



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE CIENCIAS

SISTEMA PARA LA ELABORACION DEL
ANTEPROYECTO DEL PRESUPUESTO DE
EGRESOS DE LA FEDERACION.

T E S I S

QUE PARA OBTENER EL TITULO DE:
A C T U A R I O
P R E S E N T A :
S E R G I O O L V E R A V E R A



ASESOR DE TESIS: ACT. VICTOR M. SOLIS NAJERA.

MEXICO, D. F.

1998

TESIS CON
FALLA DE ORIGEN

258688



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 está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

M. en C. Virginia Abrín Batule
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
P r e s e n t e

Comunicamos a usted que hemos revisado el trabajo de Tesis:
"SISTEMA PARA LA ELABORACION DEL ANTEPROYECTO DEL
PRESUPUESTO DE EGRESOS DE LA FEDERACION"

realizado por . SERGIO OLVERA VERA

con número de cuenta 8515884-1 , pasante de la carrera de ACTUARIO

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis Propietario	ACT. VICTOR MANUEL SOLIS NAJERA
Propietario	ACT. MA. AURORA VALDES MICHEL
Propietario	ACT. LAURA MIRIAM QUEROL GONZALEZ
Suplente	ACT. LETICIA DANIEL ORANA
Suplente	ACT. MA. SUSANA BARRERA OCAMPO

Victor Manuel Solis Najera
L. Qu. Q. 9
Leticia Daniel Orana
S.

Consejo Departamental de Matemáticas

Maestra Ma. del Pilar Alonso Reyes

Maestra Ma. del Pilar Alonso Reyes

***Sistema para la elaboración del
Anteproyecto del Presupuesto de Egresos
de la Federación***

Introducción	i
Capitulo I Sistema de Base de Datos	1
I.1 Concepto	1
I.2 Base de Datos	2
I.3 Objetivos de una Base de Datos	2
I.4 Modelos de Datos	6
I.4.1 Modelos Lógicos Basados en Objetos	6
I.4.2 Modelos Lógicos Basados en Registros	7
I.4.3 Modelo Relacional	7
I.4.4 Modelo de Red	8
I.4.5 Modelo Jerárquico	9
I.4.6 Modelos Físicos de Datos	10
I.5 Conceptos Generales	10
I.5.1 Lenguaje de Definición de Datos	10
I.5.2 Lenguaje de Manejo de Datos	10
I.5.3 Diccionario de Datos	11
I.5.4 Administrador de la Base de Datos	11
I.5.5 Usuarios de la Base de Datos	12
I.6 Estructura General de un Sistema de Base de Datos	13

I.7 El Modelo Entidad-Relación	14
I.7.1 Relaciones y Conjuntos de Relaciones	16
I.7.2 Limitantes de Mapeo	17
I.7.3 Llaves Primarias	19
I.7.4 Diagrama de Entidad-Relación (E-R)	20
I.7.5 El Enfoque Relacional	22
I.7.6 Lenguajes de Consulta	24
I.7.7 El Algebra Relacional	24
Capítulo II Arquitectura de ORACLE	26
II.1 Conceptos	26
II.2 Bases de Datos Relacionales	26
II.2.1 Espacios de Tablas (Tablespaces)	27
II.2.2 Archivos	27
II.3 Instancias	28
II.4 Estructuras Internas de la Base de Datos	28
II.4.1 Tablas y Columnas	29
II.4.2 Usuarios y Esquemas	32
II.4.3 Indices	32

II.4.4 Grupos	33
II.4.5 Grupos Hash	33
II.4.6 Vistas	34
II.4.7 Secuencias	34
II.4.8 Procedimientos, Funciones, Paquetes y Disparadores (Triggers)	34
II.4.9 Sinónimos	36
II.4.10 Privilegios y Funciones (Roles)	36
II.4.11 Enlaces de Bases de Datos	37
II.4.12 Segmentos, Extensiones y Bloques	38
II.4.13 Segmentos de Rollback	38
II.5 Estructuras Internas de la Memoria	39
II.5.1 Área Global del Sistema (System Global Area : SGA)	39
II.5.2 Búferes del Bloque de Datos	39
II.5.3 Cache del Diccionario	40
II.5.4 Búffer de Registro de Rehacer	41
II.5.5 Fondo Común SQL Compartido	41
II.5.6 Áreas de Contexto	41
II.5.7 Área Global del Programa (Program Global Area : PGA)	42
II.6 Estructuras de los Procesos	42

II.6.1 Supervisor del Sistema : SMON	42
II.6.2 Supervisor del Proceso : PMON	43
II.6.3 Escritor de Base de Datos : DBWR	43
II.6.4 Escritor de Registros : LGWR	44
II.6.5 Punto de Control : CKPT	44
II.6.6 Archivador : ARCH	44
II.6.7 Recuperador : RECO	45
II.6.8 Bloqueo : LCKn	45
II.6.9 Despachador : Dnnn	45
II.6.10 Servidor : Snnn	45
II.7 Estructuras Externas	46
II.7.1 Registros de Rehacer	46
II.7.2 Archivos de Control	46
II.8 Implementación Básica de la Base de Datos	47
II.9 Capacidades de Copia de Seguridad / Recuperación	47
II.9.1 Export / Import (Exportar / Importar)	47
II.9.2 Copias de Seguridad Fuera de Línea	48
II.9.3 Copias de Seguridad en Línea	48
II.9.4 Seguridad de las Cuentas	49

II.10 Privilegios de la Base de Datos	49
II.11 Seguridad de los Objetos	49
II.12 Auditoria	49
II.13 Ejemplo de Disposición Lógica de Base de Datos	50
II.14 Ejemplo de Disposición Física de Base de Datos	50
II.15 Comparación de los Convenios de Modelado Lógico	51
Capitulo III Presupuesto de Egresos de la Federación	53
III.1 Introducción	53
III.2 Concepto de presupuesto	56
III.3 Alcances Conceptuales del Presupuesto	57
III.3.1 Alcance Político	57
III.3.2 Alcance Administrativo	58
III.3.3 Alcance Económico y Financiero	58
III.3.4 Alcance Jurídico	59
III.4 Principios del Presupuesto	60
III.5 Proceso Presupuestario	61
III.6 Presupuesto de Egresos de la Federación	66
Capítulo IV Flujo de Datos del Sistema	70

INTRODUCCIÓN

Sistema de Anteproyecto del Presupuesto de Egresos de la Federación

Introducción

Los sistemas de bases de datos son considerados como una de las áreas de la computación y de la información que ha tenido el más rápido desarrollo. Este crecimiento fue propiciado por la evolución del hardware y software por un lado y por la creciente necesidad de procesamiento de datos por parte de los usuarios, por el otro. La historia de las bases de datos puede ser dividida en cuatro generaciones las cuales son:

La primera generación fue la de los 50's. En esos días, la tarea principal de una computadora era procesar datos bajo el control de un programa. Su uso era limitado, se empleaba para realizar funciones sencillas como contar, sumar, etc. Cada programa era suministrado con el conjunto de datos que iba a operar. En algunos casos, el programa leía los datos necesarios de la memoria secundaria a la memoria principal de la computadora, los procesaba y los regresaba a la memoria secundaria que consistía de tarjetas perforadas cintas magnéticas, las cuales solo permiten procesamiento secuencial. Este fue el primer sistema de archivos secuencial.

Para la segunda generación, las computadoras se podían usar en modo línea y en modo lote. El desarrollo de discos magnéticos como memoria secundaria propició el surgimiento de sistemas de archivos mas sofisticados, los cuales permitían accesos múltiples. Un archivo de acceso directo permite acceder un registro directamente por su dirección (en el disco), sin tener que leer todos los registros que le anteceden.

La tercera generación se caracteriza por una distinción entre la información a nivel lógico y físico y por la necesidad de administrar grandes cantidades de datos. Durante este tiempo se usaron por primera vez los módulos de datos para describir las estructuras físicas desde un punto de vista lógico.

En la cuarta generación, se hace una clara distinción entre el modelo físico y lógico de datos. Se comercializan varias bases de datos y se invade el mercado en la década de los 80's. El aspecto principal radica en que el almacenamiento físico de los datos es transparente a los usuarios, de esta manera, los cambios en el modelo físico o en el lógico no deben afectar al otro modelo. Se populariza el uso de las bases de datos relacionales.

Podemos decir que muchas empresas u organizaciones dependen de la operación continua y eficaz de un sistema de bases de datos. Un sistema de bases de datos es un sistema de mantenimiento de registros basados en computadoras; es decir, un sistema cuyo propósito general es registrar y mantener la información. Tal información puede estar relacionada con cualquier cosa que sea significativa para la empresa donde el sistema opera.

En el corazón de todo sistema de manejo de información existe una base de datos. Un *archivador metálico con registros de proveedores*, un lote de tarjetas con nombres y números de teléfono o un cuaderno con una lista de inventario de un almacén escrita a lápiz, pueden ser considerados como bases de datos. Sin embargo, el archivador o el cuaderno no constituyen en sí mismos la base de datos; lo que los convierte en bases de datos es la forma en que se organiza la información en ellos.

En una base de datos la información normalmente se organiza y se mantiene en una tabla compuesta por renglones y columnas. Los renglones en una tabla de base de datos se llaman registros o tuplas y a las columnas se les llaman campos o atributos.

El sistema que veremos en la presente estaba desarrollado en COBOL con archivos secuenciales en una plataforma IBM 4381 con sistema operativo MVS, el sistema era muy lento y además darle mantenimiento al sistema era muy costoso, motivo por el cual en la Subsecretaría de Egresos de Hacienda (antes Secretaría de Programación y Presupuesto) se tomó la decisión de migrar el sistema a un ambiente relacional para así aprovechar las relaciones entre las tablas y poder crear un sistema integral que hiciera además de elaborar el presupuesto de egresos, llevar el ejercicio del gasto así como de calendarizar el mismo.

El software que se escogió para desarrollar el Sistema Integral de Control Presupuestal (SICP) fue ORACLE bajo ambiente UNIX, la selección de este software se hizo en base a varios parámetros de licitación que solo cumplió la base de datos antes mencionada.

El SICP consta de tres grandes módulos que son :

- ANTEPROYECTO
- CALENDARIO
- EJERCICIO

El módulo de Anteproyecto, que es el que veremos en esta tesis, es el encargado de elaborar el proyecto de Presupuesto de Egresos de la Federación, el cual emite unos reportes que son todo el gasto público y que debe ser aprobado por la Cámara de Diputados.

El segundo módulo, se encarga de calendarizar el gasto público que ya fue autorizado por la Cámara.

El tercero, nos ayuda a controlar durante todo el año, el ejercicio del gasto que fue programado por el PEF.

La presente tesis esta dividida en cuatro capítulos. El capítulo I abarca los conceptos básicos acerca de las bases de datos relacionales así como un panorama general de la estructura de una base de datos.

El capítulo II, nos introduce a la base de datos ORACLE y nos da los conceptos generales relacionados con el paquete.

El capítulo III, nos introduce al concepto de presupuesto y nos relata lo que es el Presupuesto de Egresos de la Federación (PEF).

Finalmente, en el capítulo cuatro se encuentran los diagramas de flujo de los procesos del PEF así como un panorama general de la estructura del módulo de anteproyecto.

1.1 Concepto

Un sistema manejador de bases de datos (en inglés DataBase Management System, DBMS), se compone de una colección de datos interrelacionados y de un conjunto de programas que accesan dichos datos. La colección de los datos se conoce usualmente como una base de datos.

El objetivo principal de un DBMS es proveer un ambiente conveniente y eficiente para utilizar, obtener y almacenar información en una base de datos.

Los sistemas de bases de datos son diseñados para manejar grandes cantidades de información. El manejo de los datos involucra la definición de estructuras para el almacenamiento de la información así como un mecanismo para manipular dicha información.

Las funciones principales de un DBMS son:

- a) crear y organizar la base de datos
- b) establecer y mantener las trayectorias de acceso a la base de datos, de tal manera que los datos en cualquier parte de la base se puedan accesar rápidamente.
- c) manejar los datos de acuerdo con las peticiones de los usuarios
- d) mantener la integridad y seguridad de los datos
- e) registrar el uso de las bases de datos

El sistema de manejo de bases de datos es la porción más importante del software de un DBMS. Existen en el mercado varios manejadores de bases de datos como: INFORMIX, ORACLE, SYBASE, ADABAS, INGRESS, dBASE, etc. Esta tesis se enfocará al manejador Oracle ya que el sistema que se presenta está desarrollado en éste.

1.2 Bases de Datos

El contenido de base de datos se obtiene combinando datos de todas las diferentes fuentes de una organización, de tal manera que los datos estén disponibles para todos los usuarios, y los datos redundantes puedan eliminarse o al menos minimizarse. La figura 1.1 nos muestra la base de datos como un recipiente de datos a ser compartidos por varios programas. El usuario podrá recobrar datos de varias partes de la base ya que los archivos ahí almacenados, están conectados directa o indirectamente.

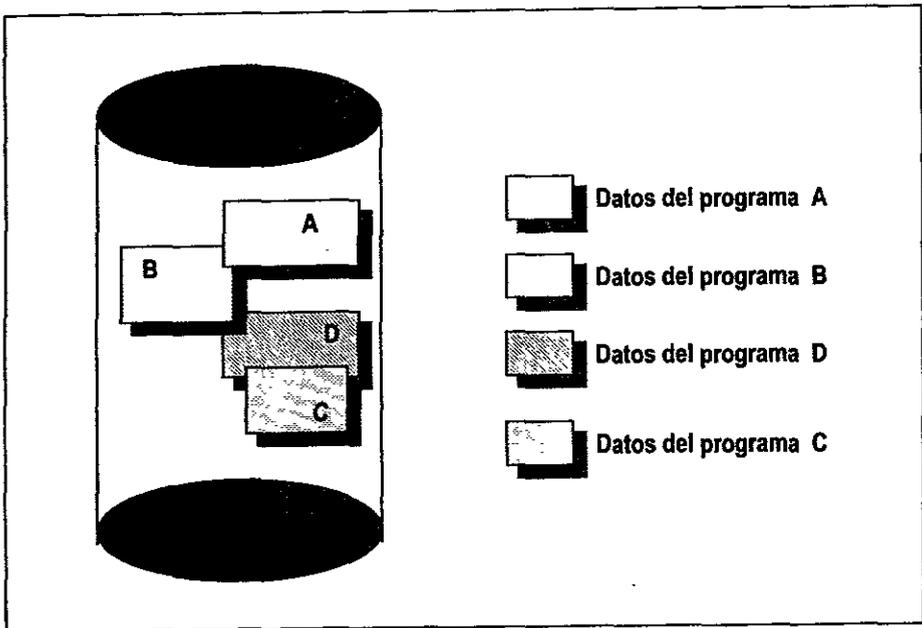


Fig. 1.1. Vista esquemática de los datos almacenados.

1.3 Objetivos de una base de datos

¿Por qué una empresa debe optar por almacenar sus datos en una base de datos?

La mejor respuesta a la pregunta anterior es la siguiente: el tener un sistema de base de datos proporciona a cualquier empresa un control centralizado de su información, la cual constituye uno de sus activos más valiosos. Los principales objetivos que se persiguen al tener un control centralizado de los datos se mencionan a continuación:

– **Reducir la redundancia.**

En empresas que no utilizan sistemas con bases de datos, o la utilizan con un mal diseño, hay una gran cantidad de datos duplicados o redundantes, lo que origina que se eleve el costo de almacenamiento y de acceso, además de incrementar la posibilidad de que exista inconsistencia en la información, es decir, que las distintas copias de la misma información no concuerden entre sí. Con las bases de datos se pretende eliminar la redundancia, aunque algunas veces se prefiere tener una redundancia controlada o mínima con el objeto de minimizar tiempos de acceso o simplificar cuestiones de direccionamiento. Podemos decir entonces que con una base de datos bien diseñada, se evita la redundancia perjudicial o innecesaria.

Al tener redundancia controlada obtenemos cuatro beneficios que son:

- Reducción de errores.
- Datos consistentes.
- Se optimiza el espacio de almacenamiento físico.
- Se reducen los requerimientos de almacenamiento para respaldos.

– **Evitar la inconsistencia de los datos.**

Se puede decir que al eliminar o controlar la redundancia, la inconsistencia en los datos no ocurrirá. Si la redundancia no se elimina, pero se controla (enterando de esto al sistema), entonces se puede garantizar que la base de datos nunca sea inconsistente, ya que cualquier cambio hecho en los datos duplicados actualizará automáticamente a los mismos en las bases de datos.

– **Compartir los datos.**

Esto no sólo significa que las aplicaciones existentes pueden compartir la información en la base de datos, sino que también es factible desarrollar nuevas aplicaciones que operen con los mismos datos almacenados, es decir, las necesidades de datos de las nuevas aplicaciones pueden atenderse sin tener que crear nuevos archivos almacenados.

Los beneficios que se logran son:

- Los datos fuentes pueden ser usados concurrentemente por múltiples aplicaciones.
- Todas las aplicaciones usarán las versiones más actualizadas de los datos.

– **Cumplir normas establecidas.**

Con un control centralizado de la base de datos, se puede garantizar que se cumplan todas las formas aplicables a la representación de los datos las cuales pueden comprender lo siguiente :

normas de la compañía, de instalación, departamentales, industriales, nacionales e internacionales. Es deseable unificar los formatos de los datos almacenados como ayuda para la migración e intercambio de datos entre sistemas.

– **Aplicar restricciones de seguridad.**

No es recomendable que todos los usuarios de un sistema de bases de datos puedan tener acceso a toda la información. Se debe asegurar que el único medio para acceder la base de datos sea a través de los canales establecidos, y por tanto, definir controles de autorización para que se apliquen cada vez que se intente el acceso a datos esenciales. Pueden establecerse diferentes controles para cada tipo de acceso a la base de datos (recuperación, modificación, supresión, etc.).

– **Conservar la integridad.**

El objetivo de la integridad es garantizar que la información almacenada en la base de datos sea exacta, o mejor dicho, consistente. La inconsistencia entre dos entradas que representan un mismo hecho, es un ejemplo de falta de integridad. La integridad de los datos combinada con una redundancia controlada nos da la habilidad de reconstruir completamente los datos si tenemos fallas en el hardware o el software. Debemos incluir procedimientos de chequeo que aseguren que los valores de los datos se ajusten a ciertas reglas establecidas.

– **Abstracción de la información.**

Uno de los objetivos principales de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de la información, esto es, *el sistema oculta ciertos detalles relativos a la forma en como los datos se almacenan y mantienen*. Sin embargo, para que el sistema sea útil, la información debe recuperarse en forma eficiente.

Los niveles de abstracción en los que puede observarse la base de datos son :

- Nivel físico o interno (Modelo Físico) : Este es el nivel más bajo de abstracción, el cual tiene por objetivo la representación y distribución física de los datos y la organización de estos en las unidades de almacenamiento. En este nivel se describen en detalle las estructuras de datos del nivel más bajo.
- Nivel conceptual (Modelo Conceptual) : En este nivel se describen los datos reales que están almacenados en la base de datos y las relaciones existentes entre los mismos. Podemos decir que es la vista lógica de los datos, completamente diferente de la organización física.
- Nivel de visión o externo (Modelo Lógico) : El nivel de visión o externo es el más cercano a los usuarios, es decir, el que atañe a la manera en que cada usuario ve los datos. Este nivel sirve para simplificar la interacción entre usuarios y el sistema el cual puede proporcionar muchas vistas diferentes de la misma base de datos. En la figura 1.2 se muestra la interrelación entre los tres niveles de abstracción.

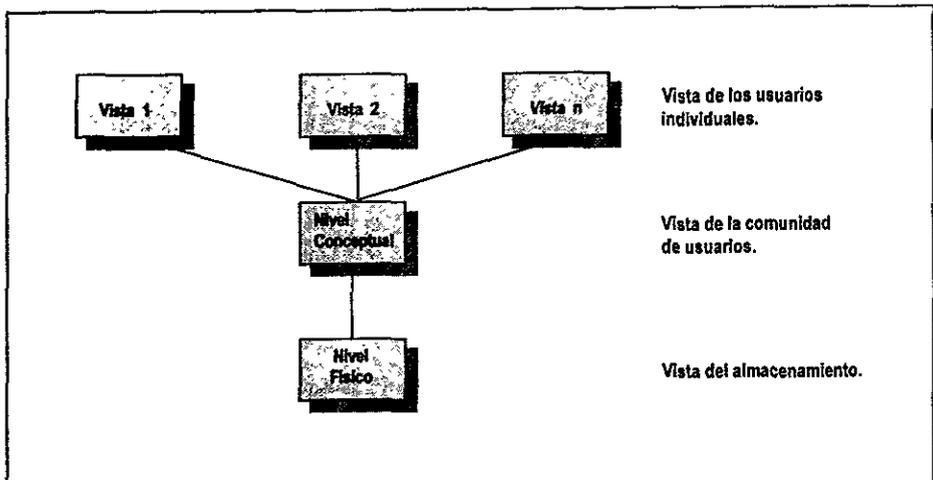


Fig. 1.2 . Niveles de abstracción.

Podemos decir que el nivel conceptual es un nivel de mediación entre los otros dos niveles. Puede considerarse como el que define una vista de la comunidad de usuarios. En otras palabras, habrá muchas vistas "externas", cada una compuesta por una representación abstracta de la base de datos en su totalidad. Así mismo, habrá una sola vista física o interna, que representa la base de datos total, tal y como está almacenada.

~ Independencia de los datos.

Para definir el concepto de independencia de los datos, revisaremos primero dos conceptos ligados al anterior. El primer concepto es el esquema de una base de datos, el cual se define como la descripción lógica de la base de datos, o sea, un diagrama de los tipos de datos que se usan y sus relaciones. El diseño general de la base de datos es un esquema de la misma. Los esquemas muy raras veces se modifican o alteran. El segundo concepto es la instancia de la base de datos, este concepto se define como el conjunto de información almacenado en la base de datos en un momento determinado.

El concepto de independencia de los datos se puede definir como sigue : es la capacidad de modificar una definición de esquema en un nivel, sin afectar la definición del esquema en el nivel inmediato superior. Existen dos niveles de tal independencia :

- Independencia física, es la capacidad de modificar el esquema físico de la base de datos sin obligar a reescribir los programas de aplicaciones. En algunas ocasiones son necesarios los cambios en el nivel físico para mejorar el rendimiento.

- *Independencia lógica*, es la capacidad de modificar el esquema conceptual sin obligar a reescribir los programas de aplicaciones. Las modificaciones en el nivel conceptual son necesarias siempre que se altera la estructura lógica de la base de datos.

Los beneficios que se obtienen al contar con independencia de los datos son los siguientes:

- Hacer cambios a los programas de aplicación y modificaciones a la base de datos es fácil.
- Dado que es fácil hacer cambios, la productividad de los programadores se incrementa.

1.4 Modelos de Datos

Es un grupo de herramientas conceptuales para describir en forma concisa los datos, sus relaciones, su semántica y sus limitantes. Estos modelos básicamente se dividen en tres grupos :

- a) modelos lógicos basados en objetos
- b) modelos lógicos basados en registros
- c) modelos físicos de datos.

1.4.1 Modelos lógicos basados en objetos

Se caracterizan por el hecho de que permiten una estructuración bastante flexible y hacen posible especificar claramente las limitantes de los datos, dentro de los modelos más conocidos de este tipo están :

- 1) modelo entidad-relación
- 2) modelo binario
- 3) modelo *semántico de datos*
- 4) modelo infológico

De estos modelos el más conocido es el modelo entidad-relación ya que ha tenido muy buena aceptación como modelo de datos. Este modelo de datos (E-R) se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades, y de las relaciones entre estos objetos.

Una entidad, es un aspecto importante acerca del cual se necesita tener o conocer información.

Una *relación*, es bidireccional y representa la asociación entre dos entidades, o entre una entidad consigo misma.

Un ejemplo del modelo entidad-relación sería una relación numdept, que asocia a un departamento con cada uno de los empleados que le pertenece. El conjunto de todas las entidades y relaciones del mismo tipo se denomina conjunto de entidades y conjunto de relaciones, respectivamente.

La estructura lógica general de una base de datos puede expresarse gráficamente por medio de un diagrama E-R que consta de los siguientes componentes :

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombos, que representan relaciones entre conjuntos de entidades.
- Líneas, que conectan los atributos a los conjuntos de entidades y los conjuntos de entidades a las relaciones.

En la figura 1.3 se muestra el ejemplo anterior en un diagrama de E-R.

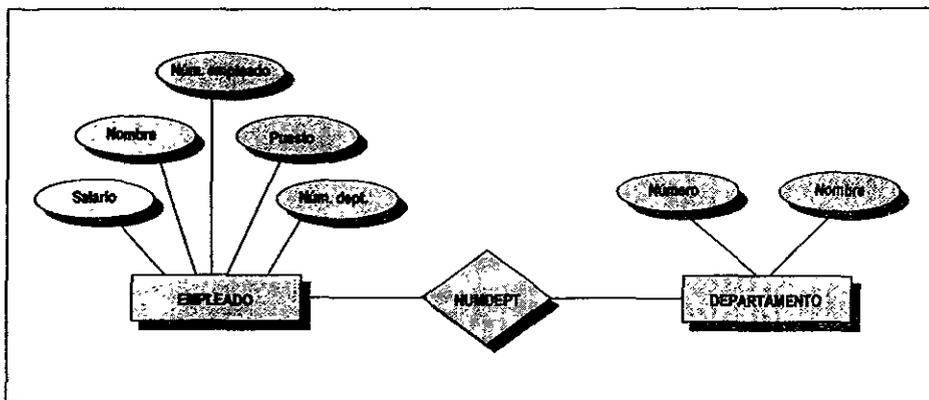


Fig. 1.3. Ejemplo de un diagrama de E-R.

1.4.2 Modelos lógicos basados en registros

Estos modelos se utilizan para describir los datos en los niveles conceptual y de visión. La diferencia entre éstos y los modelos lógicos basados en objetos, es que sirven para especificar la estructura lógica general de la base de datos como una descripción en el nivel más alto de la implantación.

Estos modelos no permiten especificar de manera precisa las limitantes de los datos.

Los modelos que han tenido más aceptación son los siguientes :

1.4.3 Modelo relacional

En el cual los datos y las relaciones entre datos se representan por medio de tablas, cada una de las cuales tiene varias columnas con nombres únicos. Los datos son representados como una

tabla de dos dimensiones. Las tablas están constituidas por columnas y renglones. Para dar un ejemplo, piénsese en una base de datos que incluye a empleados y el departamento al que pertenecen. Una muestra de una base de datos relacional se presenta en la figura 1.4. Esto indica, por ejemplo, que el empleado Jorge Hernández tiene el puesto de jefe de oficina, número de empleado 25, salario de \$8,500 mensuales y pertenece al departamento de sistemas. Nos damos cuenta que un empleado esta asociado a un solo departamento y un departamento tiene uno o más empleados.

Tabla Empleado

Num. empleado	Nombre	Apellido	Puesto	Salario	Núm de dept.
25	Jorge	Hernández	Jefe de Oficina	8500	03
15	Javier	Camarillo	Director	12000	03
20	Jorge	Ortiz	Subdirector	9000	04
21	Pedro	Marrón	Técnico	3000	02
22	José	Ortega	Analista	5000	01

Tabla Departamento

Número	Nombre
01	Finanzas
02	Mantenimiento
03	Sistemas
04	Operación

Fig. 1.4 . Ejemplo de una base de datos relacional.

1.4.4 Modelo de red

Los datos se representan por medio de conjuntos de registros y las relaciones entre los datos se representan por medio de ligas, que se pueden considerar apuntadores. En el modelo de red se permiten relaciones entre entidades 1:1 (uno a uno), 1:N (uno a muchos), M:N (muchos a muchos). La ventaja del modelo de red es que refleja el mundo real y hace más sencillo definir las estructuras de datos requeridos y sus limitantes. Su desventaja es que se pueden volver demasiado complejos. En la figura 1.5 se muestra un ejemplo de una base de datos que contiene la misma información que la de la fig. 1.4.

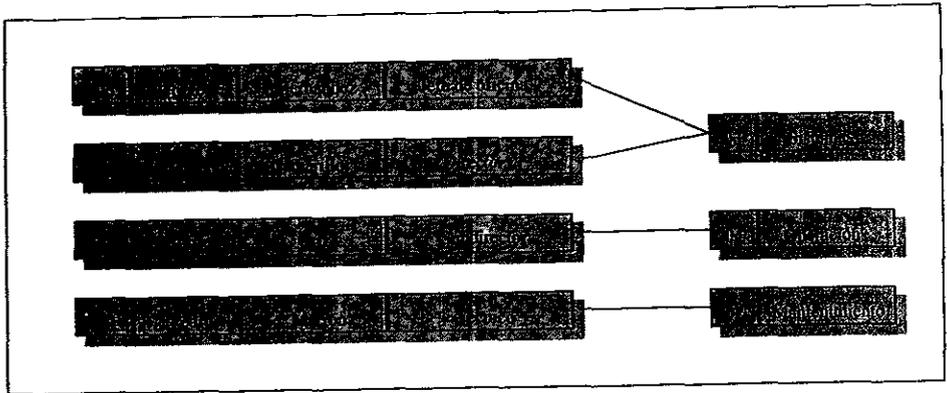


Fig. 1.5 Ejemplo de una base de datos en red.

1.4.5 Modelo jerárquico

En este modelo los datos y las relaciones entre los datos se representan por medio de registros y ligas respectivamente, de la misma forma que en el modelo de red, pero a diferencia de aquel los registros están organizados como un conjunto de árboles en vez de gráficas arbitrarias. Este modelo permite usar relaciones entre los datos de 1:1, 1:N, pero no permite usar relaciones M:N. La ventaja del modelo jerárquico radica en su sencillez y la facilidad de entenderlo. La figura 1.6 nos muestra un ejemplo con los datos de la figura 1.4.

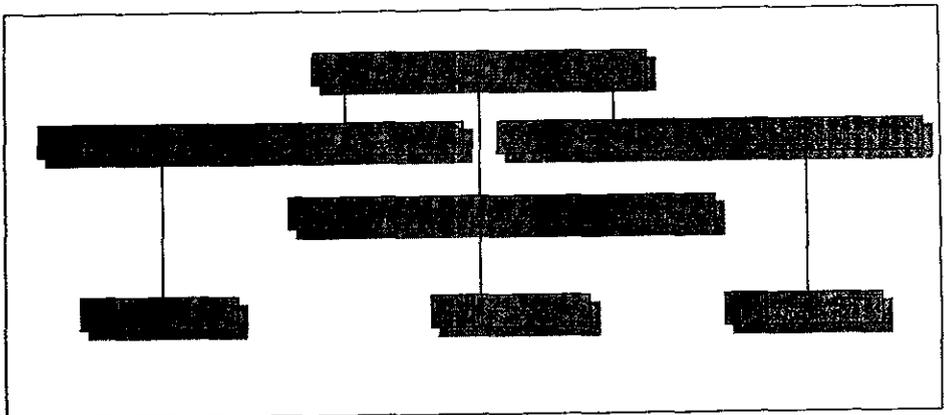


Fig. 1.6 Ejemplo de una base de datos jerárquica.

I.4.6 Modelos físicos de datos.

Los modelos físicos de datos sirven para describir los datos en el nivel más bajo y en la actualidad no son muy utilizados. Algunos de los más conocidos son:

- El modelo unificador.
- La memoria de cuadros.

1.5 Conceptos Generales

I.5.1 Lenguaje de definición de datos

Un esquema de base de datos se especifica por medio de una serie de definiciones que se expresan en un lenguaje especial llamado lenguaje de definición de datos (en inglés: DDL Data Definition Language). El resultado de la compilación de las proposiciones en DDL es un conjunto de tablas que se almacenan en un archivo especial llamado diccionario de datos. La estructura de almacenamiento y los métodos de acceso empleado por el sistema de base de datos también se especifican por medio de este lenguaje de almacenamiento y definición de los datos. El resultado de la compilación de estas definiciones es una serie de instrucciones que especifican los detalles de implantación de los esquemas de base de datos que normalmente no pueden ver los usuarios.

I.5.2 Lenguaje de manejo de datos

Los niveles de abstracción que se mencionaron en la sección anterior no solamente se aplican a la definición o estructuración de los datos, sino también a su manejo de estos; esta manipulación consiste en:

- La recuperación de la información almacenada en la base de datos.
- La inserción de información nueva en la base de datos.
- La eliminación de información de la base de datos.

En el nivel físico deben definirse algoritmos que permitan tener acceso a los datos en forma eficiente. En los niveles de abstracción más altos lo importante es la facilidad de uso. El objetivo es lograr una interacción eficiente entre las personas y el sistema.

Un lenguaje de manejo de datos (en inglés: Data Manipulation Language, DML) permite a los usuarios manejar o tener acceso a los datos que estén organizados por medio del modelo apropiado. Existen dos tipos de DML:

- De procedimientos, necesitan que el usuario especifique cuáles datos quiere y cómo deben obtenerse.
- Sin procedimientos, requieren que el usuario especifique cuáles datos quiere sin especificar como obtenerlos.

Los DML sin procedimientos son, por lo general, más fáciles de aprender y utilizar que los de procedimientos.

Una consulta (en inglés: *querie*) es una proposición que solicita la recuperación de información. La parte de un DML que implica la recuperación de información se conoce como lenguaje de consultas.

I.5.3 Diccionario de datos

Un diccionario de datos es un archivo que contiene “metadatos”, esto es, “datos acerca de los datos”. Este archivo se consulta antes de leer o modificar los datos reales en la base de datos. Un diccionario de datos completo incluye referencias cruzadas que indican, por ejemplo, que parte de datos utiliza cada programa, entre otros. De hecho, el diccionario puede integrarse incluso en la base de datos que describe y por tanto, incluir su propia descripción.

I.5.4 Administrador de la base de datos

El administrador de la base de datos (en inglés: *Data Base Administrator, DBA*) es la persona encargada de mantener el control de los datos que están almacenados en la base de datos. Las funciones del DBA son las siguientes:

- **Definición de esquema**, es decir, la creación del esquema original de la base de datos. Esto se logra escribiendo una serie de definiciones que el compilador DDL traduce a un conjunto de tablas que se almacenan permanentemente en el diccionario de datos.
- **Definición de la estructura de almacenamiento y el método de acceso**, es decir, la creación de las estructuras de almacenamiento y métodos de acceso apropiados. Esto se lleva a cabo escribiendo una serie de definiciones que posteriormente son traducidas por el compilador del lenguaje de almacenamiento y definición de datos.
- **Modificación del esquema y de la organización física**, ya sea la modificación del esquema de la base de datos o de la descripción de la organización física del almacenamiento. Estos cambios, aunque son relativamente poco frecuentes, se logran escribiendo una serie de definiciones utilizadas por el compilador de DDL o por el compilador del lenguaje de almacenamiento y definición de datos para generar modificaciones a las tablas internas apropiadas del sistema.
- **Concesión de autorización para el acceso a los datos**, es decir, conceder diferentes tipos de autorización para acceso a los datos a los distintos usuarios de la base de datos. Esto permite al administrador de la base de datos regular cuales son las partes de la base de datos a las que van a tener acceso diversos usuarios.

- **Especificación de las limitantes de integridad**, estas limitantes se conservan en una estructura especial del sistema que consulta el manejador de base de datos cada vez que se lleva a cabo un actualización en el sistema.
- **Vincularse con los usuarios**, es responsabilidad del DBA vincularse con los usuarios, **garantizar que los datos que requieran estén disponibles**.
- **Definir una estrategia de respaldo y recuperación**, en caso de daño de alguna parte de la base de datos, es esencial poder reparar los datos pertinentes a la mayor brevedad y reduciendo al mínimo las repercusiones en el resto del sistema. El DBA debe definir y poner en marcha una estrategia de respaldo y recuperación adecuada de la información.

I.5.5 Usuarios de la base de datos

Uno de los objetivos primordiales de la base de datos es crear un ambiente para la recuperación de información y para almacenar información nueva en la base de datos. Existen tres tipos diferentes de usuarios en un sistema de base de datos, y se distinguen por el modo en que ellos esperan interactuar con el sistema.

- **Programador de aplicaciones**. Estos usuarios interactúan con el sistema mediante llamadas en DML, las cuales están incrustadas en un programa escrito en un lenguaje huésped. Estos programas se denominan programas de aplicación. Puesto que la sintaxis del DML es por lo común muy diferente de la del lenguaje huésped, las llamadas en DML van precedidas, generalmente de un carácter especial, para que un preprocesador especial, llamado precompilador de DML, pueda generar el código adecuado.
- **Usuarios casuales**. Son usuarios que interactúan con el sistema sin escribir programas. En cambio, escriben sus consultas en un lenguaje de consulta de base de datos. Cada una de las consultas se maneja a través de un procesador de consultas, cuya función es tomar una proposición en DML y descomponerla en instrucciones que pueda entender el DBMS.
- **Usuarios ingenuos**. Son usuarios poco complejos que interactúan con el sistema llamando alguno de los programas de aplicaciones permanentes ya existentes.
- **Usuarios especializados**. Son usuarios complejos que escriben aplicaciones para la base de datos que no embonan en el marco tradicional de procesamiento de datos. Entre dichas aplicaciones se cuentan los sistemas de diseño ayudado por computadora, los sistemas expertos basados en conocimientos y los sistemas que almacenan información con tipos complejos de datos (audio, video, datos gráficos).

1.6 Estructura general de un sistema de base de datos.

Un sistema de base de datos se divide en módulos que se encargan de cada una de las tareas del sistema general. Dicho sistema consiste de varios componentes funcionales, entre los que se cuentan:

- *El manejador de archivos*, encargado de asignar espacio en el disco y de las estructuras de datos que se van a utilizar para representar la información almacenada en disco.
- *El manejador de la base de datos*, que constituye la interface entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y las consultas que se hacen al sistema.
- *El procesador de consultas*, que traduce las proposiciones en lenguaje de consulta a instrucciones de bajo nivel que puede entender el manejador de la base de datos.
- *El precompilador de DML*, que convierte las proposiciones en DML incrustadas en un programa de aplicaciones en llamados normales a procedimientos en el lenguaje huésped. El precompilador debe interactuar con el procesador de consultas para generar el código apropiado.

El compilador de DDL, que convierte las proposiciones en DDL en un conjunto de tablas que contienen metadatos. Dichas tablas se almacenan después en el diccionario de datos. Además se requieren varias estructuras de datos como parte de la implantación del sistema físico, incluyendo:

- *Archivos de datos*, que guardan la base de datos.
- *Diccionario de datos*, que almacenan la información relativa a la estructura de la base de datos. Se utiliza constantemente, por lo que se debe desarrollar e implantar de manera eficiente.
- *Índices*, que permiten el acceso rápido a elementos de información que contienen valores determinados.

La figura 1.7 muestra los componentes antes mencionados y las conexiones entre ellos.

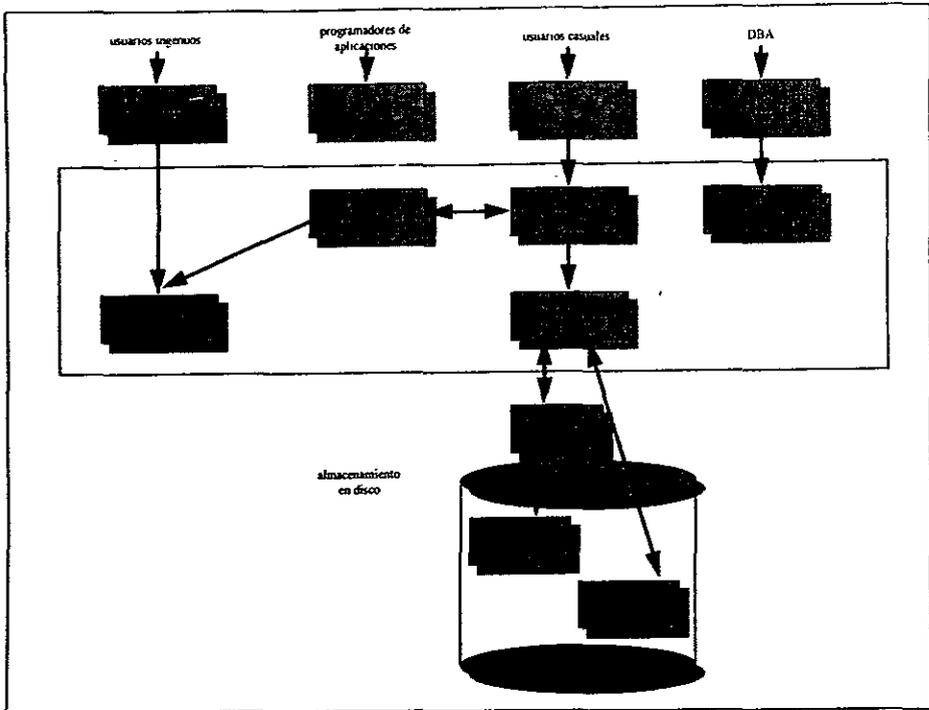


Fig. 1.7. Estructura de un sistema de base de datos.

1.7 El Modelo Entidad Relación

El modelo de datos entidad-relación (E-R) se basa en una percepción de un mundo real, el cual consiste de objetos básicos llamados entidades y de relaciones entre estos objetos.

Una entidad se puede definir como cualquier objeto que existe, es distinguible y se puede representar en la base de datos. Una entidad puede ser un objeto tangible, por ejemplo, un alumno, un artículo o un lugar, pero también puede ser un objeto intangible, tal como un suceso, el número de cuenta de un alumno, un nombre de tarea, etc.

Un conjunto de entidades es un grupo de entidades del mismo tipo, por ejemplo, el conjunto de empleados de una empresa.

No es necesario que las entidades pertenezcan exclusivamente a un solo conjunto. Por ejemplo, es posible definir el conjunto de entidades de todos los empleados de una empresa (empleado) y el conjunto de entidades de todos los clientes de la empresa (cliente). Una entidad *persona* puede ser una entidad *empleado*, una entidad *cliente*, ambas, o ninguna de las dos.

Las entidades se describen o representan por medio de atributos. Los posibles atributos del conjunto de entidades "empleado" son: nombre, número de empleado, dirección y ciudad.

Para cada atributo existe un rango de valores permitidos, llamado dominio del atributo. Por ejemplo, el dominio del atributo número de empleado de la entidad "empleado" podría ser el conjunto de todos los enteros positivos.

Formalmente, un atributo es una función que mapea un conjunto de entidades a un dominio. Así, cada entidad se describe por medio de un conjunto de parejas (atributo, valor del dato), una pareja para cada atributo del conjunto de entidades.

La figura 1.8 muestra parte de una base de datos de una institución bancaria, la cual esta compuesta por dos conjuntos de entidades: cliente y cuenta.

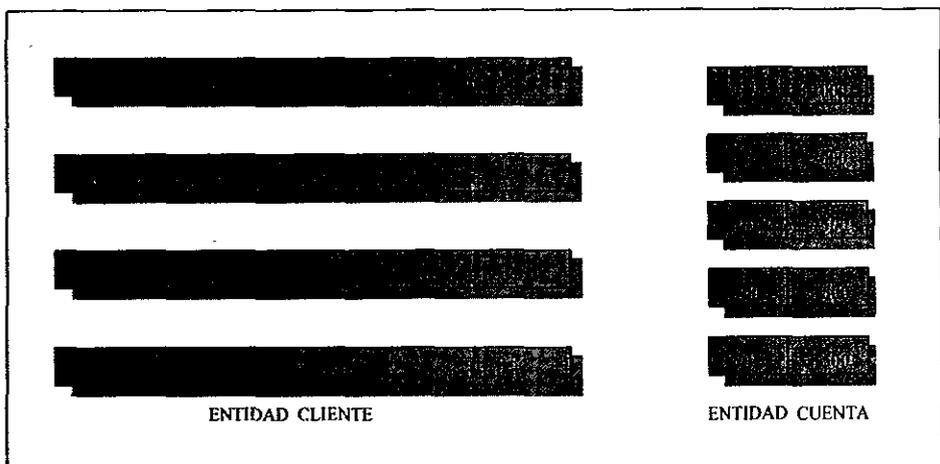


Fig. 1.8. Entidades de cliente y cuenta.

La entidad *cliente* tiene los siguientes atributos: nombre, número de cliente, dirección y ciudad donde vive.

La entidad *cuenta* tiene los siguientes atributos: número de cuenta y saldo.

1.7.1 Relaciones y Conjuntos de Relaciones.

Una relación es una asociación entre varias entidades. Por ejemplo, es posible determinar una relación que asocia al cliente "Hernández" con la cuenta 630.

Un conjunto de relaciones es un grupo de relaciones del mismo tipo. Formalmente es una relación matemática de $n \geq 2$ conjuntos de entidades. Si E_1, E_2, \dots, E_n son un conjunto de entidades, entonces un conjunto de relación R es un subconjunto de

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

donde

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

Para ilustrar lo anterior, considérense dos conjuntos de entidades, clientes y cuenta de la figura 1.8. Se definirá el conjunto de relaciones "ctecta" para denotar la asociación entre los clientes y las cuentas bancarias que tienen. Esta asociación se representa en la figura 1.9

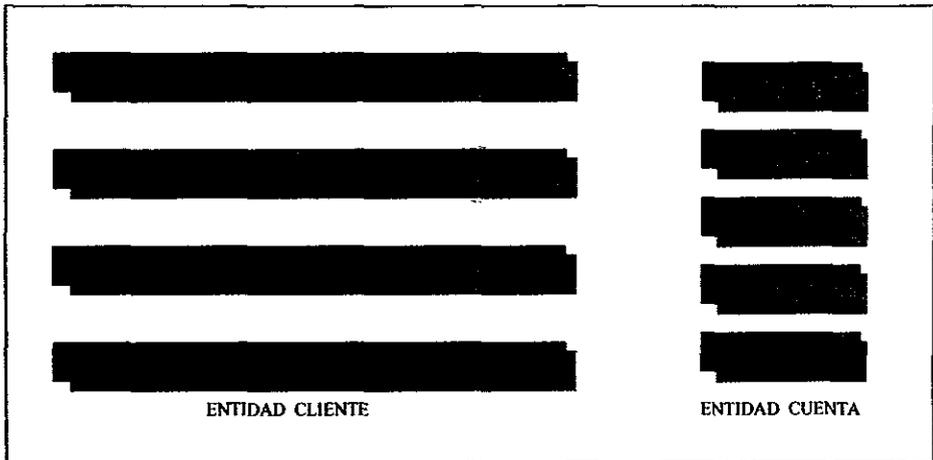


Fig. 1.9. Conjunto de relaciones entre las entidades cliente y cuenta.

Esta relación es un ejemplo de una relación binaria, es decir, que involucra a dos conjuntos de entidades. La mayor parte de las relaciones en una base de datos son binarias, pero en ocasiones existen conjuntos de relaciones que incluyen a más de dos conjuntos de entidades.

Las relaciones también pueden tener atributos descriptivos. Por ejemplo: la fecha podría ser un atributo descriptivo del conjunto de relaciones “cuenta”. Esto podría especificar la última vez que el cliente tuvo un acceso a su cuenta.

Para ilustrar esto, podemos considerar la relación (Hernández, 401) se puede describir con (fecha, 13/octubre/1996).

1.7.2 Limitantes de Mapeo

El modelo entidad relación puede definir ciertas limitantes con las que deben cumplir los datos contenidos en la base de datos. La cardinalidad de mapeo es una limitante que expresa el número de entidades con las que puede asociarse otra entidad mediante una relación.

Para ilustrar esta limitante utilizaremos los conjuntos de relaciones binarias, aunque pueden existir relaciones del tipo n -arias. Para un conjunto binario de relaciones R entre los conjuntos de entidades A y B , la cardinalidad de mapeo debe ser una de las siguientes:

- Una a una, en donde una entidad en A está asociada con sólo una entidad en B y, una entidad en B está asociada con sólo una entidad en A . (Ver figura 1.10)

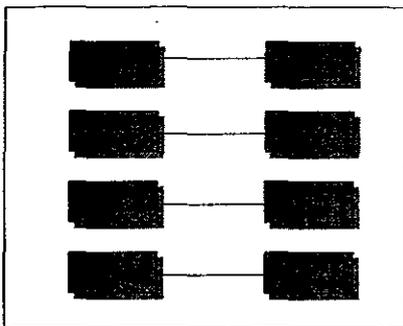


Fig. 1.10. Relación una a una.

Una a muchas, en donde una entidad en A está asociada con cualquier número de entidades en B , pero una entidad en B sólo puede estar asociada con una entidad en A . (Ver figura 1.11)

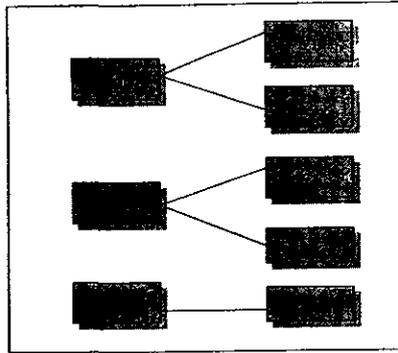


Fig. 1.11 Relación una a muchas.

Para ilustrar mejor esta relación, considérese el conjunto de relaciones "cliente", una cuenta puede pertenecer únicamente a un cliente, y un cliente puede tener varias cuentas, entonces el conjunto de relaciones es una a muchas de cliente a cuenta. Si una cuenta puede pertenecer a varios cuentahabientes (cuentas mancomunadas), el conjunto de relaciones es muchas a muchas.

- Muchas a una, una entidad A está relacionada únicamente con una entidad en B, pero una entidad en B puede relacionarse con cualquier número de entidades en A. (Ver figura 1.12)

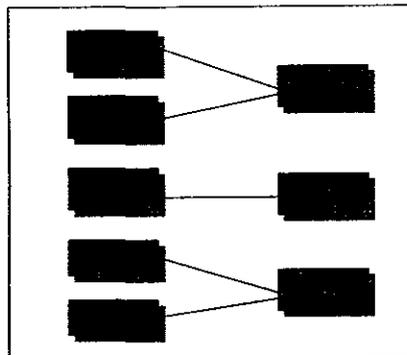


Fig. 1.12 Relación muchas a muchas.

- Muchas a muchas, en donde una entidad en A está asociada con cualquier número de entidades en B, y cualquier entidad en B puede estar asociada con cualquier número de entidades en A. (Ver figura 1.13)

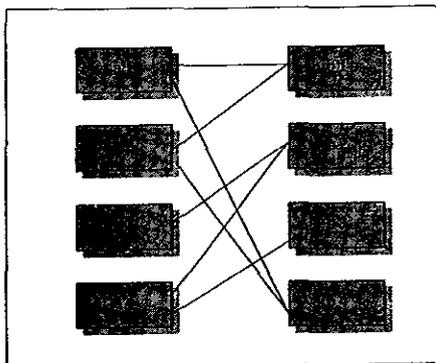


Fig. 1.13 Relación muchas a muchas.

Las dependencias de existencia constituyen otra clase de limitantes. Específicamente si la existencia de una entidad X depende de la entidad Z, se dice que X es dependiente *por existencia* de Z. Esto quiere decir que si se elimina Z, también se elimina X. Se dice que la entidad Z es dominante y que la entidad X es una entidad subordinada.

Para ilustrar lo anterior, piénsese en los conjuntos de entidades cuenta y transacción. Se forma una relación bitácora entre estos dos conjuntos que especifica que para una cuenta determinada pueden existir varias transacciones. Esta relación es una a muchas de cuenta a transacción. Cada entidad transacción debe estar relacionada con una entidad cuenta. Si se elimina una entidad cuenta, entonces deben eliminarse también todas las entidades transacción vinculadas con cuenta. Por el contrario, pueden eliminarse entidades transacción de la base de datos sin afectar ninguna cuenta. Por tanto, el conjunto de entidades cuenta es dominante y transacción es subordinada en la relación bitácora.

I.7.3 Llaves primarias.

Una tarea importante al modelar bases de datos es especificar como se van a distinguir las entidades y las relaciones.

La diferencia entre entidades y relaciones desde el punto de vista de bases de datos debe expresarse en términos de sus atributos.

Para realizar estas distinciones, se asigna una superllave a cada conjunto de entidades. La superllave es un conjunto de uno o más atributos, que juntos permiten identificar en forma única a una entidad dentro del conjunto de entidades.

El concepto de superllave no es suficiente para el modelado de la base de datos, ya que una superllave puede incluir atributos ajenos, esto es, si k es una superllave, entonces cualquier subconjunto lo sería. Lo óptimo es buscar que la superllave sea lo más pequeña posible, en donde ningún subconjunto propio sea una superllave. A estas superllaves se les llama llaves candidato. El

término llave primaria se utiliza para referirse a la llave candidato con la cual se identificará unívocamente a las entidades dentro de un conjunto de éstas.

Es posible que un conjunto de entidades no tenga suficientes atributos para formar una llave primaria. Por ejemplo, el conjunto de entidades transacción tiene tres atributos : número transacción, fecha e importe. Aunque cada entidad transacción es distinta, dos transacciones hechas en cuentas diferentes pueden tener el mismo número de transacción. Así, el conjunto de entidades no cuenta con una llave primaria. Una entidad de un conjunto de este tipo se denomina entidad débil. Una entidad que cuenta con una llave primaria recibe el nombre de entidad fuerte.

El concepto de entidades fuertes y débiles está relacionado con el de “dependencia por existencia” que se vio en el capítulo anterior. Una entidad fuerte es por definición, dominante, mientras que una entidad débil es subordinada.

Un conjunto de entidades débiles no cuenta con una llave primaria . Sin embargo, es preciso tener una forma de distinguir entre todas esas entidades aquellas que dependen de una entidad fuerte determinada. El discriminador de un conjunto de entidades débiles es un conjunto de atributos que permite hacer esta distinción. Por ejemplo, el discriminador del conjunto de entidades débiles transacción es el atributo número-transacción, ya que para cada cuenta estos números de transacción identifican en forma única cada una de las transacciones.

La llave primaria de un conjunto de entidades débiles está formada por la llave primaria de la entidad fuerte de la que dependen su existencia y su discriminador. En el caso del conjunto de entidades transacción, su llave primaria es (número-cuenta, número-transacción), donde número-cuenta identifica a la entidad dominante de una transacción, y número-transacción distingue a las entidades transacción dentro de la misma cuenta.

Los conjuntos de relaciones también tienen llaves primarias. Sus llaves primarias se forman tomando todos los atributos que constituyen las llaves primarias de los conjuntos de entidades que definen al conjunto de relaciones.

1.7.4 Diagrama Entidad-Relación (E-R)

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama entidad-relación, el cual consiste de los siguientes componentes:

- Rectángulos. Representan conjuntos de entidades.
- Elipses. Representan atributos.
- Rombos. Representan conjuntos de relaciones.
- Líneas. Conectan los atributos a los conjuntos de entidades y los conjuntos de entidades a los conjuntos de relaciones.

Por ejemplo, la figura 1.14 nos muestra el diagrama de entidad-relación que consiste en dos conjuntos de entidades, cliente y cuenta, relacionados entre si mediante un conjunto binario de relaciones ctecta. Los atributos asociados con cliente son número-cta, seguro-social, calle y ciudad-cliente. Los atributos relacionados con cuenta con número-cta y saldo.

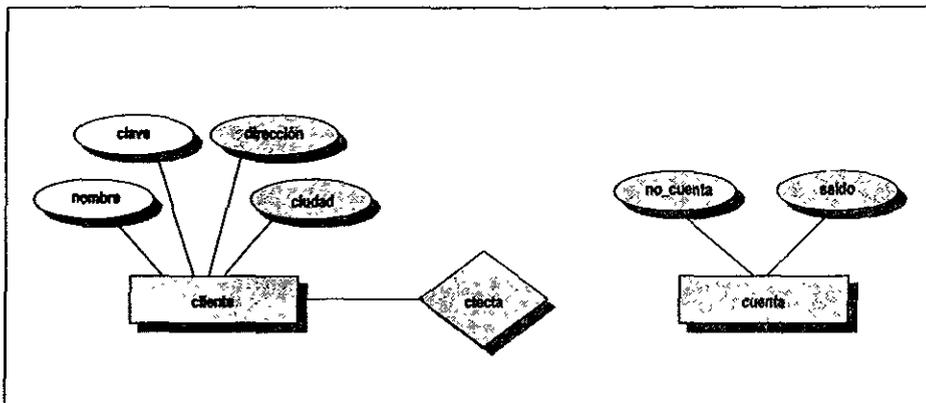


Fig. 1.14 Diagrama de Entidad-Relación.

El conjunto de relaciones “cuenta” puede ser muchas a muchas, una a muchas, muchas a una o una a una. En las figuras xx,yy y zz se ven como se dibujan estas relaciones.

Los diagramas de entidad-relación pueden representarse por medio de un conjunto de tablas. Para cada conjunto de entidades y de relaciones en la base de datos, existe una tabla única que recibe el nombre del conjunto de entidades o relaciones correspondientes. Cada tabla tiene un número de columnas las cuales también tienen un nombre único. Utilizando la figura 1.14 obtenemos lo siguiente:

NOMBRE	CLAVE	DIRECCIÓN	CIUDAD
Ortiz	100345	Taine #56	Distrito Federal
Hernández	234577	Guerrero #343	Tijuana
Martínez	784532	Taine #1987	Distrito Federal
Pedroza	213379	Edison #345	Guadalajara

ENTIDAD CLIENTE

Nº CUENTA	SALDO
259	1000
630	2000
401	1500
701	1500

ENTIDAD CUENTA

Fig. 1.15. Las tablas cliente y cuenta

1.7.5 El Enfoque Relacional

Una base de datos relacional consiste de un conjunto de tablas. La figura 1.16 muestra un pequeño ejemplo de una base de datos relacional.

Empleado

Nombre	Edad	Nº depto.
Torres	41	7
Strevel	31	7
Hernández	19	4

Departamento

Nº depto.	Gerente	Teléfono
7	Torres	2938
4	Sánchez	4068

Fig. 1.16. Una base de datos relacional.

Una columna de una tabla representa una relación entre un conjunto de valores. Puesto que una tabla es un conjunto de estas relaciones, existe una correspondencia entre el concepto de tabla y el concepto matemático de relación, del cual recibe su nombre el modelo de datos relacional.

Una de las razones por la cual el modelo de datos relacional es ampliamente aceptado es porque está basado en un modelo conceptual muy fácil de entender. Este modelo se puede resumir como sigue, una base de datos relacional consiste de tablas. Cada tabla contiene el nombre de una relación y contiene renglones y columnas. Podemos definir un esquema de relaciones como un conjunto de nombres de atributos para una relación. Se pueden crear nuevas tablas a partir de las ya existentes. El proceso de construir nuevas tablas en el modelo relacional está gobernado por las operaciones del álgebra relacional.

En bases de datos relacionales a las tablas se les denomina relaciones, los renglones de las tablas se denominan tuplas y a las columnas se les denomina atributos. Para ilustrar esto examinemos la figura 1.17 en donde vemos representados los términos anteriores.

Nº cuenta	Nombre alumno	Dirección	Edad	Sexo
8340258-8	Strevel	Guerrero # 343	26	M
8249486-5	Ordaz	Luis Enrique Erro # 54	30	M
8721894-0	Soto	Recife # 705	24	F
8234569-9	Toledo	Samuel # 32	26	F
9076298-1	González	Bosques de Asia # 13	21	M
8834219-4	Del Razo	Rivero #21 int. 4	23	M
8543256-2	Grace	Zarco # 345	22	F

Fig. 1.17. Relación alumno.

La tabla de la figura 1.17 tiene cinco atributos, *no_cuenta*, *nombre_alumno*, *dirección*, *edad* y *sexo*. Para cada atributo existe un conjunto de valores permitidos, llamado dominio. También tenemos una relación llamada "alumno" y cuenta con siete tuplas o renglones.

Para el atributo *no_cuenta*, el dominio sería el conjunto de todos los números de cuenta existentes. Sea *D1* este conjunto y sea *D2* el conjunto de todos los nombres de alumnos, *D3* el conjunto de todas las direcciones, *D4* el conjunto de todas las edades y *D5* el conjunto del tipo de sexo. Cada una de las columnas de la relación "alumno" debe componerse de una tupla de cinco (*V1*, *V2*, *V3*, *V4*, *V5*) donde *V1* es un número de cuenta, *V2* es un nombre de alumno, *V3* es una dirección, *V4* es una edad y *V5* es un tipo de sexo.

En general, la relación alumno va a contener solamente un subconjunto del conjunto de todas las columnas posibles. Por tanto, la relación alumno es un subconjunto de

$$\prod_{i=1}^5 X_i$$

En general, una tabla de *n* columnas debe ser un subconjunto de

$$\prod_{i=1}^n X_i$$

En la teoría matemática se define a una relación como un subconjunto de un producto cartesiano de un listado de dominios. Esto es similar a la definición de tablas que hemos visto, la diferencia es que nosotros asignamos nombres a los atributos y la teoría matemática se basa en nombres numéricos, usando el entero 1 para denotar el atributo cuyo dominio aparece primero en el listado de dominios, 2 para el siguiente y así sucesivamente. Debido a que las tablas son básicamente relaciones, se utilizan los términos matemáticos relación y tupla en vez de los términos tabla y columna.

En la relación alumno de la figura 1.17 hay siete tuplas. Sea la variable de tupla *t* la primera tupla de la relación. Se utilizará la notación *t*[*no_cuenta*] para denotar el valor en *t* del atributo *no_cuenta*. Así, *t*[*no_cuenta*] = 8340258-8. De manera similar, *t*[*nombre_alumno*] expresa el valor en *t* del atributo *nombre_alumno*, etc. Otra forma de notación sería escribir *t* [1] para indicar el valor en la tupla *t* del primer atributo (*no_cuenta*), *t* [2] para *nombre_alumno*, etc. Puesto que una relación es un conjunto de tuplas, se utiliza la notación matemática $t \in r$ para señalar que la tupla *t* esta en la relación *r*.

Cuando se habla de una base de datos, debe diferenciarse entre el esquema de la base de datos; es decir, el diseño lógico de la base de datos, y una instancia de la base de datos, que se constituye con la información contenida, en la base de datos en cierto momento. Así, una variable en los lenguajes de programación corresponde al concepto de una instancia de una relación.

El Esquema alumno se utiliza para indicar el esquema de la relación alumno. De esta forma, *Esquema_alumno* = { *no_cuenta*, *nombre_alumno*, *dirección*, *edad*, *sexo* }.

El esquema de una relación es una lista de atributos y sus correspondientes dominios. El hecho de que alumno es una relación con el esquema Esquema_alumno, se expresa al escribir Alumno (esquema_alumno).

Si se desea definir los dominios se utiliza la notación:
(no_cuenta:cadena, nombre_alumno:cadena, dirección:cadena, edad:entero, sexo:cadena),
para definir el esquema de relación de la relación alumno.

Una característica importante de la estructura de datos relacional es que las asociaciones entre tuplas se representan únicamente por valores de datos en columnas sacadas de un dominio común.

Ocupar atributos comunes es una forma de relacionar las tuplas de relaciones distintas.

I.7.6 Lenguajes de consulta formales.

Los lenguajes de consulta se utilizan para que el usuario obtenga información de la base de datos. Estos lenguajes se pueden clasificar en *lenguajes de procedimientos* o *sin procedimientos*.

En un lenguaje de procedimientos el usuario le ordena al sistema que realice una serie de operaciones con la base de datos para obtener el resultado deseado, en cambio, en un lenguaje sin procedimientos el usuario describe la información que desea sin indicar un procedimiento específico para obtenerla.

La mayor parte de los sistemas comerciales de bases de datos ofrece un lenguaje de consulta que incluye elementos de los dos enfoques : con procedimientos y *sin procedimientos*. En el caso de Oracle contiene los dos enfoques.

Veremos los conceptos básicos del álgebra y cálculo relacional y algunos lenguajes de consulta comerciales como SQL, QUEL.

I.7.7 El Álgebra Relacional

El álgebra relacional es un lenguaje de procedimientos el cual consiste en un conjunto de operaciones sobre las relaciones. Cada operación toma una o más relaciones como su(s) operando(s) y produce otra relación como resultado.

Este lenguaje es de procedimientos. Existen cinco operaciones fundamentales en el álgebra relacional que son: selección, proyección, producto cartesiano, unión y diferencia de conjuntos. Todas estas operaciones producen como resultado otra relación.

Las operaciones de selección y proyección se llaman operaciones unitarias, ya que actúan sobre una sola relación, las otras tres operaciones se efectúan sobre parejas de relaciones por lo que se llaman operaciones binarias.

Examinaremos las operaciones básicas del álgebra relacional, utilizando para esto la relación cliente_dirección que se muestra a continuación:

Nombre	#cliente	Calle	Ciudad
Martínez	1712	del sapo	DF
Ortiz	2487	rosa	DF
Olvera	1464	laurel	DF
Camarillo	3577	jacarandas	GUA
Hernández	2667	parmi	GUA

Fig. 1.18. Relación cliente_dirección

II.1 Concepto

Montar un puzzle puede llegar a ser una tarea difícil (incluso si todas las piezas están dadas la vuelta). Y aunque montar un puzzle con las piezas al revés es una tarea angustiosamente sencilla para algunos niños de cuatro años, la mayoría de las personas lo consiguen lentamente. Cada pieza se considera en relación con sus vecinas, en lugar de como parte de una figura más grande.

En este capítulo se presenta la gran figura de la arquitectura Oracle. También se proporcionan los detalles sobre sus piezas y sobre los conceptos de implementación básicos que dirigen su utilización. Se describen las estructuras físicas, de la memoria y de los procesos, que proporcionan datos a los usuarios. Para administrar una base de datos Oracle es necesario conocer la forma en que interaccionan estas distintas estructuras, como encajan en la gran figura y la mejor manera de personalizar el sistema para ajustarse a sus necesidades.

Es necesario entender dos conceptos básicos para sacar algún provecho de la arquitectura Oracle: bases de datos e instancias. En las dos secciones siguientes se dan descripciones de ambos conceptos y de su implementación en Oracle.

II.2 Bases de Datos Relacionales

Una *base de datos* Oracle es un conjunto de datos. Oracle proporciona la capacidad de almacenar y acceder a estos datos de forma consecuente con un modelo definido conocido como *Modelo relacional (Relational Model)*. Por ello, Oracle se conoce como un sistema de gestión de bases de datos relacionales (**RDBMS: Relational Database Management System**). La mayoría de las referencias a una «base de datos» no se refieren sólo a los datos físicos, sino también a la combinación de objetos físicos, de memoria y de proceso que se describen en este capítulo.

Los datos de una base de datos se almacenan en tablas. Las tablas relacionales se definen mediante sus *columnas*, y tienen un nombre. Los datos se almacenan como *filas* de la tabla. Las tablas

pueden estar relacionadas entre ellas, y puede utilizarse la base de datos para imponer estas relaciones. En la figura 2.1 se muestra un ejemplo de estructura de una tabla.

Una base de datos Oracle almacena sus datos en archivos. Internamente, existen estructuras de la base de datos que proporcionan una asignación lógica de los datos con los archivos, lo que permite almacenar de forma separada diferentes tipos de datos. Estas divisiones lógicas se llaman espacios de tablas.

En las dos subsecciones siguientes se describen los espacios de tablas y los archivos.

II.2.1 Espacios de tablas (tablespaces)

Un *espacio de tablas* es una división lógica de una base de datos. Cada base de datos tiene al menos un espacio de tablas (llamado espacio de tablas *SYSTEM*). Pueden utilizarse otros espacios de tablas para agrupar los usuarios o aplicaciones para facilitar el mantenimiento y mejorar el rendimiento. Ejemplos de espacios de tablas de este tipo serían *USERS* de uso general y *RBS* para los segmentos de rollback (anulación) (que se describirán en un apartado posterior de esta sección).

Un espacio de tablas sólo puede pertenecer a una base de datos.

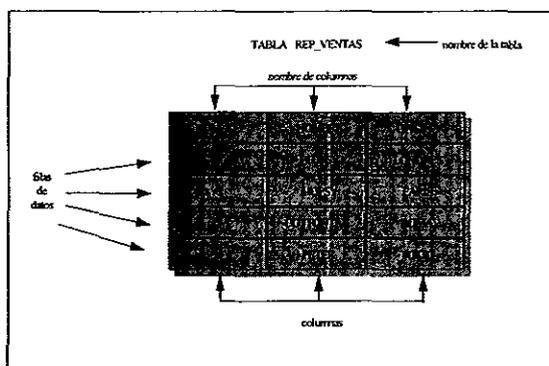


Fig. 2.1. Ejemplo de la estructura de una tabla.

II.2.2 Archivos

Cada espacio de tablas consta de uno o más archivos, llamados *archivos de datos*, en un disco. Un archivo de datos sólo puede pertenecer a un único espacio de tablas. Los archivos de datos tienen un tamaño fijo en el momento de su creación; cuando se necesita más espacio hay que añadir archivos nuevos. Para crear nuevos espacios de tablas hace falta crear nuevos archivos de datos.

Así, la división de los objetos de base de datos entre varios espacios de datos separados, que pueden situarse en discos distintos. Esta es una herramienta importante en la planificación y ajuste

de la manera en que la base de datos maneja las solicitudes de E/S que se le hacen. La relación entre espacios de tablas y archivos de datos se ilustra en la Figura 2.2.

II.3 Instancias

Para acceder a la información de la base de datos, Oracle utiliza un conjunto de procesos de fondo compartidos por todos los usuarios. Además, existen ciertas estructuras de memoria que se utilizan para almacenar la información de la base de datos más consultados recientemente. Estas áreas de memoria ayudan a mejorar el rendimiento de la base de datos al disminuir la cantidad de operaciones de E/S de acceso a los archivos de datos.

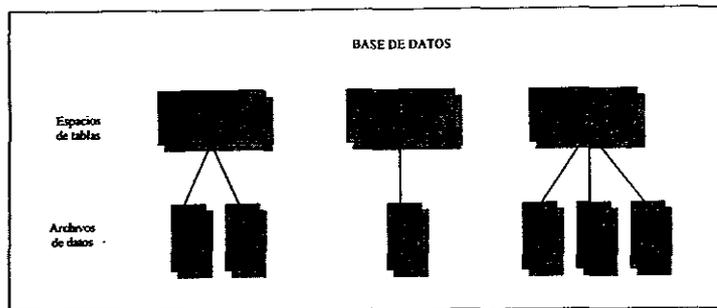


Fig. 2.2. Relación entre bases de datos, espacios de tablas y archivos de datos.

Una *instancia* de base de datos (también conocida como un *servidor*) es un conjunto de estructuras de memoria y de procesos de fondo que acceden a un conjunto de archivos de base de datos. Es posible que varias instancias accedan a una única base de datos (se trata de la opción de servidor paralelo). En la Figura 2.3 se ilustra la relación entre instancias y bases de datos.

Los parámetros que determinan el tamaño y composición de una instancia se almacenan en un archivo llamado INIT.ORA. Este archivo se lee durante el arranque de la instancia y el DBA puede modificarlo. Las modificaciones que se realicen en este archivo no surten efecto hasta el siguiente arranque que lo utilice.

II.4 Estructuras internas de las bases de datos

Partiendo de esta introducción a las bases de datos e instancias, las estructuras de base de datos de Oracle pueden dividirse en tres categorías:

- Las internas a la base de datos (tales como las tablas).
- Las internas a las áreas de memoria (incluidas las áreas de memoria compartida y los procesos).

- Las externas a la base de datos.

En las siguientes secciones se darán las descripciones de cada uno de los elementos dentro de cada categoría. Las categorías se presentarán en el orden indicado anteriormente.

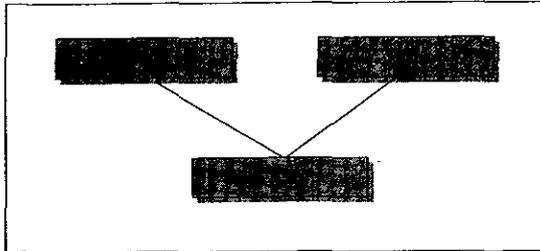


Fig. 2.3. Instancias y bases de datos en Oracle.

En primer lugar se describirán los elementos internos a la base de datos, entre los que se incluyen:

- Tablas, columnas, tipos de datos (*datatypes*) y restricciones.
- Usuarios y esquemas.
- Índices, grupos y grupos hash.
- Vistas.
- Secuencias.
- Procedimientos, funciones, paquetes y disparadores (*triggers*).
- Sinónimos.
- Privilegios y funciones(*roles*).
- Enlaces de base de datos.
- Segmentos, extensiones y bloques.
- Segmentos de rollback (anulación).

II.4.1 Tablas y columnas

Las tablas son el mecanismo de almacenamiento de la información dentro de una base de datos Oracle. Como se muestra en la Tabla 2.1, constan de un conjunto fijo de columnas, que describen los atributos de la entidad que se maneja mediante la tabla. Cada columna tiene un nombre y unas características específicas.

Las características de una columna se componen de dos partes: su *tipo de datos* y su *longitud*. En las columnas con el tipo de datos NUMBER pueden utilizarse las características adicionales de precisión y escala. La *precisión* determina el número de cifras significativas de un valor numérico, mientras que la *escala* determina la colocación de la coma decimal. Una especificación de NUMBER (9,2) para una columna indica que presenta un total de nueve cifras, con dos decimales. Tanto en Oracle Versión 6 como en Oracle 7, la precisión por omisión es de 38 cifras, que es además la precisión máxima.

En la figura 2.4 se enumeran los tipos de datos disponibles en Oracle Versión 6.

En la figura 2.5 se enumeran los tipos de datos disponibles en Oracle 7.

Las tablas propiedad del usuario SYS se llaman tablas del diccionario de datos (*data dictionary*) y proporcionan un catálogo del sistema que la base de datos utiliza para gestionarse a sí misma.

Las tablas se relacionan entre ellas mediante las columnas que tienen en común. Puede utilizarse la base de datos para imponer estas relaciones por medio de la integridad referencial (*referential integrity*). En Oracle 7, la integridad referencial se impone al nivel de la base de datos por medio de las restricciones de las tablas.

CHAR	Campo de longitud variable, con una longitud máxima de 255 caracteres.
DATE	Campo con una longitud de siete bytes que se utiliza para almacenar todas las fechas. La hora se almacena como parte de la fecha. En las consultas, la fecha aparece en el formato Día-Mes-Año, como por ejemplo, 13-APR-94 para indicar el 13 de abril de 1994.
NUMBER	Columna numérica de longitud variable. Los valores permitidos son cero y los números positivos y negativos desde 1.0E-129 hasta 9.99..E124.
LONG	Campo de longitud variable, con una longitud máxima de 65.535 caracteres.
RAW	Campo de longitud variable que se utiliza para datos binarios, con una longitud máxima de 255 caracteres.
LONG RAW	Campo de longitud variable que se utiliza para datos binarios, con una longitud máxima de 65.535 caracteres.

Fig. 2.4. Tipos de datos en Oracle Versión 6

CHAR	Campo de <i>longitud fija</i> , con una longitud máxima de 255 caracteres.
VARCHAR	En la actualidad, idéntico a VARCHAR2, aunque su funcionalidad puede cambiar en versiones futuras. Por tanto, debe utilizarse el tipo de datos VARCHAR2 para almacenar cadenas de caracteres de longitud variable.
VARCHAR2	Campo de longitud variable, con una longitud máxima de 2.000 caracteres.
DATE	Campo con una longitud de siete bytes que se utiliza para almacenar todas las fechas. La hora se almacena como parte de la fecha. En las consultas, la fecha aparece en el formato Día-Mes-Año, como, por ejemplo, 13-APR-94 para indicar el 13 de abril de 1994.
NUMBER	Columna numérica de longitud variable. Los valores permitidos son cero y los números positivos y negativos desde 1.0E-130 hasta 9.99..E125.
LONG	Campo de longitud variable, con un longitud máxima de 2 Gb.
RAW	Campo de longitud variable que se utiliza para datos binarios, con una longitud máxima de 2.000 caracteres.
LONG RAW	Campo de longitud variable que se utiliza para datos binarios, con una longitud máxima de 2 Gb.
MLSLABEL	Sólo para Trusted Oracle. este tipo de datos utiliza entre dos y cinco bytes por fila.

Fig. 2.5 Tipos de datos en Oracle7

Restricciones de las tablas

Sobre una tabla pueden imponerse *restricciones (constraints)*; en este caso, cada fila de la tabla debe satisfacer las condiciones especificadas en las cláusulas de la restricción. Supongamos la orden CREATE TABLE siguiente, que crea una tabla llamada EMPLEADO.

```
CREATE TABLE empleado
(numemp      NUMBER (10)      PRIMARY KEY,
nombre       CHAR (40)        NOT NULL,
numdept      NUMBER (2)        DEFAULT 10,
sueldo       NUMBER (7,2)      CHECK sueldo<1000000,
fecha_nac    DATE,
num_seg_soc  CHAR (9)         UNIQUE,
FOREIGN KEY (numdept) REFERENCES dept.numdept)
TABLESPACE usuarios;
```

Se aprecia, en primer lugar, que a la tabla se le asigna un nombre (EMPLEADO). Todas sus columnas tienen un nombre («NUMEMP», «NOMBRE», etc.). Cada columna tiene un tipo de datos y longitud especificados. La columna «NUMEMP» viene especificada como de tipo NUMBER, sin escala (equivalente a un entero). La columna «NOMBRE» viene especificada como CHAR(40); en Oracle Versión 6, sería una columna de longitud variable de hasta 40 caracteres. En Oracle7, es una columna de longitud fija; los valores con una longitud inferior a 40 caracteres se rellenarán con espacios hasta alcanzar dicha longitud.

La *clave primaria (primary key)* de la tabla es la columna o conjunto de columnas que hacen que cada fila de dicha tabla sea exclusiva. Una columna se identifica como clave primaria por estar definida dentro de la base de datos como *NOT NULL (NO NULA)* (lo que significa que todas las filas almacenadas en esa tabla deben tener un valor en dicha columna); no pueden dejarse en blanco (valor NULL). La restricción NOT NULL puede aplicarse a las columnas de la tabla, como en la columna NOMBRE del ejemplo anterior.

Una columna de una base de datos Oracle7 puede tener una restricción *DEFAULT (POR OMISIÓN)*, que indica que se generará un valor para dicha columna cuando se introduzca una fila en la tabla pero no se especifique ningún valor para ella. La restricción *CHECK (COMPROBACIÓN)* también es exclusiva de Oracle7; sirve para asegurar que los valores de una columna en concreto cumplan cierto criterio (en este caso, que el valor de la columna SUELDO sea menor de 1.000.000).

Existe otra restricción en Oracle7, *UNIQUE (EXCLUSIVO)*, que se utiliza para especificar la exclusividad de columnas pero no forman parte de la clave primaria. En este ejemplo, la columna «NUM_SEG_SOC» tiene una restricción UNIQUE, lo que implica que todos los registros de esta tabla deben tener un valor exclusivo en esta columna.

Una restricción de *clave externa (foreign key)* sirve para especificar la naturaleza de la relación entre tablas. Una clave externa de una tabla hace referencia a una clave primaria definida con anterioridad en cualquier otro lugar de la base de datos.

Por ejemplo, si otra tabla, llamada DEPT, tuviera una clave primaria denominada NUMDEPT, entonces los registros de esa tabla contendrían todos los valores válidos de NUMDEPT. La columna NUMDEPT de la tabla EMPLEADO del ejemplo *hace referencia* a dicha columna DEPT.NUMDEPT. Al especificar EMPLEADO.NUMDEPT como clave externa a DEPT.NUMDEPT, se garantiza que no se pueden introducir valores de NUMDEPT en la tabla EMPLEADO a menos que dichos valores ya existan en la tabla DEPT.

Las claves externas y primarias pueden definirse tanto en Oracle7 como en Oracle Versión 6. No obstante, la aplicación de estas restricciones ha cambiado entre las dos versiones. En Oracle7, la restricción se aplica a cada columna, de forma que todas las relaciones se mantienen al nivel de los datos, mientras que en Oracle Versión 6, no se aplica al nivel de los datos; la única aplicación de la restricción consiste en que una tabla que tenga una restricción de clave primaria no puede eliminarse si otras claves externas de la base de datos hacen referencia a dicha clave primaria k.

En Oracle7, las restricciones en la base de datos ayudan a asegurar la *integridad referencial* (*referential integrity*) de los datos, lo que proporciona la seguridad de que todas las referencias dentro de la base de datos son válidas y que se cumplen todas las restricciones.

II.4.2 Usuarios y Esquemas

Una cuenta de usuario no es una estructura física de la base de datos, pero sí que tiene importantes relaciones con los objetos de la base de datos: los usuarios son propietarios de los objetos de la base de datos. El usuario SYS es propietario de las *tablas del diccionario de datos* (*data dictionary tables*); en ellas se almacena información sobre el resto de las estructuras de la base de datos. El usuario SYSTEM es propietario de las vistas que permiten acceder a estas tablas del diccionario de datos, para que las utilicen el resto de los usuarios de la base de datos.

Los objetos que se crean en la base de datos como apoyo a las aplicaciones, se crean bajo cuentas de usuario. Cada una de estas cuentas puede personalizarse para que utilice un espacio de tablas específico como su espacio de tablas por omisión.

Las cuentas de la base de datos pueden conectarse a una cuenta del sistema operativo, lo que permite a los usuarios acceder a la base de datos desde el sistema operativo. Pueden entonces acceder a los objetos de su propiedad o a aquéllos a los que se les ha concedido acceso.

El conjunto de objetos que posee una cuenta de usuario se denomina *esquema* (*schema*) del usuario. En Oracle7, es posible crear usuarios que no tengan la capacidad de acceder a la base de datos. Dichas cuentas de usuario proporcionan un esquema que puede utilizarse para mantener un conjunto de objetos de base de datos separados de otros esquemas de los usuarios.

II.4.3 Índices

En una base de datos relacional, la situación física de una fila es irrelevante (a menos, por supuesto, que la base de datos tenga que encontrarla). Para que sea posible encontrar los datos, cada fila de cada tabla se etiqueta con un *ROWID* (*IDENTIFICADOR DE FILA*). Este *ROWID* le indica a la base de datos el lugar exacto en que se encuentra la fila (mediante el archivo, bloque dentro de ese archivo y fila dentro de ese bloque).

Un índice es una estructura de la base de datos que le permite al servidor localizar rápidamente una fila de una tabla. Existen dos tipos de índices: índices de grupo e índices de tabla.

Los índices contienen una lista de entradas, cada una de las cuales consta de un valor clave y de un *ROWID*. El valor clave es el valor de una columna de una fila o la combinación de los valores de varias columnas en una fila. Las entradas de los índices de tabla y de grupo de una base de datos Oracle se almacenan mediante un mecanismo de B*-tree (árbol binario), lo que garantiza un trayecto de acceso corto hasta el valor clave. Las operaciones de E/S necesarias para localizar un valor clave son mínimas, y, una vez encontrado, se utiliza el *ROWID* para acceder directamente a una fila.

Los índices se utilizan tanto en Oracle Versión 6 como en Oracle7 para mejorar el rendimiento y asegurar la exclusividad de una columna.

En Oracle Versión 6, si en una columna se ha especificado la cláusula de restricción *UNIQUE* en una orden *CREATE TABLE*, la restricción no se aplica a menos que se cree un índice de tabla para la columna.

En Oracle7, los índices se crean de forma automática siempre que se especifique una cláusula de restricción *UNIQUE* o *PRIMARY KEY* en una orden *CREATE TABLE*.

Los índices pueden crearse para una o para varias columnas de una tabla. En el ejemplo anterior de la tabla *EMPLEADO*, una base de datos Oracle7 creará de forma automática índices exclusivos para las columnas *NUMEMP* y *NUM_SEG_SOC*, ya que se han especificado como *PRIMARY KEY* y *UNIQUE*, respectivamente. En una base de datos Oracle Versión 6, los índices tendrían que crearse mediante órdenes adicionales. La eliminación de un índice no afecta a los datos de una tabla previamente indexada.

II.4.4 Grupos

Las tablas a las que se suele acceder conjuntamente pueden almacenarse físicamente juntas. Para almacenarlas juntas, se crea un grupo (*cluster*) que contenga las tablas. Los datos de las tablas se almacenan entonces juntos para minimizar el número de operaciones de E/S que deben realizarse y mejorar así el rendimiento.

Las columnas relacionadas de las tablas se denominan la clave del grupo. Esta clave del grupo se indexa mediante un índice de grupo, y su valor sólo se almacena una vez para las diversas tablas del grupo.

II.4.5 Grupos Hash

En Oracle7 se dispone de un segundo tipo de grupo. Los grupos *hash* (*hash clusters*) utilizan *funciones de hash* (*hashing functions*) sobre la clave del grupo de la fila para determinar la posición física en la que debe almacenarse la fila. Así se obtienen los mayores beneficios en el rendimiento en consultas del tipo que se muestra a continuación.

```
SELECT nombre
FROM empleado
WHERE numemp = 123;
```

En este ejemplo, se consulta en la tabla EMPLEADO una concordancia exacta en la columna NUMEMP. Si dicha tabla formara parte de un grupo hash, y NUMEMP fuera parte de la clave del grupo, la base de datos podría utilizar la función de cálculo de clave para determinar rápidamente el emplazamiento físico de los datos. La mejora del rendimiento no suele ser la misma si en la cláusula WHERE se especifica un margen de valores, como en el siguiente ejemplo.

```
SELECT nombre
FROM empleado
WHERE numemp > 123;
```

II.4.6 Vistas

Una vista (*view*) tiene el mismo aspecto que una tabla con columnas y se consulta de la misma forma que ésta. Conceptualmente, puede pensarse en una vista como en una máscara que cubre una o más tablas, de forma que las columnas de la vista se encuentran en una o más de las tablas subyacentes. Así, las vistas no almacenan físicamente los datos. La definición de una vista (en la que se incluye la consulta en la que se basa, la disposición de sus columnas y los privilegios concedidos) se almacena en el diccionario de datos.

Cuando se consulta una vista, ésta consulta a las tablas en que se basa, y devuelve los valores en el formato y orden especificados en la definición de la vista. Las vistas no pueden indexarse ya que no hay datos físicos directamente asociados a ellas.

II.4.7 Secuencias

Las definiciones de las *secuencias* (*sequences*) también se almacenan en el diccionario de datos. Las secuencias proporcionan una lista consecutiva de números exclusivos que sirve para simplificar las tareas de programación.

La primera vez que se llama a una secuencia en una consulta, devuelve un valor predeterminado. En cada consulta subsiguiente a la secuencia se obtendrá un valor aumentado en el incremento especificado. Las secuencias pueden ser cíclicas o pueden seguir creciendo hasta alcanzar un valor máximo especificado.

II.4.8 Procedimientos, Funciones, Paquetes y Disparadores (Triggers)

Un *procedimiento (procedure)* es un bloque de sentencias PL/SQL que se almacena en el diccionario de datos y al que pueden llamar las aplicaciones. Los procedimientos permiten almacenar dentro de la base de datos relaciones de las aplicaciones que se utilicen con frecuencia. Cuando se ejecuta el procedimiento, sus sentencias se ejecutan como una unidad. Los procedimientos no devuelven ningún valor al programa que los llama. No están disponibles en Oracle Versión 6.

Los procedimientos almacenados pueden ayudar a reforzar la seguridad de los datos, lo que se consigue no concediendo a los usuarios el acceso directo a las tablas de una aplicación, sino sólo la posibilidad de ejecutar un procedimiento de acceso a las tablas. Cuando se ejecuta el procedimiento, lo hará con los privilegios de su propietario. Los usuarios no podrán acceder a las tablas a no ser por medio del procedimiento.

Las *funciones (functions)*, lo mismo que los procedimientos, son bloques de código que se almacenan en la base de datos. A diferencia de éstos, las funciones pueden devolver valores al programa que las llama. No están disponibles en Oracle Versión 6.

Los *paquetes (packages)* sirven para organizar los procedimientos y las funciones en agrupamientos lógicos; sus definiciones se almacenan en el diccionario de datos de Oracle7. Los paquetes son muy útiles en las tareas administrativas necesarias para gestionar los procedimientos y las funciones.

Distintos elementos del paquete pueden definirse como «públicos» o como «privados». Los elementos públicos están accesibles para el usuario del paquete, mientras que los elementos privados están ocultos. Entre los elementos privados suelen encontrarse procedimientos a los que llaman otros procedimientos del paquete.

Los *disparadores* son procedimientos que se ejecutan cuando se produce un evento de base de datos especificado en una tabla especificada en una base de datos Oracle7. Pueden utilizarse para aumentar la integridad referencial, conseguir una seguridad adicional o mejorar las opciones de auditoría disponibles.

Existen dos tipos de disparadores:

- Disparadores de sentencia: se activan una vez por cada sentencia de disparo.
- Disparadores de fila: se activan una vez por cada fila de una tabla afectada por las sentencias.

Para cada uno de ellos puede crearse un disparador BEFORE (ANTES) y otro AFTER (DESPUÉS) para cada tipo de evento de disparo. Entre los eventos de disparo se encuentran las operaciones INSERT (Insertar), UPDATE (Actualizar) y DELETE (Eliminar).

Los disparadores de sentencia resultan útiles si el código de la acción de disparo no cuenta con los datos afectados. Por ejemplo, se puede crear un disparador de sentencia BEFORE INSERT

en una tabla para impedir operaciones de INSERT (Inserción) en ella excepto durante períodos de tiempo específicos.

Los disparadores de fila son útiles si la acción de disparo cuenta con los datos afectados por la transacción. Por ejemplo, puede crearse un disparador de fila AFTER INSERT que introduzca filas nuevas en una tabla de auditoría además de en la tabla base del disparador.

II.4.9 Sinónimos

Para realizar una identificación completa de un objeto de base de datos (como una tabla o una vista) en una base de datos distribuida Oracle, es necesario especificar el nombre de la máquina anfitriona, el nombre del servidor, el propietario del objeto y el nombre del objeto. Serán necesarios entre uno y cuatro de estos parámetros en función del emplazamiento del objeto. Los desarrolladores pueden crear sinónimos que apunten al objeto adecuado para ocultar este proceso a los usuarios, que sólo tienen que conocer el nombre del sinónimo. Los sinónimos públicos los comparten todos los usuarios de una base de datos concreta, mientras que los sinónimos privados pertenecen a propietarios de cuentas de base de datos individuales.

Por ejemplo, la tabla EMPLEADO descrita anteriormente debe ser propiedad de una cuenta (digamos que el propietario es HR). desde otra cuenta de usuario de la misma base de datos, dicha tabla podría referenciarse como HR.EMPLEADO. No obstante, esto implica que la segunda cuenta debe conocer que la tabla EMPLEADO es propiedad de la cuenta HR. Para evitar esto, puede crearse un sinónimo público llamado EMPLEADO que apunte a HR.EMPLEADO. Siempre que se haga referencia a este sinónimo apuntará a la tabla adecuada. La siguiente sentencia SQL permite crear dicho sinónimo.

```
CREATE PUBLIC SYNONYM empleado FOR hr.empleado;
```

Los sinónimos permiten proporcionar punteros a tablas, vistas, procedimientos, funciones, paquetes y secuencias. Pueden apuntar a objetos de la base de datos local o de bases de datos remotas. El apuntar a bases de datos remotas se consigue mediante la utilización de enlaces de base de datos, que se describirán más adelante en esta sección.

II.4.10 Privilegios y Funciones (roles)

Para que una cuenta pueda acceder a un objeto propiedad de otra cuenta tiene que habersele concedido primero el *privilegio (privilege)* de acceso. Lo habitual es que a los no propietarios se les conceda el privilegio de INSERT (Insertar), SELECT (Seleccionar), UPDATE (Actualizar) o DELETE (Borrar) filas de una tabla o vista. También pueden concederse privilegios para SELECT valores de secuencias o EXECUTE (Ejecutar) procedimientos y funciones. No se conceden privilegios sobre los índices o los disparadores, ya que la base de datos los utiliza durante la actividad de la tabla. Los privilegios pueden concederse a usuarios individuales o a PUBLIC (Público), que concede los privilegios a todos los usuarios de la base de datos.

En Oracle Versión 6, la gestión de los privilegios de los usuarios para una aplicación grande pueden convertirse rápidamente en una experiencia penosa, ya que es necesario conceder cada privilegio a cada cuenta de usuario según lo necesite la aplicación. En Oracle7, este proceso se simplifica mediante la utilización de las *funciones (roles)*, que son grupos de privilegios.

Los privilegios pueden concederse a una *función (role)*, y la función concederse a varios usuarios. De esta manera, el hecho de añadir nuevos usuarios a las aplicaciones se simplifica en gran medida, ya que basta con conceder o retirar funciones al usuario.

La relación entre privilegios y funciones se muestra en la Figura 2.6. En la Figura 3.6(a), aparecen como líneas los privilegios necesarios para conceder acceso SELECT sobre dos tablas a cuatro usuarios. En la Figura 2.6(b), se utiliza la capacidad de Oracle7 de manejar funciones para simplificar la administración de los privilegios. Los privilegios se conceden a una sola función, y dicha función se concede a los cuatro usuarios.

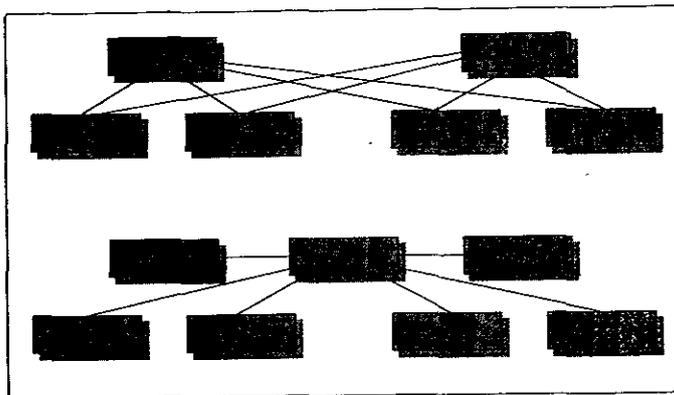


Fig. 2.6. Relación entre privilegios y funciones.

También pueden utilizarse los grupos de funciones de Oracle7 para conceder privilegios de nivel de sistema, como CREATE TABLE (Crear tabla).

II.4.11 Enlaces de base de datos

Las bases de datos Oracle que utilizan SQL*Net tienen la capacidad de hacer referencia a datos almacenados fuera de la base de datos local. En estos casos debe especificarse el nombre completo del objeto remoto. En el ejemplo del apartado dedicado a los sinónimos sólo se especificaron dos partes del nombre completo (el propietario y el nombre de la tabla). ¿Qué ocurre si la tabla se encuentra en una base de datos remota?

Para especificar una vía de acceso a un objeto situado en una base de datos remota es necesario crear un *enlace de base de datos (database link)*. Los enlaces de base de datos pueden ser tanto públicos (disponibles para todas las cuentas de esa base de datos) como privados (creados por un usuario para el uso exclusivo de esa cuenta). Las opciones disponibles al crear un enlace de base

de datos varían en función de la versión de SQL*Net que se utilice. La siguiente sintaxis funciona en ambas versiones.

```
CREATE PUBLIC DATABASE link mi_enlace
CONNECT TO hr IDENTIFIED BY puffinstuff
USING 't:ost1:db1';
```

En este ejemplo, en el enlace se especifica que cuando se utilice se va a abrir una sesión en la base de datos «db1» del servidor llamado «host1» (la «t:» especifica el controlador de TCP/IP; el controlador adecuado depende del sistema). Cuando se abra la sesión en la instancia de db1, se conectará a la cuenta de usuario HR, con la contraseña «puffinstuff». Para utilizar este enlace en una tabla, debe especificarse en la cláusula FROM, como en el siguiente ejemplo.

```
SELECT * FROM empleado@mi_enlace;
```

De esta manera se accede a la tabla descrita anteriormente. Es posible crear un sinónimo para esta tabla, como se muestra en la siguiente orden SQL.

```
CREATE SYNONYM empleado FOR empleado@mi_enlace;
```

Se observa que se ha definido la designación completa del objeto de base de datos —su anfitrión (host1), su base de datos (db1), su propietario (HR) y su nombre (EMPLEADO).

La ubicación de la tabla EMPLEADO es así completamente transparente para el usuario final. Los desarrolladores de SQL*Net V2 también pueden aprovechar los nombres de los servicios para hacer la ubicación de los datos más transparente aún. La línea «connect to» puede eliminarse si en la base de datos remota se utiliza la misma cuenta y contraseña que en la base de datos local.

II.4.12 Segmentos, Extensiones y Bloques

Los segmentos (segments) son la contrapartida física a los objetos lógicos de base de datos que almacenan los datos. En los segmentos de índice, por ejemplo, se almacenan los datos asociados con los índices. Para una gestión eficaz de los segmentos es necesario que el DBA (Administrador de bases de datos) conozca los objetos que una aplicación va a utilizar, la forma en que se van a introducir los datos en esos objetos y las maneras en que se van a recuperar.

Como un segmento es una entidad física, debe estar asignado a un espacio de tablas de la base de datos (y por tanto encontrarse en uno de los archivos de datos de dicho espacio de tablas). Un segmento consta de secciones llamadas *extensiones (extents)* (conjuntos contiguos de bloques de Oracle). Cuando las extensiones existentes en un segmento ya no pueden contener datos nuevos, el segmento consigue otra extensión. Este proceso de extensión continúa hasta agotar el espacio libre disponible en los archivos de datos del espacio de tablas, o hasta alcanzar el número máximo de extensiones por segmento. Si un segmento consta de varias extensiones, no se garantiza que dichas extensiones sean contiguas.

II.4.13 Segmentos de rollback (anulación)

Para mantener la consistencia de lectura entre varios usuarios de la base de datos y poder anular las transacciones, Oracle debe disponer de un mecanismo de reconstrucción de una «imagen anterior» de los datos para transacciones no confirmadas. Oracle utiliza *segmentos de rollback (anulación)* dentro de la base de datos para llevar a cabo esta tarea.

Los segmentos de rollback crecen hasta hacerse tan grandes como las transacciones que admiten.

II.5 Estructuras internas de la memoria

La base de datos Oracle utiliza dos tipos distintos de estructuras de memoria: áreas globales y áreas de proceso. Las áreas de proceso se describirán en la siguiente sección. Esta sección se centrará en las áreas globales de memoria que utilizan todos los usuarios de base de datos Oracle.

La implementación de las opciones de memoria disponibles presenta un amplio abanico de posibilidades en función de la opción de servidor de bases de datos que se utilice.

Los elementos que se describen son:

- Área global del sistema (*System Global Area: SGA*).
- Búferes del bloque de datos.
- Caché del diccionario.
- Búferes de registro de rehacer.
- Fondo común SQL compartido.
- Áreas de contexto.
- Área global del programa (*Program Global Area: PGA*).

II.5.1 Área global del sistema (System Global Area: SGA)

El *Área global del sistema (System Global Area: SGA)* de una base de datos Oracle sirve para lo mismo (facilita la transferencia de información entre usuarios). También mantiene la información estructural más consultada sobre la base de datos.

En Oracle Versión 6, el SGA está dividido en tres secciones principales. Estas secciones también se encuentran en Oracle7, junto con áreas compartidas adicionales que eran específicas de los procesos en Oracle Versión 6.

La composición del SGA se muestra en la Figura 2.5. En la Figura 2.5(a) se muestra la estructura de la SGA en Oracle Versión 6, y en la Figura 2.5(b), la misma estructura en Oracle7. Las áreas compartidas de las figuras se describen en las siguientes secciones.

II.5.2 Búferes del bloque de datos

Los *búferes del bloque de datos* (*data block buffers*) son un caché del SGA que se utiliza para contener los bloques de datos que se leen de los segmentos de datos de la base de datos, tales como tablas, índices y grupos. El tamaño del caché del búfer del bloque de datos viene determinado por el parámetro `DB_BLOCK_BUFFERS` del archivo `INIT.ORA` de dicho servidor de bases de datos. La gestión del tamaño del caché del búfer del bloque de datos juega un importante papel en la gestión y ajuste de la base de datos.

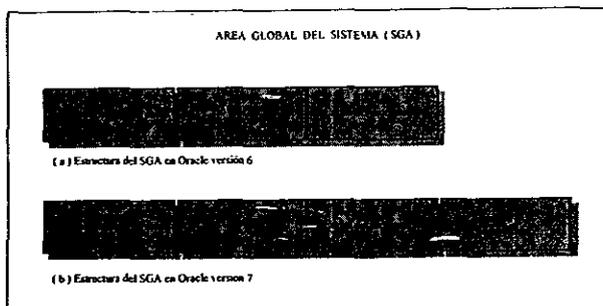


Figura 2.7. Estructuras del SGA en Oracle Versión 6 y en Oracle7.

Como este tamaño es fijo, y suele ser inferior al espacio utilizado por los segmentos de la base de datos, no se pueden conservar todos los segmentos de la base de datos en memoria simultáneamente. Oracle gestiona el espacio disponible mediante un algoritmo de *Menos utilizado recientemente* (*LRU: Least Recently Used*). Cuando se necesita espacio libre en el caché, se escriben al disco los bloques menos utilizados recientemente, siendo sustituidos por nuevos bloques de datos. De esta forma, en la memoria se conservan los datos utilizados con más frecuencia.

No obstante, si el SGA no es lo suficientemente grande, distintos objetos competirán por el espacio del caché del búfer del bloque de datos. Esto es particularmente probable cuando varias aplicaciones comparten el mismo SGA. En dicho caso, los segmentos más utilizados recientemente de cada aplicación compiten constantemente por el espacio del SGA con los segmentos más utilizados recientemente de otras aplicaciones. Como consecuencia de ello, las solicitudes de datos del caché del búfer del bloque de datos tendrán una baja relación de «aciertos» a «fallos».

II.5.3 Caché del diccionario

La información sobre los objetos de la base de datos se almacena en las tablas del diccionario de datos. En esta información se incluyen los datos de las cuentas del usuario, los nombres de los archivos de datos, los nombres de los segmentos, las ubicaciones de las extensiones, las descripciones de las tablas y los privilegios. cuando la base de datos necesita esta

información (por ejemplo para comprobar la autorización de un usuario para consultar una tabla), se leen las tablas del diccionario de datos y los datos devueltos se almacenan en el SGA, en el *caché del diccionario (dictionary cache)*.

Este caché también se gestiona mediante un algoritmo LRU. En Oracle7, la base de datos es la encargada de gestionar internamente el tamaño del caché del diccionario; forma parte del fondo común SQL compartido, cuyo tamaño se establece mediante el parámetro `SHARED_POOL_SIZE` del archivo `INIT.ORA` de la base de datos. En Oracle Versión 6, el tamaño de cada caché del caché del diccionario se establece de forma manual mediante entradas independientes en el archivo `INIT.ORA` de la base de datos (como `DC_TABLES` para el número de entradas en el caché de definición de tablas).

Si el caché del diccionario es demasiado pequeño, la base de datos tendrá que consultar reiteradamente la información que necesita en las tablas del diccionario de datos. Estas consultas se llaman *aciertos recursivos*, y se resuelven más lentamente que las consultas que pueden manejarse sólo mediante el caché del diccionario que está en memoria.

II.5.4 Búfer de registro de rehacer

Los archivos de registro de rehacer se describen en la sección «Registros de rehacer», más adelante en este capítulo. En las entradas de rehacer se describen los cambios realizados en la base de datos. Se escriben en los archivos de registro de rehacer en línea de forma que puedan utilizarse en las operaciones de reconstrucción durante las recuperaciones de la base de datos. No obstante, antes de escribirse en los archivos de registro de rehacer en línea, se almacenan primero en el SGA, en un área llamada *búfer de registro de rehacer (redo log buffer)*. La base de datos escribe después de forma periódica series de entradas de rehacer a los archivos de registro de rehacer en línea, optimizando así esta operación.

El tamaño (en bytes) de los búferes de registro de rehacer se define mediante el parámetro `LOG_BUFFER` del archivo `INIT.ORA`.

II.5.5 Fondo común SQL compartido

En las bases de datos Oracle7, el caché del diccionario se almacena en un área denominada *fondo común SQL compartido (shared SQL pool)*. En este área de memoria también se incluye información sobre sentencias ejecutadas sobre la base de datos. Así, mientras que el búfer del bloque de datos y el caché del diccionario permiten la compartición de información estructural y de los datos entre los usuarios, el *fondo común SQL compartido permite compartir las sentencias SQL* más utilizadas.

El fondo común SQL compartido contiene el plan de ejecución y el árbol de análisis de las sentencias SQL que se ejecutan sobre la base de datos. La segunda vez que cualquier usuario ejecuta una sentencia SQL idéntica, puede aprovecharse la información de análisis disponible en el fondo común SQL compartido para acelerar su ejecución.

II.5.6 Áreas de contexto

Dentro del área SQL compartida de Oracle7 existen tanto áreas públicas como privadas. Todas las sentencias SQL emitidas por un usuario necesitan un área SQL privada, que sigue existiendo hasta que se cierra el cursor correspondiente a la sentencia. La estructura análoga en Oracle Versión 6 se llama *área de contexto (context area)*; no obstante, como en Oracle Versión 6 no existen los fondos comunes SQL compartidos (y por tanto no hay áreas SQL públicas), las áreas de contexto deben gestionar toda la información de la que se ocupa el fondo común SQL compartido en Oracle7.

II.5.7 Área global del programa (Program Global Area: PGA)

El *área global del programa (PGA)* es un área de la memoria utilizada por un único proceso de usuario de Oracle. En Oracle Versión 6, contiene las áreas de contexto del usuario, además de la información de control del proceso. La memoria del PGA no se comparte.

En Oracle7, parte del PGA puede almacenarse en el SGA. Esto ocurre cuando se utiliza el servidor multiproceso (multi-threaded) de Oracle7. La arquitectura multiproceso del servidor permite que varios procesos de usuario utilicen el mismo proceso del servidor, reduciendo así los requisitos de memoria de la base de datos. Si se utiliza esta opción, la información sobre la sesión del usuario se almacena en el SGA en lugar de en el PGA.

II.6 Estructuras de los procesos

Las relaciones entre las estructuras físicas y la memoria de la base de datos se mantienen y aplican mediante *estructuras de proceso*, que son los procesos de fondo propios de la base de datos, y cuyo número varía en función de la configuración de la base de datos. Estos procesos los gestiona la base de datos y apenas necesitan trabajo administrativo.

En las secciones siguientes se describe cada uno de los procesos de fondo y el papel que juega en la gestión de la base de datos. En la Figura 2.6 se muestran las relaciones existentes entre las estructuras físicas y las estructuras de la memoria en la base de datos, junto con los procesos principales.

II.6.1 Supervisor del sistema: SMON

El proceso de fondo *SMON* sirve para las mismas funciones de supervisión tanto en Oracle7 como en Oracle Versión 6. Cuando se arranca la base de datos, SMON realiza la recuperación de la instancia según se necesite (utilizando para ello los archivos de registro de rehacer en línea). También limpia la base de datos, eliminando los objetos de las transacciones que el sistema ya no necesita.

En Oracle7, SMON tiene una función adicional: agrupa extensiones libres contiguas en extensiones libres mayores. Los DBA (Administradores de bases de datos) de Oracle Versión 6 deben realizar de forma manual la agrupación del espacio libre.

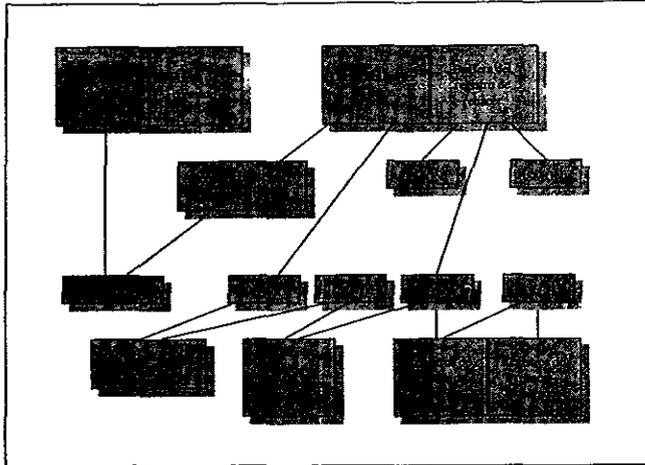


Figura 2.8 Estructuras físicas, de la memoria y de los procesos en Oracle.

II.6.2 Supervisor del proceso: PMON

El proceso de fondo *PMON* realiza una limpieza al terminar la ejecución de los procesos tanto en Oracle Versión 6 como en Oracle7. *PMON* libera los recursos que estuviera utilizando el usuario. Sus efectos se hacen patentes cuando se elimina un proceso bloqueado; *PMON* es responsable de liberar el bloqueo y hacer que esté disponible para otros usuarios. Al igual que *SMON*, *PMON* se activa de forma periódica para comprobar si es necesaria su intervención.

II.6.3 Escritor de bases de datos: DBWR

El proceso de fondo *DBWR* es responsable de gestionar el contenido del caché del búfer del bloque de datos y del caché del diccionario, lee los bloques de los archivos de datos y los almacena en el SGA, realiza series de escrituras de los bloques modificados de vuelta en los archivos de datos. El ajuste de las actividades del proceso *DBWR* (Eficiencia de las lecturas de los archivos de datos, Eficiencia de las lecturas de la memoria) es una buena parte del trabajo de ajuste que deben realizar los DBA (Administradores de bases de datos).

Aunque sólo existe un proceso *SMON* y otro *PMON* en ejecución para cada instancia de base de datos, es posible tener varios procesos *DBWR* ejecutándose a la vez, en función de la plataforma y del sistema operativo. La utilización de varios procesos *DBWR* ayuda a minimizar la contienda dentro del *DBWR* durante consultas largas que afectan a los archivos de datos. El número de procesos *DBWR* en ejecución se configura mediante el parámetro *DB_WRITERS* del archivo *INIT.ORA* de la base de datos.

II.6.4 Escritor de registros: LGWR

El proceso de fondo *LGWR* gestiona la escritura del contenido del búfer del registro de rehacer a los archivos del registro de rehacer en línea. Dicha escritura la realiza por series. Las entradas del búfer del registro de rehacer siempre contienen el estado más actualizado de la base de datos, ya que el proceso *DBWR* debe esperar antes de escribir los bloques modificados desde los búferes del bloque de datos a los archivos de datos.

Se observa que el *LGWR* es el único proceso que escribe en los archivos de registro de rehacer en línea y el único que lee directamente los búferes de registro de rehacer durante el funcionamiento normal de la base de datos. En los archivos del registro de rehacer en línea se escribe de modo secuencial, al contrario que los accesos completamente aleatorios que el *DBWR* realiza a los archivos de datos. Si los archivos de registro de rehacer en línea están replicados (como puede ocurrir en Oracle7), *LGWR* escribe simultáneamente en los conjuntos replicados de registros.

II.6.5 Punto de control (Oracle7): CKPT

Los puntos de control (*checkpoints*) ayudan a reducir la cantidad de tiempo necesaria para realizar la recuperación de las instancias. Los puntos de control provocan que el *DBWR* escriba en los archivos de datos todos los bloques que se hayan modificado desde el último punto de control y que actualice las cabeceras de los archivos de datos y los archivos de control para registrar el punto de control. Se producen de forma automática cuando se llena un archivo de registro de rehacer en línea; puede utilizarse el parámetro `LOG_CHECKPOINT_INTERVAL` del archivo `INIT.ORA` de la instancia de la base de datos para configurar un punto de control más frecuente. En Oracle Versión 6, el único método de modificar la frecuencia del punto de control es mediante la modificación de `LOG_CHECKPOINT_INTERVAL`.

Sólo en Oracle7, puede crearse un proceso de fondo adicional, *CKPT*, para separar las dos funciones de *LGWR* (Señalar puntos de control y copiar la entrada de rehacer) en dos procesos de fondo. El proceso de fondo *CKPT* se habilita definiendo como `TRUE` (Verdadero) el parámetro `CHECKPOINT_PROCESS` del archivo `INIT.ORA` de la instancia de la base de datos. Este proceso no es necesario a menos que la base de datos experimente un elevado volumen de transacciones, lo que provoca retardos en las conmutaciones de registro.

II.6.6 Archivador: ARCH

El proceso de fondo *LGWR* escribe en los archivos de registro de rehacer en línea de forma cíclica; después de rellenar el primer archivo empieza a escribir en el segundo, hasta que lo rellena, y empieza entonces a escribir en el tercero. Cuando se rellena el último archivo de registro de rehacer en línea, *LGWR* empieza a sobrescribir el contenido del primero.

Cuando Oracle se ejecuta en modo `ARCHIVELOG`, la base de datos realiza una copia de cada uno de los archivos de registro de rehacer antes de sobrescribirlo. Dichos archivos suelen

escribirse en un dispositivo de disco. aunque también pueden escribirse directamente en un dispositivo de cinta magnética, aunque esto último suele ocupar mucho tiempo de operador.

La función de archivado la realiza el proceso de fondo ARCH. Las bases de datos que utilizan esta opción experimentarán problemas de contienda en su disco de registro de rehacer durante los periodos con altas tasas de transacción de datos, ya que LGWR intentará escribir en un archivo de registro de rehacer mientras ARCH intenta leer de otro. También pueden producirse bloqueos en la base de datos si se llena el disco de destino de registro del archivo. En este caso, ARCH se detiene, lo que impide que LGWR siga escribiendo, lo que a su vez impide que se produzcan más transacciones en la base de datos hasta que se libere espacio para los archivos de registro de rehacer archivados.

II.6.7 Recuperador: RECO

El proceso de fondo RECO sirve para corregir fallos en las bases de datos distribuidas de Oracle7. Este proceso intenta acceder a las bases de datos implicadas en transacciones dudosas y resolver dichas transacciones. Este proceso sólo se crea si la plataforma de trabajo admite la Opción distribuida (*Distributed option*) de Oracle7 y el parámetro DISTRIBUTED_TRANSACTIONS del archivo INIT.ORA está definido con un valor superior a cero.

II.6.8 Bloqueo (Servidor paralelo): LCKn

Existen varios procesos LCK, denominados LCK0 a LCK9, que sirven para el bloque entre instancias cuando se utilizan las opciones de Servidor paralelo de Oracle7 y Oracle 6.2. En Oracle Versión 6.2, el número de procesos LCK se define por medio del parámetro MI_BG_PROCS del archivo INIT.ORA de la base de datos; en Oracle7, se define mediante el parámetro GC_LCK_PROCS.

II.6.9 Despachador : Dnnn

Los procesos despachadores (dispatcher) forman parte de la arquitectura de servidor multiproceso (MTS) de SQL*Net V2; al manejar varias conexiones ayudan a minimizar las necesidades de recursos. Debe crearse al menos un proceso despachador para cada protocolo que admita el servidor de la base de datos. Los procesos despachadores se crean durante el arranque de la base de datos, basándose en la configuración de SQL*Net, y pueden crearse o eliminarse mientras la base de datos permanezca abierta.

II.6.10 Servidor: Snnn

Los procesos del servidor se crean para gestionar las conexiones con la base de datos que necesiten un servidor dedicado. Son típicos en las conexiones anteriores a SQL*Net V2.

II.7 Estructuras externas

Los archivos de datos de la base de datos, como se ha descrito anteriormente en el apartado «Archivos» de la sección de introducción de este capítulo, proporcionan el almacenamiento físico de los datos de la base de datos. Por tanto, son estructuras tanto «internas», por estar ligadas directamente a los espacios de tablas, como «externas», por tratarse de archivos físicos.

Los siguientes tipos de archivos, aunque están relacionados con la base de datos, están separados de los archivos de datos.

- Registros de rehacer.
- Archivos de control.

II.7.1 Registros de rehacer

Oracle conserva registros de todas las transacciones que afectan a la base de datos. Estas transacciones se registran en un archivo denominado archivo de registro de rehacer en línea, y se utilizan para recuperar las transacciones de la base de datos en el orden adecuado en caso de que se produzca un fallo de la base de datos. La información de registro de rehacer se almacena externamente a los archivos de datos de la base de datos.

Los archivos de registro de rehacer también permiten que Oracle coordine la forma en que se escribe los datos al disco. Cuando se produce una transacción en la base de datos, se introduce en los búferes de registro de rehacer, mientras que los bloques de datos afectados por la transacción no se escriben de manera inmediata al disco. Esto permite que la base de datos realice series de escrituras al disco, optimizando así el rendimiento de esta función.

Todas las bases de datos Oracle tienen dos o más archivos de registro de rehacer en línea. Oracle escribe en ellos de manera cíclica: después de rellenar el primer archivo de registro escribe en el segundo, hasta rellenoarlo. Cuando se rellenan todos los archivos de registro de rehacer en línea, vuelve al primero y empieza a sobrescribir su contenido con nuevos datos de transacciones. Si la base de datos está trabajando en modo ARCHIVELOG, se hará una copia de los archivos de registro de rehacer en línea antes de sobrescribirlos. Esta copia puede utilizarse para recuperar parte de la base de datos al estado que tuviera en cualquier instante.

En Oracle7, los archivos de registro de rehacer pueden replicarse dentro de la base de datos, lo que permite que el DBA replique los archivos de registro de rehacer sin confiar en el sistema operativo o en las posibilidades hardware del entorno operativo.

II.7.2 Archivos de control

La arquitectura física general de una base de datos se mantiene por medio de sus archivos de control, en los que se registra la información de control sobre todos los archivos de la base de

datos. Se utilizan para conservar la consistencia interna y servir de guía en las operaciones de recuperación.

Como los archivos de control son fundamentales para la base de datos, se guardan varias copias en línea. Estos archivos suelen almacenarse en discos separados para minimizar las consecuencias perjudiciales de un posible fallo del disco. La base de datos creará y mantendrá los archivos de control especificados durante la creación de la base de datos.

II.8 Implementación básica de la base de datos

En su forma más elemental, una base de datos Oracle consta de:

- Uno o más archivos de datos.
- Uno o más archivos de control.
- Dos o más registros de rehacer en línea.

Internamente, dicha base de datos contiene:

- Varios usuarios/esquemas.
- Uno o más segmentos de rollback (anulación).
- Uno o más espacios de tablas.
- Tablas del diccionario de datos.
- Objetos de usuario (tablas, índices, vistas, etc.)

El servidor que accede a esa base de datos consta de (como mínimo):

Un SGA (que incluye el caché del búfer del bloque de datos, el caché del diccionario, el caché del búfer de registro de rehacer y, en Oracle7, el fondo común SQL compartido).

- El proceso de fondo SMON.
- El proceso de fondo PMON.
- El proceso de fondo DBWR.
- El proceso de fondo LGWR.
- Procesos de usuario con PGA asociados.

Esta es la configuración básica; todo lo demás es opcional o depende de la versión de Oracle y de las opciones que se utilicen.

II.9 Capacidades de copia de seguridad/recuperación

La base de datos Oracle incorpora varias opciones de copia de seguridad y de recuperación.

II.9.1 Export/Import (Exportar/Importar)

La utilidad *Export* (*Exportar*) consulta la base de datos y almacena su respuesta en un archivo binario. Es posible personalizar las partes de la base de datos que se leen con esta utilidad.

Se le puede indicar que lea la base de datos completa, un usuario o un conjunto de esquemas de usuario o un conjunto de tablas específico. También dispone de opciones que permiten exportar sólo las tablas que se han modificado desde la última exportación (a esto se le llama exportación *incremental*) o desde la última exportación del sistema completo (llamada exportación *acumulativa*).

Las exportaciones del sistema completo leen también las tablas del diccionario de datos completas. Por tanto, pueden utilizarse para recrear por entero una base de datos, ya que en el diccionario de datos se realiza el seguimiento de los usuarios, archivos de datos y objetos de la base de datos. Suele utilizarse para eliminar la fragmentación de la base de datos.

Export realiza una lectura lógica de la base de datos. Para leer la información contenida en el archivo de volcado binario creado por la exportación debe utilizarse la utilidad *Import (Importar)*. Import puede elegir de forma selectiva los objetos o usuarios del archivo de volcado que se deseen importar. Intentará entonces introducir esos datos en la base de datos (en lugar de sobrescribir los registros existentes).

II.9.2 Copias de seguridad fuera de línea

Además de copias de seguridad lógicas (exportaciones) de la base de datos también pueden realizarse copias de seguridad físicas de sus archivos. Se dispone de dos opciones para llevar a cabo una copia de seguridad física de la base de datos: *copias de seguridad en línea (online backups)* y *copias de seguridad fuera de línea (offline backups)*. Para realizar copias de seguridad fuera de línea debe apagarse primero la base de datos; puede realizarse entonces una copia de seguridad de los archivos que componen la base de datos en un dispositivo de almacenamiento (copias de disco a disco o escrituras en cinta magnética). La base de datos puede volver a abrirse una vez terminada la copia de seguridad. No es posible realizar copias de seguridad físicas *incrementales de la base de datos Oracle*.

Aunque no se trata de la principal opción de copia de seguridad y recuperación disponible, sigue siendo una buena idea realizar una copia de seguridad fuera de línea de la base de datos de forma periódica (como cuando el anfitrión en que se encuentra tiene que someterse a las operaciones de mantenimiento rutinarios).

II.9.3 Copias de seguridad en línea

Las bases de datos que trabajan en modo ARCHIVELOG (descrito en la sección de proceso «Archivador: ARCH») disponen de copias de seguridad en línea, que permiten realizar copias de seguridad físicas de la base de datos mientras la base de datos se encuentra abierta. Esto se consigue poniendo los espacios de tablas en un estado de copia de seguridad de forma temporal y restableciéndolos a continuación a su estado normal, una vez realizada la copia de seguridad de sus archivos. Esta posibilidad se introdujo con Oracle Versión 6.

II.9.4 Seguridad de las cuentas

Las cuentas de la base de datos pueden protegerse mediante una contraseña. Esta protección es independiente de la protección mediante contraseña del sistema operativo. También se pueden crear cuentas con capacidad de acceso automático, lo que permite a los usuarios que han accedido a una cuenta del anfitrión acceder a una cuenta de base de datos relacionada sin tener que introducir una contraseña de base de datos. Disponer de una cuenta o de privilegios en una base de datos no le concede al usuario una cuenta o privilegios en otras bases de datos.

II.10 Privilegios de base de datos

En Oracle Versión 6, sólo existen tres niveles de privilegios de sistema en la base de datos: CONNECT (Conexión), RESOURCE (Recursos) y DBA (Administrador de bases de datos). Todos los usuarios necesitan el privilegio CONNECT, y los que van a crear segmentos necesitan tener el privilegio RESOURCE.

En Oracle7, pueden crearse funciones (*roles*) de nivel de sistema a partir de todo el conjunto de privilegios de nivel de sistema —como CREATE TABLE (Crear tabla), CREATE INDEX (Crear índice), SELECT ANY TABLE (Seleccionar cualquier tabla)— para aumentar el conjunto básico de funciones de nivel de sistema. En Oracle7 siguen existiendo CONNECT, RESOURCE y DBA, aunque ahora son funciones en lugar de privilegios.

II.11 Seguridad de los objetos

Los usuarios que han creado objetos pueden conceder privilegios sobre ellos a otros usuarios por medio de la orden GRANT (Conceder). También pueden conceder a otros usuarios la posibilidad de realizar concesiones sobre sus objetos. En Oracle7, el privilegio GRANT ANY TABLE (Conceder cualquier tabla) le da a las cuentas la capacidad de conceder privilegios sobre cualquier objeto a cualquier usuario.

II.12 Auditoría

Las actividades de los usuarios que afectan a objetos de la base de datos pueden supervisarse mediante la orden AUDIT (Auditoría). Entre dichas actividades se incluyen los accesos a tablas, los intentos de iniciar una sesión y las actividades que son privilegio del DBA. El resultado de estas auditorías se almacena en un archivo de auditoría dentro de la base de datos. Oracle7 ofrece el conjunto completo de posibilidades de auditoría de Oracle Versión 6, además de opciones adicionales como la utilización de disparadores (triggers) de base de datos para complementar las auditorías ordinarias.

II.13 Ejemplo de disposición lógica de base de datos

La disposición lógica de una base de datos Oracle tiene una enorme importancia en las opciones de administración de que dispone el DBA. La disposición del espacio de tablas que se muestra en la figura 2.9 se basa en las consideraciones de diseño, «Disposiciones lógicas de las bases de datos». Esta disposición permite aislar segmentos de base de datos en función de su utilización y de sus características.

Espacio de tablas	Utilización
SYSTEM	Diccionario de datos.
DATA	Tablas de uso normal de una aplicación.
DATA_2	Tablas estáticas que se utilizan durante el funcionamiento normal de una aplicación.
INDEXES	Índices de las tablas de funcionamiento normal de una aplicación.
INDEXES_2	Índices de las tablas estáticas de la aplicación.
RBS	Segmentos de rollback (anulación) de uso normal.
RBS_2	Segmentos de rollback especializados que se utilizan en las cargas de datos.
TEMP	Segmentos temporales de uso normal.
TEMP_USER	Segmentos temporales creados por un usuario concreto.
TOOLS	Tablas de herramientas de RDBMS.
TOOLS_1	Índices de las tablas de herramientas de RDBMS.
USERS	Objetos de los usuarios en las bases de datos en desarrollo.

Fig. 2.9 Ejemplo de disposición lógica de base de datos.

II.14 Ejemplo de disposición física de base de datos

La disposición física adecuada de los archivos de base de datos depende de la propia base de datos; no obstante, existen ciertas reglas generales, que pueden aplicarse para separar correctamente archivos de base de datos cuyas solicitudes de operaciones de E/S entren en conflicto entre ellas. La configuración que se muestra en la Figura 2.10 se da como ejemplo de un sistema de Producción con 12 discos. Es conveniente seguir los procedimientos que se proporcionan en el Capítulo 4 para determinar la distribución adecuada de archivos que cumpla sus necesidades.

Disk	Contents
1	Oracle software
2	SYSTEM tablespace, Control file 1
3	RBS tablespace, RBS_2 tablespace, Control file 2
4	DATA tablespace, Control file 3
5	INDEX tablespace
6	TEMP tablespace, TEMP_USER tablespace
7	TOOLS tablespace, INDEX_2 tablespace
8	Online Redo logs 1, 2, and 3
9	Application software
10	DATA_2
11	Archived redo log destination disk
12	Export dump file destination disk

II.15 Comprensión de los convenios de modelado lógico

En una de las secciones anteriores dedicada a las restricciones de las tablas se definieron varios términos de modelado de datos. En esta sección se mostrará la manera de representar gráficamente las relaciones que implican dichos términos.

Una clave primaria (*primary key; PK*) es la columna o conjunto de columnas que hacen que cada registro de una tabla sea exclusivo. Una *clave externa (foreign key; FK)* es un conjunto de columnas que hacen referencia a una clave primaria existente.

Las tablas pueden estar relacionadas entre ellas por medio de tres tipos de relaciones: *de una a una*, *de una a muchas* y *de muchas a muchas*. En una relación de una a una (1:1), las tablas comparten una clave primaria común. En una relación de una a muchas (1:M), un único registro de una tabla está relacionado con varios registros de otra tabla. En una relación de muchas a muchas (M:M), varios registros de una tabla están relacionados con varios registros de otra tabla.

Relaciones de una a una

Es raro que dos tablas compartan una clave primaria exactamente igual. En caso de ocurrir, suele ser por motivos de rendimiento o de seguridad. Por ejemplo, Oracle recomienda que, cuando se utilice un tipo de datos LONG en una tabla, se almacene en una tabla aparte, relacionada con la primera de modo 1:1.

Consideremos la tabla REP_VENTAS de la Figura 2.1 ¿Qué ocurriría si hubiera que añadir una columna adicional, RESUMEN, de un tipo de datos LONG, a los datos ya almacenados? Como existe un resumen para cada representante de ventas, la columna RESUMEN debería almacenarse en la tabla REP_VENTAS. No obstante, esto forzaría a la base de datos a leer el valor LONG cada vez que se realizara una consulta a la tabla, incluso si sólo se busca el campo NOMBRE.

Para mejorar el rendimiento, es conveniente crear una segunda tabla, denominada RESUMEN_REP_VENT. Esta tabla tendrá la misma clave primaria (NUMERO—REP) y una columna adicional (RESUMEN). De esta forma, las dos tablas tendrán una relación 1:1, tal como se muestra gráficamente en la Figura 2.11.

La línea continua entre las dos entidades indica que la relación es obligatoria. en caso de que la relación fuera opcional la línea sería discontinua.



Fig. 2.11. Diagrama de relaciones entre entidades para una relación 1:1.

Relaciones de uno a muchos

Las relaciones de una a una son raras. Es mucho más habitual que las relaciones sean del tipo de una a muchas (1:M). En este tipo de relación, un registro de una tabla está relacionado con varios registros de otra.

Consideremos de nuevo la tabla REP_VENTAS. Según se desprende de los registros que aparecen en la Figura 2.1, sólo hay un representante de ventas por oficina. No obstante, es posible que el análisis de los datos revele que varios representantes de ventas pueden informar a la misma oficina. En esta caso, debería crearse una nueva entidad, OFICINA. La columna OFICINA de la tabla REP_VENTAS sería entonces una clave externa a esta nueva tabla.

Como varios representantes de ventas (registros de la tabla REP_VENTAS) pueden informar a una sola oficina (registro de la tabla OFICINA), existe una relación 1:M entre estas tablas, como se muestra gráficamente en la Figura 2.9. Se aprecian dos diferencias en la línea de conexión: la adición de una «bifurcación» (*crow-foot*) en el lado de «muchas» de la relación, y la utilización de una línea discontinua en el lado de «una». La línea a rayas se utiliza para dar a entender que la relación no es obligatoria en dicho lado (en otras palabras, que es posible que exista una oficina que no tenga asignados representantes de ventas).

Relaciones de muchas a muchas

También puede ser posible que varias filas de una tabla estén relacionadas con varias filas de otra tabla. Consideremos la tabla REP_VENTAS (Figura 2.1) de nuevo. En este ejemplo, supongamos que el análisis de los datos revela que los representantes de ventas trabajan con varias empresas. Además, una única empresa puede servirse de varios representantes de ventas. Por tanto, existe una relación de muchas a muchas entre la entidad REP_VENTAS y la entidad EMPRESAS.

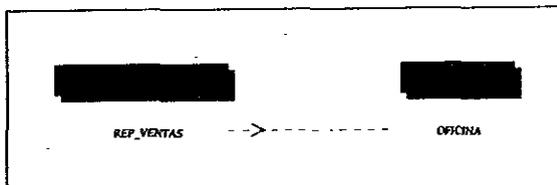


Fig. 2.12. Diagrama de relaciones entre entidades para una relación 1:M.

Para entender esta relación, debe observarse que un único representante de ventas (registro de la tabla REP_VENTAS) puede tener correspondencia con varias empresas (registros de la tabla EMPRESAS), y que también se da el caso contrario. Esta relación se muestra gráficamente en la Figura 2.13

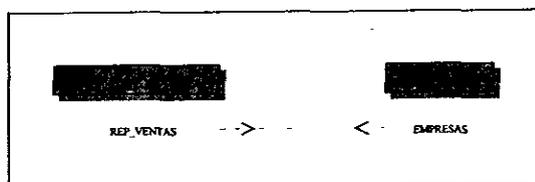


Fig. 2.13. Diagrama de relaciones entre entidades para una relación M:M.

CAPÍTULO III

Presupuesto de Egresos de la Federación

III.1 Introducción

El Presupuesto del Sector Público es el instrumento del Gobierno que se constituye en el principal movilizador de la administración del Estado y alrededor de él se desenvuelve gran parte de la vida económica del país. Son pocas las decisiones cotidianas de la política y administración que no están vinculadas con los ingresos o con los egresos del presupuesto. Cuando el Gobierno crea o suprime impuestos, o varía las tasas de los mismos; cuando aporta recursos para el sostenimiento de las instituciones educativas; cuando aumenta sus puestos de trabajo o mejora las remuneraciones de sus empleados; cuando decide llevar adelante un proyecto de inversión, está influyendo de distintas formas en el desarrollo de las actividades productivas y financieras del país y contribuye al bienestar económico y social de la población. De allí la importancia del presupuesto y su gravitación en el desarrollo nacional.

Cada día es mayor la preocupación por el imparable aumento del gasto público. Su constante incremento provoca fuertes presiones a la Secretaría de Hacienda. Ante esta presión, las autoridades toman las medidas para frenar el gasto público hasta lo imposible, para reducir el déficit y su impacto negativo en el financiamiento de la economía.

Esta política presupuestaria de restricción prolongada es un arma de doble filo; aunque puede tener un efecto positivo e inmediato sobre el déficit presupuestario, también supone un gradual deterioro en la prestación de servicios públicos, sobre todo en aquellas áreas de gestión del Gobierno que son altamente sensibles al comportamiento de la sociedad (Servicios Sociales).

La preparación y elaboración del presupuesto del sector público constituye un “*acto político*” en el cual se revisan, orientan y se asignan recursos para el cumplimiento de los fines del Estado. Para este propósito la decisión de “*asignar recursos escasos*” a “*finés crecientes*”, requiere de un cuidadoso análisis que otorgue al político los mayores y mejores elementos informativos para la toma de decisiones; para ello, la forma y contenido del presupuesto se presenta como la alternativa que pueda hacer compatible la solución cálida y duradera a la necesidad de cobertura a ambos déficits, el de financiación y el de prestación de servicios públicos.

El principal fundamento del nuevo enfoque del presupuesto para el sector público, persigue brindar un esquema que permita racionalizar las decisiones en materia presupuestaria, para un manejo más transparente, ordenado y disciplinado del ingreso y del gasto público.

En la actualidad, mediante **Leyes Anuales de Presupuesto**, el legislativo responsabiliza al funcionario público de mantenerse bajo los límites del gasto específicamente autorizado. Este concepto no hace relevantes la oportunidad y el propósito con que se utilizan los fondos públicos, lo cual significa que la responsabilidad no es por el objetivo obtenido, sino por la honestidad y la legalidad con que se gestionan los fondos públicos.

El nuevo Proceso de Administración Financiera Integrada redefine el papel esperado del funcionario público en el cumplimiento de su gestión. Más allá del cumplimiento de formalidades y paternalismos administrativos, la administración se siente obligada a responder a enormes y variadas demandas socioeconómicas, las que deberán atenderse bajo la siempre situación de escasez de recursos obligando a la necesidad de aplicar criterios de economicidad, transparencia y sana administración en la gestión de los fondos públicos. Este concepto exige que los organismos rectores, integrantes de la administración financiera, deben disminuir su paternalismo e intervencionismo en la gestión y pasar a ser órganos de normatividad, seguimiento y evaluación de los instrumentos a su cargo; para lo cual deberá aplicarse el principio de la normatividad centralizada y la descentralización operativa y de procedimientos. Lo anterior en forma aparente se ve como algo sencillo, pero es necesario considerar que se estarían cambiando procedimientos y tutelajes de treinta o más años, lo que obligará a ser cautelosos en la instrumentación y tener una estrategia adecuada a las circunstancias.

Otro aspecto que es necesario considerar, es la necesidad de tener la clara concepción de que el cambio de la estructura y metodología del presupuesto por si solo, no traerá como consecuencia el cambio automático en el comportamiento de los participantes de la gestión del presupuesto; es necesario que el cambio técnico vaya acompañado de esfuerzos claros y ejemplares de una nueva política presupuestaria orientada a acabar con el incrementalismo, la aceptación de una estructura rígida del gasto, la asignación de recursos en función de las presiones existentes, y en su lugar dar los pasos de una estrategia de reorientación gradual del gasto, incorporación de criterios de eficacia y eficiencia para la asignación de recursos y, en general, cambiar la idea de que el órgano del presupuesto cumple en forma exclusiva el rol de guardián de los fondos públicos, orientándolo a un organismo de análisis y de apoyo a la gestión presupuestaria del sector público.

También es necesario transformar al presupuesto en un instrumento útil a la gestión pública, lo cual implica orientar los trabajos de la reforma presupuestaria en el sentido de acercar más el presupuesto al trabajo de los directivos del sector público, a fin de que sea utilizado en la toma de decisiones, pero no sólo en el acto formal de presentar un anteproyecto, sino como un instrumento de seguimiento y evaluación de los objetivos y resultados previstos. Para esto es necesario simplificar la estructura y métodos del presupuesto hasta ahora enmarcados en una excesiva e inútil especialización que complica la comunicación, creando aislamiento entre los ejecutores y los especialistas en presupuesto.

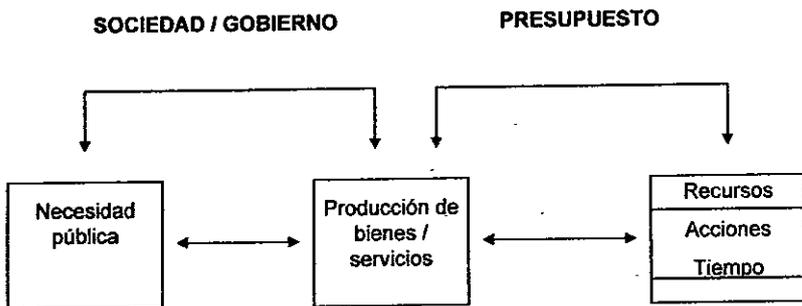
Para cumplir este objetivo será necesario mejorar la ejecución presupuestaria a través de métodos y procedimientos que impliquen su descentralización operativa, bajo el concepto de que los ejecutores de las unidades presupuestarias asumen un rol más directo en la gestión y administración de sus recursos. Este procedimiento será acompañado de un seguimiento y evaluación presupuestaria de sencilla aplicación y de manejo oportuno.

El diseño y aplicación del modelo presupuestario debe entenderse como un proceso progresivo y flexible que en su instrumentación vaya ajustándose a las variaciones del contexto; no considerarlo así puede llevar a precipitar decisiones inadecuadas que crearían graves problemas en el sector público.

III.2 Concepto de Presupuesto

El Presupuesto es un instrumento en el cual se asigna recursos para el cumplimiento de los objetivos establecidos en los fines o funciones de una institución. Si aplicamos este concepto al ámbito del sector público: “El presupuesto será la herramienta que le permite cumplir con la producción de bienes y servicios públicos para la satisfacción de las necesidades de la población al rol asignado al estado en la economía y sociedad del país”.

Fig. 1



De la definición empleada se extrae que el presupuesto moderno tiene dos elementos fundamentales, uno es los “*Propósitos o Resultados Esperados*” y el otro “*La Cuantificación de los Recursos Necesarios*”. Esta ecuación *Propósitos y Recursos* es la que da sentido al presupuesto; de no ser así, sólo es un listado de compras del Gobierno que debe realizar en forma anual.

Otro aspecto que es necesario considerar en esta concepción del presupuesto moderno es que no se trata del cumplimiento de un proceso de cálculo de los recursos que tiene un plazo determinado en el tiempo (Proceso de elaboración de proyectos de presupuestos) sino más bien es un proceso “*vivo, permanente y flexible*” que exige otra dinámica al “*ritual*” presupuestario de una sola etapa de trabajo.

La elaboración de un presupuesto para un año objetivo (n+1) se hace tomando en cuenta los resultados del último ejercicio anual cerrado y también en base a los resultados “*parciales*” del presupuesto en vigencia de la manera como se muestra en la figura 2.

Figura 2



Por lo expresado, el presupuesto debe constituir un elemento fundamental de la gestión pública que la permita al funcionario:

- Estar informado de los resultados previstos.
- Conocer la cuantía de sus recursos en cualquier momento.
- Poder tomar medidas correctivas en función de su avance de resultados.
- Reprogramar los objetivos no alcanzados, en ejercicios presupuestarios futuros.

III.3 Alcances Conceptuales del Presupuesto

III.3.1 Alcance Político

Gobernar implica tomar decisiones por medio de políticas globales y específicas para dirigir la acción del sector público.

Estas políticas pueden ubicarse en dos planos, uno que tiene que ver con la producción de los servicios públicos que le corresponden al estado, y otro con las políticas vinculadas al manejo de su administración presupuestaria; entre los primeros, se puede citar a la política económica, fiscal, sectorial, regional; y entre los segundos, a la política salarial, presupuestaria, de inversión pública, de crédito público.

Ambos tipos de política encuentran en el presupuesto el instrumento que interrelaciona sus alcances y hacer factible su coherencia a través del análisis de las magnitudes de sus intenciones con las posibilidades de recursos existentes en la economía del sector público.

III.3.2 Alcance Administrativo

Definido el marco político del presupuesto, es necesario encontrar su expresión "*operativa*" que signifique la concreción de las acciones y recursos necesarios para la obtención de los objetivos explicitados en las políticas de Gobierno.

El concepto moderno de la administración rebasa la función operativa tradicional de tomar decisiones de conformidad a los asuntos que se presentan; en su lugar el administrativo moderno deberá: planear, dirigir, coordinar, informarse, supervisar, evaluar y presupuestar las actividades a su cargo, de tal manera que disponga en forma previa a la ejecución de las acciones de los elementos necesarios para racionalizar sus decisiones y cumplir sus responsabilidades con eficacia y eficiencia.

El presupuesto cumple un rol fundamental para la administración al constituir una "*guía de gestión*" que elimine o minimice la necesidad de tomar decisiones improvisadas.

III.3.3 Alcance Económico y Financiero

A través de sus variables básicas, ingreso, gasto y funcionamiento, el presupuesto tiene efectos económicos y financieros en el sistema nacional.

En lo referente a lo económico, el ingreso público, principalmente los tributarios, tiene en primer lugar una función de contracción de la demanda de los agentes de la economía, al retirárseles un porcentaje de sus recursos a través de los diferentes impuestos. Pero también existe la función de "*redistribución*" al emplear ese "*retiro de recursos*" en la satisfacción de necesidades básicas de la población.

Por otro lado, el sector público es el mayor empleador del país; con esa fuerza de trabajo, además de la ocupación moviliza la producción de la economía, puesto que genera un instrumento de la demanda de bienes y servicios. También, al realizarse las obras públicas a través de la ejecución de los proyectos de inversión, se genera, en primer lugar, una *“acumulación o ampliación del capital nacional”*, y en segundo lugar *“se amplía la capacidad de producción del sector público”*.

Desde el punto de vista financiero el presupuesto significa origen y destino de flujos financieros como contrapartida al movimiento económico generado por sus variables reales. Estos flujos tienen un efecto principal en el financiamiento de la economía a través de la política monetaria y crediticia del país; por lo tanto, las dimensiones de ingreso y gasto público deben hacerse por medio de un cuidadoso manejo que no ponga en riesgo un sano financiamiento de la economía.

III.3.3 Alcance Jurídico

El presupuesto finalmente se traduce en una norma legal que proviene de las entidades legislativas que representan la voluntad popular del país; por lo tanto, la sanción del presupuesto *significa un análisis, reajuste y regulación de las acciones que el poder ejecutivo tiene previsto realizar en un período de tiempo.*

En otras palabras, el sentido jurídico del presupuesto implica los límites de las acciones que debe realizar la administración pública para el cumplimiento de las funciones que le corresponden.

También en el dispositivo de aprobación del presupuesto se establecen *“normas”* que regulan la administración de los recursos públicos, así como se fijan las respectivas responsabilidades por ese manejo administrativo.

III.4 Principios del Presupuesto

Programación

Por su propia naturaleza el presupuesto tiene un contenido y forma de programación, es decir, se deben expresar con claridad los objetivos seleccionados, las acciones necesarias para alcanzarlas, los recursos estimados expresados en unidades respectivas y traducirlos a variables monetarias, lo que origina la necesidad de asignaciones presupuestarias.

Equilibrio

Se refiere a la cobertura financiera del presupuesto. El presupuesto debe formularse en condiciones que el total de sus egresos sean equivalentes al total de los ingresos estimados a recolectarse y cualquier diferencia debe ser financiada por recursos provenientes de la capacidad de endeudamiento en los términos y niveles más realistas y concretos.

El no cumplimiento de este principio significa dimensiones falsas del presupuesto que pueden originar problemas económicos y monetarios al país.

Racionalidad

Relacionado con la “austeridad” en el manejo de los recursos a través de elaborar presupuestos en condiciones de evitar el gasto suntuario, así como impedir el desperdicio y mal uso de los recursos. Por otro lado, se relaciona con la aplicación de la “economicidad” en la satisfacción de las necesidades públicas, es decir, tratar de obtener lo más que se pueda gastando lo menos y mejor posible.

Ambito Universal

Está referido al alcance institucional del presupuesto, en el sentido de que se debe incorporar al presupuesto todo lo que es materia de él, es decir, no debe quedar ninguna institución pública en forma extrapresupuestaria. Por otro lado, se refiere a la “unidad” del documento en cuanto a la integración de sus componentes para darle una “coherencia total”, y también a la unidad metodológica empleada en el proceso presupuestario.

Transparencia

Tiene que ver con la claridad, acuciosidad y especificación con que se expresan los elementos presupuestarios. Los documentos presupuestarios deben expresar en forma ordenada y clara todas las acciones y recursos necesarios para cumplir con los objetivos.

En lo referente a la acuciosidad, el presupuesto debe expresarse con profundidad, sinceridad y honestidad.

Por último, el presupuesto debe “*especificar*” con precisión y realismo todos los flujos de ingresos y egresos, así como el financiamiento posible.

Flexibilidad

El presupuesto no debe contener “*rigideces*”, ni en su estructura ni tampoco en sus componentes. En su estructura, está debe ser capaz de ser reorientada, modificada de acuerdo a la evaluación de los resultados, lo cual implica que no deben existir ni ingresos rígidos, ni gastos no modificables.

El otro sentido de este principio tiene que ver con la “*Unidad de Caja*”, es decir, los ingresos deben constituir un sólo fondo de cobertura de los egresos y la asignación debe hacerse en base a prioridades y competencias, lo que implica que no existan ingresos “*pre-asignados*” (afectando al gasto).

Difusión

El contenido del presupuesto debe ser ampliamente difundido ya que constituye el esfuerzo del gobierno por atender los servicios que le han sido encomendados y los cuales son “*financiados*” con las contribuciones de todos los ciudadanos, a través del pago de sus impuestos y tarifas de servicios públicos.

Estos principios constituyen los “*postulados*” básicos para que el presupuesto encuentre su pleno desarrollo en todo su proceso.

III.5 Proceso Presupuestario

Está conformado por un conjunto de etapas de naturaleza flexible, dinámica y continua, que están interconectadas, son interdependientes e interactuantes, con lo cual este proceso constituye una unidad y el tratamiento por separado de sus etapas es sólo para fines de mejorar el análisis y facilitar la coordinación con los otros sistemas que tienen que ver con el presupuesto en la administración financiera.

Una Visión del Proceso Presupuestario se presenta en la figura siguiente:

Figura 3
PROCESO PRESUPUESTARIO



Formulación del Presupuesto

En forma aparente esta fase constituye el primer paso para el desarrollo de los trabajos del proceso presupuestario, lo cual en realidad no es así. En efecto, para realizar una correcta programación y formulación presupuestaria es necesario realizar un minucioso análisis de los resultados de ejercicios presupuestarios anteriores que sustenten la estimación de los elementos del “nuevo” presupuesto; también es necesario evaluar qué está pasando con el presupuesto vigente a efectos de relacionar lo deseable con lo posible (ejecución presupuesto vigente). De esta relación, incluso, pueden salir medidas correctivas tanto para el ejercicio en vigencia como para su incorporación en el presupuesto futuro.

Metodológicamente, formular un presupuesto significa cumplir algunos pasos que implica efectuar trabajos de tipo especializado de acuerdo a las características institucionales y también de conformidad al marco legal. En general se pueden distinguir los siguientes:

- Estimación del contexto económico y financiero del presupuesto. (Marco Macroeconómico).
- Determinación de la política presupuestaria.

- Elaboración de los proyectos de presupuesto de cada Dependencia.
- Elaboración del proyecto de ley del presupuesto.

Estimación del Contexto Económico y Financiero del Presupuesto

La determinación, en una primera aproximación, del marco económico y financiero del presupuesto permitirá enmarcar todos los trabajos de formulación en estas orientaciones. A esta actividad también se la conoce como **La Programación Presupuestaria**.

En cuanto a los aspectos económicos, en primer lugar será muy importante estimar el comportamiento de la economía, con sus principales variables, indicadores, políticas y orientaciones que sirven para determinar el ámbito de las acciones del sector público.

Derivados de los objetivos, metas y políticas de la economía, se determinan los recursos financieros que dan soporte a la estimación del comportamiento económico. En este sentido será indispensable disponer de los montos de recursos financieros que serán destinados al sector público, procedentes de la tributación, de los precios y tarifas de los servicios públicos y, por captación y movilización de ahorro, tanto del sector externo como del mercado interno de capitales.

Con estos elementos se podrá realizar en una forma exploratoria la compatibilización Ingresos, Gastos y Financiamiento para definir el nivel global del presupuesto preliminar. Esta actividad permite dar orientación y consistencia a los pasos siguientes en la formulación del presupuesto.

Definición de la Política Presupuestaria

La política presupuestaria debe determinar las orientaciones prioridades, techos y variables básicos, así como las normas, métodos y procedimientos para la elaboración de los proyectos de presupuesto en cada institución. Para definir la política presupuestaria es necesario realizar una serie de trabajos técnicos y consultas políticas a efecto de que el presupuesto se haga sobre bases firmes y consistentes.

En estos trabajos se incluye la estimación de los ingresos del sector público, la que debe incluir los instrumentos legales en caso de variación de sus principales componentes; la estimación del gasto total y su estructura, tanto en términos de variables financieras como destino institucional y/o sectorial; la determinación de la posible captación de financiamiento interno y externo, y las orientaciones de prioridades en razón de los planes de gobierno y de desarrollo económico y social. En síntesis, una política presupuestaria completa debe precisar:

- Políticas y orientaciones de tipo fiscal.
- Techos o rangos de gasto, de ingresos y de financiamiento.
- Prioridades nacionales e inter e intra-sectoriales.
- Normas para presupuestación (regulaciones para ingresos y gastos).
- Métodos y procedimientos (formatos e instrucciones).
- Indicadores socioeconómicos para la formulación.
- Cronogramas de trabajo y responsabilidades.

Elaboración de los Proyectos de Presupuesto de cada Dependencia

La elaboración de los proyectos de presupuestos en las instituciones es un paso que moviliza a todo el aparato administrativo del sector público en una época determinada de cada año, que implica una serie de actividades y negociaciones en diferentes niveles de las organizaciones administrativas. Esta actividad comprende los pasos siguientes:

- Estimaciones preliminares realizadas por las instituciones antes de conocer la política presupuestaria. (*Programación Institucional*).
- Conocimiento, difusión y aplicación de la política presupuestaria.
- Análisis y compatibilización de las estimaciones preliminares y la política presupuestaria.
- Formulación de reajustes institucionales, así como orientación y asesoría para la elaboración de los proyectos en cada unidad presupuestaria.
- Elaboración de los proyectos en cada unidad presupuestaria.
- Revisión, reajustes e integración del proyecto de presupuesto institucional.
- Remisión a la Secretaría de Hacienda del proyecto de presupuesto de la institución.

Por otro lado, en cada unidad presupuestaria se deben realizar los trabajos técnicos siguientes:

- Evaluación de los resultados efectuados del último año cerrado contablemente (Año n-1); esta evaluación debe comprender la revisión y análisis de los resultados físicos (propósitos) y de la administración de los recursos (ingresos y gastos).
- Análisis de la ejecución presupuestaria del año en curso (Año n), que tendrá el mismo alcance de la evaluación anual pero referida al avance logrado en el año hasta la fecha en la que se elabora el proyecto.
- Estimación inicial de los propósitos a lograr para el presupuesto del año objetivo (Año n+1), que comprende determinar los objetivos, metas y recursos financieros de acuerdo a una primera aproximación del plan de trabajo para el año n+1.
- Comparación de las estimaciones preliminares con los resultados de las evaluaciones efectuadas (Año n-1 y Año n) y determinación de los ajustes y medidas correctivas a incorporarse en el proyecto de presupuesto.
- Elaboración en la unidad ejecutora del proyecto de presupuesto reajustado y de conformidad con la política presupuestaria.

Estas acciones deben ser apoyadas por las unidades del sistema de presupuesto en lo referente a: difusión y capacitación de la política presupuestaria, asesoría en cuanto al manejo de los formatos e instructivos y para la coordinación interna y externa de la institución.

El proceso de elaboración de los proyectos de presupuesto en las instituciones, inicialmente tiene la característica de ser descendente en cuanto a los niveles de la organización. Esto se debe al sentido de orientación de los niveles más altos hacia los de tipo operativo; la respuesta de estos niveles son las propuestas de los proyectos que se hacen con sentido ascendente y dependerá de las negociaciones y reajustes al número de “*subidas y bajadas*” de los proyectos de las unidades ejecutoras.

III.6 Proyecto de Presupuesto de Egresos de la Federación

De los capítulos anteriores obtenemos un concepto general de presupuesto que es el siguiente “El presupuesto se define como el calculo anticipado del conjunto de los gastos e ingresos previstos para cubrir las obligaciones, compromisos y las operaciones del sector público, para un ejercicio fiscal”.

El Proyecto de Presupuesto de Egresos de la Federación (PPEF) es la propuesta de asignación de recursos públicos que prepara el Poder Ejecutivo y somete a consideración de la H. Camara de Diputados (HCD), para avanzar en satisfacer las aspiraciones de bienestar de la sociedad, expresadas en el Plan Nacional de Desarrollo y sus Programas Sectoriales.

El presupuesto de Egresos de la Federación es el documento por medio del cual la HCD autoriza a los tres poderes de la unión a ejercer los recursos públicos, de acuerdo a una distribución con cargo a la recaudación y al financiamiento. El uso de dichos recursos permite dar cumplimiento a los objetivos de las políticas públicas.

La formulación del Proyecto de Presupuesto de Egresos (PPEF) es una tarea que le corresponde a la Secretaría de Hacienda, quien se apoya técnicamente en la Unidad de Política y Control Presupuestal (UPCP). Básicamente el trabajo consiste en la revisión de los proyectos de presupuesto en cuanto a que su contenido está de acuerdo en la política presupuestaria; en caso negativo, se deben realizar reuniones de discusión y análisis para determinar las causas que motivan la no consideración de la política presupuestaria en la formulación de los proyectos; una vez efectuados los reajustes de los proyectos, se procede a elaborar PPEF que en términos básicos debe contener lo siguiente:

- Mensaje o exposición de motivos.
- Proyecto de ley de aprobación del presupuesto de ingresos y egresos para cada institución.
- Normas que regulan la ejecución presupuestaria.
- Cuadros, resúmenes y anexos.

Aprobación del Proyecto de Presupuesto de Egresos de la Federación

Esta es una fase eminentemente política que concluye con la sanción del documento presupuestario. Sin embargo, es recomendable que la aprobación del nuevo presupuesto se haga en un periodo que deje un adecuado espacio de tiempo para el conocimiento del presupuesto por las unidades ejecutoras y para la preparación de la ejecución presupuestaria en forma oportuna.

En cuanto al nivel de aprobación, éste debe ser el que otorgue la adecuada transparencia del contenido presupuestario, pero manteniendo un margen para la flexibilidad operativa a la administración durante la ejecución presupuestaria.

Ejecución Presupuestaria

La ejecución del presupuesto constituye la movilización de los recursos a efecto de llevar a cabo las acciones y procesos previstos en el plan de trabajo para lograr los resultados en términos de objetivos y metas que figuran en el presupuesto aprobado.

Desde el punto de vista financiero, la ejecución presupuestaria comprende al presupuesto de ingresos como al presupuesto de egresos. La coordinación de ambos permitirá que los resultados se obtengan en la forma más adecuada y oportuna.

El nuevo enfoque del presupuesto, plantea que esta fase se realice siguiendo el principio de *"Centralización Normativa y Descentralización Operativa"*.

La centralización normativa implica dar orientación e integrar desde el punto de vista de políticas, normas y procedimientos del funcionamiento de cada uno de los subsistemas, la cual se realizará por medio de los órganos rectores de la Administración Pública, quienes garantizarán que se establezcan las normas emitidas en cada una de las instituciones del sector público, constituyendo este paso que brinda un adecuado soporte al control interno.

La descentralización operativa implica delegar en cada institución pública la aplicación de la normatividad emanada de los órganos rectores, con lo cual se persigue *"desburocratizar"* la ejecución del presupuesto para agilizarla, de tal manera que se garantice la obtención de los resultados con una adecuada administración de los recursos.

El método que da una respuesta precisa al principio mencionado es la Programación de la Ejecución Presupuestaria (PEP).

La PEP consiste en previsiones para cada sub-período del ejercicio presupuestario mensual, trimestral, semestral del avance esperado en el plan de trabajo y de las necesidades de recursos financieros; lo que permitirá compatibilizar las necesidades de recursos con las posibilidades financieras y aportar para dichos períodos asignaciones por cada unidad de gestión presupuestaria en forma previa a la ejecución misma. Todas las acciones administrativas de control interna para la ejecución de los egresos serán de responsabilidad de las unidades ejecutoras.

Otro aspecto de la ejecución presupuestaria a considerar en su mejoramiento, es relacionar las posibles modificaciones del presupuesto aprobado, por trasposos y/o esfuerzos, a los resultados del seguimiento y evaluación de la ejecución presupuestaria.

Seguimiento y Evaluación de la Ejecución Presupuestaria

Esta fase comprende el acompañamiento de la ejecución presupuestaria a través de la verificación de los resultados parciales que se van obteniendo en un período de la PEP, así como su análisis al finalizar este período. La finalidad es determinar el comportamiento de los elementos del presupuesto para detectar *desviaciones en la ejecución* y, en caso necesario, aplicar medidas correctivas y oportunas, y realimentar el proceso de la formulación presente y futura del presupuesto. Por lo tanto, el *seguimiento y evaluación* comprenderá las etapas siguientes:

- Medición de resultados de la ejecución presupuestaria.
- Determinación y análisis de las desviaciones presentadas.
- Selección de las medidas correctivas.
- Aplicación de las medidas correctivas.

Tal como se aprecia, el seguimiento y evaluación de la ejecución presupuestaria es un proceso que tiene un ciclo perfectamente marcado donde es necesario dar cumplimiento de todas

sus etapas, porque de otra manera el sistema queda a nivel informativo y sus objetivos se *"diluyen en el tiempo"*.

La determinación de responsabilidades y supervisión de la aplicación de las medidas correctivas podría ser un proceso del sistema de control.

En el capítulo siguiente veremos como se elabora el Proyecto de Presupuesto de Egresos de la Federación mediante el Sistema Integral de Control Presupuestal (SICP) mediante su módulo anteproyecto, que es donde accesan los usuarios para la elaboración del PPEF.

CAPÍTULO IV

Flujo de datos del sistema

IV.1 Flujo de la información

En el presente capítulo se describe el flujo que llevan los procesos de la información así como de los reportes que se emiten para que sean revisados en la Cámara de Diputados.

En los diagramas de flujo se describen las direcciones (en Unix) en las que actualmente están los programas, formas, reportes, listados de reportes etc.

Diagrama general del Anteproyecto del PEF

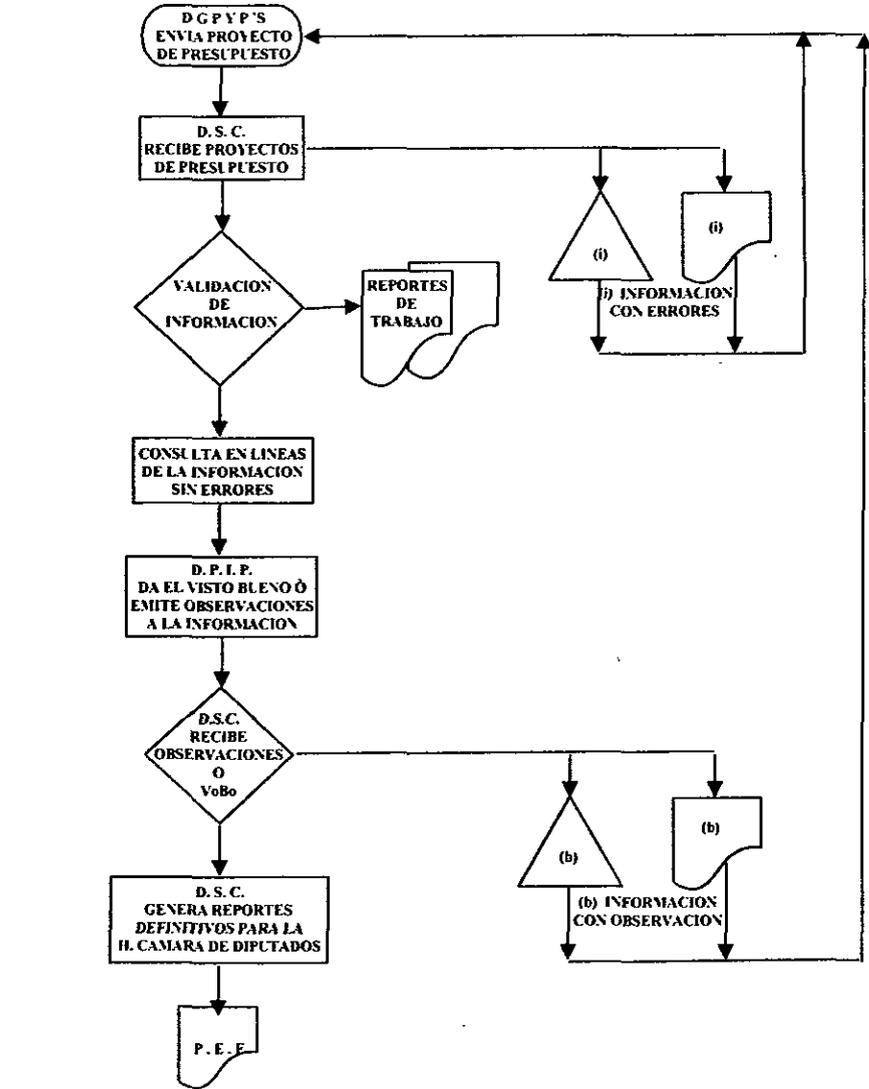


Diagrama general del Anteproyecto del PEF

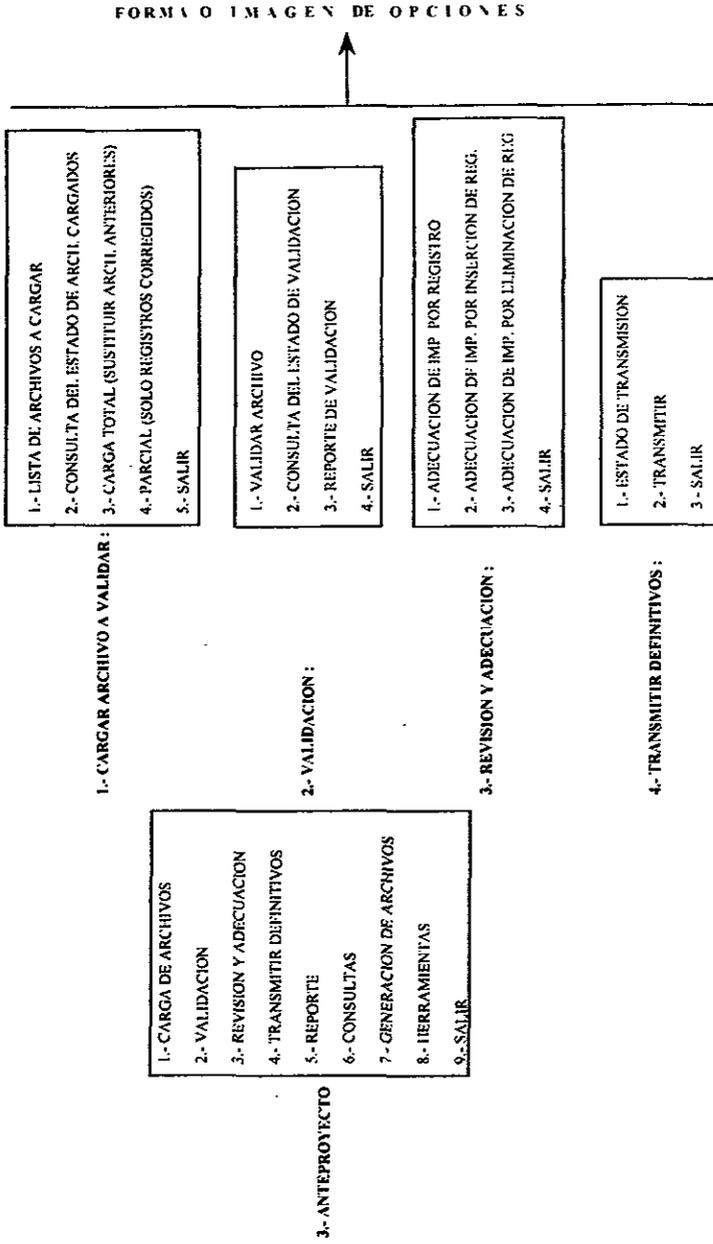


Diagrama general del Anteproyecto del PEF

- 1.- PARTICIPACION DE LA UNIDAD POR CAPITULOS (GENERICO)
- 2.- PARTICIPACION DE LA UNIDAD POR CAPITULOS (RAMO 22)
- 3.- PARTICIPACION DE LA UNIDAD POR CAPITULOS (RAMOS 24 Y 29)
- 4.- RESUMEN PROGRAMATICO ECONOMICO FINANCIERO
- 5.- RESUMEN PROGRAMATICO ECONOMICO FINANCIERO (PARCIAL RAMOS 12)
- 6.- RESUMEN PROGRAMATICO ECONOMICO FINANCIERO (RAMO 12, TP=8)
- 7.- RESUMEN PROGRAMATICO ECONOMICO FINANCIERO (RAMO 22)
- 8.- RESUMEN PROGRAMATICO ECONOMICO FINANCIERO (RAMOS 24 Y 29)
- 9.- RESUMEN CUANTITATIVO DEL RAMO (GENERICO)
- 10.- RESUMEN CUANTITATIVO DEL RAMO (RAMO 22)
- 11.- RESUMEN CUANTITATIVO DEL RAMO (RAMOS 24 Y 29)
- 12.- ASIGNACION PRESUPUESTAL POR PROGRAMA (GENERICO)
- 13.- ASIGNACION PRESUPUESTAL POR PROGRAMA (RAMO 22)
- 14.- SALIR

1.- REPORTES A LA CAMARA :

5.- REPORTES :

- 1.- POR UNIDAD
- 2.- POR UNIDAD CON DESCRIPCION
- 3.- POR PROGRAMA
- 4.- POR PROGRAMA CON DESCRIPCION
- 5.- RESUMEN DE ENTIDAD POR CONCEPTO E IDENTIFICACION DEL GASTO
- 6.- RUBROS DE GASTO
- 7.- SALIR

2.- REPORTES DE TRABAJO :

1.- ESTADO DEL PROCESO :

6.- CONSULTAS :

- 1.- TEMPORAL
- 2.- DEFINITIVO
- 3.- SALIR

- 1 - ESTADO DEL ANTEPROYECTO
- 2 - TOTAL GASTO FEDERAL DE LOS RAMOS
- 3 - RUBROS GLOBALES DEL GOBIERNO FEDERAL
- 4 - TIPO DE PAGO E IDENTIFICADOR DE GASTO POR RAMO
- 5 - CONCEPTO E IDENTIFICACION DE GASTO POR RAMO
- 6 - SERVICIOS OFICIALES (3806)
- 7 - SALIR

2.- INFORMACION :



Diagrama general del Anteproyecto del PEF

1.- GLOBAL
2.- SUBSIDIOS 4200
3.- APOYOS 4300
4.- RESUMEN ADMINISTRATIVA DE TRANSFERENCIAS
5.- SALIR

1.- TRANSFERENCIAS
2.- GENERA SECUENCIAL DE ANTEP. DEFINITIVO
3.- GENERA SECUENCIAL DE PAQUETE SALARIAL
4.- SALIR

7.- GENERACION DE ARCHIVOS:

8.- HERRAMIENTAS :

1.- TEMPORAL
2.- DEFINITIVO
3.- SALIR

1.- RESTAURACION DE AREAS :

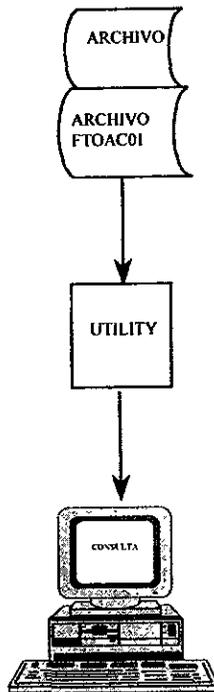
9.- SALIR



CARGA DE ARCHIVOS

LISTA DE ARCHIVOS A CARGAR

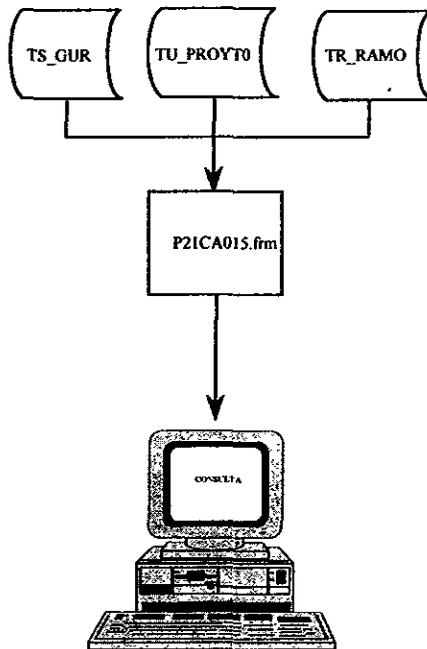
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21CA010



CARGA DE ARCHIVOS

CONSULTA DEL ESTADO DE ARCHIVOS CARGADOS (ACTUAL)

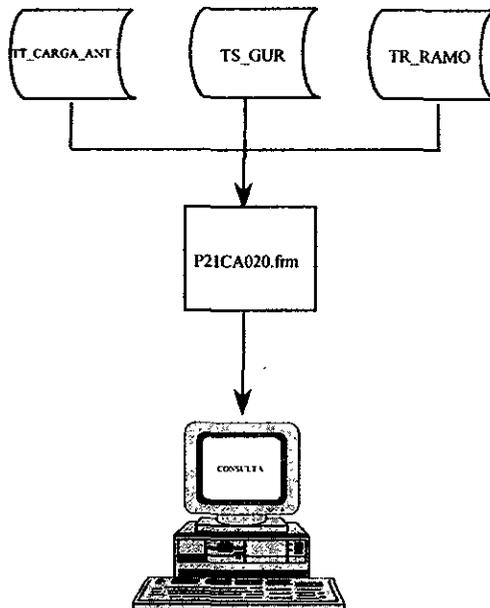
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21CA015



CARGA DE ARCHIVOS

CONSULTA DEL ESTADO DE ARCHIVOS CARGADOS (POR ENVIO)

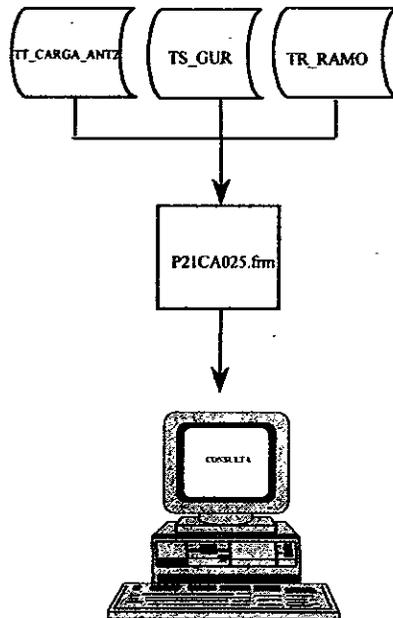
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21CA020



CARGA DE ARCHIVOS

CONSULTA DEL ESTADO DE ARCHIVOS CARGADOS (POR DIA)

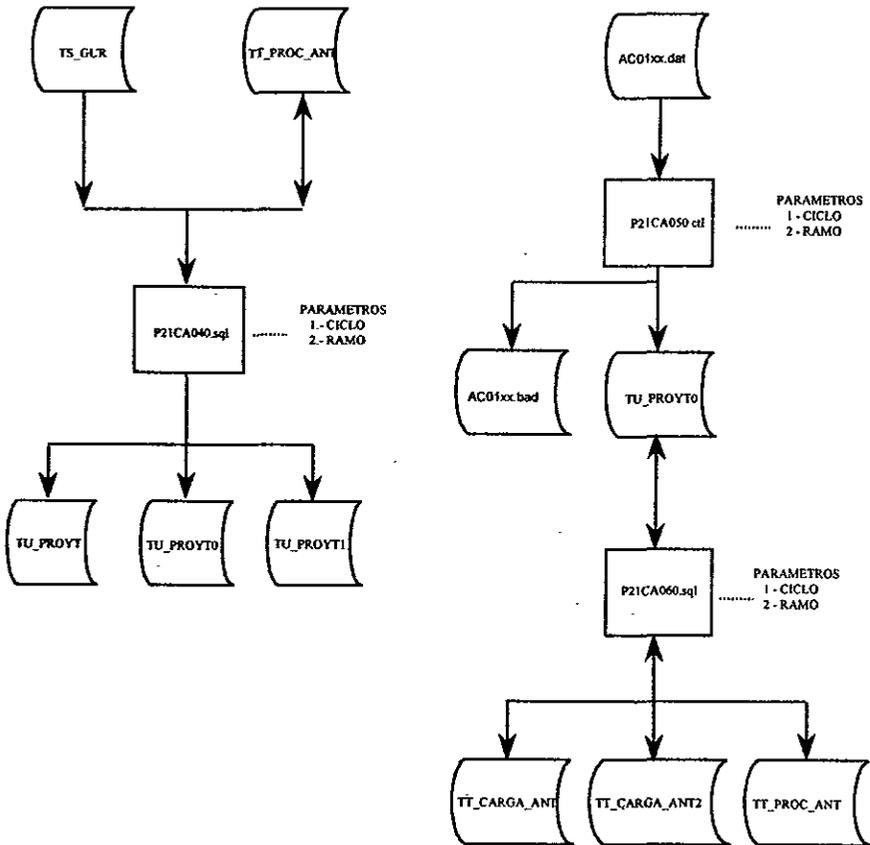
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21CA025



CARGA DE ARCHIVOS

CARGA TOTAL

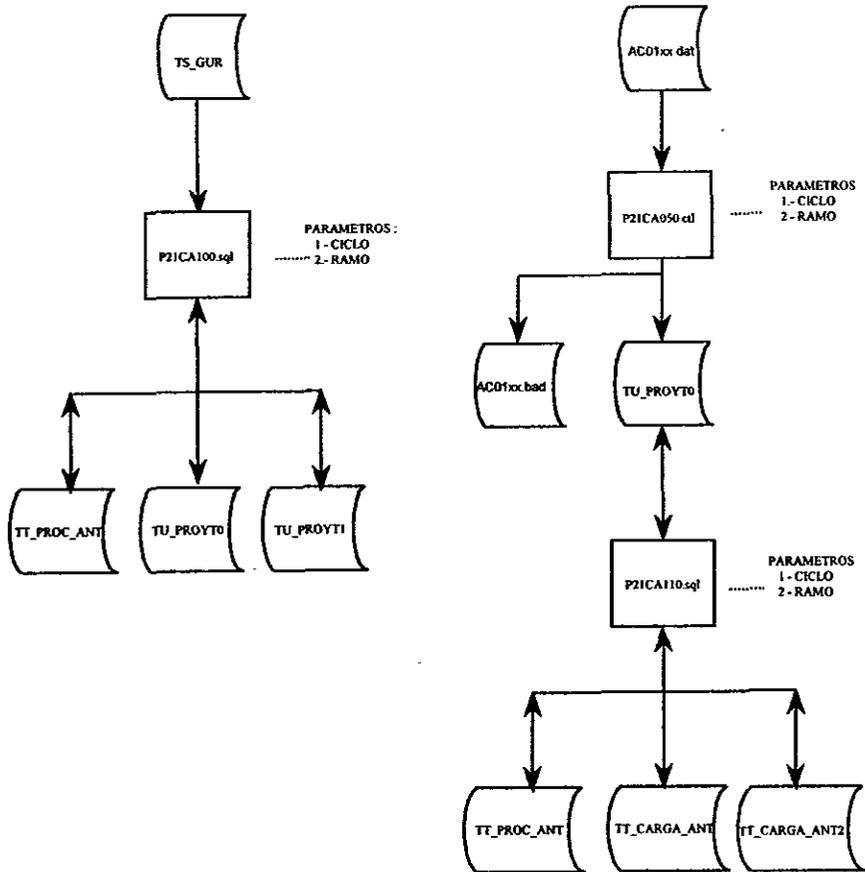
RUTA DE ACCESO /uno/dgpp/prod/shells/P21CA030



CARGA DE ARCHIVOS

CARGA PARCIAL (SOLO REGISTROS CORREGIDOS)

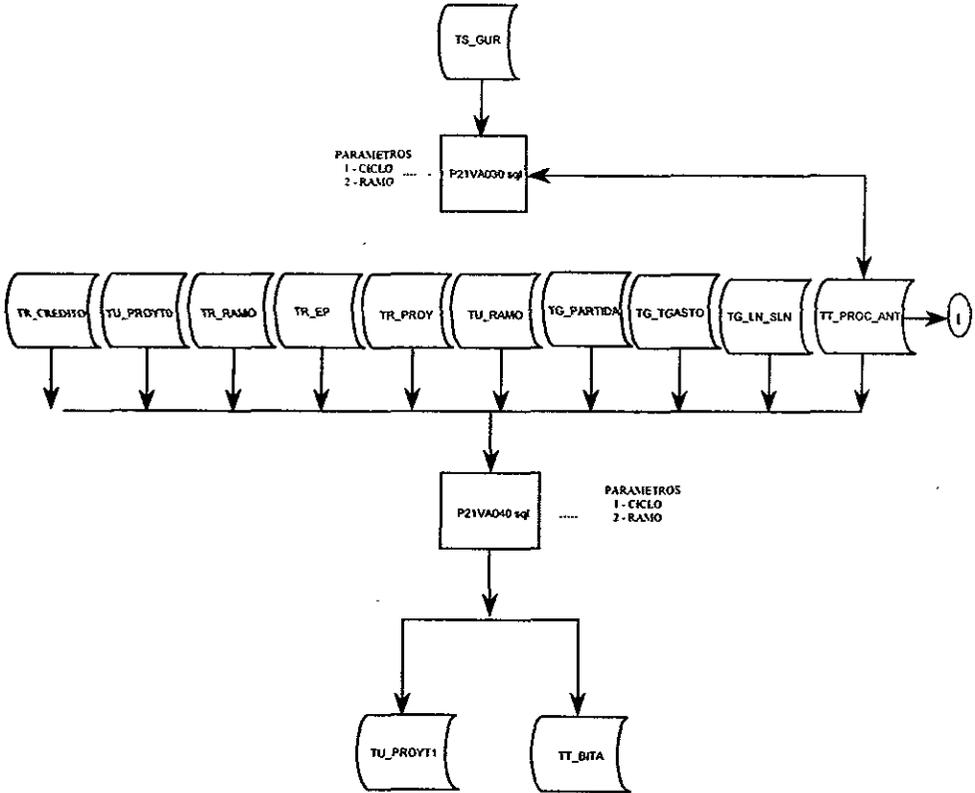
RUTA DE ACCESO /uno/dgpp/prod/shells/P21CA090



VALIDACION

VALIDAR ARCHIVO

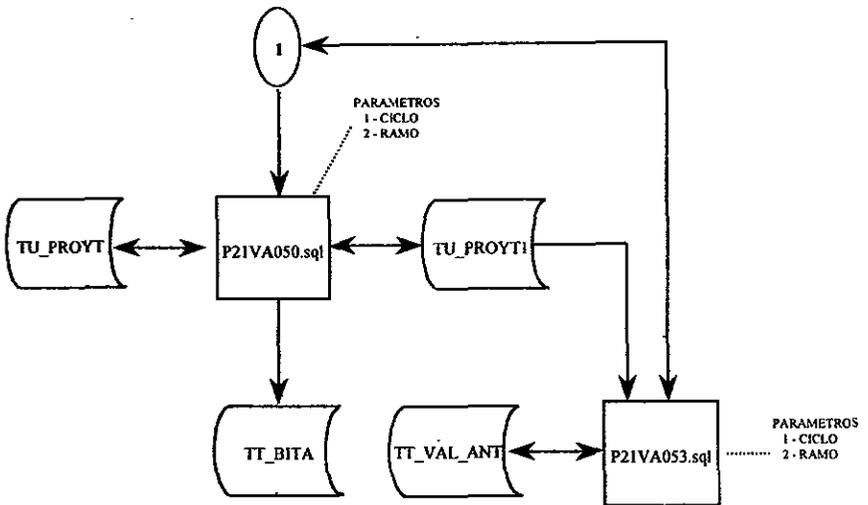
RUTA DE ACCESO /uno/dgpp/prod/shells/P21VA020



VALIDACION

VALIDAR ARCHIVO

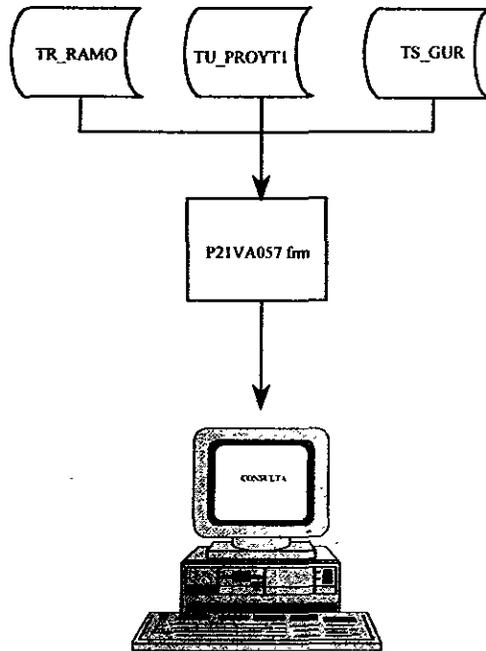
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21VA020



VALIDACION

CONSULTA DEL ESTADO DE VALIDACION (ACTUAL)

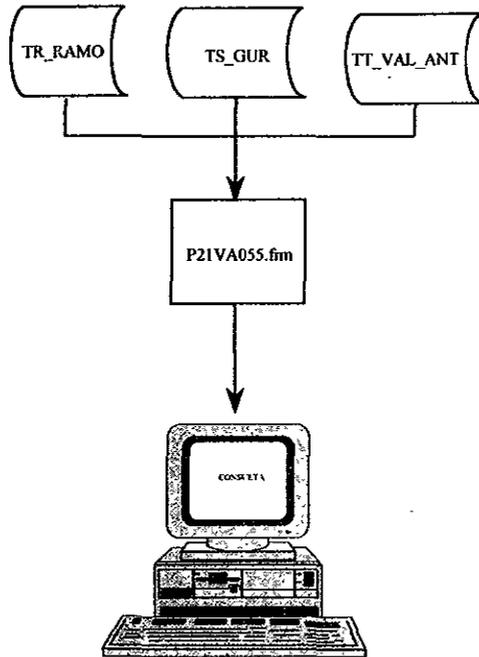
RUTA DE ACCESO : </uno/dgpp/prod/formas/P21VA057>



VALIDACION

CONSULTA DEL ESTADO DE VALIDACION (POR ENVIO)

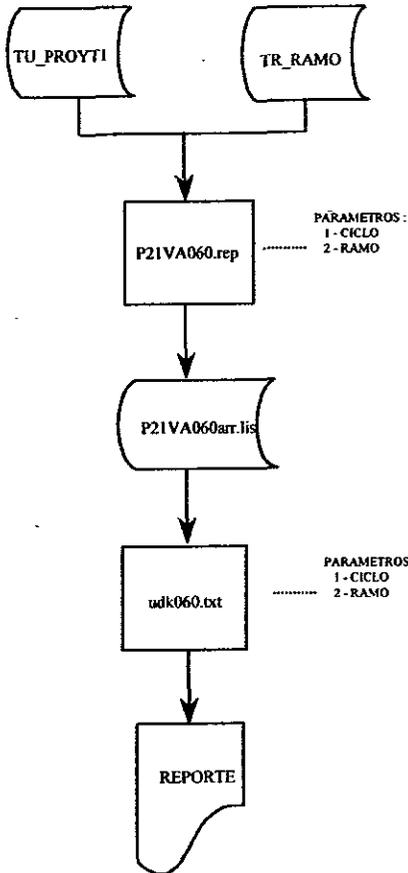
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21VA055



VALIDACION

REPORTE DE VALIDACION

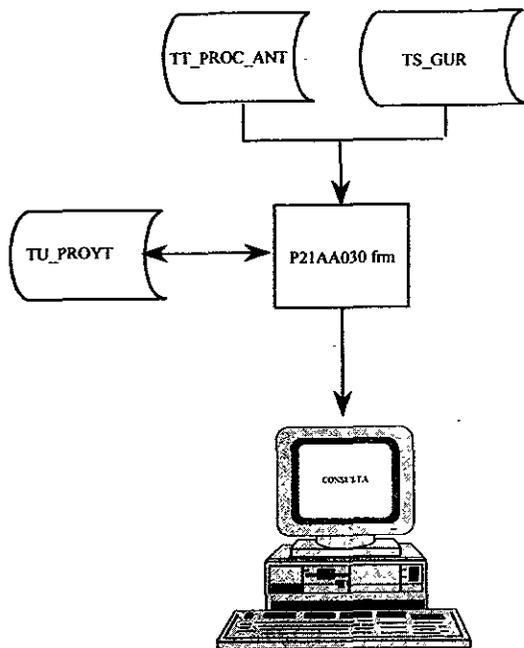
RUTA DE ACCESO : /uno/dgpp/prod/reportes/P21VA060



REVISION Y ADECUACION

ADECUACION DE IMPORTES POR REGISTRO

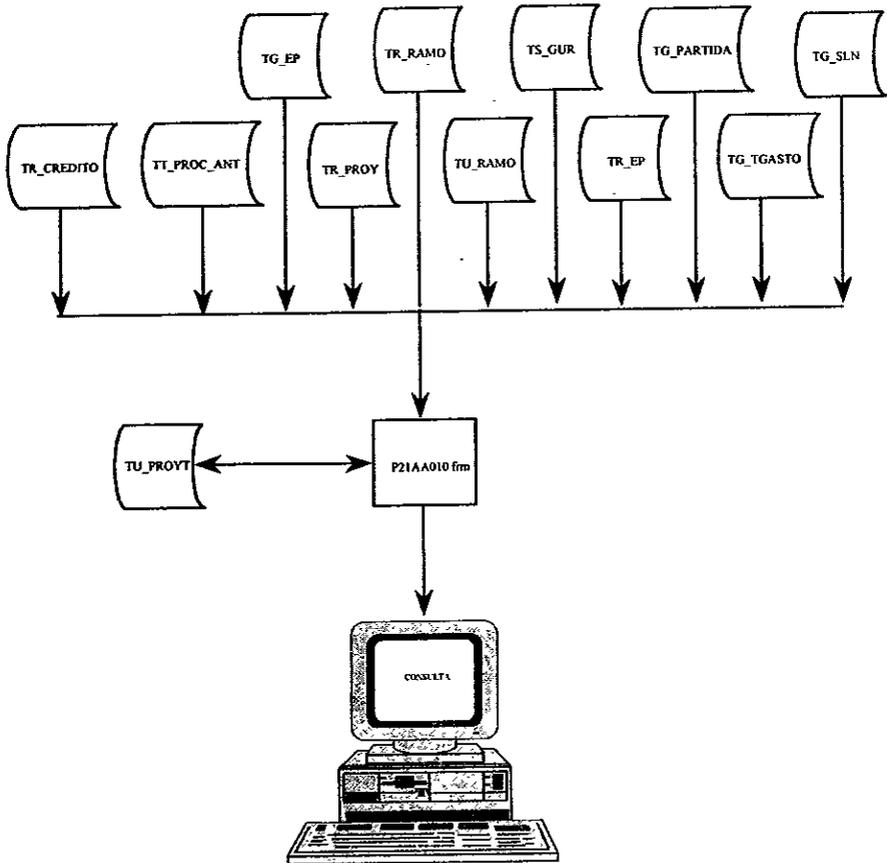
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21AA030



REVISION Y ADECUACION

ADECUACION DE IMPORTES POR INSERCIÓN DE REGISTROS

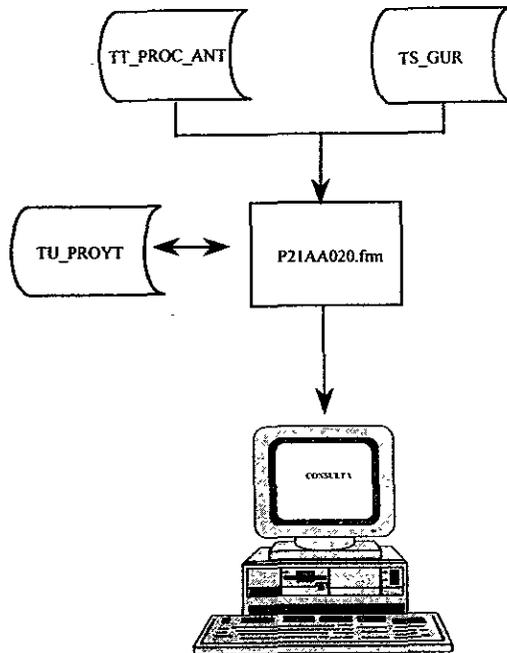
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21AA010



REVISION Y ADECUACION

ADECUACION DE IMPORTES POR ELIMINACION DE REGISTROS

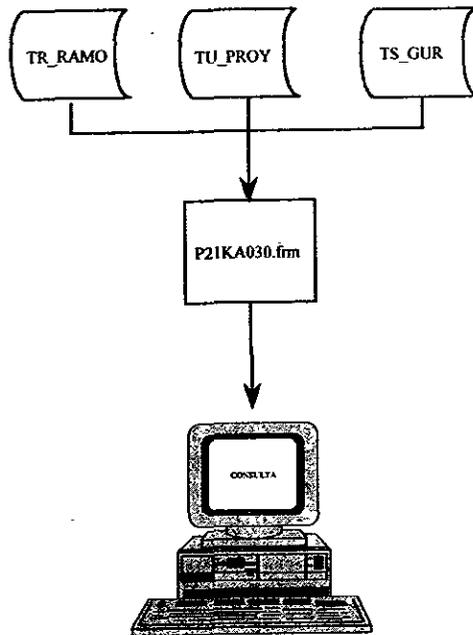
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21AA020



TRANSMITIR DEFINITIVOS

CONSULTA DEL ESTADO DE TRANSMISION

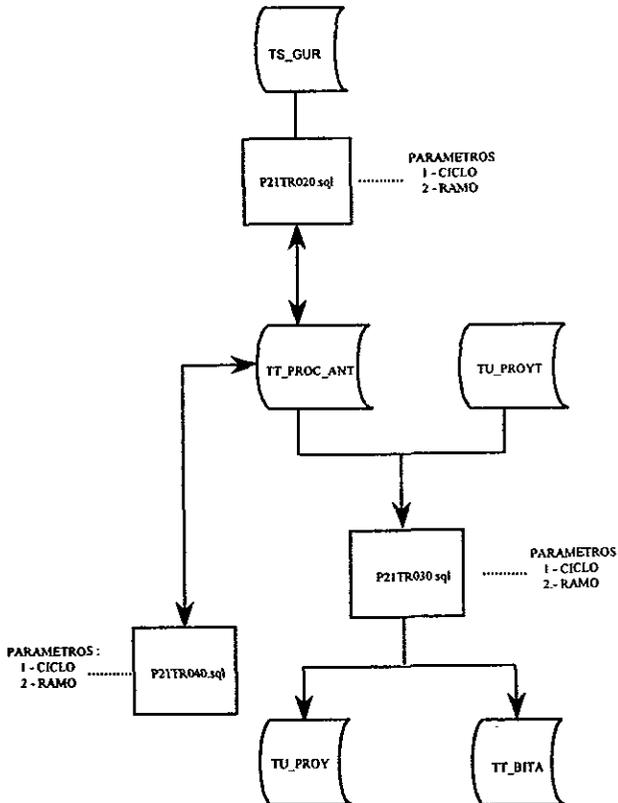
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21KA030



TRANSMITIR DEFINITIVOS

TRANSMITIR

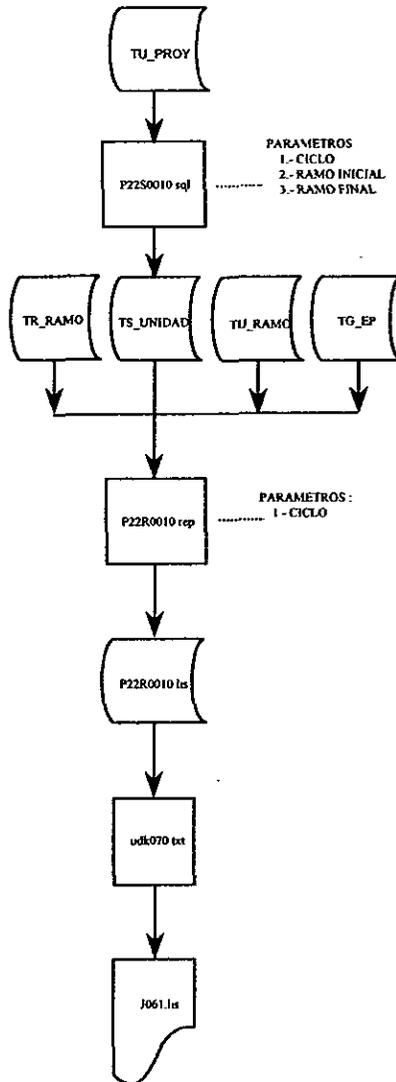
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21TR010



REPORTES A LA CAMARA

PARTICIPACION DE LA UNIDAD POR CAPITULOS (GENERICO)

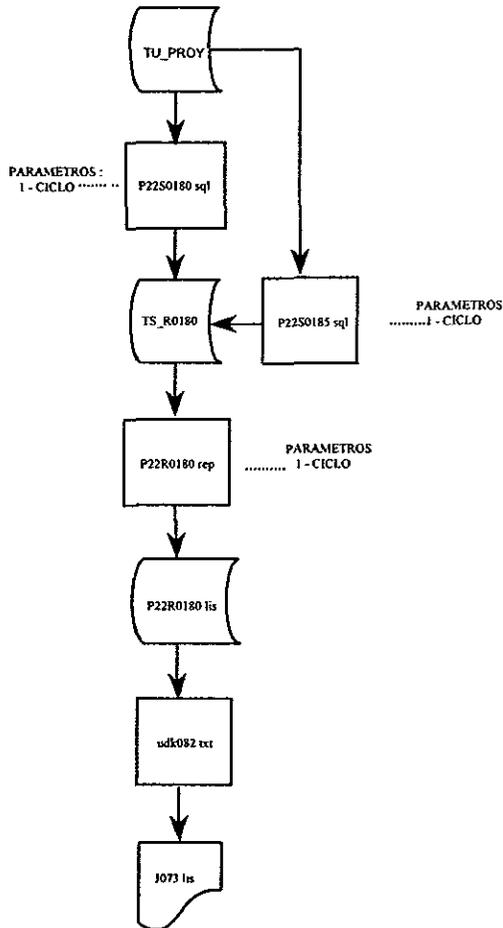
RUTA DE ACCESO : /uno/dgpp/prod/shells/prepp2



REPORTES A LA CAMARA

PARTICIPACION DE LA UNIDAD POR CAPITULOS (RAMO 22)

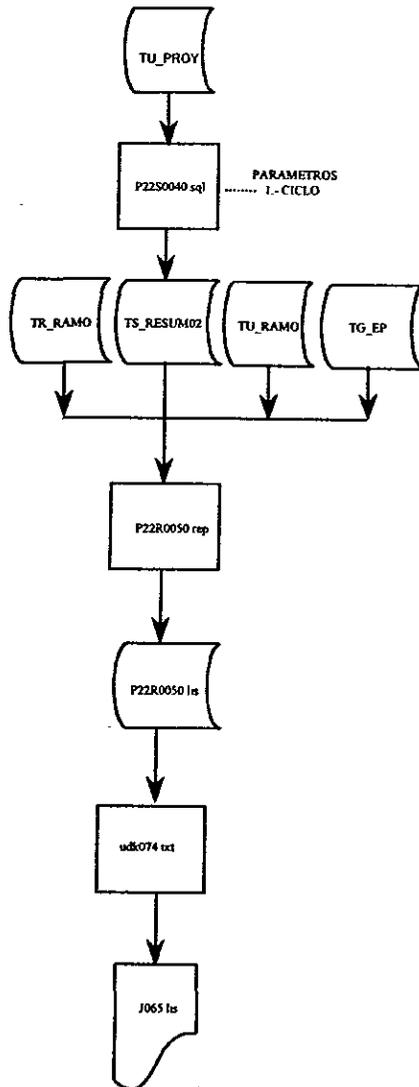
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0180



REPORTES A LA CAMARA

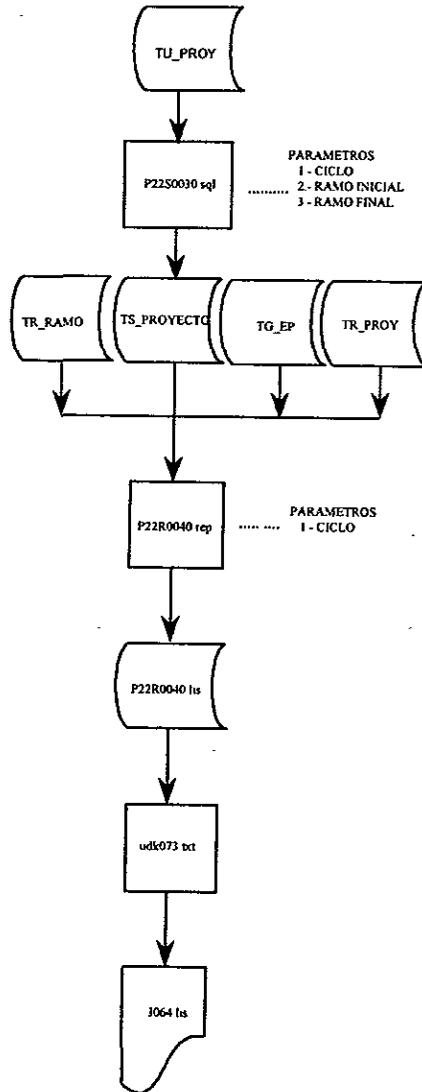
PARTICIPACION DE LA UNIDAD POR CAPITULOS (RAMOS 24 Y 29)

RUTA DE ACCESO : /uno/dgpp/prod/shells/prepp6



REPORTES A LA CAMARA
RESUMEN PROGRAMATICO ECONOMICO FINANCIERO

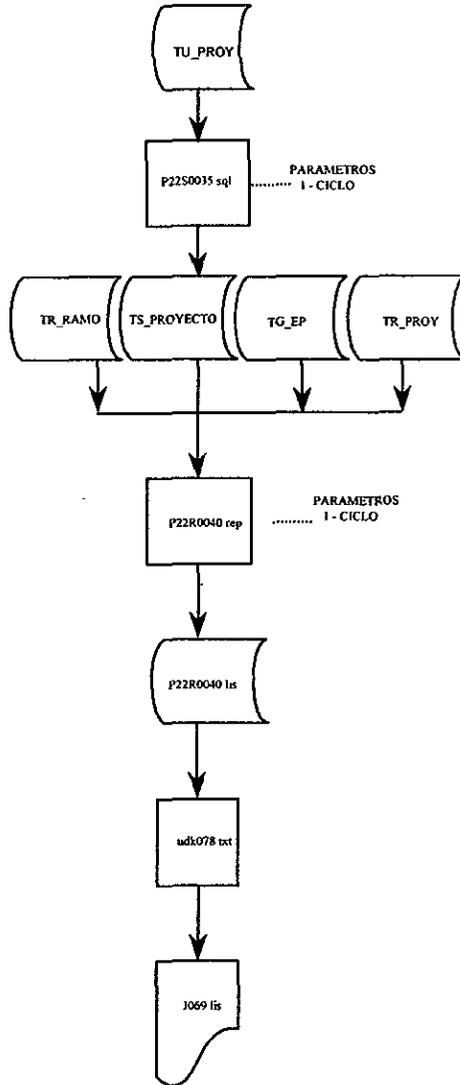
RUTA DE ACCESO : /uno/dgpp/prod/shells/prepp



REPORTES A LA CAMARA

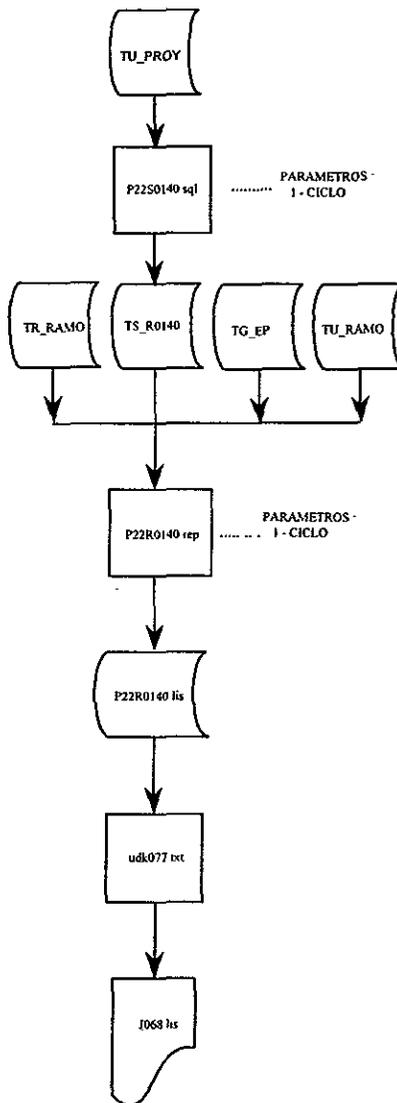
RESUMEN PROGRAMATICO ECONOMICO FINANCIERO (PARCIAL RAMO 12)

RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0035



REPORTES A LA CAMARA
RESUMEN PROGRAMATICO ECONOMICO FINANCIERO
(RAMO 12, TP = 8)

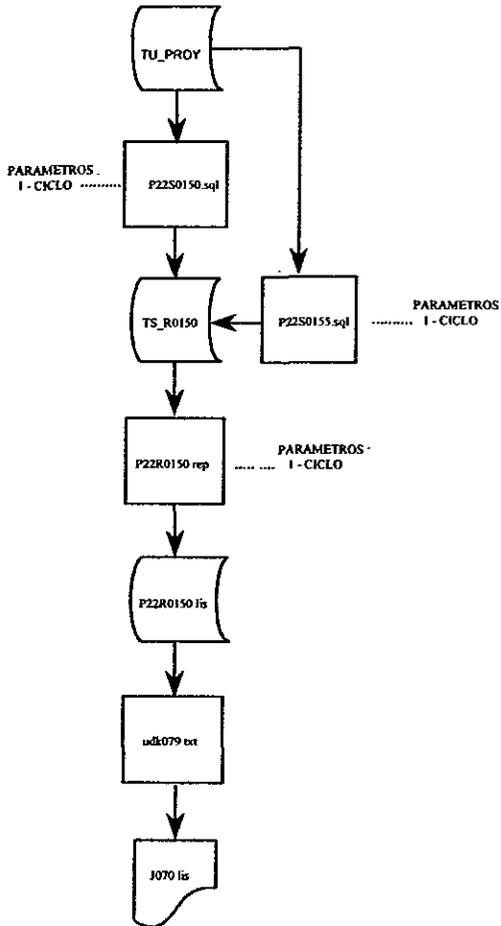
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0140



REPORTES A LA CAMARA

RESUMEN PROGRAMATICO ECONOMICO FINANCIERO (RAMO 22)

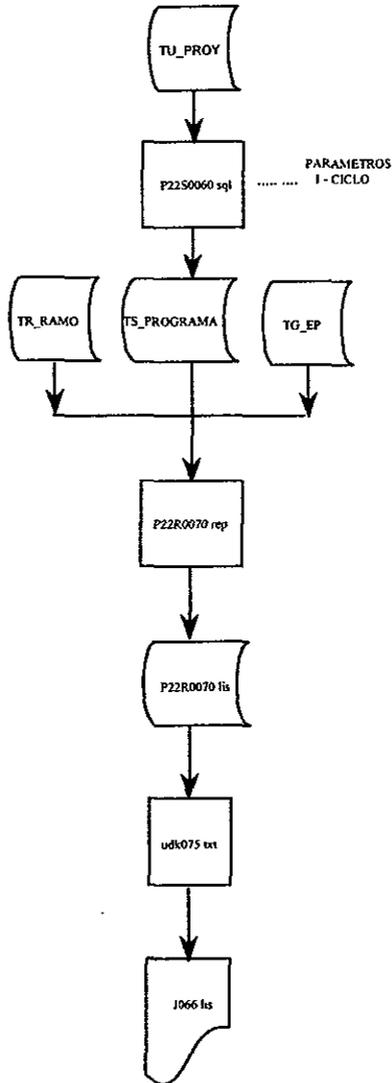
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0150



REPORTES A LA CAMARA

RESUMEN PROGRAMATICO ECONOMICO FINANCIERO (RAMOS 24 Y 29)

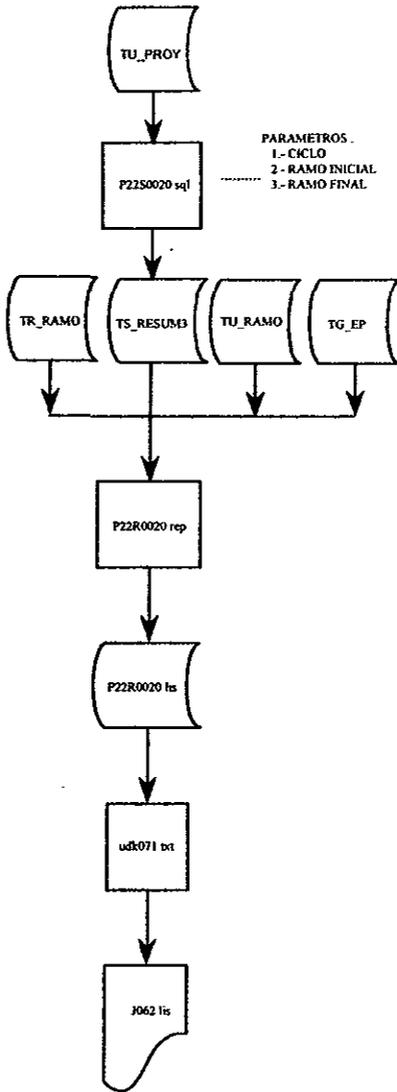
RUTA DE ACCESO : /uno/dgpp/prod/shells/prepp1



REPORTES A LA CAMARA

RESUMEN CUANTITATIVO DEL RAMO (GENERICO)

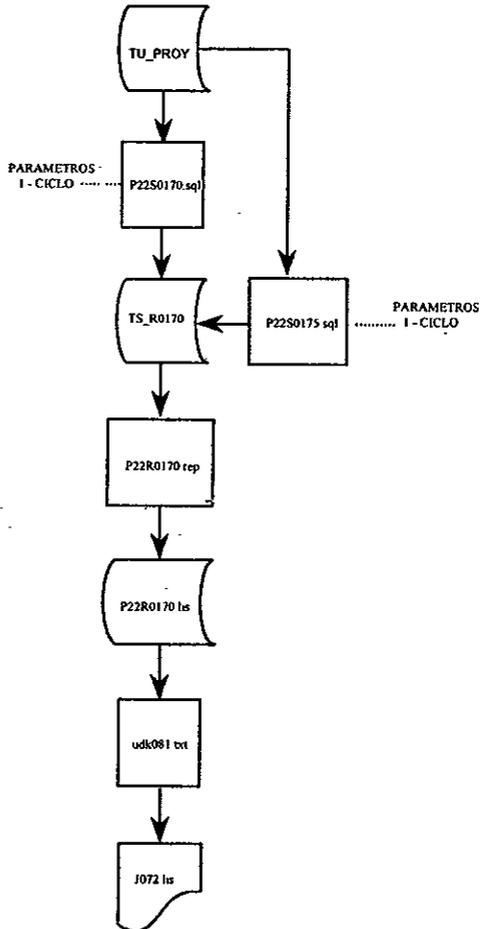
RUTA DE ACCESO : /uno/dgpp/prod/shells/prepp3



REPORTES A LA CAMARA

RESUMEN CUANTITATIVO DEL RAMO (RAMO 22)

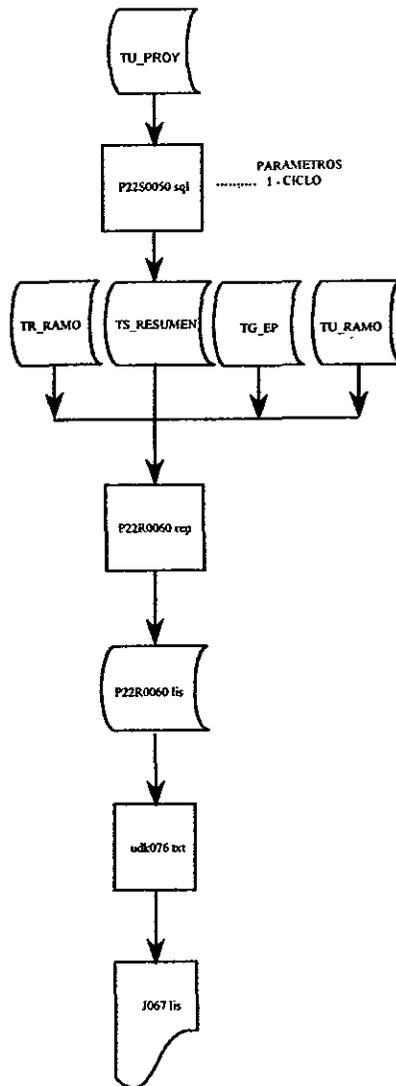
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0170



REPORTES A LA CAMARA

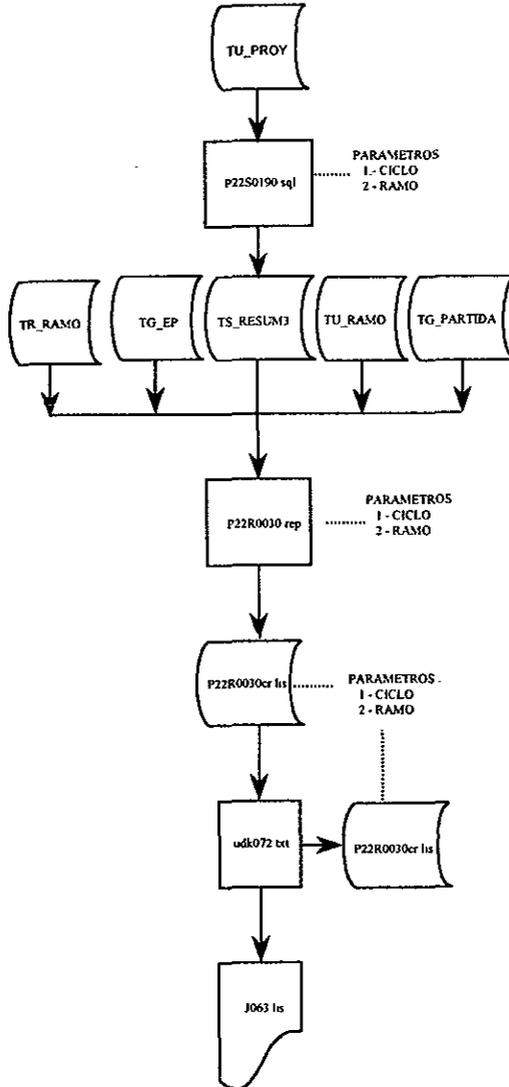
RESUMEN CUANTITATIVO DEL RAMO (RAMOS 24 Y 29)

RUTA DE ACCESO : /uno/dgpp/prod/shells/prepp4



REPORTES A LA CAMARA
ASIGNACION PRESUPUESTAL POR POGRAMA (GENERICO)

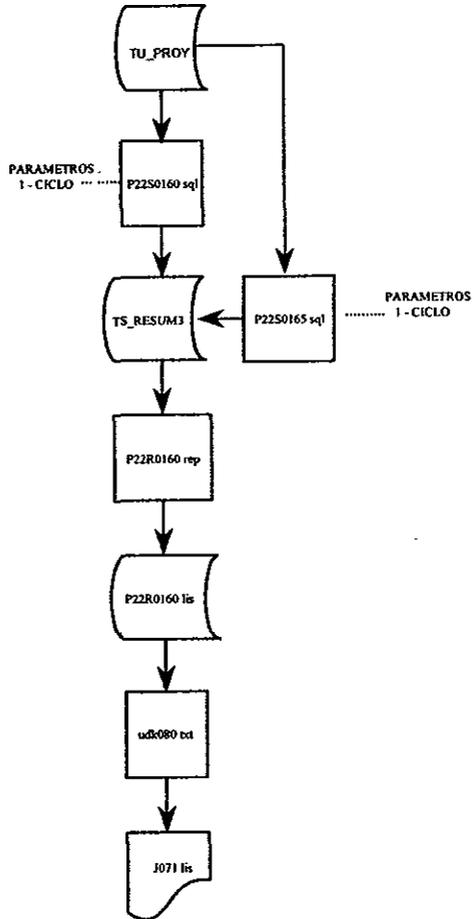
RUTA DE ACCESO : /uno/dgpp/prod/shells/prepp5



REPORTES A LA CAMARA

ASIGNACION PRESUPUESTAL POR PROGRAMA (RAMO 22)

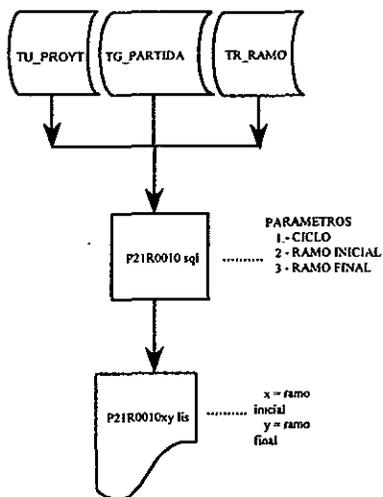
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0160



REPORTES DE TRABAJO

POR UNIDAD

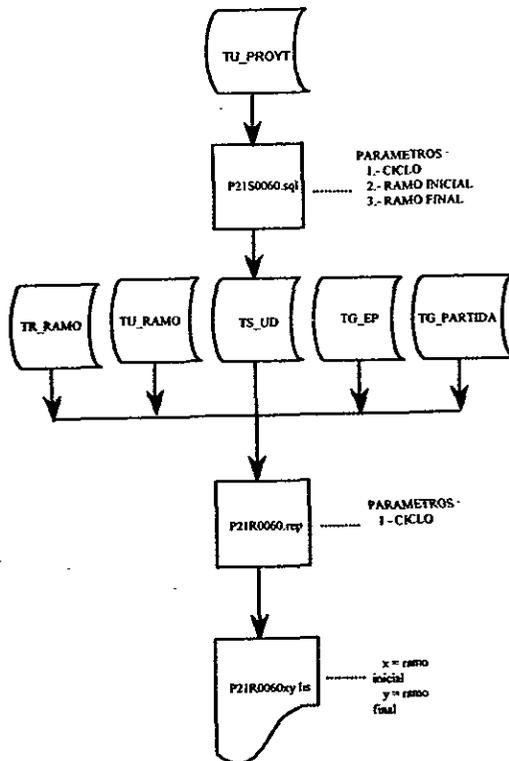
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0010



REPORTES DE TRABAJO

POR UNIDAD CON DESCRIPCION

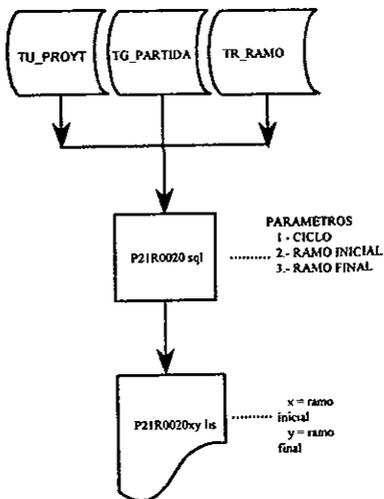
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0060



REPORTES DE TRABAJO

POR PROGRAMA

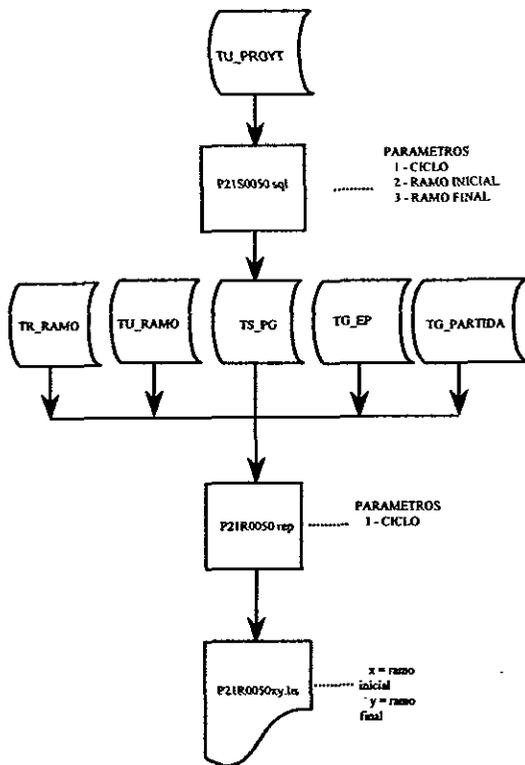
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0020



REPORTES DE TRABAJO

POR PROGRAMA CON DESCRIPCION

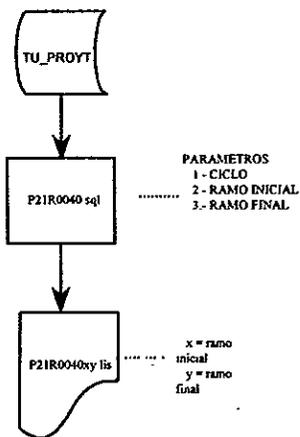
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0050



REPORTES DE TRABAJO

RESUMEN DE ENTIDAD POR CONCEPTO E IDENTIFICADOR DE GASTO

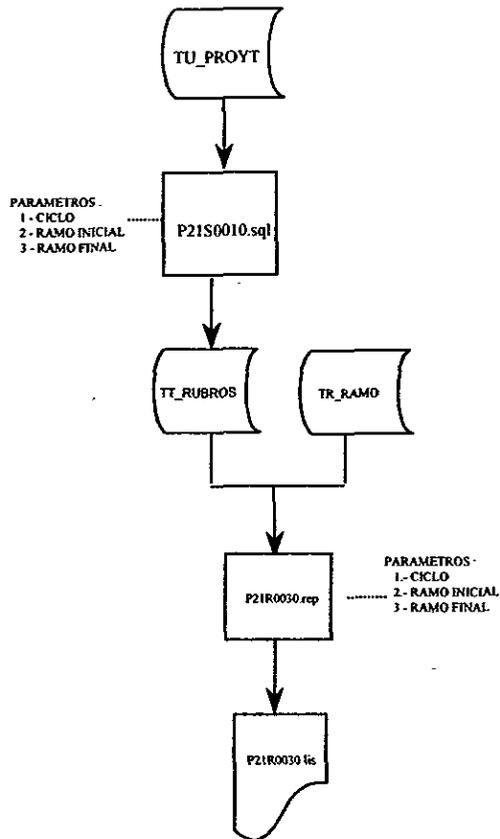
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0040



REPORTES DE TRABAJO

RUBROS DE GASTO (TEMPORAL)

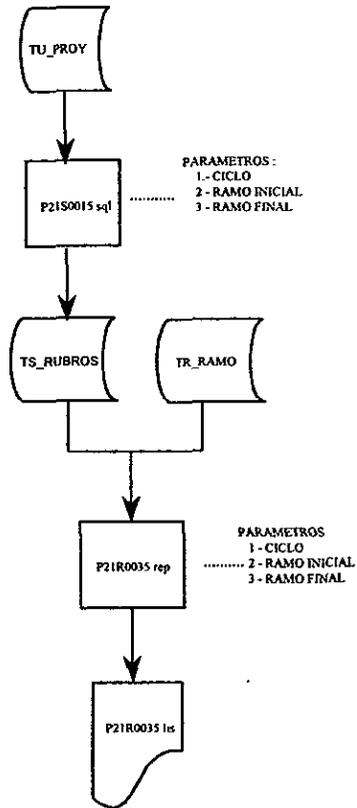
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0030



REPORTES DE TRABAJO

RUBROS DE GASTO (DEFINITIVO)

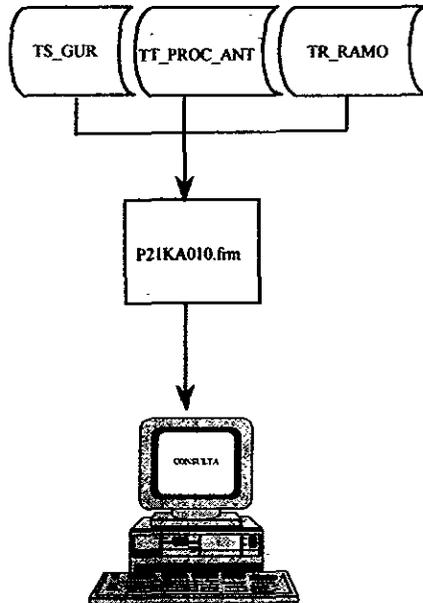
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0035



CONSULTAS

ESTADO DEL PROCESO

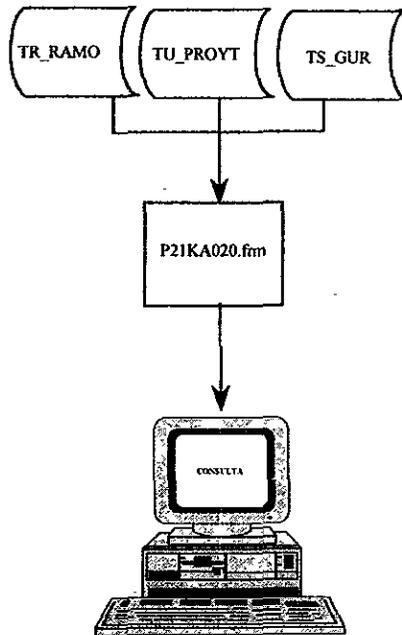
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21KA010



CONSULTAS

ESTADO DEL ANTEPROYECTO (TEMPORAL)

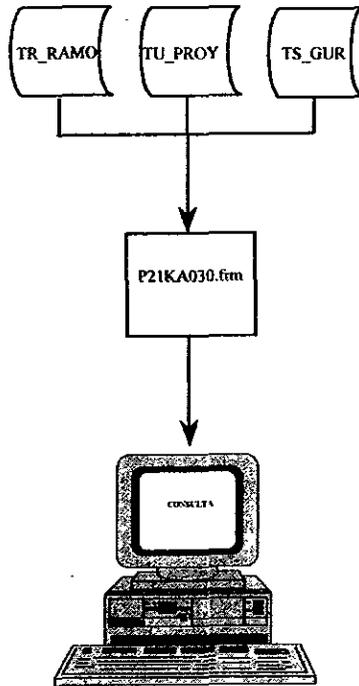
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21KA020



CONSULTAS

ESTADO DEL ANTEPROYECTO (DEFINITIVO)

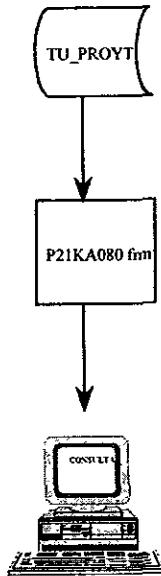
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21KA030



CONSULTAS

TOTAL GASTO FEDERAL DE LOS RAMOS (TEMPORAL)

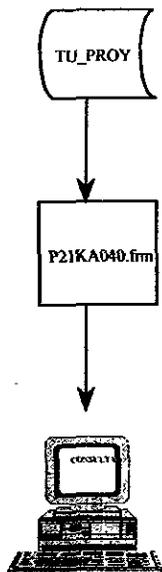
RUTA DE ACCESO : </uno/dgpp/prod/formas/P21KA080>



CONSULTAS

TOTAL GASTO FEDERAL DE LOS RAMOS (DEFINITIVO)

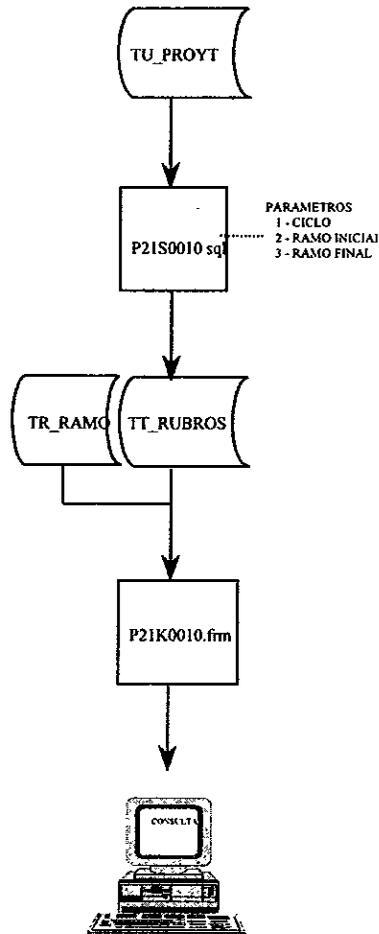
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21KA040



CONSULTAS

RUBROS GLOBALES DEL GOBIERNO FEDERAL (TEMPORAL)

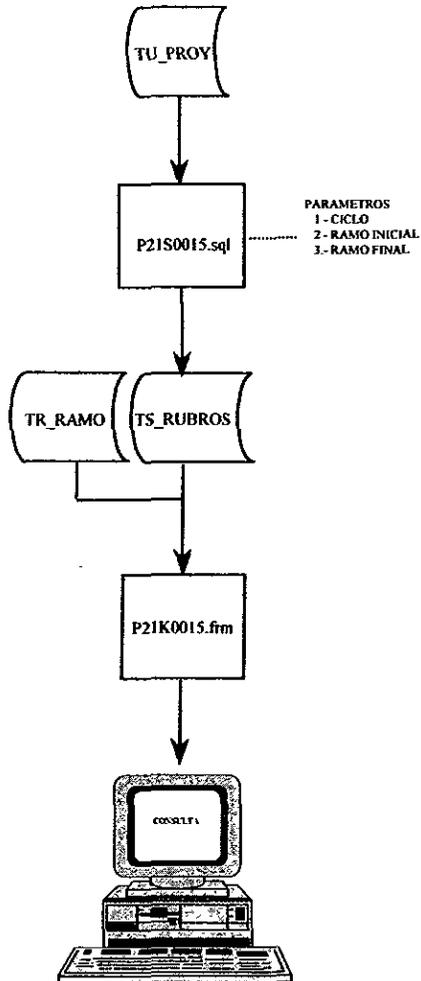
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21K0010



CONSULTAS

RUBROS GLOBALES DEL GOBIERNO FEDERAL (DEFINITIVO)

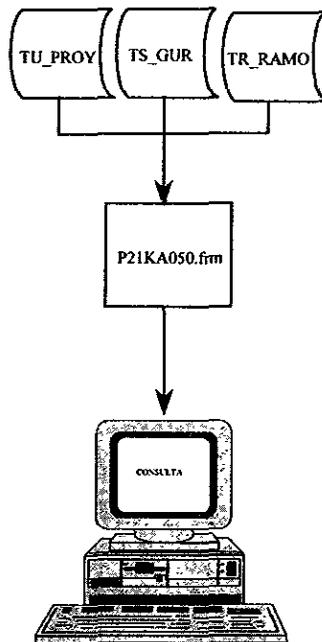
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21K0015



CONSULTAS

TIPO DE PAGO E IDENTIFICADOR DE GASTO POR RAMO

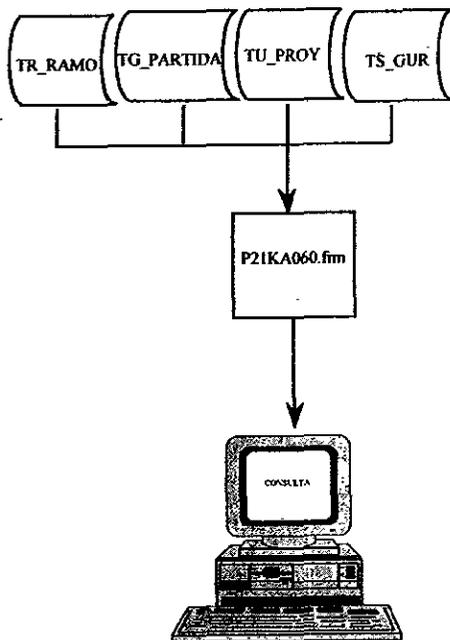
RUTA DE ACCESO : `/uno/dgpp/prod/formas/P21KA050`



CONSULTAS

CONCEPTO E IDENTIFICADOR DE GASTO POR RAMO

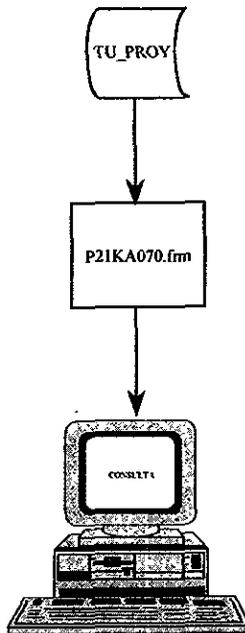
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21KA060



CONSULTAS

SERVICIOS OFICIALES (3806)

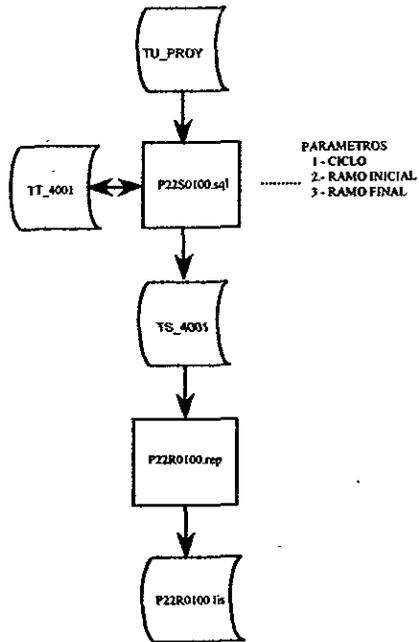
RUTA DE ACCESO : </uno/dgpp/prod/formas/P21KA070>



GENERACION DE ARCHIVOS

TRANSFERENCIAS (GLOBAL)

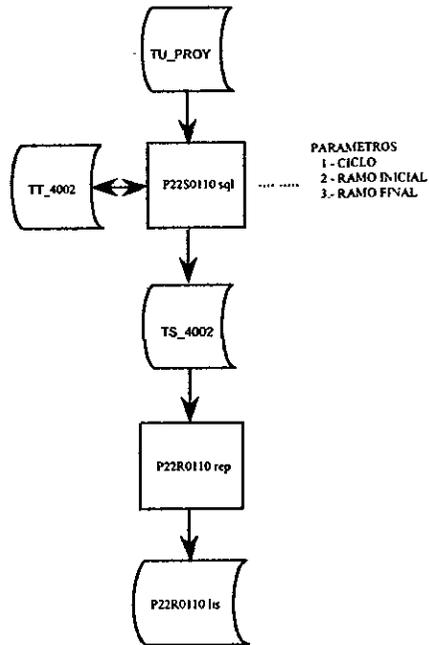
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0100



GENERACION DE ARCHIVOS

TRANSFERENCIAS (SUBSIDIOS 4200)

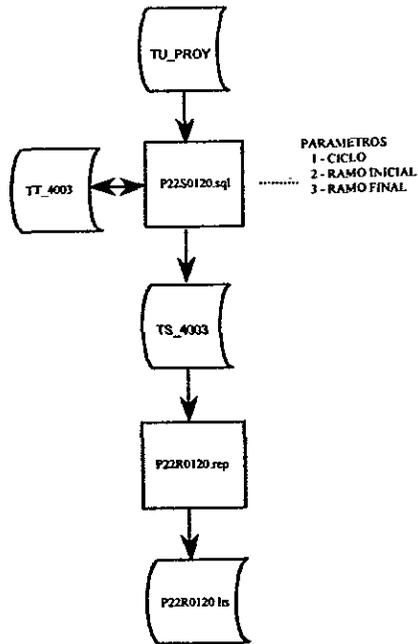
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0110



GENERACION DE ARCHIVOS

TRANSFERENCIAS (APOYOS 4300)

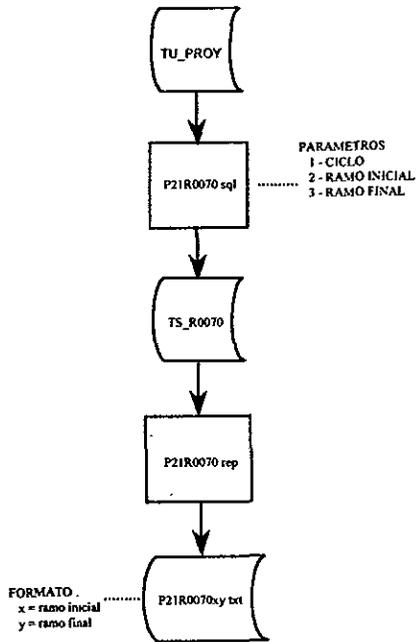
RUTA DE ACCESO : /uno/dgpp/prod/shells/P22R0120



GENERACION DE ARCHIVOS

TRANSFERENCIAS RESUMEN ADMINISTRATIVO DE TRANSFERENCIAS

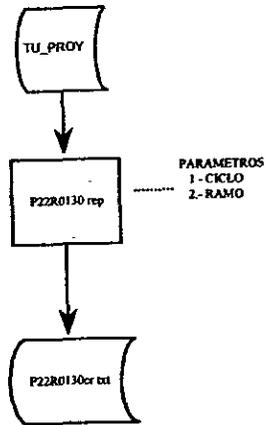
RUTA DE ACCESO : /uno/dgpp/prod/shells/P21R0070



GENERACION DE ARCHIVOS

GENERA SECUENCIAL DE ANTEPROYECTO (DEFINITIVO)

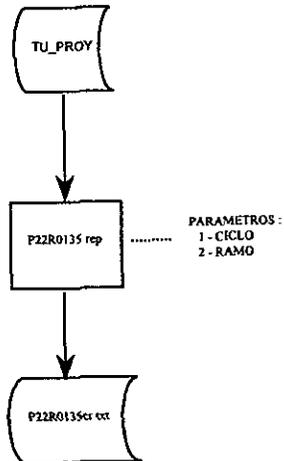
RUTA DE ACCESO : /uno/dgpp/prod/reportes/P22R0130



GENERACION DE ARCHIVOS

GENERA SECUENCIAL DE PAQUETE SALARIAL

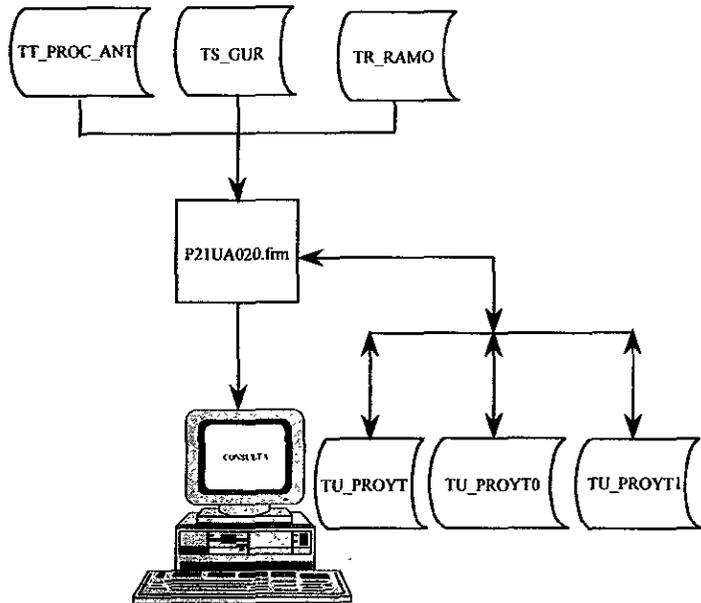
RUTA DE ACCESO : /uno/dgpp/prod/reportes/P22R0135



HERRAMIENTAS

RESTAURACION DE AREAS (TEMPORAL)

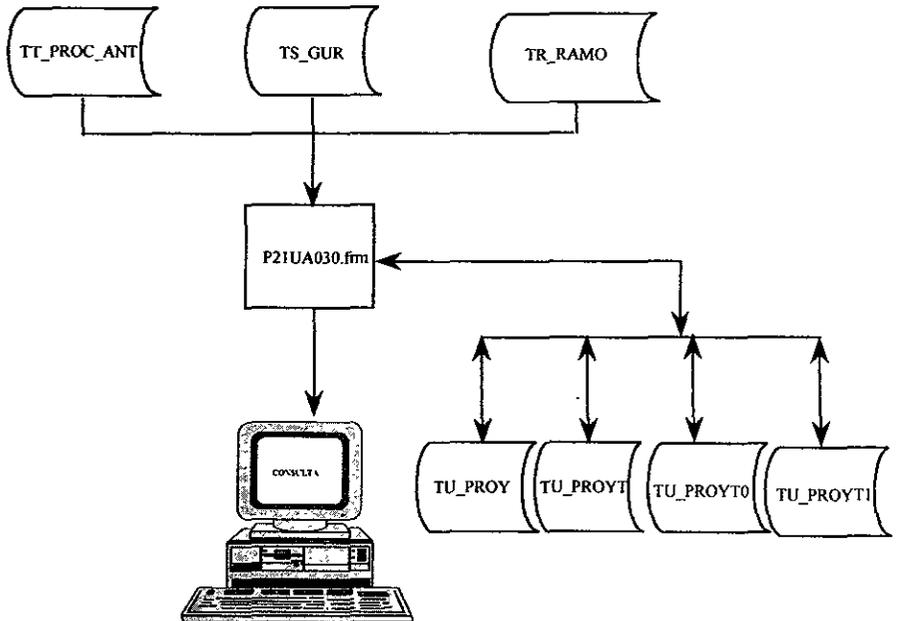
RUTA DE ACCESO : /uno/dgpp/prod/formas/P21UA020



HERRAMIENTAS

RESTAURACION DE AREAS (DEFINITIVO)

RUTA DE ACCESO : /uno/dgpp/prod/formas/P21UA030



Bibliografía

Bases de datos: Métodos Prácticos en maxi y miniordenadores / David Martin

Database : Principles, programing, performance / Patrick O'Neil

Sistemas de Administración de Bancos de Datos / Alfonso F. Cardenas

Database Management and Design / Gary W. Lansen

Database Management: Objectives, System Functions and Administration / Gordon C. Everest

Manual del curso de Administración I de ORACLE / Centro educacional de ORACLE

Fundamentals of Database Systems / Ramez Elmasri

Manual de Normatividad para el gasto Publico / SHCP, Subsecretaría de Egresos

Ley de Presupuesto, Contabilidad y Gasto Público / SHCP-UPCP

Manual de Programación y Presupuesto / SHCP
