifica un campo de <i>lista de campos</i> o <i>registro de pantalla</i> .
EXIT INPUT	Son palabras reservadas para salir inmediatamente de la declaración INPUT.
END INPUT	Son palabras reservadas que terminan la declaración INPUT.

LET

Asigna un valor a una variable de programa.

Sintaxis:

LET *variable* = *expr*

Explicación:

LET	Es una palabra reservada requerida
<i>variable</i>	Es el identificador de una variable de programa.
<i>expr</i>	Es una expresión que contiene un valor.

MAIN

Indica el inicio del bloque principal del programa.

Sintaxis:

```
MAIN
  declaración
  . . .
END MAIN
```

Explicación:

MAIN	Es una palabra reservada requerida. Todo programa en Informix-4GL debe tener un bloque principal.
<i>declaración</i>	Es cualquier tipo de declaración de Informix-4GL excepto MAIN.
END MAIN	Son palabras reservadas requeridas que indican la terminación del programa principal.

OPEN FORM

Asocia un identificador Informix-4GL con una forma de pantalla precompilada.

Sintaxis:

```
OPEN FORM nombre de la forma FROM "archivo-forma"
```

Explicación:

OPEN FORM	Son palabras reservadas requeridas.
<i>nombre de la forma</i>	Es un identificador de Informix-4GL.
"archivo-forma"	Es la ruta del nombre de la forma compilada. Debe ir entre comillas.

OPTIONS

Modifica los atributos de entrada para formas de pantalla y cambia las teclas para la operación en pantalla.

Sintaxis:

```
OPTIONS { INPUT { WRAP | NO WRAP } |
        ACCEPT KEY nombre de la tecla
}
```

Explicación:

OPTIONS
INPUT WRAP

Es una palabra reservada.

Son palabras reservadas opcionales que indican al cursor se mueva en la lista de los campos del INPUT durante la ejecución de un INPUT, hasta que la tecla de aceptación es presionada. El default es INPUT NO WRAP.

INPUT NO
WRAP
ACCEPT KEY

Son palabras reservadas que indican que el INPUT termina con un RETURN después del último campo.

Son palabras reservadas opcionales que especifican la tecla con la cual termina una declaración INPUT. El default es la tecla ESC.

nombre de la tecla

Es la tecla con la cual se termina una declaración INPUT.

RETURN

Abandona una función y regresa valores a la rutina llamada.

Sintaxis:

RETURN [*lista de expresiones*]

Explicación:

RETURN
lista de expresiones

Es una palabra reservada requerida.

Es una lista de una o más expresiones, separadas por comas, deben ser del mismo tipo a la lista de argumentos en la cláusula RETURNING de la declaración CALL.

Creación de una forma.

Es una pantalla que se despliega en la terminal, soporta entradas o salidas con un programa de aplicación 4GL para introducir, extraer, modificar o borrar datos.

La forma consiste de tres secciones requeridas (DATABASE, SCREEN y ATTRIBUTES) y dos secciones opcionales (TABLES e INSTRUCTIONS).

La estructura de una forma es la siguiente:

DATABASE

{ *nombre de la base de datos* | FORMONLY }

SCREEN [SIZE *líneas* [BY *columnas*]]

{ *texto, caracteres gráficos y*

[*etiqueta de campo*]

}

[END]

[TABLES

[*alias de la tabla* = [*dueño.*] *tabla*] ...

[END]]

ATTRIBUTES

etiqueta de campo =
{ *nombre de la tabla.columna*
| FORMONLY.*nombre del campo*
| TYPE [*tipo de dato* [NOT NULL] | LIKE *nombre de la tabla.columna*] } ...
[, *lista de atributos*] [= ...] [;] [= ...] ;

...

[END]

[INSTRUCTIONS

[DELIMITERS "ab"]

[SCREEN RECORD *nombre del registro* [[*n*]]

({ *nombre de la tabla.* *

| *nombre de la tabla.columna1* THRU *nombre de la tabla.columna2*

| *nombre de la tabla.columna n* } [, ...])

...]

[END]]

Sección DATABASE.

Identifica la base de datos con la cual la forma va a trabajar.

Explicación:

DATABASE

Es una palabra reservada requerida que indica el inicio de la sección.

nombre de la base de datos

Es el nombre de una base de datos que contiene columnas usadas para definir campos desplegados de la forma.

FORMONLY

Es una palabra reservada para indicar que la forma no está asociada con ninguna base de datos.

Sección SCREEN

Especifica las dimensiones vertical y horizontal de la pantalla, y la posición de los campos a desplegar e información que aparecerá en la pantalla.

Explicación:

SCREEN

Es una palabra reservada que indica el inicio de la sección.

SIZE

Es una palabra reservada opcional que especifica el tamaño vertical y horizontal de la ventana.

líneas

Es un entero que especifica el número total de líneas que la pantalla puede desplegar. El default es 24 líneas.

BY

Es una palabra reservada opcional para especificar cuantos caracteres por línea se pueden desplegar.

columnas

Es un entero que especifica el ancho de la ventana.

texto y caracteres gráficos

Son cadenas de caracteres ASCII que se desea aparezcan en la pantalla.

[]	Son delimitadores para un campo. El ancho del campo es el número de caracteres que pueden ser plasmados entre los corchetes.
<i>etiqueta de campo</i> { }	Es el identificador del campo a desplegar en pantalla. Las llaves son requeridas para indicar el inicio y fin de lo que se despliega en pantalla.
END	Es una palabra reservada opcional para finalizar la sección.

Sección TABLES

Lista todas las tablas que hacen referencia en la forma.

Explicación:

TABLES	Es una palabra reservada que indica el inicio de la sección.
<i>alias de la tabla</i>	Es el alias que se utiliza en la forma.
<i>dueño</i>	Es el nombre del usuario que creó la tabla.
<i>tabla</i>	Es el identificador o sinónimo de <i>tabla</i> en la base de datos.
END	Es una palabra reservada opcional para finalizar la sección.

Sección ATTRIBUTES

Asocia un identificador y un tipo de dato con cada campo en la sección SCREEN. Describe el comportamiento y apariencia de cada campo como Informix-4GL lo desplegará, especifica un valor por default, el límite de los valores que pueden ser introducidos y otros parámetros.

Explicación:

ATTRIBUTES	Es una palabra reservada requerida que marca el inicio de la sección.
<i>etiqueta de campo</i>	Es la etiqueta del campo especificada en la sección SCREEN.
<i>nombre de la tabla.columna</i>	Es un nombre de una columna de la tabla especificada en la sección TABLES.
FORMONLY	Es una palabra reservada indicando que el campo no corresponde a una columna de una tabla en la base de datos.
<i>nombre del campo</i>	Es un identificador SQL para el nombre del campo.
TYPE	Es una palabra reservada para especificar un tipo de dato en Informix-4GL.
<i>tipo de dato</i>	Es cualquier tipo de dato excepto SERIAL.
LIKE	Es una palabra reservada para asociar el campo de la forma con el tipo de dato de una columna de una tabla.
NOT NULL	Son palabras reservadas para requerir un valor en el campo.

lista de atributos

Es una lista de uno o más atributos desplegados en el campo, separados por comas. Los más usuales son: UPSHIFT (convierte letras minúsculas a mayúsculas), REVERSE (despliega el campo en video inverso), AUTONEXT (el cursor avanza automáticamente durante el INPUT al siguiente campo cuando el campo actual sea tecleado completamente), WORDWRAP COMPRESS (se utiliza en campos de línea múltiple para unir una cadena de caracteres larga), COMMENTS (despliega un mensaje en la última línea de la forma, es desplegado cuando el cursor se mueve al campo asociado y se borra cuando se mueve a otro campo)

Sección INSTRUCTIONS.

Especifica los delimitadores en los campos, define registros y arreglos de la forma.

Explicación:

DELIMITERS	Es una palabra reservada para especificar los delimitadores en el campo.
<i>a</i>	Es el delimitador de campo que abre.
<i>b</i>	Es el delimitador de campo que cierra.
SCREEN RECORD	Son palabras reservadas para definir una lista de campos como un registro o arreglo de la forma.
<i>nombre del registro</i>	Es un identificador SQL para el registro de la forma.
<i>n</i>	Es un entero que especifica el número de renglones en el arreglo de la forma.
<i>nombre de la tabla. *</i>	Es el nombre de la tabla, alias o sinónimo (. * indica todas las columnas de la tabla).
<i>columna1,</i> <i>columna2,</i> <i>columna n</i>	Son nombres de campos que se definieron en la sección ATTRIBUTES.
THRU	Es una palabra reservada opcional que especifica campos consecutivos.
END	Es una palabra opcional que especifica el final de la sección.

PROGRAMA DE APLICACIÓN

A continuación se lista el código fuente que ejemplifica como crear una base de datos, una tabla y una forma.

MAIN

```
CREATE DATABASE prueba
```

```
DATABASE prueba
```

```
CREATE TABLE contrib
```

```
(  
  rfc          CHAR(13),  
  oficina      CHAR(3),  
  nombre       VARCHAR(80),  
  domicilio    VARCHAR(150),  
  fec_alta     DATE  
)
```

```
CREATE UNIQUE INDEX ixcontrib ON contrib( rfc )
```

END MAIN

Creación de formas:

```
DATABASE FORMONLY
```

```
SCREEN
```

```
{
```

```
  **  MENU  **
```

```
  \[1\] A L T A S  
  \[2\] B A J A S  
  \[3\] C A M B I O S  
  \[4\] C O N S U L T A  
  \[9\] S A L I D A
```

```
  TECLEE SU OPCION :[x]
```

```
}
```

```
ATTRIBUTES
```

```
x = FORMONLY.vx, REVERSE,
```

```
  COMMENTS=" Presione el numero de la Opcion deseada ( 1 a 4 y 9 ) ";
```

```
INSTRUCTIONS
```

```
DELIMITERS " "
```

DATABASE prueba
SCREEN
{

R.F.C. : [r1]

OFICINA : [o1]

NOMBRE : [n1]
 [n1]

DOMICILIO : [d1]
 [d1]
 [d1]

FECHA DE
ALTA : [f1]
 [x]

}

TABLES
contrib
END

ATTRIBUTES

r1 = contrib.rfc, UPSHIFT, REVERSE, AUTONEXT;
o1 = contrib.oficina, REVERSE, AUTONEXT;
n1 = contrib.nombre, UPSHIFT, REVERSE, WORDWRAP COMPRESS, AUTONEXT;
d1 = contrib.domicilio, REVERSE, WORDWRAP COMPRESS,AUTONEXT;
f1 = contrib.fec_alta, REVERSE, COMMENTS = "FORMATO DE CAPTURA AA/MM/DD";
x = FORMONLY.vx;
END

INSTRUCTIONS

DELIMITERS " "

SCREEN RECORD sc_rg(contrib.rfc THRU contrib.fec_alta)

A continuación se detalla el código fuente que ejemplifica las altas, bajas, cambios y consultas de un contribuyente:

```
DATABASE prueba
GLOBALS
  DEFINE r_cont RECORD LIKE contrib.*,
        vx CHAR(1)
END GLOBALS
*****
MAIN

  OPEN FORM fcontrib FROM "contrib"
  OPEN FORM fmenu FROM "menu"
  CLEAR SCREEN

  WHILE TRUE
    OPTIONS INPUT NO WRAP
    DISPLAY FORM fmenu

    INPUT BY NAME vx
    AFTER INPUT
      IF ( vx NOT MATCHES "[12349]" ) THEN
        ERROR " OPCION INVALIDA "
        NEXT FIELD vx
      END IF
      IF ( vx = "9" ) THEN
        EXIT PROGRAM(0)
      END IF
    END INPUT

    OPTIONS ACCEPT KEY FB, INPUT WRAP
    CASE vx
      WHEN "1" CALL altas()
      WHEN "2" CALL bajas()
      WHEN "3" CALL cambios()
      WHEN "4" CALL consultas()
    END CASE

  END WHILE

END MAIN
*****
FUNCTION altas()

  DISPLAY FORM fcontrib

  WHILE TRUE
    INITIALIZE r_cont.* TO NULL
    DISPLAY "Opciones : <F6> Regresar al MENU " AT 2,3
    DISPLAY "          : <F7> Dar de ALTA el R.F.C." AT 3,3
```

```

LET r_cont.fec_alta = TODAY
INPUT r_cont.* WITHOUT DEFAULTS FROM sc_rg.*
ON KEY (F7)
    DISPLAY "" AT 2,3
    RETURN
AFTER INPUT
    IF ( ( LENGTH(r_cont.rfc) = 0 ) OR ( LENGTH(r_cont.rfc) < 13 ) ) THEN
        ERROR "R.F.C. INCORRECTO."
        NEXT FIELD sc_rg.rfc
    END IF
    IF ( LENGTH(r_cont.oficina) = 0 ) THEN
        ERROR "SE DEBE CAPTURAR LA CLAVE DE A.L.R."
        NEXT FIELD sc_rg.oficina
    END IF
    IF ( LENGTH(r_cont.nombre) = 0 ) THEN
        ERROR "SE DEBE CAPTURAR EL NOMBRE DEL CONTRIBUYENTE"
        NEXT FIELD sc_rg.nombre
    END IF
    IF ( LENGTH(r_cont.domicilio) = 0 ) THEN
        ERROR "SE DEBE CAPTURAR EL DOMICILIO"
        NEXT FIELD sc_rg.domicilio
    END IF
    IF ( ( r_cont.fec_alta IS NULL ) or ( r_cont.fec_alta > TODAY ) ) THEN
        ERROR "FECHA DE ALTA INCORRECTA"
        NEXT FIELD sc_rg.fec_alta
    END IF
END INPUT

INSERT INTO contrib VALUES (r_cont.*)
ERROR "REGISTRO DADO DE ALTA."
SLEEP 02
ERROR ""
END WHILE

END FUNCTION # Fin de la Funcion ALTAS
#*****
FUNCTION bajas()

    DISPLAY FORM fcontrib

    WHILE TRUE
        CLEAR FORM
        DISPLAY " " AT 2,3
        DISPLAY " " AT 3,3
        DISPLAY "Opciones : <F7> Desplegar Datos del R.F.C." AT 2,3
        DISPLAY " : <F6> Regresar al MENU " AT 3,3

        INPUT r_cont.rfc FROM rfc
        ON KEY(F7)
            DISPLAY "" AT 2,3
            RETURN
    
```

```

AFTER INPUT
  IF ( r_cont.rfc IS NULL ) THEN
    ERROR " SE REQUIERE TECLEAR EL R.F.C. "
    NEXT FIELD rfc
  END IF
  SELECT * INTO r_cont.* FROM contrib
  WHERE rfc = r_cont.rfc
  IF ( STATUS = NOTFOUND ) THEN
    ERROR " EL R.F.C. NO EXISTE EN LA BASE DE DATOS "
    NEXT FIELD rfc
  END IF
END INPUT

DISPLAY " " " AT 2,3
DISPLAY " " " AT 3,3
DISPLAY "Opciones : <F7> Realizar la BAJA del R.F.C." AT 2,3
DISPLAY " : <F8> Cancelar la BAJA " AT 3,3
DISPLAY BY NAME r_cont.*

INPUT BY NAME vx
ON KEY(F9)
  CLEAR FORM
  CONTINUE WHILE
END INPUT

DELETE FROM contrib WHERE rfc = r_cont.rfc
DISPLAY " R.F.C. DADO DE BAJA " AT 23,3
SLEEP 2
DISPLAY "" AT 23,3
END WHILE

END FUNCTION # Fin de la Funcion BAJAS
#*****
FUNCTION cambios()

  DISPLAY FORM fcontrib

  WHILE TRUE
    CLEAR FORM
    DISPLAY " " " AT 2,3
    DISPLAY " " " AT 3,3
    DISPLAY "Opciones : <F7> Desplegar Datos del R.F.C." AT 2,3
    DISPLAY " : <F6> Regresar al MENU " AT 3,3

    INPUT r_cont.rfc FROM rfc
    ON KEY(F7)
      DISPLAY "" AT 2,3
      RETURN

  AFTER INPUT
    IF ( r_cont.rfc IS NULL ) THEN
      ERROR " SE REQUIERE TECLEAR EL R.F.C. "

```



```

        NEXT FIELD rfc
    END IF
    SELECT * INTO r_cont.* FROM contrib
    WHERE rfc = r_cont.rfc
    IF ( STATUS = NOTFOUND ) THEN
        ERROR " EL R.F.C. NO EXISTE EN LA BASE DE DATOS "
    NEXT FIELD rfc
    END IF
END INPUT

DISPLAY " " AT 2,3
DISPLAY " " AT 3,3
DISPLAY "Opciones : <F7> Realizar el CAMBIO " AT 2,3
DISPLAY " : <F8> Cancelar el CAMBIO " AT 3,3
DISPLAY r_cont.fec_alta TO fec_alta

INPUT r_cont.oficina, r_cont.nombre, r_cont.domicilio WITHOUT DEFAULTS
FROM oficina, nombre, domicilio
ON KEY(F9)
    CLEAR FORM
    CONTINUE WHILE
AFTER INPUT
    IF ( r_cont.oficina IS NULL ) THEN
        ERROR " SE REQUIERE TECLEAR LA OFICINA "
    NEXT FIELD oficina
    END IF
    IF ( r_cont.nombre IS NULL ) THEN
        ERROR " SE REQUIERE TECLEAR EL NOMBRE DEL CONTRIBUYENTE "
    NEXT FIELD nombre
    END IF
    IF ( r_cont.domicilio IS NULL ) THEN
        ERROR " SE REQUIERE TECLEAR EL DOMICILIO DEL CONTRIBUYENTE "
    NEXT FIELD domicilio
    END IF
END INPUT

UPDATE contrib SET oficina = r_cont.oficina,
                 nombre = r_cont.nombre,
                 domicilio = r_cont.domicilio
WHERE rfc = r_cont.rfc
DISPLAY " CAMBIO REALIZADO " AT 23,3
SLEEP 2
DISPLAY "" AT 23,3
END WHILE
END FUNCTION # Fin de la Funcion CAMBIOS
#*****
FUNCTION consultas()

DISPLAY FORM fcontrib

WHILE TRUE
    DISPLAY "Opciones : <F6> Regreso a MENU " AT 2,3

```

```

DISPLAY "      : <F7> CONSULTA de R.F.C." AT 3,3
INITIALIZE r_cont.* TO NULL
CLEAR FORM

INPUT r_cont.rfc FROM sc_rg.rfc
ON KEY (F7)
  DISPLAY "" AT 2,3
  RETURN
AFTER INPUT
  IF ( LENGTH (r_cont.rfc) != 13 ) THEN
    ERROR "R.F.C. INCORRECTO"
    INITIALIZE r_cont.rfc TO NULL
    NEXT FIELD sc_rg.rfc
  END IF
  SELECT * INTO r_cont.* FROM contrib WHERE rfc = r_cont.rfc
  IF ( STATUS = NOTFOUND ) THEN
    ERROR "R.F.C. NO ENCONTRADO EN LA BASE DE DATOS."
    INITIALIZE r_cont.rfc TO NULL
    NEXT FIELD sc_rg.rfc
  END IF
END INPUT

DISPLAY "      " AT 3,3
DISPLAY "Opcion : <F7> CONSULTA de otro R.F.C." AT 2,3
DISPLAY r_cont.* TO sc_rg.*

INPUT BY NAME vx
ON KEY (F8)
  CONTINUE WHILE
END INPUT

END WHILE
END FUNCTION # Fin de la Funcion CONSULTAS
#*****

```

BIBLIOGRAFÍA

Roger S. Pressman
Ingeniería del software
Editorial Mc Graw Hill

Henry F. Korth
Abraham Silberschatz
Fundamentos de Bases de Datos
Editorial Mc Graw Hill

Harvey M. Deitel
Introducción a los Sistemas Operativos
Editorial Addison Wesley

Sandra L. Emerson
The practical SQL Handbook Using Structured Query Language
Editorial Addison Wesley

Informix-4GL Reference Manual
Volume One & Two
Informix

Larry R. Newcomer
Select ... SQL
The Relational Database Language
Editorial Mawmillan

Philip Steele
Robert Heydt
Dbase IV 1.1 y SQL
Programación para expertos
Editorial Megabyte Grupo Noriega Editores

Rick F. van der Lans
Introduction to SQL
Editorial Addison Wesley

Manual de usuario.