

42  
2EJ



**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO**

**FACULTAD DE CIENCIAS**

**"SISTEMA AUTOMATIZADO PARA EL  
CONTROL DEL ACERVO EN LAS  
BIBLIOTECAS DE LA UNAM"**

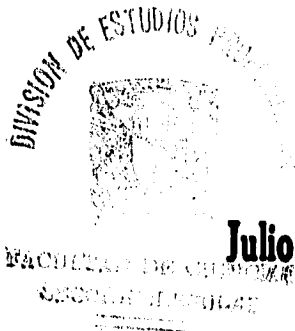
**T E S I S**  
Que para obtener el Título de  
**A C T U A R I O**  
p r e s e n t a

**GONZALEZ TRAPAGA MARIA ILEANA**



México, D. F.

Julio de 1995



**FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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



UNIVERSIDAD NACIONAL  
AVENIDA DE  
MEXICO

**M. EN C. VIRGINIA ABRIN BATULE**  
Jefe de la División de Estudios Profesionales  
Facultad de Ciencias  
Presente

Los abajo firmantes, comunicamos a Usted, que habiendo revisado el trabajo de Tesis que realiz(ó)ron LA pasante(s) PASANTE GONZALEZ TRAPAGA MA. ILEANA

con número de cuenta 7006436-6 con el Título: "SISTEMA AUTOMATIZADO PARA EL CONTROL DEL ACERVO EN LAS BIBLIOTECAS DE LA UNAM"

Otorgamos nuestro **Voto Aprobatorio** y consideramos que a la brevedad deberá presentar su Examen Profesional para obtener el título de ACTUARIO

GRADO	NOMBRE(S)	APELLIDOS COMPLETOS	FIRMA
M. en C. Director de Tesis	MA. GUADALUPE ELENA IBARGUENGOITIA	GONZALEZ	<i>[Firma]</i>
MTRD.	ADOLFO RODRIGUEZ	GALLARDO	<i>[Firma]</i>
M en C.	GUSTAVO ARTURO MARQUEZ	FLORES	<i>[Firma]</i>
DRA.	MA. DEL CARMEN LOPEZ	LAISECA	<i>[Firma]</i>
Suplente MAT.	ANA LUISA SOLIS	GONZALEZ COSIO	<i>[Firma]</i>
Suplente			

La realización de este trabajo la dedico

**A MIS PADRES**

a quienes les debo todo en la vida

**A MI PADRE**

A quien hasta hoy,  
si viviera, vería  
que su aliento  
para conmigo  
no fue en vano

**A MI MADRE**

Con todo mi  
AMOR,  
gratitud y  
admiración

**A GINA, ROLANDO  
ARCELIA Y SERGIO**

**por su confianza,  
estímulo y cariño**

**A EDUARDO, ANA PAOLA  
CARITO Y NURIA**

**con todo mi amor**

**A MARA, JUANIS,  
DELFINO Y ALEJANDRO**

**por su confianza,  
estímulo y  
estimación**

**A MIS AMIGOS**

**por su amistad  
y su cariño**

## AGRADECIMIENTOS

Agradezco muy especialmente a la M. en C. Lupita Ibarquengoitia por la dirección de este trabajo, así como su apoyo, paciencia, confianza y amistad depositados en mí, no solamente durante la realización de este trabajo sino a lo largo de mi formación académica.

A los miembros del jurado: M. en C. Gustavo Arturo Márquez Flores, Dra. Ma. del Carmen López Laiseca y Mat. Ana Luisa Solís González Cosío por su revisión crítica y sugerencias que mejoraron este trabajo.

Mi agradecimiento al Mtro. Adolfo Rodríguez Gallardo por el apoyo recibido en la realización de este trabajo y en el desarrollo del sistema; así como por la revisión crítica del mismo y por participar como miembro del jurado.

Agradezco al Mat. Alejandro Ramírez Nieto por su apoyo en la realización de este trabajo y en el desarrollo del sistema.

Al Lic. Rafael Ibarra Contreras todo mi agradecimiento por el apoyo que me brindó con las sugerencias que mejoraron mi trabajo, por sus correcciones de ortografía y de estilo acerca del escrito y por la orientación en las dudas de traducción del inglés al español de algunos términos técnicos.

A la bióloga Arcelia González Trápaga por su ayuda en el diseño y digitación de algunas figuras.

De igual forma mi agradecimiento a la Srita. Julieta Navarrete por la digitación de diagramas.

Agradezco al Ing. Isaác Vivas Escobedo por su apoyo en la realización de este trabajo.

A todos aquellos compañeros y amigos de la DGB que en el transcurso de los años me han ofrecido una amistad sincera y desinteresada.

Agradezco también a mis amigos Felipe, Martha, Irma, Rosario, Rocío, Alberto, Luzma, Silvia, Carmelita, Enna, Sandra, Sra. Amalia, Jaime, Elizabeth, Licha, July, Lino, Jovv, Juan, Francisco, Bety y Luz María por su apoyo moral y de aliento para la realización de este trabajo.

A mi familia agradezco el estímulo, confianza, comprensión, cariño y apoyo moral que siempre me han brindado.

---

---

**CONTENIDO**

---

---

**INTRODUCCION** **X**

---

**CAPITULO 1**

**CONCEPTOS BASICOS DE BASES DE DATOS** **2**

---

1.1 INTRODUCCION (GENERALIDADES)	2
1.2 ¿ QUE ES UN SISTEMA DE BASES DE DATOS ?	4
1.3 ¿ POR QUE UTILIZAR BASES DE DATOS ?	8
1.4 INDEPENDENCIA DE LOS DATOS	10
1.5 ARQUITECTURA DE UN SISTEMA DE BASES DE DATOS	12
1.6 MODELO DE DATOS RELACIONAL	15
1.6.1 RELACIONES	15
1.6.2 DOMINIOS Y ATRIBUTOS	16
1.6.3 LLAVES	16
1.7 ALGEBRA RELACIONAL	18
1.8 CALCULO RELACIONAL	23
1.9 NORMALIZACIONES	24
1.10 CONSIDERACIONES PARA BASES DE DATOS GRANDES	29

**CAPITULO 2**

**INGENIERIA DEL SOFTWARE** **35**

---

2.1 INTRODUCCION	35
2.2 SOFTWARE E INGENIERIA DEL SOFTWARE	35
2.3 ANALISIS DE REQUERIMIENTOS DEL SISTEMA Y DEL SOFTWARE	39
2.4 PRINCIPIOS DEL ANALISIS	41
2.5 REQUERIMIENTOS DE LAS BASES DE DATOS	44
2.6 DISEÑO DEL SOFTWARE	46
2.7 FUNDAMENTOS DEL DISEÑO DE SOFTWARE	48
2.8 MANTENIMIENTO DEL SOFTWARE	54

**CAPITULO 3**

**DESCRIPCION DEL MEDIO AMBIENTE DEL SISTEMA DE INVENTARIOS** **57**

---

3.1 INTRODUCCION	57
3.2 ANTECEDENTES DEL SISTEMA DE INVENTARIOS	59
3.3 ¿ EN QUE CONSISTE EL SISTEMA DE INVENTARIOS ?	62
3.4 DESCRIPCION DEL SUBSISTEMA DE INVENTARIOS	62
3.4.1 Análisis	62
3.4.2 Descripción del proceso de registro de los libros (generalidades)	69
3.4.3 Descripción del proceso de registro de Inventarios	77
3.5 HERRAMIENTAS	86
3.5.1 BRITTON LEE Intelligent Database Machine (IDM)	86
3.5.2 EL MODELO RELACIONAL DE DATOS EN IDM	91
3.5.3 ALGEBRA Y CALCULO RELACIONAL CON IDM	93



3.5.4 AM-2000-10 y AM-1062.	100
3.5.4.1 Alpha Microsystem 2000-10	100
3.5.4.2 Alpha Microsystem 1062	101

#### **CAPITULO 4**

<b>DESCRIPCION DEL SISTEMA</b>	<b>104</b>
4.1 INTRODUCCION	104
4.2 ESTRUCTURA DE LA BASE DE DATOS	104
4.2.1 Sistema LIBRUNAM: Generalidades	104
4.2.2 La información y sus elementos (etiquetas)	106
4.2.3 Estructura de la Base de Datos "libros"	110
4.2.4 Estructura de la relación "inventarios"	117
4.2.5 Esquema general del Sistema LIBRUNAM	120
4.3 PROCEDIMIENTOS GENERALES EN LIBRUNAM	122
4.3.1 Captura y procesos de validación de la información	122
4.3.2 Procesamiento y almacenamiento de la información bibliográfica en la base de datos "libros"	123
4.3.3 Procesos para el uso en línea del banco de datos LIBRUNAM (generalidades)	123
4.4 ACTUALIZACION DE LOS INVENTARIOS	129
4.4.1 Descripción del proceso de registro de Inventarios	129
4.4.1.1 Altas de la información	131
4.4.1.2 Actualización de la información	142
4.4.2 Generación de listados detallados del acervo bibliográfico dado de alta y de baja en el inventario	149
4.4.3 Generación de etiquetas del acervo bibliográfico dado de alta en el inventario.	153
<b>CONCLUSIONES</b>	<b>159</b>
<b>REFERENCIAS</b>	<b>164</b>
<b>BIBLIOGRAFIA CONSULTADA</b>	<b>168</b>

---

---

**INTRODUCCION**

---

---

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

El diseñar un sistema basado en la automatización de un proceso manual no garantiza su éxito, pero el llevar a cabo un desarrollo automatizado profesional de un sistema, donde el análisis y la planificación tienen sus bases en estudios e investigaciones bien establecidos en el área a nivel mundial sí lo garantiza. Sobre todo cuando se habla de automatizar procesos que manejan grandes volúmenes de información donde se pierde el control de los procesos manuales y los tiempos se vuelven insuficientes para efectuarlos.

El mundo en constante cambio, ha favorecido el desarrollo científico, tecnológico e industrial de la sociedad, que a su vez ha requerido la formación de cuadros capacitados para hacer frente a sus necesidades cada vez más sofisticadas. Es por ello que las universidades y otros centros de enseñanza superior han crecido rápidamente en lapsos muy cortos, y en consecuencia el acceso a servicios e información ha tenido que desarrollarse constantemente para brindarles atención inmediata.

Las bibliotecas universitarias no se sustraen a esta tendencia, por lo que el automatizar todos los procesos que las lleven a suministrar un mejor servicio se vuelve necesario.

Es tarea primordial para la Dirección General de Bibliotecas el apoyar a todas las bibliotecas de la UNAM a solventar sus problemas y a proporcionar los elementos necesarios para lograrlo. Por ello se ha preocupado en desarrollar sistemas automatizados que agilicen sus servicios siendo el principal y más importante el sistema LIBRUNAM, que para controlar los libros que ingresan al banco de datos se apoya en el sistema INVENTARIOS.

Este sistema es el que se encarga de controlar el acervo bibliográfico que todas las bibliotecas de la UNAM ponen a disposición de los universitarios.

En el presente trabajo se describen la creación, el desarrollo, y las aplicaciones de este sistema así como las definiciones y nociones básicas sobre la teoría utilizada.

El desarrollo del sistema INVENTARIOS se basa en los conceptos establecidos en las *bases de datos relacionales*, ya que son las más adecuadas para el almacenamiento y la explotación de la información que se maneja. Para este desarrollo se analizaron todas las necesidades y experiencias que se obtuvieron a partir del control manual del acervo en las bibliotecas de la UNAM, y se les aplicó la metodología establecida en la *ingeniería del software* para la definición,

desarrollo y mantenimiento del software (programación) que sostiene el sistema y sus aplicaciones.

El documento está dividido en cuatro capítulos, y finaliza con las conclusiones.

En el primer capítulo se presentan los conceptos básicos que se manejan dentro de las bases de datos, haciendo hincapié en la utilización del álgebra y el cálculo relacionales como herramientas básicas para el desarrollo de *las bases de datos relacionales*.

El segundo capítulo se refiere a los conceptos básicos de la *ingeniería del software*; ya que ésta nos proporciona los elementos que se requieren para desarrollar un software confiable y de calidad para la explotación de cualquier sistema automatizado y en particular de cualquier base de datos.

En el tercer capítulo, se hace la descripción del medio ambiente del sistema INVENTARIOS, es decir, se presentan los antecedentes del sistema, se describen la problemática y los procesos de registro tanto de los libros en forma general y en particular dentro del control del inventario de la biblioteca, como el del registro de los inventarios cuando se llevan a cabo levantamientos de inventarios físicos en ella. En este capítulo se incluye también la descripción el equipo de cómputo utilizado.

En el cuarto y último capítulo, se detalla el sistema automatizado INVENTARIOS, su estructura, todos los procesos involucrados y sus productos.

Al término de los cuatro capítulos, se presentan las conclusiones del trabajo.

Y para finalizar, se detallan las referencias que apoyan los conceptos utilizados durante el desarrollo de la presente tesis y la bibliografía consultada.

**CAPITULO 1**

---

---

**CONCEPTOS BASICOS  
DE BASES DE DATOS**

---

---

# CONCEPTOS BASICOS DE BASES DE DATOS

## 1.1 INTRODUCCION (GENERALIDADES)

En este capítulo se presentan los conceptos básicos que se manejan en las bases de datos y en los sistemas de bases de datos.

En una época como la actual, el acelerado desarrollo de la ciencia, las artes, y la tecnología se fundamenta en el conocimiento que el hombre obtiene y desarrolla del universo que lo rodea, y para lograr esto, la consecución de la información se convierte en una necesidad primordial.

La información cobra una vital importancia para la organización de una empresa (término empleado para designar cualquier organización comercial, científica, técnica o de otra clase que posea un nivel razonable de autosuficiencia) y consecuentemente para proporcionar un mejor servicio al público.

Un mensaje es un conjunto de ideas vertidas por un ser humano para que otro las reciba; y las ideas representadas por un mensaje son la información de éste.

Usando una terminología más común en informática, cada mensaje es un registro y cada registro contiene varias palabras cuyo significado está ligado a su posición dentro del registro. La posición que dentro de cada registro ocupa un dato se llama campo; y los registros que tienen campos con el mismo significado predefinido se suelen agrupar en conjuntos llamados archivos. [1]

El volumen de datos almacenados y la complejidad de su organización hoy en día están aumentando a pasos agigantados. El adecuado aprovechamiento de estas tremendas cantidades de datos almacenados exige un especial cuidado en:

- Su almacenamiento
- La transmisión de los datos (capacidad de acceso a la base de datos desde localidades remotas.)
- El diálogo entre el hombre y la computadora (que permite al usuario hacer todo tipo de averiguaciones en los archivos, además de modificar, agregar o borrar datos en ellos.)

Todo esto y en especial las tecnologías desarrolladas para el almacenamiento y explotación de grandes volúmenes de datos ha llevado a la definición y desarrollo de lo que conocemos como diseño y manejo de BASES DE DATOS, y es en el presente, sin lugar a duda, una de las actividades más trascendentales en el campo de la informática.

Podemos entonces definir una **base de datos** como una *colección grande y organizada de información a la que se accede mediante el software y que es una parte integral del funcionamiento del sistema.*

Es importante que en el diseño de la base, los datos sean almacenados de manera que se puedan utilizar indiferentemente para una amplia variedad de aplicaciones y que a la vez pueda cambiarse fácil y rápidamente la manera de utilizarlos; y es por eso que se vuelve prioritario cuidar en este diseño lo siguiente:

- Los datos deben ser independientes de los programas que los utilizan, de modo que se puedan enriquecer y reestructurar sin que sea necesario modificar la programación existente, además de que puedan ser utilizados indistintamente para una amplia variedad de aplicaciones.

- Debe ser posible interrogar y explorar la base de datos sin necesidad de recurrir a la tediosa operación de escribir programas utilizando lenguajes convencionales de programación.

- La eliminación de toda imprecisión en la definición y el uso de los datos debe constituir un objetivo primordial a tenerse en cuenta en todas las etapas del diseño y la integración de la base de datos.

- Es necesario también, tomar en cuenta las intrincadas concertaciones a que hay que llegar entre los diversos aspectos del diseño, por ejemplo: entre el aprovechamiento del espacio de almacenamiento y el aprovechamiento del tiempo, entre el tiempo de respuesta y la complejidad de las estructuras de datos, entre el diseño capaz de facilitar las averiguaciones imprevistas y el diseño adecuado para satisfacer requerimientos operacionales bien definidos.

[2]

## 1.2 ¿ QUE ES UN SISTEMA DE BASES DE DATOS ?

Un sistema de bases de datos es en esencia...." un sistema de mantenimiento de registros basado en computadoras, es decir, un sistema cuyo propósito general es registrar y mantener información ". [3]

También podríamos decir que un *sistema de bases de datos es una colección de bases de datos que coexisten dentro de un sistema donde los contenidos de estas bases son independientes y disjuntos.* [4]

Son independientes porque la información contenida en una base de datos no depende de la contenida en cualquiera otra base del sistema, y son disjuntos porque las bases de datos del sistema no tienen elementos en común, es decir, la información contenida en una base de datos en particular del sistema no puede estar localizada en ninguna otra base del sistema.

Los componentes principales de un Sistema de Bases de Datos son: datos, hardware, software, usuarios.

**Datos** .- Los datos almacenados en el sistema se dividen en una o más bases de datos.

"Una base de datos es un repositorio de datos almacenados, y, en general, es tanto *integrada* como *compartida*." [3]

"Por *integrada* se entiende que la base de datos puede considerarse como una unificación de varios archivos de datos independientes donde se eliminan de redundancias entre los mismos." [3]

"Por *compartida* se entiende que partes individuales de los datos en la base pueden compartirse entre varios usuarios distintos y cada uno puede acceder a las mismas partes con propósitos diferentes." [3]

Se deriva el hecho de "*compartimiento concurrente* a la oportunidad de que diversos usuarios accedan en realidad la base de datos -tal vez la misma parte de la base de datos inclusive- al mismo tiempo. Un sistema así en ocasiones se llama *sistema de usuarios múltiples*." [3].

**Hardware** .- Son los dispositivos electrónicos (por ejemplo CPU, memoria) que proporcionan la capacidad de cómputación, y los dispositivos electromecánicos (por ejemplo sensores, motores, bombas) que proporcionan las funciones del mundo exterior. [5]



**Software** .- Está constituido por los programas de computadora, estructuras de datos y la documentación asociada que sirven para realizar el método lógico, procedimiento o control requerido para el manejo de la información dentro de la computadora [5].

Entre la base de datos física en sí (es decir, el almacenamiento real de los datos) y los usuarios del sistema existe un nivel de software que recibe el nombre de **Sistema de Administración de Bases de Datos** o **SABD**; el cual maneja las solicitudes de acceso a la base formuladas por los usuarios.

Una función del SABD es proteger a estos usuarios contra los detalles a nivel hardware; por lo tanto el SABD ofrece una vista de la base de datos por encima del nivel de hardware y apoya las operaciones que éstos realizan. [6]

**Usuarios** .- Son los individuos que utilizan y operan el software y el hardware [5].

Se tienen tres tipos de usuarios a saber: [6].

- a) **Programador de aplicaciones** - Es el que se encarga de escribir programas de aplicación que utilicen bases de datos; mismos con los que se recupera información, se crea nueva información, y se suprime o cambia la existente.
- b) **Usuario final** - Es el que accesa la(s) base(s) desde una terminal, utilizando o un lenguaje de consulta proporcionado como parte integral del sistema o un programa de aplicación escrito por un usuario programador.
- c) **Administrador de bases de datos o ABD.** - Es el que administra y da mantenimiento a las bases de datos, además de controlar su uso; es decir, es la persona (o grupo de personas) que controla la estructura general de los datos además de ser el responsable de su seguridad y control; por lo que consecuentemente es el que se encarga del control general del sistema de bases de datos.

Como elementos también incluidos en un sistema de bases de datos se tienen:

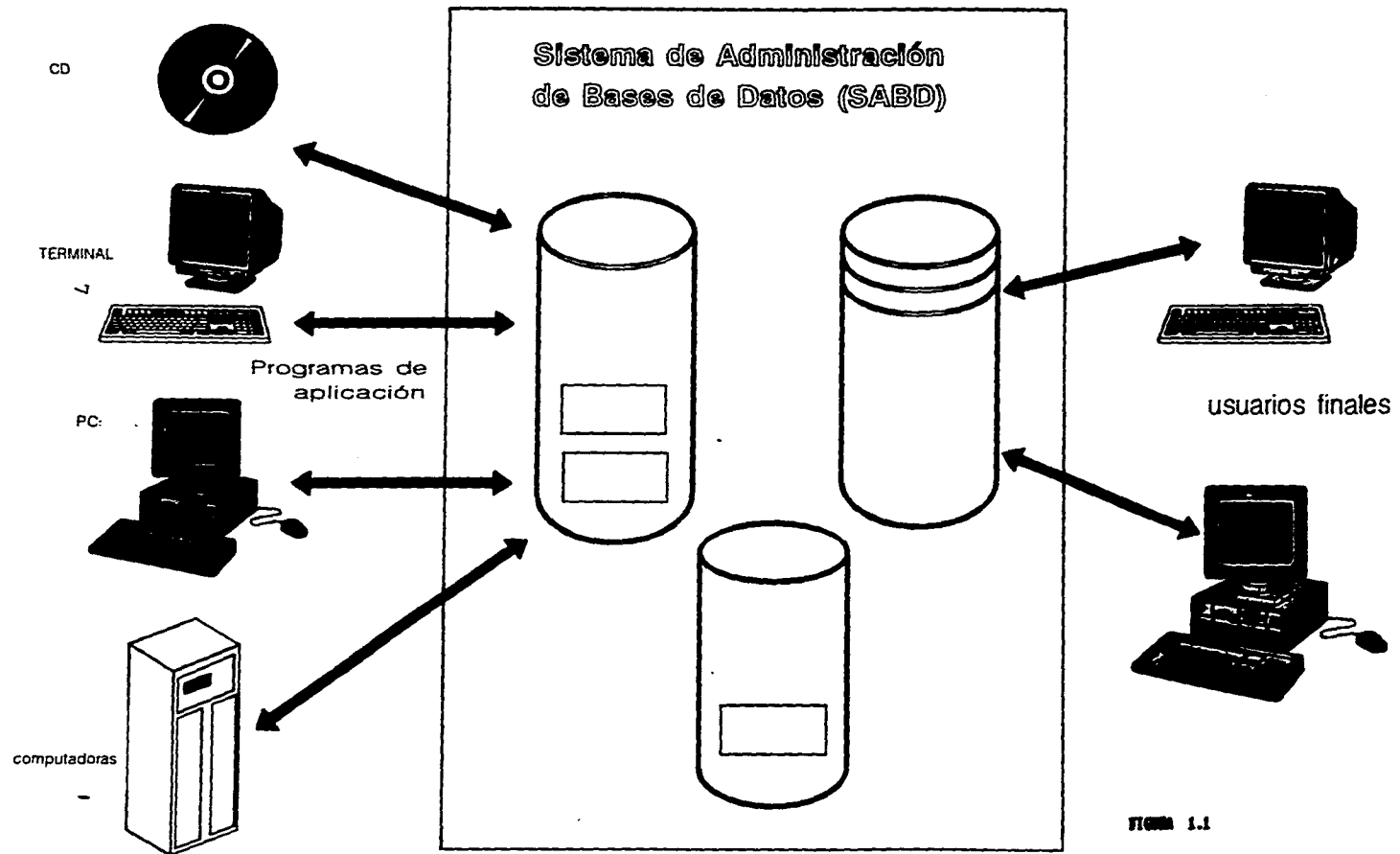
**Información** .- Es lo que nos proporciona un conocimiento, y es consecuencia del manejo de los datos; por lo que se puede definir como una colección organizada de datos a la cual se accede mediante el software y que finalmente es una parte integral del funcionamiento del sistema.

**Documentación** .- Los manuales, los impresos y otra información descriptiva que explica el uso, y/o la operación del sistema.

**Procedimientos** .- Los pasos que definen el uso específico de cada elemento del sistema. [7].

En la siguiente figura se muestra una representación simplificada de un sistema de bases de datos. (Figura 1.1)

# SABD



### 1.3 ¿ POR QUE UTILIZAR BASES DE DATOS ?

En los primeros días de procesos de datos, los problemas se resolvían creando y controlando archivos independientes; pero al crecer las necesidades y por consiguiente el volumen de información se empezó a dificultar el control de la misma y fue entonces cuando se comenzaron a estructurar las ideas para las bases de datos.

Un sistema de bases de datos proporciona a la empresa un control centralizado de sus datos de operación.

Todo sistema de bases de datos debe tener una persona específica que la controle y por lo tanto controle los datos de operación. Esta persona es el **Administrador de la Base de Datos** conocido como **ABD**.

Entre las responsabilidades del administrador de bases de datos (ABD) están:

- Decidir el contenido de la información de la base de datos.
- Decidir la estructura de almacenamiento y la estrategia de acceso.
- Vincularse con los usuarios.
- Definir los controles de autorización y los procedimientos de validación (los que son extensiones lógicas del esquema conceptual.)
- Definir la estrategia de respaldo y recuperación.
- Controlar el desempeño y responder a los cambios de requerimientos (mantenimiento del sistema.)

A continuación se dan unos ejemplos de los programas de utilería que maneja el ABD para llevar a cabo estos controles:

- Rutinas de carga (para crear la versión inicial de la base de datos)
- Rutinas de reorganización (reordenar la base de datos para recuperar espacios.)
- Rutinas de registros de eventos diarios (bitácoras de rutinas de usuarios, además del registro de los estados anterior y posterior de cada operación.)

- Rutinas de recuperación (restaurar la base de datos a un estado anterior a la falla.)
- Rutinas de análisis estadístico (para ayudar a controlar el desempeño.)
- Y por último, una utilidad que puede ser uno de los recursos más importantes del ABD es el **diccionario de datos** que es una base de datos que contiene "datos acerca de datos"; (es decir, descripciones de otros objetos del sistema.)

Las ventajas de tener un control centralizado de los datos son las siguientes:

■ Se puede reducir la **redundancia** de los datos y por lo tanto ahorrar espacio de almacenamiento de información. No siempre se debe eliminar toda redundancia porque a veces conviene tener repeticiones, pero sí debe estar controlada.

■ Se puede evitar la **inconsistencia**. La inconsistencia es consecuencia de la redundancia, es decir si el sistema no controla la redundancia puede entonces por ejemplo duplicarse un dato; y al hacerse una actualización (creación, supresión, modificación), llevarse a cabo solamente en uno de los datos, y entonces presentar una inconsistencia. En estos casos se dice que la base de datos es inconsistente.

Cuando la redundancia no está eliminada pero sí controlada, en cualquier cambio efectuado en un dato, el sistema en forma automática efectúa la actualización en el(los) duplicado(s). A esto se le llama **propagación automática**; es decir, el SADB deberá controlar la redundancia y asumir la responsabilidad de propagar las actualizaciones evitando con esto proporcionar información errónea.

■ Se pueden **compartir** datos. Los datos pueden ser compartidos por las aplicaciones existentes además de poder hacerlo en un futuro con las nuevas aplicaciones a desarrollar (sin necesidad de crear nuevos archivos almacenados.)

■ Se puede hacer cumplir las normas establecidas. Con un control central de la base de datos, el Administrador de la base de datos puede garantizar que se cumplan todas las formas aplicables a la representación de los datos. También es deseable unificar los formatos de los datos almacenados como ayuda para el intercambio o migración de los mismos a otros sistemas.

■ Se pueden aplicar restricciones de seguridad por el Administrador de la base de datos:

- para accesos por los canales establecidos
- para definir controles de autorización (de estos accesos.)

Estas restricciones deben de establecerse para cada aplicación.

■ Se puede conservar la **integridad**. Esto es, que los datos de la base sean exactos.

La inconsistencia es un ejemplo de falta de integridad (por redundancia no controlada.) Otro ejemplo de ausencia de integridad es la presencia de datos incorrectos por falta de validación en las entradas de los datos o en las actualizaciones.

■ Se pueden equilibrar los requerimientos contradictorios. Para lograrlo el ABD debe conocer los requerimientos globales de la empresa.

■ Se puede proveer la independencia de los datos. Por su importancia, este concepto se explica en la siguiente sección.

#### **1.4 INDEPENDENCIA DE LOS DATOS**

La mayoría de las aplicaciones actuales son dependientes de los datos por lo que cualquier cambio en la estructura de almacenamiento o la estrategia de acceso afecta a las aplicaciones y en muchos casos en forma muy severa.

Así tenemos que en un sistema de bases de datos y concretamente en cualquier base de datos no se debe permitir que las aplicaciones sean dependientes de los datos por las razones siguientes:

- Aplicaciones diferentes pueden requerir de vistas diferentes de los mismos datos.
- El ABD no puede tener la libertad de modificar la estructura de almacenamiento y/o la estrategia de acceso según las necesidades que se presenten sin tener que alterar las aplicaciones existentes, lo que implica un desperdicio de esfuerzos de reprogramación para el mantenimiento del sistema.

Por lo tanto la independencia de los datos es un objetivo y se define como la "inmunidad de las aplicaciones a los cambios de la estructura de almacenamiento y de la estrategia de acceso" [8].

Para una mejor comprensión de los elementos que se manejan dentro del lenguaje de bases de datos se tienen las siguientes definiciones: [8]

**Campo almacenado.**- Es la unidad de datos con nombre más pequeña que se halla almacenada en la base de datos. Esta contendrá en general muchas ocurrencias o instancias de cada uno de los diversos tipos de campo almacenado.

**Registro almacenado.**- Es un conjunto con nombre de campos almacenados asociados.

**Ocurrencia o instancia.**- La ocurrencia o instancia de un registro almacenado se compone de un grupo de ocurrencias de campos almacenados relacionados (y representa una asociación entre ellas.)

En la mayoría de las bases de datos la ocurrencia de registro almacenado es la unidad de acceso a la base de datos, es decir la unidad que el SABD puede recuperar o almacenar en un acceso.

**Archivo almacenado.**- es el conjunto (con nombre) de todas las ocurrencias de un tipo de registro almacenado.

**Materialización de los datos.**- El campo lógico que percibe una aplicación corresponde a algún campo almacenado único; por lo que el proceso de materialización (es decir, la construcción de una ocurrencia del campo lógico a partir de la ocurrencia del campo almacenado correspondiente y su presentación ante la aplicación) es directo.

El campo lógico también puede ser materializado por medio de un cómputo efectuado sobre un conjunto de varias ocurrencias de campos almacenados y así tener un campo virtual; aquí se considera que el proceso de materialización es indirecto.

La diferencia entre campo real y virtual es que el último no acepta modificaciones (al menos no de manera directa) porque es el resultado de una materialización indirecta.

**Estructura de los registros almacenados** - Un registro lógico de una aplicación puede contener campos de varios registros almacenados.

**Estructura de archivos almacenados** - Un archivo almacenado específico puede realizarse físicamente de muchas maneras; por ejemplo, puede estar contenido en un volumen de almacenamiento o distribuirse en diversos; puede presentar o no un ordenamiento físico; los registros almacenados pueden agruparse o no en bloques; pero todo esto no debe afectar las aplicaciones; es decir, la base de datos debe crecer sin afectar las aplicaciones existentes.

Entonces, permitir que la base de datos crezca sin perjudicar las aplicaciones existentes es la razón principal para proporcionar la independencia de los datos.

Con los conceptos descritos hasta esta parte podemos concluir lo siguiente:

*"La Base de Datos puede definirse como una colección de datos interrelacionados, almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados. Dícese que un sistema comprende una colección de bases de datos cuando éstas son totalmente independientes desde el punto de vista estructural."* [9]

### **1.5 ARQUITECTURA DE UN SISTEMA DE BASES DE DATOS**

La arquitectura de un sistema de bases de datos se divide en tres niveles: interno, conceptual y externo. [10]

El nivel interno es el más cercano al almacenamiento físico, es decir, es el que concierne a la manera como los datos se almacenan en realidad; el nivel externo es el más cercano a los usuarios, es decir, es el que atañe a la manera cómo cada usuario ve los datos; y por último, el nivel conceptual es un "nivel de mediación" entre los otros dos. [11]

Una **vista** sirve para representar el contenido total de la información de la base de datos.

Llamamos **esquema** a la descripción lógica de la base de datos. [12]



La vista de un usuario individual está relacionada con el nivel externo, por lo que se le llama vista externa; la cual está definida por un esquema externo.

La vista conceptual es una representación del contenido total de la información de la base de datos, o sea, una vista de los datos "como son en realidad" y que puede ser diferente a como se almacenan. La vista conceptual es definida por medio de un esquema conceptual.

Hablar de registro interno es similar a hablar de registro almacenado.

La vista interna es una representación de nivel muy bajo de la base de datos en su totalidad.

La vista interna se describe por medio del esquema interno el cual define los diversos tipos de registros almacenados, los índices que existen, de qué manera se representan los campos almacenados, y en qué secuencia física se hallan los registros almacenados.

Hablar de esquema interno es similar a hablar de definición de la estructura de almacenamiento.

Cada usuario individual a menudo sólo se interesa en alguna parte de la base de datos total, y cada uno con una vista externa dependiendo de la aplicación que esté utilizando; por lo tanto, habrá muchas "vistas externas", cada una compuesta por una representación más o menos abstracta de alguna parte de la base de datos, y por consiguiente varios esquemas externos; y habrá una sola "vista conceptual" con una representación también abstracta de la base de datos en su totalidad, y por ende un solo esquema conceptual.

Asimismo, habrá una sola "vista interna" que representa la base de datos total tal como está almacenada, y un solo esquema interno.

El concepto *representación abstracta* se refiere a una construcción orientada al usuario (como registros y campos lógicos.)

En la siguiente figura se muestran los tres niveles generales de la arquitectura de un sistema de bases de datos.

## LOS TRES NIVELES DE LA ARQUITECTURA.

### Nivel externo

VISTAS DE LOS USUARIOS INDIVIDUALES



### Nivel conceptual

VISTA DE LA COMUNIDAD DE USUARIOS



### Nivel interno

VISTA DEL ALMACENAMIENTO



FIGURA 1.2

Los usuarios de los que se ha hablado, son programadores o usuarios de terminales en línea cuyo nivel de destreza varía.

Cada usuario tiene un lenguaje a su disposición, y este lenguaje varía dependiendo del tipo de usuario de que se trate, por ejemplo se puede tener un lenguaje de consulta para un usuario de terminal, o un lenguaje de programación para el programador de aplicaciones, etc.; lo importante del lenguaje del usuario es que incluye un *sublenguaje de datos (DSL)*, es decir un subconjunto del lenguaje total que concierne a los objetos y a las operaciones de la base de datos.

Cualquier *sublenguaje de datos (DSL)* es una combinación de un *lenguaje de definición de datos (DDL)* y un *lenguaje de manipulación de datos (DML)*.

El *DDL* permite la definición o descripción de los objetos de la base de datos tal como los percibe el usuario.

El *DML* apoya el manejo o procesamiento de esos objetos.

Las operaciones del usuario se expresan en términos de registros externos, y el *SABD* debe convertirlas en las operaciones correspondientes sobre los registros internos o almacenados. Estas últimas operaciones deben convertirse a su

vez en operaciones al nivel real del hardware, es decir, en operaciones sobre registros físicos o bloques. La componente responsable de esta conversión interna/física se llama **método de acceso**. [13]

El método de acceso se compone de un conjunto de rutinas cuya función es ocultar al SADB todos los detalles dependientes de los dispositivos y presentarle una **interfaz de registros almacenados**. De esta manera, la interfaz de registros almacenados corresponde al nivel interno, así como la interfaz con el usuario corresponde al nivel externo. [13]

Toda esta información se especificará como parte de la definición de la estructura de almacenamiento.

### **Bases de Datos Distribuidas.**

Se termina esta sección con un comentario sobre bases de datos distribuidas; ya que la tecnología de las mismas constituye un avance relativamente reciente dentro del campo general de las bases de datos.

"Una **base de datos distribuida** es, por lo común, una base de datos no almacenada en su totalidad en un solo lugar físico; sino que se distribuye a lo largo de una red de computadoras geográficamente separadas que se conectan por medio de enlaces de comunicación." [14]

## **1.6 MODELO DE DATOS RELACIONAL**

### **1.6.1 RELACIONES**

El modelo o estructura básica en las bases de datos relacionales es la relación.

**Definición** Dada una serie de conjuntos  $D_1, D_2, \dots, D_n$  (no necesariamente distintos), se dice que  $R$  es una **relación** sobre estos  $n$  conjuntos si es un conjunto de  $n$  tuplas ordenados  $\langle d_1, d_2, \dots, d_n \rangle$  tales que  $d_1$  pertenece a  $D_1$ ,  $d_2$  pertenece a  $D_2$ , ...,  $d_n$  pertenece a  $D_n$ . Los conjuntos  $D_1, D_2, \dots, D_n$  son los **dominios** de  $R$ . El valor  $n$  es el **grado** de  $R$ . [15]

Es conveniente representar una **relación** en forma de una **tabla**; cada renglón de la tabla representa un **tupla** (un  $n$  tuple) de la relación.

El número de tuplas de una relación se llama **cardinalidad** de la relación.

Finalmente diremos que la información de una base de datos relacional, se organiza y almacena en una tabla con renglones y con columnas; cada renglón está relacionado con los otros, porque todos ellos tienen el mismo tipo de información. Diferentes tablas pueden estar relacionadas por los campos comunes.

### 1.6.2 DOMINIOS Y ATRIBUTOS

Un atributo representa el uso de un dominio dentro de una relación. [16]

Uno o más atributos pueden pertenecer a un mismo dominio, por lo que para normar esto, se introduce la idea de normalización. [16]

Esto es, que en toda base de datos relacional se requiere que todas las relaciones satisfagan la condición siguiente:

"Que todo valor en la relación - es decir, cada valor de atributo en cada tuple - sea atómico." [16]

En otras palabras, en cada intersección de un renglón y una columna de la tabla, siempre hay exactamente un valor, nunca un conjunto de valores. Una relación que satisface la condición anterior, se dice que está **normalizada**. Asimismo, se dice que la relación está en la **primera forma normal** [16].

En el enfoque relacional sólo se admiten relaciones normalizadas porque: [16]

- Esta selección no impone ninguna restricción real sobre lo que puede representarse.
- La simplificación resultante en la estructura de los datos conduce a simplificaciones correspondientes en muchas otras áreas.

### 1.6.3 LLAVES

En una relación específica, hay un atributo cuyos valores son únicos dentro de la misma y, por tanto, puede ser usado para identificar los tuples de esa relación. Este atributo se conoce como la **llave primaria** de la relación [17].

No toda relación tendrá una llave primaria de un solo atributo; pero sí cada relación tendrá una combinación de

atributos que, tomados en conjunto, tienen la propiedad de identificación única.

Cuando en una relación existe más de una combinación de atributos que poseen la propiedad de identificación única, se dice que esta relación tiene más de una **llave candidata**.

Se puede encontrar a veces una relación donde hay más de una llave que posee la propiedad de identificación única y que no es la llave primaria, esta llave recibe el nombre de **llave alterna** [17].

Cuando los valores de una llave son valores de una llave primaria en otra relación, entonces a esta llave se le conoce como **llave foránea**.

Esto lleva a imponer las reglas siguientes:

#### **Regla de integridad I (Integridad de la entidad)**

"Ningún componente de un valor de una llave primaria puede ser nulo." [18]

Ya que, las llaves primarias realizan la función de identificación única en una base de datos relacional.

**Dominio primario** .- "Un dominio específico puede designarse como **primario** si y sólo si existe alguna llave primaria de un solo atributo definida sobre ese dominio". [18]

Cualquier relación que incluya un atributo que se defina sobre un dominio primario debe obedecer la restricción siguiente:

#### **Regla de Integridad II (Integridad de referencia o transitividad)**

"Sea  $D$  un dominio primario, y sea  $R_1$  una relación con un atributo  $A$  que se define sobre  $D$ . Entonces, en cualquier instante dado, cada valor de  $A$  en  $R_1$  debe ser o bien (a) nulo, o bien (b) igual a  $V$ , por ejemplo, donde  $V$  es el valor de la llave primaria de algún tuple de alguna relación  $R_2$ . ( $R_1$  y  $R_2$  no son por fuerza distintas) con llave primaria definida sobre  $D$ ." [18]

Se llama a veces **llave foránea** a un atributo como  $A$ . [18]

Finalmente diremos que el término "base de datos relacional" significa una base de datos para la cual los operadores a disposición del usuario actúan sobre estructuras relacionales.

El álgebra relacional y el cálculo relacional sirven como base para que a través de un sublenguaje de datos de alto nivel se puedan generar nuevas relaciones de otras existentes; los dos presentan muchas ventajas para los lenguajes usados para acceder bases de datos, y expresan resultados en términos de conjuntos de datos.

### 1.7 ALGEBRA RELACIONAL

Álgebra relacional es un conjunto de operaciones sobre las relaciones. Cada operación toma una o más relaciones como su(s) operando(s) y produce otra relación como su resultado.

Codd definió originalmente un conjunto de tales operaciones, y mostró que esas operaciones eran **relacionalmente completas** en el sentido de que proporcionaban al menos la potencia de recuperación del cálculo relacional.

El álgebra se compone de dos grupos de operadores:

- 1) Los tradicionales de la teoría de conjuntos.
- 2) Los operadores relacionales especiales.

Las **operaciones tradicionales** sobre conjuntos son:  
[19]

UNION	U
INTERSECCION	$\cap$
DIFERENCIA	-
PRODUCTO CARTESIANO	X

Para todas las operaciones tradicionales sobre conjuntos con excepción del producto cartesiano las dos relaciones operando deben ser compatibles con respecto a la unión, es decir, deben ser del **mismo grado** y tener los **mismos dominios**.

A continuación se da una breve descripción de estas operaciones.

### UNION

Los dominios de A son los mismos de B.

**A UNION B** = {x| x es un tuple de A o de B o de los dos}

es el conjunto de todos los tuples x que pertenecen a A, o a B, o a los dos.

### INTERSECCION

Los dominios de A son los mismos de B.

**A INTERSECCION B** = {x| x es un tuple de A y de B}

es el conjunto de todos los tuples x que pertenecen a A y a B.

### DIFERENCIA

Los dominios de A son los mismos de B.

**A DIFERENCIA B** = {x| x es un tuple de A y x no está en B}

es el conjunto de todos los tuples x que pertenecen a A y no a B.

### PRODUCTO CARTESIANO

**A PRODUCTO B** = {x| x es una concatenación de un tuple de A y un tuple de B en ese orden}

(otra forma de escribirlo es: **A VECES B**)

es el conjunto de todos los tuples x tales que, x es la concatenación de un tuple de A y un tuple de B en ese orden.

Para todas estas operaciones se considera que los atributos deben estar de izquierda a derecha dentro de una relación; y se depende de los nombres-de-atributos para efectos de referencia, exigiéndose que ninguna relación contenga dos atributos con el mismo nombre.

Para generar nombres de atributos para las relaciones derivadas, es decir, relaciones representadas por expresiones en lugar de nombres o alias se tiene lo siguiente:

"Sea **S** un alias para **R**, introducido por medio de una definición de alias:

**S APODA R;**

(**S** sencillamente es otro nombre para la relación **R**.) Cuando se hace referencia a la relación dada por el nombre **R**, se considera que el atributo **A** tiene el nombre con calificación R.A.

Cuando se hace referencia a la relación dada por el alias **S**, se considera que el atributo **A** tiene el nombre con calificación S.A. . Entonces de esta manera un atributo de una relación declarada siempre tiene un nombre con calificación" [20].

En la unión, la intersección y la diferencia el resultado tiene los mismos nombres-de-atributo con calificación que los primeros operandos.

Los atributos de **A VECES B** (producto cartesiano extendido) tienen exactamente los nombres-de-atributos con calificación de **A** y **B**. [20]

Las operaciones relacionales especiales son [21]:

SELECCION  
PROYECCION  
REUNION  
DIVISION

A continuación se da una breve descripción de estas operaciones.

#### **SELECCION**

El operador algebraico de selección produce un subconjunto "horizontal" de una relación específica; es decir el subconjunto de los tuples de la relación dada para el cual se cumple un predicado específico (es decir, una condición.) El predicado se expresa como una combinación booleana de términos, donde cada término es una comparación simple que se puede establecer como verdadera o falsa para un tuple dado inspeccionando ese tuple por separado.

**SELECCION A DONDE** <predicado> = {**x** | **x** es un tuple de **A**  
y predicado es verdadero para **x**}



## PROYECCION

El operador algebraico de proyección produce un subconjunto "vertical" de una relación dada; es decir, el subconjunto obtenido al seleccionar los atributos especificados, en un orden especificado de izquierda a derecha y eliminando los tuples duplicados en los atributos seleccionados.

Ningún atributo se puede especificar más de una vez en una operación de proyección.

La omisión de la lista de nombres-de-atributo es equivalente a especificar una lista que contiene todos los nombres-de-atributos de la relación dada, en el mismo orden (de izquierda a derecha); en otras palabras, tal proyección es idéntica a la relación especificada.

**PROYECTA A SOBRE <lista de atributos> =**

El operador de proyección produce un subconjunto vertical de una relación dada.

## REUNION

Por definición es equivalente a tomar el producto cartesiano extendido de las dos relaciones dadas y luego realizar una selección adecuada sobre ese producto.

La reunión de la relación **A** sobre el atributo **X** con la relación **B** sobre el atributo **Y** es el conjunto de todos los tuples **x** tales que **x** es la concatenación de un tuple **a** que pertenece a **A** y un tuple **b** que pertenece a **B**, donde **x** cumple con una condición.

**REUNE (A VECES B) DONDE <predicado > = {x | x es una concatenación de un tuple de A y un tuple de B y se cumple que predicado es verdadero para x}**

A la reunión donde la "condición de reunir" se fundamenta en la igualdad entre valores de la columna común se le conoce como **equirreunión** [22].

## DIVISION

El operador de división divide una relación dividendo **A** de grado  $m+n$  entre una relación divisor **B** de grado  $n$ , y produce una relación resultado de grado  $m$ . El  $(m+i)$ -ésimo

atributo de **A** y el *i*-ésimo atributo de **B** (*i* en el rango de 1 a *n*) deben estar definidos sobre el mismo dominio.

Considérense los primeros *m* atributos de **A** como un solo atributo compuesto **x**, y los últimos *n* como otro atributo compuesto **y**; entonces definimos el resultado de dividir **A** entre **B** como sigue:

**A DIVIDENTRE B** = {**x** | el par  $\langle x, y \rangle$  aparece en **A** para todos los valores **y** que aparecen en **B**}

Los atributos del resultado tienen los mismos nombres con calificación que los primeros *m* atributos de **A**.

#### **Notaciones finales acerca del álgebra relacional.**

El resultado de cualquier operación de recuperación puede considerarse una tabla, en otras palabras, el proceso de recuperación es un proceso de construcción de tablas [23]; por ello, el álgebra relacional es en esencia un lenguaje de recuperación.

La importancia del álgebra estriba en que proporciona un patrón con respecto al cual se pueden medir otros lenguajes.

Un lenguaje es completo si incluye equivalentes de los operadores algebraicos ya que el álgebra es relacionalmente completa.

También podemos decir que sienta las bases para la investigación de varios aspectos de la administración de Bases de Datos, tales como el diseño de bases de datos, la definición de vistas y la reestructuración.

Se usa además como soporte para trabajos sobre técnicas de optimización.

Para finalizar esta sección y señalando la importancia que tiene el álgebra relacional, diremos que las condiciones para definir el modelo de bases de datos relacionales constan de dos componentes principales:

- Estructura de datos relacionales.
- Álgebra relacional.

Un sistema de bases de datos se puede llamar totalmente relacional si soporta:

- Bases de datos relacionales (incluidos los conceptos de dominio y llave y las 2 reglas de integridad)
- Un lenguaje que sea al menos tan potente como el álgebra relacional.

Un sistema que soporta bases de datos relacionales, pero que tiene un lenguaje menos potente que el álgebra se le llama semirelacional.

### 1.8 CALCULO RELACIONAL

"El cálculo relacional es una notación que sirve para definir una relación que ha de derivarse de las relaciones ya existentes en la base de datos." [24]

En otras palabras, a través tanto del cálculo relacional, como del álgebra relacional se puede describir una relación derivada de relaciones existentes; la gran distinción entre las dos notaciones es que en el cálculo no se especifica la secuencia de operaciones; es decir, no es un lenguaje procedimental por lo que con él se expresa lo que se quiere obtener, no cómo obtenerlo. La forma general de una expresión de cálculo relacional es:

**<lista> donde <predicado>**

donde **<lista>** es una lista de especificaciones de dominio para que la relación resultado sea expresada en términos de los atributos de relaciones existentes, y **<predicado>** es una condición o expresión booleana en términos similares.

Un aspecto fundamental del cálculo y de los lenguajes basados en él, es la noción de variable de tupla.

En el cálculo relacional se crea una variable de tupla tal que es una variable que "varía sobre" alguna relación con nombre, es decir, una variable cuyos únicos valores permitidos son tuplas de esa relación.

"Se dice que un lenguaje es relacionalmente completo si posee la propiedad de que cualquier relación definible por medio de las expresiones del cálculo se puede recuperar mediante proposiciones pertinentes de ese lenguaje." [25]

Un ejemplo de cómo se manejan las variables de tuplas en el cálculo relacional es el siguiente:

Sea **inventario** el nombre de una relación declarada; se define una variable de tuple cualquiera por ejemplo **i**, con la expresión:

**range of i is inventario**

Donde **i** es sencillamente un alias temporal declarado para poder vincularnos a la relación **inventario** durante la sesión que estamos llevando a cabo; y sea **matriz** el nombre de uno de los atributos contenidos en la estructura de esta relación. Entonces, se hace referencia a la información almacenada en este campo a través del alias de la relación, de tal forma que el atributo **matriz** tiene el nombre con calificación **i.matriz** para los procesos que se ejecuten en la relación **inventario** durante la sesión de trabajo.

En otras palabras, la variable de tuple **i** varía sobre la relación **inventario**; y por medio de ella se utilizan los nombres con calificación para los atributos que se desean trabajar dentro de esa relación en el manejo del cálculo relacional. Por ejemplo, si deseamos obtener todos los tuples con la información de los libros que se dieron de alta en la relación mencionada, en una fecha determinada (en nuestro ejemplo el mes de febrero), escribiremos la siguiente expresión de cálculo relacional:

**retrieve (i.matriz) where i.fecha="9402"**

Esta expresión se traduce como sigue: "para cada valor posible de la variable **i**, recupere el componente **matriz** de ese valor si y sólo si el componente **fecha** tiene el valor del mes de febrero del año 1994".

Como se ve en este caso, se tiene un conocimiento previo de como se está almacenando la información, ya que el componente **matriz** es numérico y el componente **fecha** es una cadena de cuatro caracteres (año y mes.)

## 1.9 NORMALIZACIONES

Dado un conjunto de datos a representar... ¿cómo se opta por una estructura lógica adecuada para esos datos? En otras palabras ¿cómo se decide qué relaciones se necesitan y qué atributos deben tener?..... Esto es el problema del *diseño de bases de datos*.

"...Así pues, quizás un buen criterio de diseño sea el de <<cada hecho en un lugar>> (es decir, evitar la redundancia en tanto sea posible.)" [26]

La teoría de la normalización es una formalización que tiene aplicación práctica en el área de diseño de bases de datos.

La teoría de la normalización es útil en el proceso de diseño de bases de datos, pero no es una panacea. Se aconseja familiarizarse con las técnicas de normalización básicas pero no se sugiere que el diseño se base sólo en los principios de normalización.

La teoría de la normalización permite reconocer cuando una relación posee propiedades indeseables a pesar de estar normalizada e indica cómo tales relaciones se pueden convertir a una forma más deseable.

La teoría de la normalización está basada en el concepto de **formas normales**.

#### **Formas normales [26]**

Se dice que una relación está en una forma normal particular si satisface cierto conjunto específico de restricciones. Estas restricciones serán presentadas más adelante.

La normalización es un proceso de paso a paso que permite reemplazar la correspondencia entre datos con relaciones bidimensionales. Las tablas deberán organizarse de forma tal que no se pierda ninguna de las relaciones existentes entre datos de tal suerte que en el proceso de normalización se eliminen todas las estructuras no simples, y se conviertan a la forma de tablas bidimensionales. Es ésta la que llamamos **primera forma normal**.

Dicho de otra manera, para obtener la primera forma normal hay que reducir todas las estructuras de datos que no sean bidimensionales a relaciones bidimensionales.

El proceso de normalización subsiguiente parte las relaciones que se tienen en primera forma normal en relaciones aún más simples; primero se reducen los datos a la segunda forma normal y posteriormente a la tercera.

Podemos continuar con una mayor descomposición y pasar a la cuarta forma normal y posteriormente a la quinta, pero para el propósito de este documento con la descripción de hasta la tercera forma normal es suficiente por lo que ya no se detallan éstas últimas.

## **Definiciones:**

### **Primera forma normal**

"Una relación R está en **primera forma normal (1FN)** si y sólo si todos los dominios subyacentes sólo contienen valores atómicos." [27]

Para definir la 2a. forma normal primero se define lo que significa funcionalmente dependiente:

### **Dependencia Funcional**

"Dada una relación R, el atributo Y de R es **funcionalmente dependiente** del atributo X de R si y sólo si cada valor de X en R tiene asociado a él exactamente un valor de Y en R (en cualquier instante.)" [28]

Una dependencia funcional es una forma de restricción de integridad.

Para obtener la 2a. forma normal hay que eliminar cualquier dependencia incompleta de atributos no primos de llaves primarias.

Un atributo es **no primo** si no participa en la llave primaria.

Se debe especificar la restricción de dependencia funcional en la definición de la base de datos; es decir, desde el esquema conceptual de tal forma que SABD pueda hacerla cumplir.

Dos atributos son mutuamente independientes si ninguno es funcionalmente dependiente del otro.

La 2a. forma normal, se define de la siguiente manera:

### **Segunda forma normal**

"Una relación R está en **segunda forma normal (2FN)** si y sólo si está en 1FN y cada atributo no primo es completamente dependiente de la llave primaria." [29]

Aquí vemos que es en la forma 2FN donde se identifica(n) la(s) llave(s), prestando especial atención en detectar los campos que dependen de la llave primaria; que no necesariamente es un solo campo, ya que puede ser una llave compuesta, asegurando con esto la dependencia funcional.

Una relación que esté en 1FN y no en 2FN se puede reducir siempre a un conjunto equivalente de relaciones en 2FN. La reducción consiste en reemplazar las relaciones por *proyecciones* adecuadas, de tal forma que al efectuar una *reunión natural* (equirreunión con la duplicación de columnas eliminada) en el conjunto de estas proyecciones se recuperará la relación original. En otras palabras, el proceso es reversible. [29]

Cuando se hace una reducción de una relación a un conjunto de relaciones equivalentes a través de proyecciones como se acaba de describir, de tal forma que no se pierda información en estas relaciones resultantes, se dice que se hace una **descomposición sin pérdidas**. [29]

Para obtener la 3a. forma normal hay que eliminar cualquier dependencia transitiva de atributos no primos de llaves primarias; y se define de la siguiente manera:

#### **Tercera forma normal**

"Una relación R está en **tercera forma normal (3FN)** si y sólo si está en 2FN, y todo atributo no primo no es dependiente transitivamente de la llave primaria." [29]

La 3a. forma normal (3FN) a veces se llama Forma Normal de Boyce/Codd, y se abrevia FNBC.

A esta forma se le llama también dependencia transitiva y siempre que se desee convertir una relación de la forma 2FN a la 3FN se tiene que ver primero si existe transitividad, específicamente transitividad de campo en campo.

Antes de decir si una relación en un instante dado está o no en 3FN es necesario conocer el significado de los datos, esto es, las dependencias implícitas.

El SABD no puede asegurar que una relación se mantenga en 3FN (o cualquiera otra excepto 1FN) sin tener información de todas las dependencias pertinentes; sin embargo, para una relación en 3FN, todo lo que se necesita informar al SABD sobre esas dependencias es una indicación de el(los) atributo(s) que constituye(n) la llave primaria. [29]

#### **Relaciones con más de una llave candidata**

Se llama *determinante funcional* a un atributo, tal vez compuesto, del cual depende funcionalmente en forma completa algún otro atributo. Entonces se define la Forma Normal de Boyce/Codd (FNBC) como sigue:

"Una relación R está en Forma Normal de Boyce/Codd (FNBC) si y sólo si cada determinante es una llave candidata." [29]

Dos llaves candidatas se traslapan si comprenden dos o más atributos cada una y si tienen algún atributo en común.

Una relación que no se puede descomponer en componentes independientes, se dice que es atómica.

#### Conclusiones acerca de las normalizaciones:

El objetivo principal de la normalización es utilizar las técnicas de *descomposición sin pérdidas* como ayuda en el diseño de bases de datos relacionales. La idea es comenzar con alguna relación dada, junto con una declaración de ciertas restricciones e ir reduciendo en forma sistemática a un conjunto de relaciones que sean equivalentes a la original, pero en alguna forma preferible a ella, usando las restricciones para guiarse en el proceso de reducción como sigue:

- a) Tomar proyecciones de la relación original en 1FN para eliminar cualquier dependencia funcional no completa, generando así un conjunto de relaciones en 2FN.
- b) Tomar proyecciones de estas relaciones en 2FN para eliminar cualquier dependencia transitiva, generando así un conjunto de relaciones en 3FN.
- c) Tomar proyecciones de estas relaciones en 3FN para eliminar cualquier dependencia funcional donde el determinante no sea una llave candidata, generando así un conjunto de relaciones en FNBC forma de Boyce/Codd.

Estos tres pasos se condensan en: "Tomar proyecciones de la relación original para eliminar las dependencias funcionales en las que el determinante no sea una llave candidata".

El objetivo principal del proceso de normalización es reducir la redundancia y, por tanto, evitar ciertos problemas con las operaciones de actualización.

Las nociones de dependencia y normalización atañen al significado de los datos. Por el contrario, los lenguajes tales como el álgebra relacional y cálculo relacional conciernen únicamente a los valores reales de los datos. En particular, estos lenguajes no requieren que las relaciones



sobre las que operan se encuentren en alguna forma normal distinta de la 1FN. La normalización se puede entender principalmente como una disciplina con la que el diseñador de bases de datos puede capturar una parte, aunque sea pequeña, de la semántica del mundo real de la empresa que la base de datos representa. [30]

En la siguiente y última sección del presente capítulo, se habla de manera muy general sobre las consideraciones que se deben de observar en el manejo de Bases de Datos Grandes.

### **1.10 CONSIDERACIONES PARA BASES DE DATOS GRANDES**

La definición de una "base de datos muy grande" depende de las capacidades y limitaciones de un Sistema Manejador de Bases de Datos dado (SABD). Esta sección considera una base de datos relacional muy grande si al menos una relación contiene algunos millones de tuples, o donde al menos una relación contiene múltiplos en gigabytes de datos. (Gigabytes -mil megabytes- es la mejor medida del volumen de una relación para representar tanto el tamaño del tuple como la cuenta de tuples.)

Algunos ejemplos de sistemas con bases de datos muy grandes ocurren en los siguientes dominios: la Industria de los Seguros, la de los Servicios Financieros, la de las Telecomunicaciones, la de la Mercadotecnia en gran masa, las Bibliotecas Públicas, etc.

Se deben considerar cinco condiciones generales en el manejo de bases de datos grandes: [31]

#### **1. Eficiencia en la Recuperación.**

En una base de datos relacional muy grande, las búsquedas y los ordenamientos pueden ser muy lentos. Es necesario seleccionar mecanismos de acceso apropiados, las indexaciones y los ordenamientos pueden reducir o eventualmente eliminar grandes ordenamientos y búsquedas. También es necesario tener relaciones de muestra para que los usuarios y los desarrolladores puedan hacer pruebas en ellas; y por último, implementar relaciones de resúmenes que quizás puedan convertir corridas largas de solicitudes en accesos cortos y simples.

#### **2. Manejo de Transacciones.**

Existen muchas transacciones por segundo, por lo cual es necesario contar con:

- *Opciones de Bloques.*
- *Agrupamientos de los Datos.*
- *Particionamiento o Segmentación las tablas.*
- *Control de Procesos.*
- *Segmentación de actualizaciones.*

### **3. Procedimientos para Insertar y Borrar.**

Insertar o borrar millones de registros puede ser una ejecución mala, por lo que es necesario contar con que se debe:

- *Usar una utilería para carga masiva de los datos; ya que están especialmente diseñadas para manejar grandes cantidades de registros.*
- *Inhabilitar las entradas o accesos; ya que muchas veces esto es necesario para agilizar los procesos de actualización durante una carga de información.*
- *Deshechar temporalmente y después recrear índices; ya que si el número de inserciones y/o actualizaciones es muy grande, el deshechar y recrear índices puede ser más rápido que su mantenimiento durante las actualizaciones de la relación.*

### **4. Consideraciones para el punto de verificación y de reiniciación en procesos muy largos.**

Se debe de cuestionar qué tanto tiempo de ejecución se necesita para acceder grandes volúmenes de datos. Se deben considerar los efectos y consecuencias de la reiniciación de procesos para las Bases de Datos relacionales muy grandes y asegurar que los que involucran accesos largos a la base puedan ser recuperados y reestablecidos en lapsos de tiempos razonables. Se puede por ejemplo, dividir un proceso muy largo en varios procesos que se lleven a cabo en lapsos más cortos, de tal forma que sean salvados los resultados de cada uno de estos procesos pequeños en relaciones o archivos.

## 5. Utilerías.

Los procesos de utilería pueden requerir de tiempo excesivo para su ejecución. Es necesario acondicionar estos procesos para:

- Favorecer la ejecución paralela de utilería; particionando o segmentando las tablas, como por ejemplo para efectuar respaldos, reorganizaciones, recuperaciones, etc.. Es necesario evaluar las circunstancias para utilizar los procesos de utilería en paralelo.
- Sacar ventaja de utilerías parciales. Ver por ejemplo, la posibilidad de alternar respaldos completos con respaldos parciales, donde un respaldo parcial copia sólo los bloques físicos que han cambiado desde el último respaldo; además, los respaldos parciales pueden ejecutarse más rápidamente.
- Las utilerías que acumulan estadísticas del funcionamiento se programen sólo cuando sean necesarias. Minimizar las frecuencias de utilerías que actualicen catálogos de estadísticas.
- Asegurar tiempos de recuperación. Evaluar y ajustar el tiempo de recuperación de una base de datos grande y sus índices asociados es importante, ya que es un proceso que puede llegar a durar días o semanas si se trata de una base muy grande

Por otro lado, tenemos que las bases de datos en general no son objetos estáticos, las relaciones contenidas en ellas varían con el tiempo por lo que se tienen que ajustar y acondicionar a los cambios que se vayan requiriendo. Estos cambios pueden afectar no sólo valores de los datos sino también definiciones de columnas, relaciones y reglas asociadas con las relaciones. Muchos son relativamente fáciles de implementar, aunque también se presentan cambios que resultan más difíciles porque afectan programas y vistas a usuarios.

El desafío es entonces entender la planeación e implementación asociados con la ejecución de cambios en un medio ambiente relacional, y en particular en uno donde se manejen grandes volúmenes de información.

Se direccionan estos objetivos a tres tipos de cambios en una base de datos relacional grande: eliminar, agregar y modificar objetos.

A continuación se describen estas tres acciones:

### **1. Eliminar Objetos.**

Se refiere a la acción de borrar un objeto y su definición o descripción. Existen dos razones para borrar un objeto:

- i) ya no se necesita más el objeto
- ii) se necesita modificar el objeto de tal manera que primero se tiene que borrar y luego redefinir.

Antes de borrar un objeto deben estar claras las implicaciones de eliminarlo; se deben planear y programar cuidadosamente estas operaciones. Analizar qué otros objetos también deben ser borrados. Estar seguros de una referencia antes de borrar el objeto para obtener una lista de objetos relacionados, y por lo último, notificar al usuario.

En general, al trabajar con una base de datos grande se debe estar consciente de las modificaciones estructurales que requieren el eliminar, redefinir, y recrear objetos existentes.

Es responsabilidad del encargado de efectuar estos procesos, el estar seguro de que al borrar un objeto no se dañe la integridad de la base de datos.

En un medio ambiente donde al anular un objeto automáticamente se genera una cascada de objetos dependientes también eliminados, o donde los objetos no pueden ser borrados hasta que todos los objetos dependientes no se hayan suprimido, hay que establecer mecanismos o procedimientos para identificar y subsecuentemente recrear tales objetos dependientes.

Se puede borrar un objeto sólo si se mira alrededor de éste y se crea otro similar, quizás para implementar cambios que no se podían aplicar al objeto original. O también, se puede necesitar posteriormente restaurar uno erróneamente borrado, por lo que hay que estar seguros de mantener copias del lenguaje de definición de datos para crear todos los objetos. En estos casos, también se necesitará la autoridad apropiada para recrear la definición del objeto y para restaurar el contenido del(los) mismos; además de reestablecer los privilegios de acceso eliminados.

## **2. Agregar Objetos.**

En general, agregar objetos a una base de datos relacional es una operación simple. Se necesita considerar permisos, propietarios, seguridad y autorizaciones.

Para relaciones, índices o productos específicos de bases de datos también se necesita calcular los requerimientos de espacio.

Si se aumenta una relación a la base de datos, se debe identificar sus vínculos con las tablas existentes, e identificar además las llaves primarias, las llaves foráneas y las reglas para insertar y borrar.

## **3. Modificar Objetos.**

Modificar objetos puede ser más complejo que agregar nuevos.

Los cambios pueden ser de dos tipos: cambios en la representación física y cambios en la estructura lógica.

Los cambios en la representación física involucran las estructuras de almacenamiento.

Los cambios lógicos en la estructura involucran objetos visibles para los usuarios (relaciones, vistas, columnas) y las reglas de integridad.

Es importante hacer estos cambios tan transparentes como sea posible para los usuarios y para las aplicaciones existentes.

Si se quieren renombrar objetos, se deben usar sinónimos.

Si se van a agregar columnas a una relación se deben considerar las implicaciones con respecto a los dominios y la secuencia de las columnas dentro de ella.

Se deben documentar también los cambios diseñados en el diccionario de datos. Incluir en él una descripción del diseño original, del diseño de los cambios y de la razón de los cambios.

Para finalizar, diremos también que se debe de crear un escenario para incorporar los cambios antes de intentar implementarlos.

**CAPITULO 2**

---

---

**INGENIERIA  
DEL  
SOFTWARE**

---

---

# INGENIERIA DEL SOFTWARE

## 2.1 INTRODUCCION

En este capítulo se presentan los conceptos básicos de la **ingeniería del software**, que hoy en día está reconocida como una disciplina derivada de una investigación seria y un estudio minucioso y que proporciona los elementos que se requieren para desarrollar un software confiable y de calidad para la explotación de cualquier sistema basado en computadora.

## 2.2 SOFTWARE E INGENIERIA DEL SOFTWARE

El software se ha convertido en un elemento clave de la evolución de los sistemas y productos informáticos. En su historia ha pasado de ser una resolución de problemas especializada y herramientas de análisis de la información, a una industria por sí misma, pero en esta evolución se han creado también un conjunto de problemas que hoy en día aún persisten, y es por medio de la ingeniería del software que se dan pautas para limitar y/o preveer su presencia en los sistemas informáticos. [32]

El rápido desarrollo de la informática ha intensificado la crisis del software de la siguiente forma: [33]

1. La sofisticación del hardware ha dejado desfasada nuestra capacidad de construir software que pueda explotar el potencial del hardware.
2. Nuestra capacidad de construir nuevos programas no puede dar abasto a la demanda de éstos.
3. Nuestra capacidad de mantener los programas existentes está amenazada por el mal diseño y el uso de recursos inadecuados.

La crisis del software se refiere a un conjunto de problemas encontrados en el desarrollo del software de computadoras. Esta crisis no sólo se limita al software que "no funciona adecuadamente" sino que abarca las dificultades asociadas con cómo desarrollar y mantener el volumen creciente de software existente y cómo satisfacer su demanda también creciente.

Y la respuesta para encontrar soluciones a esta crisis del software es la **ingeniería del software**; ya que estos problemas pueden corregirse dando un enfoque de ingeniería al

desarrollo del software, junto con la mejora continua de técnicas y herramientas. [34]

En realidad, el software es un elemento lógico en vez de físico del sistema; a diferencia del hardware, cuyo proceso creativo se traduce finalmente en forma física. Por lo tanto, el software tiene las siguientes características considerablemente distintas a las del hardware: [35]

- El software es desarrollado; no es fabricado en un sentido clásico.

- El software no se "estropea", es decir, no es susceptible de males del entorno como el hardware; aunque por cambios sufridos por mantenimiento *si se va deteriorando*. También el deterioro se va presentando por errores de diseño y traducción en código de máquina o por la presencia de virus en la máquina donde se trabaja.

- La mayoría del software se construye a la medida, en vez de ensamblar componentes existentes; aunque nuevas tecnologías plantean lo contrario.

#### **Definición de ingeniería del software**

Una primera definición fue propuesta por Fritz Bauner en la primera conferencia importante dedicada al tema [36], y determina la ingeniería del software como:

*"El establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales."* [37]

Aunque se han propuesto muchas definiciones globales, todas refuerzan la importancia de una disciplina de ingeniería para el desarrollo del software.

*"La ingeniería del software abarca un conjunto de tres elementos claves; - métodos, herramientas y procedimientos - que facilitan al gestor controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva."* [37]

■ Los métodos de ingeniería del software suministran el "cómo" construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen:



- *planificación y estimación de proyectos;*
- *análisis de los requerimientos del sistema y del software;*
- *diseño de estructuras de datos, arquitectura de programas y procedimientos algorítmicos;*
- *codificación;*
- *prueba y mantenimiento.*

■ **Las herramientas de la ingeniería del software** suministran un soporte automático o semiautomático para los métodos.

■ **Los procedimientos de la ingeniería del software** son el "pegamento" que une a los métodos y herramientas y facilita un desarrollo racional y oportuno del software de computadora.

Por otro lado, tenemos que la ingeniería del software está compuesta de **pasos** que incluyen los métodos, herramientas y procedimientos acabados de mencionar. Estos pasos se denominan frecuentemente **paradigmas de la ingeniería del software**. Un paradigma para la ingeniería del software se elige basándose en la naturaleza del proyecto y de la aplicación, los métodos y herramientas a usar y los controles y entregas requeridos. [37]

El proceso de desarrollo del software contiene tres fases genéricas independientemente del paradigma elegido. Las tres fases son: **definición, desarrollo y mantenimiento**; mismas que se encuentran en todos los desarrollos de software, independientemente del área de aplicación, tamaño del proyecto o complejidad. [38]

■ **La fase de definición** se enfoca sobre el **qué**. Esto es, durante la definición el que desarrolla el software intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué interfaces han de establecerse, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto.

Por tanto, han de identificarse los requerimientos claves del sistema y del software. Aunque los métodos aplicados durante la fase de definición variarán dependiendo del paradigma de ingeniería del software (o combinación de

paradigmas) aplicado, de alguna forma se producirán los tres siguientes pasos específicos:

- *Análisis del sistema*
- *Planificación del proyecto de software*
- *Análisis de requerimientos*

■ La **fase de desarrollo** se enfoca sobre el **cómo**. Esto es, durante esta fase, el que desarrolla el software intenta descubrir cómo han de diseñarse las estructuras de datos y la arquitectura del software, cómo han de implementarse los detalles procedimentales, cómo ha de traducirse el diseño a un lenguaje de programación (o a un lenguaje no procedimental) y cómo ha de realizarse la prueba.

Los métodos aplicados durante la fase de desarrollo variarán dependiendo del paradigma de ingeniería del software (o combinación de paradigmas) aplicado. Sin embargo, de alguna forma se producirán los tres siguientes pasos concretos:

- *Diseño del software*
- *Codificación*
- *Prueba del software*

■ La **fase de mantenimiento** se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas por la evolución del entorno del software y a las modificaciones debidas a los cambios de los requerimientos del cliente dirigidos a reforzar o a ampliar el sistema.

Durante la fase de mantenimiento se encuentran tres tipos de cambios:

- *Corrección*
- *Adaptación*
- *Aumento*

Las tres fases genéricas y sus pasos relacionados descritos se complementan con las siguientes actividades:

Las **revisiones** que se realizan durante cada paso para asegurar que se mantiene la calidad; la **documentación** que se desarrolla y controla para asegurar que toda la información

sobre el sistema y el software estará disponible para un uso posterior; y el control de los cambios que se instituye de forma que los cambios puedan ser mejorados y registrados. [39]

Para finalizar esta sección, concluiremos que la **ingeniería del software** es una disciplina que integra métodos, herramientas y procedimientos para desarrollar software de alta calidad para sistemas basados en computadora, y siempre se define primero, luego se desarrolla y finalmente se mantiene. [40]

### **2.3 ANALISIS DE REQUERIMIENTOS DEL SISTEMA Y DEL SOFTWARE**

#### **Consideraciones sobre el elemento humano**

Un sistema basado en computadora casi siempre tiene un elemento humano. Una persona puede interactuar directamente con el hardware y el software, a fin de conducir las funciones del sistema; y en todos los casos la gente es la responsable del desarrollo, soporte o mantenimiento del sistema.

Para que el ingeniero de sistemas pueda asignar una función al elemento humano, se debe especificar antes la interacción que es necesaria para poder realizar la función. Para hacerlo, se deben entender los "componentes" del elemento humano como lo son: la memoria humana y la representación del conocimiento, el pensamiento y el razonamiento, la percepción visual y la construcción del diálogo humano.

El ingeniero de sistemas especifica una interfaz hombre-máquina tal que permite a la persona desarrollar heurísticas (pautas, reglas y estrategias) para la interacción hombre-máquina. En general, estas heurísticas deben permanecer invariables a lo largo de diferentes dominios de interacción.

La mayoría de las interacciones hombre-máquina se realiza a través de un medio visual (informes impresos, gráficas, pantallas, etc.), y el ingeniero de sistemas debe tomar en cuenta las limitaciones físicas y de conocimiento del elemento humano, logrando con esto una interfaz "amigable".

Además de los "componentes" del elemento humano antes mencionados, es importante tener en cuenta la capacidad individual y la sofisticación de los usuarios del sistema a la hora de asignar procesos. Una interfaz que es

completamente aceptable para un ingeniero podría ser inadecuada para un trabajador sin preparación. [41]

La **ingeniería humana** es una actividad multidisciplinaria que aplica un conocimiento derivado de la psicología para especificar y diseñar interacciones hombre-máquina de alta calidad.

El proceso de ingeniería humana comprende los siguientes pasos:

■ **Análisis de actividad.** - Se evalúa cada actividad asignada a un elemento humano en el contexto de la interacción requerida con otros elementos.

■ **Análisis y diseño semántico.** - Se define el significado preciso de cada acción requerida por el usuario y de cada acción producida por la máquina.

■ **Diseño léxico y sintáctico.** - Se identifica y representa la forma específica de las acciones y las órdenes, luego se diseña la implementación del software y del hardware de cada acción u orden.

■ **Diseño del entorno del usuario.** - El hardware, el software, y otros elementos del sistema se combinan para formar un entorno del usuario.

■ **Creación de prototipos.** - Es indispensable el utilizar un prototipo para especificar una interacción hombre-máquina, con el fin de poder evaluar esta interacción desde una perspectiva humana.

#### **Consideraciones de la base de datos**

No todos los sistemas basados en computadora hacen uso de una base de datos, pero para aquellos que sí lo hacen este almacenamiento de la información a menudo es crucial para el funcionamiento general del sistema.

"La **ingeniería de bases de datos** (análisis, diseño e implementación de bases de datos) es una disciplina técnica que se aplica una vez que se ha definido el ámbito de la información. Por ello, el papel del ingeniero de sistemas es el de definir la información que va a contener la base de datos, los tipos de peticiones que se podrán procesar, la manera en que se accederá a los datos y la capacidad de la base de datos." [42]

### **Análisis del sistema**

El análisis del sistema es una actividad que engloba a todos los elementos del sistema.

El **análisis del sistema** define el papel de cada elemento de un sistema informático, asignando finalmente al software el papel que va a desempeñar. [43]

El análisis del sistema se realiza teniendo presente los siguientes objetivos: [44]

- Identificar las necesidades del cliente.
- Evaluar la viabilidad del sistema.
- Realizar un análisis técnico y económico
- Asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema.
- Establecer restricciones de costo y de tiempo.
- Crear una definición del sistema que sea la base para todo el trabajo de ingeniería subsecuente (tanto en software como en hardware, en ingeniería humana y en bases de datos.)

### **2.4 PRINCIPIOS DEL ANALISIS**

Todos los métodos de análisis están relacionados por un conjunto de principios fundamentales: [45]

1. *"El dominio de la información, así como el dominio funcional de un problema debe ser representado y comprendido."* [46]

Todas las aplicaciones del software pueden llamarse *procesamiento de datos*; es decir, el software se construye para procesar datos.

El ámbito de información contiene tres planteamientos diferentes de los datos y del control a medida que son procesados por un programa de computadora:

- **el flujo de información** representa la manera en que los datos y el control cambian conforme se mueven a través de un sistema

- **el contenido de información** representa los elementos de datos individuales que componen otros elementos mayores de información
- **la estructura de información** representa la organización interna de los distintos elementos de datos y de control

2. "El problema debe subdividirse de forma que se descubran los detalles de una manera progresiva (o jerárquica.)" [46]

Para poder comprender los problemas como un todo es necesario particionarlos (en partes que se puedan entender fácilmente) y establecer interfaces entre las partes, de forma que se realice la función global.

Durante el análisis de requerimientos, son particionados tanto el dominio funcional como el dominio de la información del software. En esencia, la partición descompone un problema en sus partes constituyentes. Conceptualmente, establecemos una representación jerárquica de la función o de la información y luego descomponemos el elemento superior de la siguiente forma:

- exponiendo cada vez más detalles, al movernos verticalmente por la jerarquía.
- o
- descomponiendo funcionalmente el problema, al movernos horizontalmente por la jerarquía.

En las relaciones de control entre elementos, un elemento que controla a otro elemento se dice que es *superior* a él, e inversamente, un elemento controlado por otro se dice que es un *subordinado* del controlador.

3. "Deben desarrollarse las representaciones lógicas y físicas del sistema." [46]

La visión lógica de los requerimientos del software presenta las funciones que han de realizarse y la información que ha de procesarse, independientemente de los detalles de implementación.

La visión física de los requerimientos del software presenta la manifestación en el mundo real de las funciones de procesamiento y de las estructuras de información.

El análisis de requerimientos del software debe enfocarse sobre lo que el software debe realizar, en vez de cómo se va a implementar el procesamiento.

El análisis debe ser conducido independientemente del paradigma de ingeniería del software aplicado; sin embargo la forma que ese análisis tomará puede variar.

En unos casos es posible aplicar los principios fundamentales del análisis y derivar a una especificación del software en papel a partir del cual se pueda desarrollar el diseño; en otros casos, se va a una recolección de los requerimientos, se aplican los principios de análisis y se construye un modelo de software, denominado **prototipo**. Otras veces, hay circunstancias en la que se requiere la construcción de un prototipo al comienzo del análisis, ya que el modelo es el único medio a través del cual se pueden obtener los requerimientos en forma efectiva.

La especificación de los requerimientos del software se produce en la culminación de la tarea del análisis. Los requisitos se representan de forma que conduzcan finalmente a una correcta implementación del software. La función y el comportamiento asignados al software se refina, estableciendo una descripción completa de la información, una descripción funcional detallada, una indicación de los requerimientos de rendimiento y de las restricciones de diseño, unos criterios de validación apropiados y otros datos pertinentes sobre los requerimientos.

El que desarrolla el software y el usuario/cliente deben realizar una cuidadosa revisión de la *Especificación de los Requerimientos del Software* (y del prototipo), ya que en ésta se fundamenta la fase de desarrollo.

#### **Conclusiones acerca de los fundamentos del Análisis de Requerimientos y sus Principios:**

El análisis de requerimientos es el primer paso técnico del proceso de ingeniería del software.

El análisis debe enfocarse tanto en los dominios de la información como en el funcional y de comportamiento del problema. Para comprender mejor lo que se solicita, el problema se subdivide y se representa en enfoques lógicos y físicos.

En muchos casos, no es posible especificar completamente un problema en una etapa tan temprana. La construcción de prototipos ofrece un método alternativo, que da como resultado un modelo ejecutable del software, a partir del cual se pueden refinar los requerimientos.

Como resultado del análisis se desarrolla la especificación de requerimientos del software la cual es necesario sea revisada con mucho cuidado tanto por el cliente como por el que hace el desarrollo.

## **2.5 REQUERIMIENTOS DE LAS BASES DE DATOS**

El análisis de requerimientos para una base de datos incorpora las mismas tareas que el análisis de requerimientos del software. Es necesario un contacto estrecho con el cliente; es esencial la identificación de las funciones e interfaces; se requiere la especificación del flujo, estructura y asociatividad de la información y debe desarrollarse un documento formal de los requerimientos. [47]

### **Características de las bases de datos**

Ya hemos hablado de lo que es una base de datos y solamente diremos que contiene entidades de información que están relacionadas vía organización y asociación. La arquitectura lógica de una base de datos se define mediante un esquema que representa las definiciones de las relaciones entre las entidades de información. La arquitectura física de una base de datos depende de la configuración del hardware residente; sin embargo, tanto el esquema (descripción lógica) como la organización (descripción física) deben adecuarse para satisfacer los requerimientos funcionales y de comportamiento para el acceso a análisis y creación de informes.

### **Pasos del análisis**

Antes de que pueda comenzar la evaluación de los requerimientos de una base de datos, el analista debe comprender los objetivos y ámbitos globales del sistema para el que se va a desarrollar la base de datos.

Posteriormente se desarrolla un modelo de información completo y con gran detalle.

El siguiente paso en el análisis de una base de datos es definir las características lógicas y físicas de la base de datos.

Con el modelo de información y la especificación del sistema como guía, el analista, trabajando junto con el diseñador de la base de datos, define la organización lógica de los datos; y posteriormente la organización física.



En la organización lógica debe considerarse los requerimientos de acceso, modificación, asociatividad de los datos, así como otros aspectos orientados al sistema; y una vez que ha sido establecida la organización lógica de los datos, debe desarrollarse la organización física.

La organización física de una base de datos define la estructura de archivos, los formatos de los registros, las características de procesamientos dependientes del hardware y las características del Sistema Administrador de Bases de Datos (SABD).

Finalmente ha de realizarse una completa revisión del esquema y características físicas.

Los aspectos sobre los que puede ser necesario hacer compromisos en el diseño final de una base de datos incluyen la especificación frente a la generalización, el grado de asociatividad, el potencial de expansión y características operacionales.

Una base de datos especializada se enfoca sobre los requerimientos específicos de la información de un sistema, a diferencia de una base de datos generalizada que es aplicable a una clase más amplia de requerimientos de información.

La generalidad se consigue a expensas de un mayor procesamiento en la interface de la información, una mayor sobrecarga asociada con la adaptabilidad a un rango más amplio de problemas y una mayor complejidad interna en la organización de las estructuras de datos.

La estructura, el tamaño y el diseño lógico de una base de datos puede tener un impacto significativo en la organización física, hardware, métodos de acceso y rendimiento.

### **Normalización**

Como ya vimos, la técnica llamada normalización se utiliza para simplificar la estructura lógica de los datos, identificando para eliminar los que sean redundantes; determinando las llaves necesarias para el acceso a los elementos de datos y ayudando a establecer las relaciones necesarias entre los elementos de datos.

Pueden aplicarse tres niveles de normalización, llamados formas normales (1FN, 2FN, 3FN), ver Capítulo 1.

### **Análisis de la capacidad**

Los procedimientos para el análisis de bases de datos son análogos a los que se aplican para el análisis de los

requerimientos de programas. Sin embargo, actividades como la normalización y análisis de la capacidad deben también considerarse.

Para finalizar, diremos que el análisis de una base de datos requiere un exámen del uso proyectado para la base de datos, así como de su contenido y organización.

## 2.6 DISEÑO DEL SOFTWARE

El diseño es el primer paso de la fase de desarrollo de cualquier producto o sistema de ingeniería; y se define como:

*"...el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física". [48]*

El objetivo del diseñador es desarrollar un modelo combinando: la intuición y los criterios en base a la experiencia de construir entidades similares, un conjunto de principios y/o heurísticas que guían la forma en la que se desarrolla el modelo, un conjunto de criterios que permiten discernir sobre la calidad y un proceso de iteración que conduce finalmente a una representación del diseño final.

El diseño del software se asienta en el núcleo técnico del proceso de ingeniería del software y se aplica independientemente del paradigma de desarrollo utilizado; y es la primera de tres actividades técnicas - *diseño, codificación y prueba*.

Los requisitos del programa, establecidos por medio de los modelos de información, funcional y de comportamiento, alimentan el paso del diseño. Mediante metodologías de diseño se realiza el diseño de datos, el diseño arquitectónico y el diseño procedimental.

El *diseño de datos* transforma el modelo del campo de información, creado durante el análisis, en las estructuras de datos que se van a requerir para implementar el software. El *diseño arquitectónico* define las relaciones entre los principales elementos estructurales del programa. El *diseño procedimental* transforma los elementos estructurales en una descripción procedimental del software. Se genera el código fuente y, para integrar y validar el software, se llevan a cabo las pruebas.

Además de estos diseños, muchas aplicaciones modernas requieren una actividad distinta de *diseño de la interfaz*. El

diseño de la interfaz establece la disposición y los mecanismos para la interacción hombre-máquina.

Estas fases de diseño, codificación y prueba absorben el 75% o más del costo de la ingeniería del software (excluyendo el mantenimiento.)

Es en la fase de diseño de software donde se toman decisiones que afectarán finalmente al éxito de la implementación del programa y a la facilidad del mantenimiento del mismo.

Lo que se considera más importante en esta fase es la calidad, es decir, es por medio del diseño como podemos evaluar la calidad del software, además de ser la base de todas las etapas posteriores del desarrollo y de la fase de mantenimiento.

El diseño del software es un proceso por medio del cual se traducen los requisitos en una representación del software.

En el proceso de diseño en la ingeniería del software se obtiene un buen modelo mediante la aplicación de principios fundamentales de diseño, de una metodología sistemática y de una concienzuda revisión.

Por otro lado, todas las metodologías de diseño de software tienen las siguientes características comunes:

- 1) un mecanismo para la traducción de la representación del campo de información en una representación de diseño.
- 2) una notación para representar los componentes funcionales y sus interfaces
- 3) heurísticas para el refinamiento y la partición
- 4) criterios para la valoración de la calidad

Independientemente de la metodología de diseño usada, un ingeniero de software debe aplicar un conjunto de conceptos *fundamentales* al diseño de datos, arquitectónico y procedimental.

## 2.7 FUNDAMENTOS DEL DISEÑO DE SOFTWARE

Los conceptos fundamentales para el diseño del software proporcionan la base necesaria para que un programa no solamente funcione bien, sino que además **funcione correctamente**. [49]

A continuación se presentan estos conceptos:

### **Abstracción**

La *abstracción* permite concentrarse en un problema a un nivel de generalización, independientemente de los detalles irrelevantes de bajo nivel. El uso de la abstracción también permite trabajar con conceptos y términos que son familiares al entorno del problema, sin tener que transformarlos a una estructura no familiar.

Cuando se considera una solución modular para cualquier problema, pueden formularse muchos niveles de *abstracción*. En el nivel superior de abstracción, se establece una solución en términos amplios usando el lenguaje del entorno del problema. En los niveles inferiores de abstracción se toma una orientación más procedimental. La terminología orientada al problema se acompaña con una terminología orientada a la implementación. En el nivel más bajo de abstracción, se establece la solución de modo que pueda implementarse directamente.

### **Refinamiento**

El *refinamiento sucesivo* es una estrategia de diseño descendente. La arquitectura de un programa se desarrolla en niveles sucesivos de refinamiento de los detalles procedimentales.

En cada paso (del refinamiento), una o varias instrucciones del programa, se descomponen en instrucciones más detalladas en forma sucesiva hasta llegar a las sentencias del lenguaje de programación.

Cada paso de refinamiento implica algunas decisiones de diseño.

### **Modularidad**

En la modularidad, el software se divide en componentes con nombres y ubicaciones determinados, que se denominan *módulos* y que se integran para satisfacer los requisitos del problema.

En otras palabras, un problema complejo es más fácil de resolver si se divide en trozos más manejables.

Cualquier sistema se puede diseñar de forma modular.

### **Arquitectura del software**

La arquitectura del software se refiere a dos características importantes del software de computadora:

- (1) la estructura jerárquica de los componentes procedimentales (módulos)
- (2) la estructura de los datos.

### **Jerarquía de control**

La jerarquía de control, también denominada estructura del programa, representa la organización (frecuentemente jerárquica) de los componentes del programa (módulos) e implica una jerarquía de control. No representa aspectos procedimentales del software, tales como la secuencia de procesos, la ocurrencia u orden de decisiones o la repetición de operaciones.

### **Ocultamiento de información**

El principio de ocultamiento de información dice que los módulos se han de "caracterizar por decisiones de diseño que los oculten unos a otros"; es decir, los módulos deben especificarse y diseñarse de forma que la información (procedimientos y datos) contenida dentro de un módulo sea inaccesible a otros módulos que no necesiten tal información.

Por lo tanto, el ocultamiento implica que para conseguir una modularidad efectiva hay que definir un conjunto de módulos independientes, que se comuniquen con los otros sólo mediante la información que sea necesaria para realizar la función del software.

Los fundamentos de diseño que se han descrito hasta este punto, sirven todos para incentivar los diseños modulares.

Un diseño modular reduce la complejidad, facilita los cambios y produce como resultado una implementación más sencilla; al mismo tiempo que facilita el desarrollo paralelo de las diferentes partes de un sistema.

Para definir módulos en una arquitectura de software, se utilizan la abstracción y el ocultamiento de información. [50]

### **Independencia funcional**

El concepto de *independencia funcional* es una derivación directa del de modularidad y de los conceptos de abstracción y ocultamiento de información.

La independencia funcional es la clave de un buen diseño y el diseño es la clave de la calidad del software.

La independencia funcional se adquiere desarrollando módulos con "una clara" función y una "aversión" a una excesiva interacción con otros módulos. Dicho de otro modo, se trata de diseñar software de forma que cada módulo se centre en una subfunción específica de los requisitos y tenga una interfaz sencilla.

La independencia se mide con dos criterios cualitativos: la *cohesión* y el *acoplamiento*.

### **Cohesión**

"La cohesión es una extensión del concepto de ocultamiento de información. Un módulo cohesivo ejecuta una tarea sencilla de un procedimiento de software y requiere poca interacción con procedimientos que ejecutan otras partes de un programa" [51]; es decir, un módulo cohesivo sólo hace (idealmente) una cosa.

La cohesión puede ser representada como un "espectro" donde el punto medio es aceptable (aquí la cohesión es casi tan buena como una gran cohesión); no así la cohesión baja, donde es mucho peor que una de la mitad del espectro.

Un módulo que realice un conjunto de tareas que están débilmente relacionadas entre sí (es decir, un módulo de cohesión baja), es *coincidentalmente cohesivo*. Uno que realice tareas relacionadas en forma lógica es *logicamente cohesivo*. Uno que contenga tareas que están relacionadas por el hecho de que se ejecutan en el mismo momento, se dice que exhibe *cohesión temporal*.

Los niveles moderados de cohesión están relativamente cercanos unos a otros en su grado de independencia modular. Cuando los elementos de procesamiento de un módulo están relacionados y deben ejecutarse en un orden específico, existe *cohesión procedimental*. Cuando todos los elementos de procesamiento se concentran sobre un área de una estructura de datos, se presenta una *cohesión de comunicación*. Una

cohesión alta se caracteriza porque el módulo realiza una tarea procedimental determinada.

En la siguiente figura se muestra un "espectro" de cohesión.

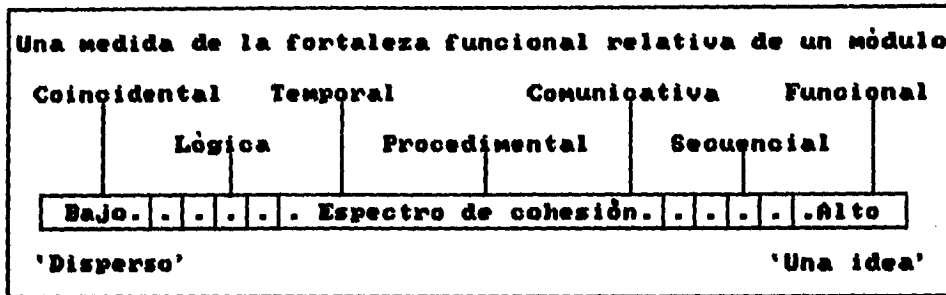


Figura 2.1 Espectro de cohesión

A continuación se da un conjunto de criterios sencillos para establecer el grado de cohesión: [52]

Una técnica útil para determinar si un módulo está funcionalmente con cohesión es escribir una frase que describa la función (propósito) del módulo y luego examinar dicha frase. Puede hacerse la siguiente prueba:

1. Si la frase resulta ser una sentencia compuesta, contiene una coma o contiene más de un verbo, probablemente el módulo realiza más de una función; por tanto, probablemente tiene cohesión secuencial o de comunicación.

2. Si la frase contiene palabras relativas al tiempo, tales como "primero", "a continuación", "entonces", "después", "cuándo", "al comienzo", etc., entonces probablemente el módulo tiene una cohesión secuencial o temporal.

3. Si el predicado de la frase no contiene un objeto específico sencillo a continuación del verbo, probablemente el módulo esté acotado lógicamente. Por ejemplo, editar todos los datos tiene una cohesión lógica; editar sentencia fuente puede tener cohesión funcional.

4. Palabras tales como "inicializar", "limpiar", etc., implican cohesión temporal.

Los módulos funcionalmente con cohesión siempre se pueden describir en función de sus elementos usando una sentencia compuesta. Pero si no se puede evitar el lenguaje anterior, siendo aún una descripción completa de la función del módulo, entonces probablemente el módulo no tenga cohesión.

### Acoplamiento

El acoplamiento es una medida de la interconexión entre los módulos de una estructura de programa. El acoplamiento puede ser representado como un "espectro" al igual que la cohesión, y depende de la complejidad de las interfaces entre los módulos, del punto en el que se hace una entrada o referencia a un módulo y de los datos que pasan a través de la interfaz.

En el diseño de software, buscamos el más bajo acoplamiento posible. La conectividad sencilla entre los módulos da como resultado un software que es más fácil de comprender y menos propenso a que los errores se propaguen a lo largo del sistema.

Cuando existe una lista de argumentos sencilla (es decir, cuando se pasan datos simples y existe una correspondencia uno a uno entre elementos), se exhibe en esta porción de la estructura, un bajo acoplamiento (acoplamiento de datos en el espectro.) Una variación del acoplamiento de datos, denominado *acoplamiento por estampado*, se presenta cuando se pasa una porción de una estructura de datos (en vez de argumentos simples) a través de una interfaz de módulo (parámetros.)

En niveles moderados, el acoplamiento se caracteriza por el paso de control entre módulos. En el *acoplamiento de control*, el control se pasa mediante un "indicador" sobre el que se toman las decisiones en un módulo subordinado o superior.

Los niveles relativamente altos de acoplamiento se producen cuando los módulos están ligados a un entorno externo al software; por ejemplo, la entrada y salida (E/S) acopla un módulo con dispositivos, formatos y protocolos de comunicación específicos. El *acoplamiento externo* es esencial, pero debe limitarse a un número pequeño de módulos dentro de una estructura.

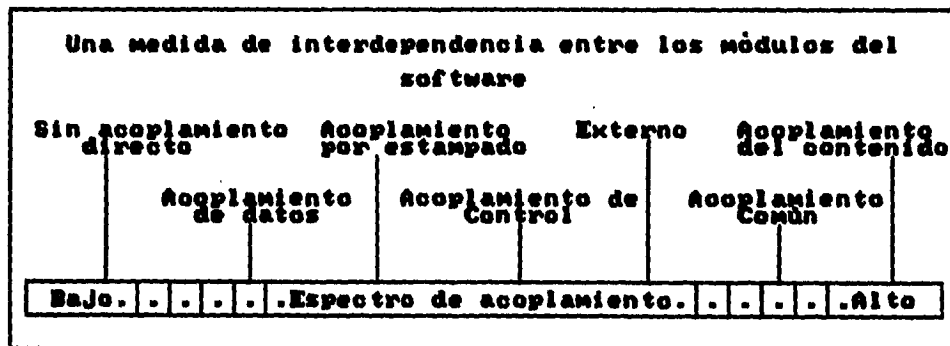


Quando varios módulos se referencian a un área de datos global se denomina *acoplamiento común*; y éste es también un alto acoplamiento.

El diagnóstico de los problemas que se presentan en estructuras con considerable acoplamiento común consume mucho tiempo y es difícil. Sin embargo, el uso de datos globales no es necesariamente malo, por lo que el diseñador sólo debe tener cuidado y estar consciente de las posibles consecuencias del acoplamiento común.

Quando un módulo utiliza información de datos o de control contenida dentro de los límites de otro módulo se denomina *acoplamiento por contenido*, y es el que presenta el mayor grado de acoplamiento; razón por la cual debe evitarse. También se produce este tipo de acoplamiento cuando se realiza una bifurcación hacia la mitad de un módulo.

En la siguiente figura se representa un espectro de acoplamiento.



**Figura 2.2 Espectro de acoplamiento**

Todos estos modos de acoplamiento surgen por decisiones de diseño que se toman en el desarrollo de la estructura; sin embargo, pueden introducirse durante la codificación variantes de acoplamiento externo.

**Conclusiones acerca del diseño del software y de sus fundamentos:**

El diseño es técnicamente la parte central de la ingeniería del software. Durante el diseño se desarrollan, se revisan y se documentan los progresivos refinamientos de las

estructuras de datos, de la estructura del programa y de los detalles procedimentales.

El diseño da como resultado representaciones del software cuya calidad se puede evaluar.

Los conceptos de modularidad (tanto de programa como de datos) y de abstracción permiten al diseñador simplificar y reutilizar los componentes del software. El refinamiento es un mecanismo que permite representar sucesivas capas de detalle funcional. Las estructuras del programa y de los datos contribuyen a la visión general de la arquitectura del software, mientras que el procedimiento proporciona los detalles necesarios para la implementación de los algoritmos. El ocultamiento de información y la independencia funcional son heurísticas que permiten conseguir una modularidad efectiva.

El diseño del software puede verse desde las perspectivas técnicas y de gestión del proyecto.

Desde el punto de vista técnico, el diseño comprende cuatro actividades: diseño de los datos, arquitectónico, procedimental y de interfaces.

La notación de diseño, junto con los conceptos de la programación estructurada, permite al diseñador representar los detalles procedimentales, facilitando su traducción al código. Las herramientas disponibles son gráficas, tabulares y textuales.

Finalmente no hay que precipitarse, **el diseño merece la pena.**

En la siguiente y última sección del presente capítulo, se habla de manera general sobre las consideraciones que se deben de observar en el mantenimiento al software desarrollado en cualquier sistema automatizado.

## **2.8 MANTENIMIENTO DEL SOFTWARE**

El mantenimiento es la última fase del proceso de ingeniería del software, y se lleva la mayor parte del presupuesto destinado al software de computadora. [53]

En la construcción de sistemas basados en computadora, el cambio es inevitable; por ello, debemos desarrollar mecanismos de evaluación, control e implementación de modificaciones.

Sobre el software de computadora se llevan a cabo cuatro tipos de mantenimiento, a saber:

El *mantenimiento correctivo* actúa para corregir errores que no han sido descubiertos antes de poner en uso el software.

El *mantenimiento adaptativo* se aplica cuando los cambios del entorno externo precipitan las modificaciones del software.

El *mantenimiento perfectivo* incorpora mejoras solicitadas por la comunidad de usuarios.

Finalmente, el *mantenimiento preventivo* mejora la futura facilidad de mantenimiento y la fiabilidad como base para las futuras mejoras.

La *facilidad de mantenimiento* se puede definir cualitativamente como la facilidad de comprender, corregir, adaptar y/o mejorar el software.

La *facilidad de mantenimiento* es un fin clave que guía los pasos de cualquier metodología de ingeniería del software.

La *facilidad de mantenimiento* que se consiga para el software se ve afectada por muchos factores. Una falta de cuidado en el diseño, en la codificación o en la prueba tiene un impacto obviamente negativo sobre nuestra capacidad de mantener fácilmente el software. Una pobre configuración del software puede tener un similar impacto negativo incluso cuando se hayan seguido cuidadosamente los ya mencionados pasos técnicos.

Las tareas asociadas con el mantenimiento del software comienzan mucho antes de que se haga una petición de mantenimiento. Inicialmente, se debe establecer una organización de mantenimiento; se deben prescribir procedimientos de evaluación y de información, y se debe definir una secuencia estandarizada de sucesos para cada petición de mantenimiento. Además se debe establecer un sistema de registro de información de las actividades de mantenimiento y definir criterios de revisión y de evaluación.

Para terminar esta última sección del capítulo diremos que las tareas desarrolladas durante el proceso de ingeniería del software definen la *facilidad de mantenimiento* y tienen un gran impacto sobre el éxito de cualquier método de mantenimiento.

**CAPITULO 3**

---

---

**DESCRIPCION DEL  
MEDIO AMBIENTE  
DEL SISTEMA DE  
INVENTARIOS**

---

---

## DESCRIPCION DEL MEDIO AMBIENTE DEL SISTEMA DE INVENTARIOS

### 3.1 INTRODUCCION

En este capítulo se hace la descripción del medio ambiente del sistema INVENTARIOS. Primero se da una explicación general del ámbito donde se encuentra implantado el sistema de inventarios, se presentan los antecedentes del sistema, se describen la problemática que llevó a desarrollarlo y los procesos de registro de los libros tanto en forma general dentro del sistema LIBRUNAM como en forma particular dentro del control del inventario de una biblioteca dada. También se detalla el proceso de registro de los inventarios cuando se llevan a cabo levantamientos de inventarios físicos en las bibliotecas universitarias. Se incluye al final del capítulo la delimitación del equipo de cómputo utilizado para el desarrollo del sistema.

La Universidad Nacional Autónoma de México es una de las principales instituciones de educación superior a nivel nacional, por lo que sus objetivos primordiales son el de apoyar, fomentar, desarrollar y difundir la enseñanza, y por ende, la investigación científica y la cultura en todas sus ramas.

Para lograr estos propósitos tiene que contar con una variedad de apoyos básicos entre los que destacan los que ofrecen sus bibliotecas; siendo la biblioteca uno de los principales medios de recopilación, recuperación y transmisión de la información que se genera a través del quehacer universitario.

Por lo anterior, el progreso de una biblioteca se efectúa paralelamente al de todos los campos abarcados por el conocimiento humano.

Mientras más grande sea el tamaño de la colección contenida en una biblioteca, se vuelve más imperioso el elaborar una infraestructura administrativa eficiente que soporte los incrementos del material sin mermar el acceso a la información; ya que también es responsabilidad de toda biblioteca ofrecer al usuario la consulta de las obras más adecuadas a sus necesidades en el menor tiempo posible desde su edición.

En la Universidad Nacional Autónoma de México, cada escuela, colegio, facultad, instituto y centro de investigación disponen, al igual que diversas dependencias administrativas y académico-administrativas, de una o más bibliotecas; las que en su conjunto (cerca de 200) reúnen y ponen a disposición de los universitarios la colección

bibliográfica más grande, heterogénea e importante del país destinada a la docencia, investigación y difusión de la cultura; por lo que su control, mantenimiento y organización hacen que se convierta en un sistema bibliográfico automatizado de relevante importancia.

Es la Dirección General de Bibliotecas la que se encarga de coordinar, apoyar y homogeneizar todos los servicios que ofrecen a la comunidad universitaria estas bibliotecas.

#### **¿Qué o quién es la Dirección General de Bibliotecas?**

Antes de que se concediese la autonomía a la UNAM en 1929, ésta dependía de la Secretaría de Educación Pública, la cual en 1922 establece el Departamento de Bibliotecas, al que se incorporan en 1924 las bibliotecas anexas a las facultades universitarias. Dos años más tarde, se establece dentro de la misma dependencia, la Biblioteca de Bibliografía y Catálogos.

Al convertirse en autónoma la Universidad Nacional de México, esta biblioteca se fusiona con el Departamento de Bibliotecas Universitarias, dando origen al Departamento Técnico de Bibliotecas; el cual, fue hasta 1954 que se albergó en el edificio de la Biblioteca Central de Ciudad Universitaria después de haber permanecido en otras varias localidades. La Biblioteca Central inicia sus labores en 1956. Posteriormente, es en 1966 que el Departamento Técnico de Bibliotecas se transforma en lo que hoy se conoce como la **Dirección General de Bibliotecas (DGB)**, y la Biblioteca Central permanece con este carácter hasta 1975, año en que se constituye en el Departamento de Servicios al Público de la Dirección General de Bibliotecas. [54]

La *Dirección General de Bibliotecas* es el organismo perteneciente a la Universidad Nacional Autónoma de México que se encarga de normar el desarrollo (en todos sentidos) del sistema bibliotecario. Es decir, se encarga de dirigirlo y coordinarlo; de conseguir el establecimiento de normas jurídicas universitarias que regulen la actividad bibliotecaria. Se encarga asimismo de realizar con prontitud y calidad los procesos técnicos de los materiales adquiridos por las bibliotecas del sistema y de mantener un sistema de información sobre los mismos. También se hace cargo de coadyuvar en la vigilancia de la utilización de los recursos presupuestarios y de todo tipo que se destinen a los servicios bibliotecarios; de establecer los lineamientos generales para la utilización racional de los recursos bibliotecarios disponibles; de prestar servicios bibliotecarios en sus propias unidades; de opinar respecto a la creación, fusión, edificación, ampliación o remodelación de bibliotecas; de difundir los planes, programas e informes que genere el sistema bibliotecario; de promover la

capacitación y el mejoramiento profesional del personal bibliotecario; y en general en cada uno de los aspectos que admita el proceso global de la administración en esta materia. [55] [56]

### 3.2 ANTECEDENTES DEL SISTEMA DE INVENTARIOS

El acelerado adelanto de la tecnología ha logrado que los sistemas automatizados aplicados a las bibliotecas compuestas por un acervo grande, representen la mejor herramienta para ayudar a solucionar los problemas administrativos de las mismas.

Las bibliotecas de la UNAM constituyen en su conjunto una de las fuentes documentales más importantes de América Latina; y por esto y por la gran cantidad de usuarios que utilizan sus servicios se volvió imprescindible el establecimiento de un sistema automatizado que ayudara a llevar a cabo todas las tareas inherentes a ellas; ya que se requería manejar el gran volumen de información con que cuenta la Universidad en forma rápida, oportuna y confiable, para que los resultados obtenidos con su manejo llegasen a cada una de las personas que los solicitaran.

Fue entonces que, en la Dirección General de Bibliotecas se hizo un estudio a conciencia de todas estas necesidades, se analizó la problemática y se escogió una solución de entre varias propuestas, sobre la que se diseñó y se desarrolló el sistema integral de manejo de información de libros denominado **LIBRUNAM**. El diseño de este sistema se hizo por módulos y fue llevado a cabo por un grupo multidisciplinario de personas; entendiéndose desde el principio que debía ser compatible con otros sistemas.

Las primeras fichas catalográficas<sup>1</sup> fueron incorporadas a la base de datos desarrollada, a principios de 1978; y posteriormente se fueron implementando todos los demás módulos del sistema, adecuándolos a las modificaciones y cambios de estrategia necesarios al implantar un nuevo sistema, hasta que en 1979 quedó ya conformado de modo tal que ya se podía considerar como completo. Después de esto, se capacitó al personal académico, al bibliotecario y al no profesional para su utilización. [57]

1. Ficha catalográfica se refiere a una ficha que pertenece a un libro cualquiera (i.e. una ficha bibliográfica) a la que ya se le aplicó el proceso de catalogación, por lo que a partir de este punto se usarán en el doc. indistintamente los 2 términos

El manejo de este sistema mejoró y/o transformó muchas áreas del sistema bibliotecario universitario; y pronto se ganó un reconocimiento tanto nacional como internacional por las facilidades y beneficios que se obtenían con su uso, además de ser un sistema muy amigable con el usuario y muy adecuado a sus necesidades.

Pronto se generaron más requerimientos a partir de su explotación, las cuales llevaron a la creación e integración de nuevos módulos; además de que su diseño sirvió como base para el desarrollo de otros sistemas que también prestan servicio en la DGB.

Actualmente LIBRUNAM está integrado por alrededor de 580,000 registros de títulos diferentes de libros que se encuentran en las bibliotecas de la UNAM y que corresponden aproximadamente a 3,370,000 volúmenes.

La recuperación de la información es muy flexible y puede obtenerse por cualquiera de los elementos de la ficha como lo son: autor, título, tema, clasificación, editorial, ISBN, número de título (matriz), o cualquier palabra que se encuentre dentro del texto de la ficha (o combinación de ellas.) Estos elementos se constituyen en las llaves de acceso para la recuperación de la información bibliográfica. También se puede refinar la búsqueda de una ficha en la base de datos a través de una combinación de estas llaves.

Una de las características más importantes de este sistema y que lo distingue de entre otros, es la recuperación fonética de la información, esto es que, si se desconoce la ortografía de algún vocablo buscado y se incurre en un error ortográfico al hacer la búsqueda, la información que se solicita de igual forma será recuperada.

Podemos decir que LIBRUNAM es un sistema basado en computadoras que contiene datos bibliográficos. También podemos decir que es un sistema seguro, confiable y compatible con la mayor parte de los catálogos automatizados del mundo, además de que contiene rutinas que agilizan la adquisición y proceso de los libros; duplica y desarrolla juegos de tarjetas catalográficas; genera todas la serie de impresiones para el pago y la distribución de los libros; así como un conjunto de productos que sirven de apoyo a las bibliotecas universitarias y a la propia DGB. Sirve también para controlar la producción y rendimiento del personal involucrado con el sistema; además de llevar el control de la calidad del banco de datos. También contiene rutinas que agilizan los procesos de adquisición de libros, impresión y ordenamiento de fichas catalográficas y todas las formas necesarias para la distribución de los libros (catálogos electrónicos, discos compactos, etc..)



Es así como se ha desechado el concepto obsoleto de la biblioteca como almacén de libros o como sala de lectura; y se ha dado paso a una visión de biblioteca como una parte dinámica de los procesos de educación, investigación y desarrollo de la comunidad universitaria y del país.

A continuación se muestra un diagrama con los sistemas automatizados y su interrelación que fueron desarrollados en la DGB para apoyar al sistema bibliotecario de la UNAM.

**SISTEMAS AUTOMATIZADOS QUE SE LLEVAN A CABO EN LA DIRECCION GENERAL DE BIBLIOTECAS**

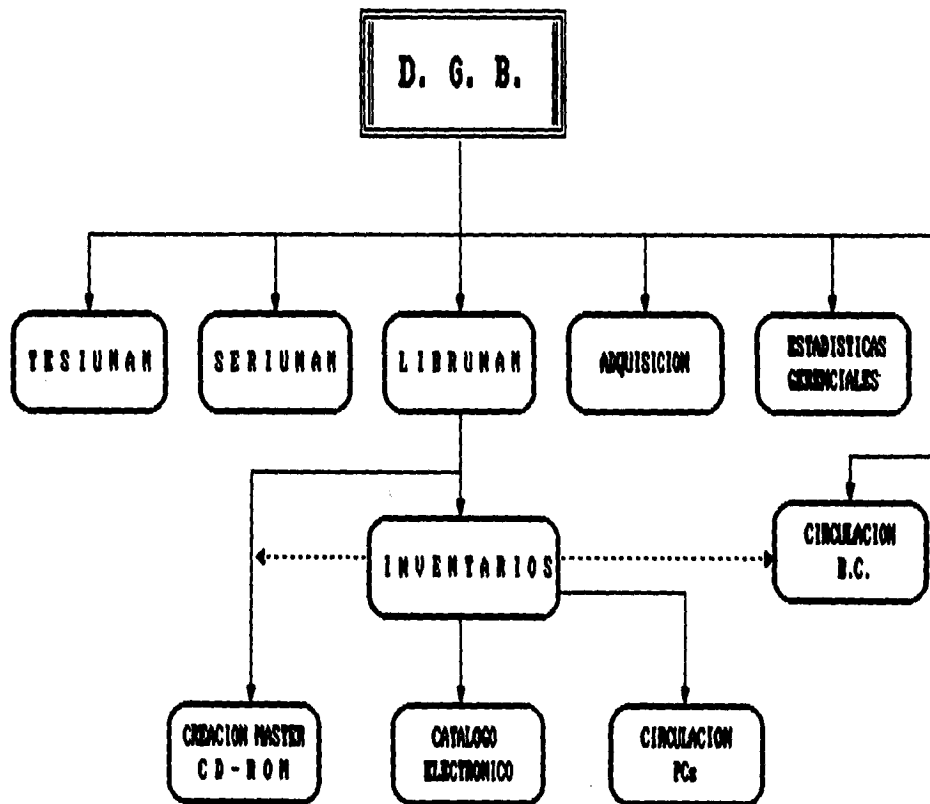


FIGURA 3.1

### **3.3 ¿ EN QUE CONSISTE EL SISTEMA DE INVENTARIOS ?**

En la UNAM cada biblioteca departamental selecciona el material que necesita para satisfacer al universo de usuarios que atiende; sin embargo, los procesos de adquisición y técnicos de los libros se realizan en forma centralizada en la Dirección General de Bibliotecas; por consiguiente, es importante que dentro del sistema se lleve un estricto control de cada libro adquirido, esto comprende el control centralizado que debe tener la DGB y el que debe tener cada biblioteca de la UNAM sobre su acervo.

Esta necesidad ha llevado a que se le dé una importancia prioritaria a uno de los módulos que conforman el sistema LIBRUNAM, llamado **INVENTARIOS** y que es el que se encarga de este control, que además, probablemente sea la única forma viable para unificar las funciones de los servicios que se prestan dentro de las bibliotecas. También, podemos decir que es el medio que se utiliza para la importante tarea de cuantificar el acervo universitario y poder llevar parámetros estadísticos confiables sobre las necesidades de la población universitaria.

Otro de los objetivos del sistema de Inventarios es garantizar la uniformidad de los catálogos (tanto en tarjetas como electrónicos) de las bibliotecas departamentales en relación con la DGB y saber con exactitud la ubicación de los libros para facilitar su acceso a todo tipo de usuarios.

El sistema **INVENTARIOS**, que en realidad puede ser considerado como un subsistema del sistema LIBRUNAM, fue creado mediante análisis e investigaciones previas y desde su inicio ha marcado los lineamientos que deberán seguir las bibliotecas para optimizar los servicios que proporcionan.

### **3.4 DESCRIPCION DEL SUBSISTEMA DE INVENTARIOS**

#### **3.4.1 Análisis**

La problemática que llevó en un principio al desarrollo del Subsistema de Inventarios fue el deficiente servicio de préstamo de libros, siendo sus principales fallas las siguientes:

- Inexistencia de catálogos.
- Catálogos deficientes por falta de tarjetas.

- Catálogos deficientes por falta de calidad técnica en lo referente a la clasificación y catalogación descriptiva.
- Un gran porcentaje de ubicación equívoca de libros.
- Material desaparecido que continúa registrado en catálogos.
- Falta de personal y/o capacitación del mismo, entre otros.

Debido a estos y más problemas, el resultado es un servicio en la bibliotecas departamentales que no satisface las necesidades del usuario.

La automatización del subsistema de INVENTARIOS como un módulo del sistema LIBRUNAM se volvió entonces necesaria y básica para poder crear la infraestructura de cómputo que ayudara a implantar los sistemas locales de circulación automatizada (préstamo de libros) y los sistemas globales de préstamo interbibliotecario.

Para su diseño como sistema fueron necesarios un período de investigación de factores y análisis de la problemática, para después llegar a una síntesis del nuevo sistema en sus formas:

- 1) general
- 2) detallada

Para el análisis del sistema fue necesario:

- a) definir el/los objetivos del sistema
- b) recopilar y describir las salidas, y con esto nos referimos a los listados que, como producto del módulo se requieren, por ejemplo: archivos, tarjetas catalográficas, etc.

Estas dos medidas que se llevaron a cabo ayudaron a establecer los límites del subsistema.

Para determinar las salidas se hizo un estudio del contenido y formato de las mismas, así como de los tiempos de entrega de éstas y de la frecuencia de actualización de la información proporcionada por el sistema.

También se tomó en cuenta:

- A quiénes se les tenía que distribuir estas salidas (las bibliotecas)
- El uso que se les iba a dar
- Y para un futuro, se pensó en la elaboración de resúmenes incidentales, los cuales debieran ser alimentados mas o menos en forma periódica (como pueden ser: reportes de avances de trabajo, listas de errores, etc..)

El módulo automatizado de Inventarios se trabajó desde sus inicios hasta antes de 1985 en una computadora Burroughs B6800, la cual debido a diversas circunstancias, llegó al punto en que ya no cumplía con todos los requerimientos que se habían determinado para su desarrollo; por lo que se hizo imperativa la necesidad de encontrar una nueva solución, tal que no sólo resolviera los problemas técnicos que se estaban presentando, sino que además permitiera tanto la autoadministración del equipo de cómputo, como el proveer a un futuro un mayor grado de descentralización de las bibliotecas; con un control limitado que les concediera cierta independencia de funciones, sin dejar de seguir los lineamientos generales coordinados por la Dirección General de Bibliotecas.

Esta insuficiencia no solamente afectaba al subsistema INVENTARIOS, sino que ya representaba un problema serio para el manejo de todos los sistemas controlados en la DGB.

Se decidió entonces adquirir un nuevo equipo de cómputo, optando por uno pionero en México: The Intelligent Database Machine, conocida como IDM de la serie 500 y fabricada por la compañía Britton Lee, la cual se accesa por medio de una máquina Alpha-Micro (minicomputadora multi-usuario.)

La IDM es una máquina configurada como manejadora de bases de datos desde el hardware, lo cual hace que el manejo de las bases de datos creadas en ella (en forma de árboles), sea mucho más rápido que si utilizáramos un paquete de los existentes en el mercado.

Esta máquina es muy potente en cuanto a sus capacidades, siendo además relativamente fácil su manejo, lo que facilitó el transporte del sistema LIBRUNAM a esta nueva plataforma.

Los objetivos primordiales que se consideraron para la adquisición de este equipo fueron:

- Resolución de los problemas técnicos en el manejo tanto del equipo de cómputo como de los sistemas.
- Autoadministración de los recursos de cómputo.
- Descentralización a un futuro no lejano de las bibliotecas departamentales con una cierta independencia de funciones.
- Establecimiento de una red automatizada de bibliotecas de la UNAM, siendo la Dirección General de Bibliotecas la administradora general y en donde además se albergara la base de datos maestra del sistema.

Una vez adquirido este equipo, se comenzó a desarrollar el nuevo sistema de INVENTARIOS para implantarlo en él.

#### **Desarrollo del nuevo Sistema Automatizado de Inventarios**

Para desarrollar el nuevo sistema de Inventarios, que es el que está actualmente funcionando, se siguieron las siguientes etapas:

1) Estudio del proyecto inicial del sistema automatizado de Inventarios para examinar el estado en que se encontraba y los logros que se tenían a la fecha.

Se recopilaron y detallaron todas las salidas (es decir, los archivos, listados de errores, etc.) revisando el contenido y formato de éstas, las cuales se sintetizaron y adecuaron posteriormente a las nuevas salidas y necesidades del sistema.

2) Desarrollo del módulo para el traslado de la información de la base de datos de Inventarios de la Burroughs B6800 hacia la Britton-Lee, constituido por las siguientes fases:

- Se investigaron las cintas magnéticas ubicadas en el IIMAS (Instituto de Investigación en Matemáticas Aplicadas y en Sistemas) y se recuperaron las que contenían la información referente a los inventarios ya elaborados.
- La información contenida en estas cintas se bajó al disco para ser estudiada y trabajada. Esta información estaba compuesta por tres tipos de archivos: los archivos que contenían la información de los números asignados a los volúmenes en los procesos de alta en el banco de datos de LIBRUNAM,

los archivos creados por los usuarios de LIBRUNAM al hacer los cargos de los libros encontrados durante sus búsquedas en línea y los archivos de las dependencias inventariadas en sus distintas fases.

Los archivos con las altas y los cargos comprendían la información procesada desde el inicio del sistema LIBRUNAM en el año 1979 hasta la fecha en que se llevó a cabo este procedimiento (se llama cargo al proceso de incrementar un registro con los datos que identifican al libro como único dentro de la UNAM en el archivo donde se localiza el acervo de cada biblioteca.)

- Esta información se revisó, se recopiló y se procesó a través de programas creados en la B6800 de acuerdo a los distintos formatos que fue presentando, mismos que a su vez la validaron y la alojaron dentro de archivos con formatos uniformizados para su transportabilidad a la IDM. Estos archivos posteriormente fueron grabados en cintas magnéticas.
- Estas cintas se trasladaron a la Dirección General de Bibliotecas, donde la información que contenían se bajó al disco de la Alpha-Micro. Esta información se almacenó en archivos que luego fueron procesados para, por último, darse de alta en el nuevo diseño de la base de datos previamente creada en IDM.
- Dentro del diseño de la nueva base de datos, se creó una relación denominada "inventario", para la cual la llave de acceso es la combinación de la dependencia y el número de adquisición (número único asignado en forma consecutiva a cada volumen adquirido por una biblioteca específica), ya que esta unión debe ser única para cada libro en el sistema bibliotecario de la UNAM además de que no permite que se den equivocaciones en la asignación de este número al momento de dar de alta la información en la base. Otra de las llaves de acceso a la relación es el número de matriz, que es otro de los campos del registro de inventario, la cual se constituye como llave foránea y conecta esta relación con la que contiene los datos completos de cada ficha, por lo que esta llave permite detectar fácilmente y cuando se necesiten los demás datos contenidos en la ficha bibliográfica.
- Se hizo una doble protección en cintas magnéticas de cada archivo creado en Alpha-Micro, (cabe

mencionar que estas cintas son salvaguardadas por la Subdirección de Informática de la Dirección General de Bibliotecas.)

### **Objetivos del nuevo Sistema de Inventarios**

Los objetivos del desarrollo de este nuevo sistema fueron en gran parte los mismos que se tenían desde el inicio del sistema automatizado de INVENTARIOS, pero aplicados a los nuevos recursos y con el propósito de dar un mejor y más eficiente servicio a los usuarios de las bibliotecas de la UNAM.

Las aplicaciones que se buscaban obtener con el manejo de este sistema son las siguientes:

- Conocer el material bibliográfico con que cuenta cada biblioteca de la UNAM.
- Conocer en cual o cuales de las bibliotecas de la UNAM se encuentra disponible un determinado libro.
- Tener herramientas para hacer inventarios periódicos en las bibliotecas y así controlar el patrimonio universitario y apoyar las rutinas de selección y adquisición de materiales.
- Ofrecer catálogos locales o UNION<sup>1</sup> a todas las bibliotecas del sistema.
- Promover la cooperación entre bibliotecas y fomentar el préstamo interbibliotecario al poder duplicar catálogos locales de bibliotecas o catálogos UNION de la UNAM.
- Ofrecer información sobre las colecciones de cada biblioteca para normar criterios de adquisición y descarte y/o convenios de cooperación entre bibliotecas con colecciones similares y/o complementarias.
- Crear la infraestructura de cómputo para implantar:
  - Sistemas locales de circulación automatizada (para lo cual es forzoso que previamente sea inventariado el acervo, ya en que caso contrario se corre el peligro de perder el control.)

1. Catálogos UNION se refiere a un catálogo que contenga todo el acervo localizado en las bibliotecas de la UNAM.

• Sistemas globales automatizados de préstamo interbibliotecario (aún no son una realidad.)

• Bibliotecas con estantería abierta al público (que igualmente, sin el inventario se tiene el riesgo de perder el control.)

- Utilizar los recursos modernos para agilizar cualesquiera de las fases del proceso de inventarios o las rutinas de mantenimiento, empleando por mencionar como ejemplo, las etiquetas ópticas.

#### **Situación actual del proyecto de Inventarios**

De las bibliotecas que han sido inventariadas o que se encuentran en alguna fase del mismo, se han detectado variedad de problemas de entre los que se destacan los siguientes:

1) escasez o ausencia de personal de apoyo para el levantamiento físico del inventario en la bibliotecas departamentales;

2) aspectos de orden laboral;

3) errores ocasionados por la falta de capacitación del personal;

4) falta de actualización o carencia de catálogos topográficos;

5) alto porcentaje de material clasificado en sistema Dewey y no en formato Marc/DGB;

6) importante porcentaje de material mal clasificado;

7) donaciones valiosas sin clasificar;

8) limitación en el tiempo en el que se lleva a cabo el levantamiento físico del inventario ya que se busca no afectar al préstamo de los libros (por lo general, se procura aprovechar los tiempos vacacionales para hacerlo y así afectar menos a los usuarios);

9) suspensión de los procesos de inventarios por cambios de administración o de políticas en las bibliotecas;

entre otros.



Por otra parte, en la Dirección General de Bibliotecas se adolece de la infraestructura adecuada desde el punto de vista de recursos tanto humanos como materiales para afrontar de manera integral el proyecto de inventarios; ya que a pesar de los esfuerzos para allegarse de estos recursos, de antemano se sabe que es imposible contar con todo lo que se requiere para llevar a cabo el proyecto en más de 170 bibliotecas y casi 3,370,000 volúmenes; por lo que se ha convenido que parte de los procesos involucrados, en la mayoría de los casos y en forma descentralizada, sean las bibliotecas con sus propios medios las que los efectúen, con una previa orientación o capacitación por parte de la DGB y sin que ella deje de tener el control de estos procesos.

Actualmente se siguen fijando nuevos límites dentro del sistema, incluso ya se está haciendo el estudio para el transporte del mismo a una nueva plataforma de cómputo; siempre respetando la filosofía original, pero además tratando de resolver los problemas que presentan las necesidades derivadas de la dinámica actual de las bibliotecas y de sus usuarios, y buscando también aprovechar las facilidades y los beneficios que proporcionan las nuevas alternativas de equipo de cómputo que constantemente están evolucionando, por lo que pueden adecuarse más a las propuestas de solución con el menor costo posible. En la definición de estos límites, se considera que no sólo el contenido y formato de las salidas es importante, sino de igual modo, los tiempos de entrega de las mismas y la frecuencia de actualización y acceso de la información procesada.

#### **3.4.2 Descripción del proceso de registro de los libros (generalidades)**

Para poder entender el sistema de Inventarios, se necesita primero comprender el de LIBRUNAM (en forma general), y para lograrlo es necesario mirar brevemente sobre los procesos que se controlan en forma global en la Dirección General de Bibliotecas, por lo que en esta sección se da una resumida explicación de ellos.

En general las bibliotecas están organizadas en:  
[58]

- 1) Servicios técnicos
- 2) Servicios al público

- 1) Los servicios técnicos tienen la responsabilidad de adquirir, organizar, controlar y mantener el material bibliográfico; es decir, se encargan de todos los procesos que hacen que el material esté más accesible al usuario. Estos procesos comprenden: selección de este material, adquisición, catalogación, clasificación, generación de encabezamientos de materia, elaboración de índices y resúmenes, y por último, colocación del mismo en los estantes.
- 2) Los servicios al público son los que permiten al usuario localizar el material con la mayor precisión y el menor esfuerzo a través de un sistema adecuado de información.

#### **Descripción breve de los procesos que involucran los servicios técnicos**

Como ya se dijo, el registro, el pago y los procesos técnicos de los libros que son adquiridos por las bibliotecas del sistema bibliotecario de la UNAM se realizan en forma centralizada en la Dirección General de Bibliotecas por personal académico especializado.

A continuación se da una breve explicación de todas las etapas que comprenden el proceso de los libros:  
[59]

#### **Adquisición**

La selección de los libros se realiza en cada biblioteca departamental; la solicitud al proveedor del material deseado, se lleva a cabo indistintamente por la biblioteca o por la DGB; y el registro de propiedad y trámite de pago ante las autoridades administrativas universitarias lo efectúa la DGB.

#### **Proceso técnico**

La biblioteca departamental una vez que ya ha recibido del proveedor el material solicitado, presenta a la DGB la factura de compra y fotocopias de las portadas de los libros y de su contenido para que sean procesados dentro del sistema LIBRUNAM.

Las fases que comprenden el proceso técnico de cada libro son:

1) *Localización de la obra en el catálogo oficial de la UNAM* - Es la operación de investigar en el sistema LIBRUNAM si la obra ya está registrada dentro del banco de datos del acervo bibliográfico.

En esta parte se pueden presentar dos opciones:

i. Los libros que al hacer la investigación en el banco de datos no se encuentren, significa que son libros que ingresan por vez primera al sistema LIBRUNAM y por consiguiente no existe un título igual registrado en todo el sistema bibliotecario de la UNAM.

A estas obras de nuevo ingreso se les hace el proceso completo del libro previo a su registro en el banco, para lo que se remiten a la sección de catalogación (fase 2)

ii. Los libros que, al hacer la investigación se detectan registrados en el banco de datos se manejan como ejemplares duplicados, aun cuando se trate de un título nuevo para la biblioteca que está haciendo la adquisición, y en ese caso deben quedar registrados los datos solamente en el inventario de esta biblioteca.

2) *Catalogación y clasificación del material de nueva adquisición* - A los libros de nueva adquisición en el sistema de bibliotecas de la UNAM, se les efectúa todo el proceso como ya se dijo; esto implica que se remiten a la sección de clasificación, en donde se hace la búsqueda del título en diferentes fuentes catalográficas: nacionales y de la Biblioteca del Congreso de EUA. Si la obra se encuentra en estas fuentes, el material localizado únicamente se adapta a las necesidades de la UNAM; y en caso contrario, es necesario hacer la clasificación completa en base a reglas y estructuras establecidas internacionalmente.

3) *Distribución del material a las bibliotecas departamentales* - En esta etapa se imprime toda la papelería generada del proceso de los libros como son las tarjetas para los catálogos de los libros (para las bibliotecas que aún las requieren) y sus remisiones; y posteriormente son entregadas a las bibliotecas propietarias.

4) *Actualización de inventarios* - Cada nuevo libro adquirido por cualquiera de las bibliotecas de la UNAM se registra en los inventarios donde se controla qué biblioteca lo tiene y cuántas copias posee; no importando si se trata de un título nuevo o no. Esto se registra en el banco durante la fase 2) si se trata de un título nuevo o durante la 1) si se trata de una obra ya existente en el banco.

5) *Actualización de catálogos* - Cada biblioteca se encarga de insertar en sus catálogos las tarjetas generadas

por la DGB; en las que aún lo mantienen, o en otro caso, hace la solicitud a la DGB para que se le haga la actualización en su Catálogo Electrónico.

Se hace notar que cuando se mantiene un catálogo en tarjetas, es necesario disponerlas en un orden alfabético previo a su inserción en el catálogo de la biblioteca.

A continuación se muestra un diagrama del proceso de registro de los libros.

**PROCESOS QUE SE CONTROLAN EN LA DIRECCION GENERAL DE BIBLIOTECAS**

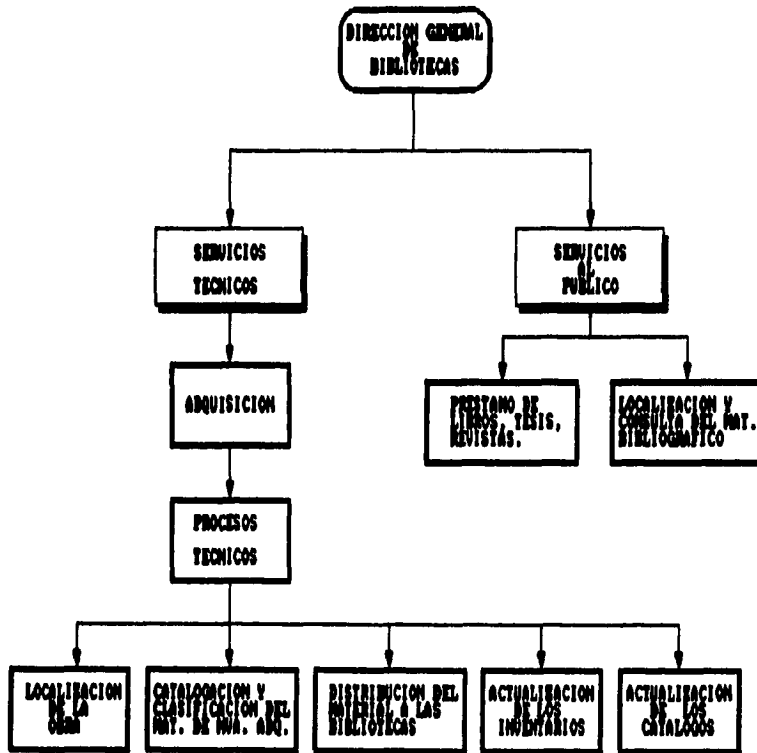


FIGURA 3.2

### **Características generales de la información**

Los datos que permiten reconocer un libro y localizarlo físicamente en los anaqueles de una biblioteca están contenidos en las fichas catalográficas.

Estas fichas se encuentran o en los catálogos en forma de tarjetas de cartón, o bien, en forma electrónica instalados en Computadoras Personales, de manera individual en algunos casos o en una red de computadoras conectadas vía un servidor (Server) en otros. También se pueden encontrar las tarjetas almacenadas en microfichas en aquellas bibliotecas que no cuentan con computadoras personales.

Las fichas catalográficas de la UNAM se construyen por personal profesional especializado en esta área en base a reglas y estructuras establecidas internacionales, pero adaptadas a las propias necesidades.

En el desarrollo del sistema LIBRUNAM, para construir las fichas catalográficas del banco de datos y saber con exactitud la información que debían contener se analizaron las reglas de catalogación descriptiva (I y II), las normas para la construcción de los temas, las especificaciones al registrar los autores, los títulos, las notas de serie y la estructura de la clasificación oficial de la UNAM. (Ver la sección *clasificación y codificación de la información* más adelante.) [60]

En general se presentan tres partes distintivas en una ficha: i) Asiento principal - donde se encuentra el autor de la obra, o en su ausencia, un título. ii) El cuerpo de la ficha - que está siempre en la parte superior, y comprende la descripción de la obra. iii) El registro de la ficha - que comprende los temas que abarca la obra, los coautores, etc. . [61]

Esta información es completamente variable, en su estructura, longitud y frecuencia (pueden presentarse fichas formadas desde tres hasta más de treinta elementos diferentes), razón por la cual se tuvo que estudiar con cuidado el diseño para poder desarrollar la base relacional "libros" que maneja el sistema LIBRUNAM.

### **Proceso automatizado de los libros**

En esta sección, se describe en forma general la automatización de los procesos concernientes a los servicios técnicos que acabamos de describir, dejando en una sección aparte los procesos involucrados con el registro de los inventarios.

El banco de datos LIBRUNAM se actualiza con la producción diaria por lo que se pone especial cuidado en salvaguardar la información procesada ya que todos los módulos que conforman el sistema dependen de ello, y todos los procesos que se llevan a cabo con él requieren, necesariamente, que esta información esté actualizada para su confiabilidad.

Los pasos que se siguen en el proceso automatizado de un libro son los siguientes:

1) **Adquisición** .- En esta etapa se efectúa el registro de las solicitudes de compra de libros en la base de datos "adquisicion". Es también en esta etapa donde se llevan a cabo todas las rutinas que involucran el control de las partidas presupuestales de las bibliotecas en la adquisición de su material bibliográfico (asignación presupuestal, movimientos contables, registro de facturas, control de proveedores, generación de todas las salidas para pagos y estados de cuenta, etc.), además de que ofrece información tanto en línea como en listados impresos de todos estos procesos.

2) **Localización en línea** .- Para realizar la localización de las obras dentro del sistema LIBRUNAM se utiliza la recuperación en línea de la información del banco de datos. Los localizadores investigan una obra a partir de palabras significativas dentro del texto, en forma fonética (recuperación libre), o por asientos completos (recuperación estricta.) Esto incluye hacer la recuperación de una ficha catalográfica en base a cualquiera de las llaves de acceso (o combinación de ellas) definidas en la base "libros" y que están asentadas en los principales campos que conforman la ficha (autor, título, tema, número de matriz, serie, ISBN, editorial, etc..) Mientras más se afine la recuperación con estas llaves más rápido se hace la intersección de los datos y consecuentemente más rápido se obtiene la información buscada. A partir de estas localizaciones se decide, como ya se dijo, si el libro requiere de un proceso completo por ser un título nuevo para el banco de datos, y entonces se remite a la sección de clasificación, o si se trata de un cargo al inventario de una biblioteca específica por tratarse de un título que ya existe dentro del banco. Si este último es el caso, automáticamente la máquina asigna al libro localizado un número consecutivo de volumen dentro del acervo de la

biblioteca (número de adquisición.) Este proceso se registra en la base de datos dentro de la relación "inventario" y se guarda en archivos especiales para la generación posterior de juegos de tarjetas y de las remisiones que las amparan.

3) *Clasificación y codificación de la información* .- La **clasificación** de la obra se lleva a cabo, como ya se mencionó, en base a distintas fuentes catalográficas (incluyendo LIBRUNAM), y adaptándola según los cambios necesarios. Esta clasificación se hace en hojas de codificación especialmente diseñadas para el sistema LIBRUNAM.

En la **codificación** se adoptaron las reglas del "Formato MARC II" para monografías (Machine Readable Cataloging), que es un formato internacional especialmente diseñado por la Biblioteca del Congreso de Estados Unidos para material bibliográfico, y que está regido por las Reglas Angloamericanas de Catalogación para el intercambio de información bibliográfica legible e identificable por computadora, y que además prevé todas las opciones que presenta este tipo de información, sin limitaciones para su registro. [61]

Dicho formato asocia un prefijo numérico denominado etiqueta, a cada elemento o campo de una ficha, y separa los subcomponentes de cada elemento con un símbolo preestablecido.

Para facilitar las rutinas de codificación, captura y procesamiento de la información se seleccionaron las etiquetas del formato MARC II que se consideraron más adecuadas para este fin, estableciéndose entonces el formato MARC/DGB para la codificación del material bibliográfico de la UNAM; se diseñó además un sistema de símbolos simplificados que se adhieren a la información fuente e identifican cada campo de la misma. [61]

4) *Captura, validación y procesamiento de la información* .- La captura de la información puede ser hecha en cualquier equipo, siempre respetando la codificación y el tamaño del registro del archivo de datos que se genera; en la DGB se programaron rutinas para capturar esta información desde las terminales de una de las computadoras anfitrionas que accesan la IDM, además de que estas rutinas también sirven para orientar al capturista.

Los datos son validados en forma estricta por programas alojados en una de las computadoras anfitrionas de la IDM para garantizar la calidad del banco y que se encargan de: normalizar la información a las características requeridas, revisar sintácticamente cada campo de datos, y generar

archivos y listados de error para su posterior corrección y reproceso.

Los datos se procesan y se almacenan en nodos de longitud variable dentro de la base "libros". IDM se encarga de organizar la estructura de la información en forma de árboles, siendo éste un proceso transparente para los usuarios del sistema.

El sistema permite revisar los datos visualmente (a través de los listados de error o a través de diskettes con la información procesada previa a su inserción al banco de datos) para su corrección antes de ser dados de alta en la base de datos "libros".

El registro de inventario de cada ejemplar se hace, como ya se ha mencionado, en una relación llamada "inventario" de la base "libros" en nodos de longitud fija y aquí, al igual que en el cargo a un volumen de título duplicado, se le asigna un número que es consecutivo dentro de cada biblioteca, a cada ejemplar que se da de alta en esta relación.

5) *Generación de papelería y distribución* .- A partir del procesamiento de la información contenida en las formas codificadas, se utilizan programas para la generación e impresión de los juegos de tarjetas para los catálogos de las bibliotecas. Cabe mencionar que estas tarjetas son entregadas en orden topográfico y listas para su inserción en los catálogos de las bibliotecas. Cuando se trata de obras ya registradas en el banco el sistema también genera los duplicados de tarjetas necesarios.

Estas tarjetas siempre se acompañan de sus remisiones que las respaldan (listados que contienen la información procesada y concentrada por dependencia bibliotecaria.)

LIBRUNAM también permite la generación de diferentes listados de la información que contiene, como lo son: los de índices (por autor, título, tema, No. de adquisición, etc.); listas de nuevas adquisiciones, etc..

Por último, el sistema permite además, obtener estadísticas de producción para apoyar al mejoramiento de su administración.

A continuación, se muestra un diagrama de los procesos involucrados en el sistema LIBRUNAM, que ilustra lo que se acaba de describir.



## PROCESO AUTOMATIZADO DE LOS LIBROS

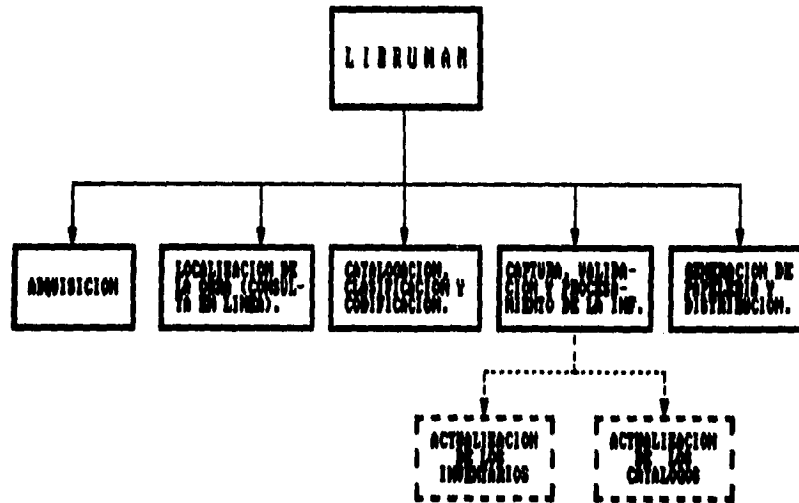


FIGURA 3.3

### 3.4.3 Descripción del proceso de registro de Inventarios

El objetivo de esta sección es el describir los procesos que se llevan a cabo para el levantamiento y control de los inventarios en las bibliotecas de la UNAM.

En general el proceso de registrar inventarios se divide en tres formas de llevarlo a cabo, que unidas permiten el control integral del acervo de una biblioteca dada. Estos tipos de inventario son:

- a) Registro del acervo existente (registro retrospectivo.)
- b) Registro de las nuevas adquisiciones (registro actual.)
- c) Cotejo periódico de inventarios y actualización del mismo (registro periódico.)

En cualquiera de estos casos, los elementos básicos que se requieren para la actualización del acervo de cada biblioteca alojado en el banco de datos LIBRUNAM son los campos llave del número de adquisición y del número de matriz, mismos que nos relacionan a todos los demás datos que se requieren para dar información sobre un libro y su registro en la base de datos; por lo que todos los procesos involucrados en el inventario, finalmente buscan la consecución de estos dos elementos para establecer el control.

A la postre, en casi todas las circunstancias, el objetivo no sólo es uniformizar el control en combinación con la DGB, sino además se espera el apoyo de ésta para automatizar el préstamo de los libros, y/o controlarlos automáticamente con un catálogo electrónico local.

Por otro lado, para ofrecer un mejor servicio a los usuarios, se vuelve necesario el abrir la estantería al público, y si no se tiene inventariado el acervo, es probable que llegue el momento en que se pierda el control del mismo.

A continuación describiremos cada una de estos casos, los cuales se ilustran en la siguiente figura:

#### PROCESO DE REGISTRO DE INVENTARIOS

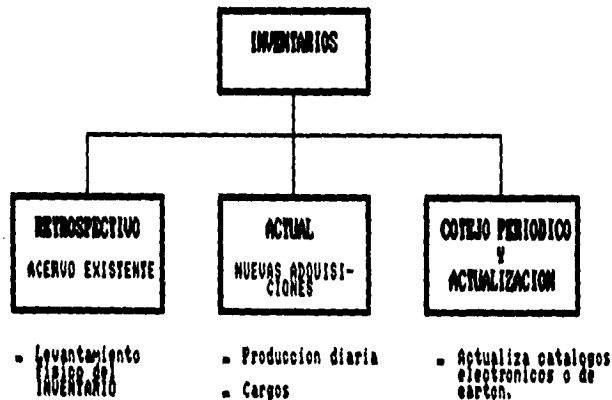


FIGURA 3.4

### ■ Retrospectivo

En este tipo de inventario es donde se registra el acervo existente de la biblioteca que lo solicite a la DGB.

El registro retrospectivo del acervo existente en el sistema bibliotecario, requiere de un levantamiento físico del inventario de cada una de las bibliotecas de la UNAM y de su inclusión en el banco de datos.

Este proceso implica en forma general:

- Preparación de documentos.
- Cotejo físico del acervo contra los documentos en los casos en que ya exista información registrada en el banco de datos.
- Proceso masivo de búsqueda en línea y/o captura para el registro de inventario por volumen.
- Corrección de la información que se está procesando (altas, bajas, cambios.)
- Impresión de etiquetas.
- Corrección de errores en base a las etiquetas generadas (altas, bajas, cambios.)
- Impresión de nuevos catálogos.

Los procesos involucrados en este tipo de inventario pueden ser considerados como la parte medular de los inventarios de las bibliotecas, ya que es aquí donde se sientan las bases para el control real del acervo de cada una de ellas, además de dejar todo preparado para facilitar los consecutivos levantamientos periódicos de sus inventarios ( cotejo periódico .)

Las etapas que se siguen en esta forma de inventario son los siguientes:

1. El primer paso que se presenta es la solicitud de apoyo a la DGB por parte de la biblioteca que desea hacer su inventario.

2. El siguiente paso es el diagnosticar la situación de la biblioteca. En esta etapa:

1) se determina el tipo de catálogo con que cuenta la biblioteca,

ii) se da una estimación aproximada del número de volúmenes que están alojados en la biblioteca,

iii) se conoce al personal de apoyo con que cuenta la biblioteca y se les da una rápida capacitación,

iv) se elabora el calendario de trabajo

En base a este diagnóstico se determina si el procedimiento va a ser el que usualmente se sigue para todas las bibliotecas, o se va a utilizar algún otro recurso, o se necesita elaborar programación especial que lo apoye; ya que la situación en muchas de las bibliotecas del sistema universitario varía considerablemente de las restantes por diversos factores (administrativos, directivos, políticos, situaciones especiales como lo son por ejemplo, bibliotecas que se forman de colecciones de otras; bibliotecas cuyo presupuesto está incluido dentro de otras partidas presupuestales, bibliotecas que se subdividen en otras, etc.) y por consiguiente quedan afectadas en los controles que deben de tener de su acervo.

También en base a este diagnóstico se determinan los tiempos de procesos, de elaboración del material de apoyo, los recursos de máquina que se van a requerir, la papelería necesaria y sus costos (por lo general, parte del gasto lo absorbe la biblioteca departamental y el resto la DGB) y por último se da una estimación de la fecha de término del mismo.

3. El siguiente paso es la elaboración por parte de la DGB, de los listados de la información que tiene registrada la biblioteca en su inventario dentro del banco de datos.

Estos listados se generan por cuadruplicado, cada uno en distinto orden, de forma que faciliten el trabajo al personal que está llevando a cabo el inventario.

Los elementos incluidos en estos listados y que proporcionan información sobre cada libro son los siguientes: la clasificación de la obra, el autor, el título, el número de matriz que corresponde al título y el número de adquisición que corresponde a ese libro dentro de la biblioteca. En estos listados cada renglón corresponde a la información de un solo volumen, para que el que investiga los libros pueda registrar claramente la situación de cada uno de ellos.

4. En esta etapa la biblioteca comienza a trabajar lo que se conoce como el levantamiento del inventario físico; es decir, lo que corresponde a la confrontación física de los listados recibidos y repartidos al personal que va a llevar a cabo la investigación con los libros en la estantería.

Es imprescindible asegurarse antes de comenzar, que la colección esté ordenada topográficamente (es decir, ordenada en base a la clasificación de los libros, que es la que define su lugar en la estantería); y también, es importante señalar que se considera necesario en esta etapa el cerrar el acceso al préstamo de los libros, ya que en caso contrario se puede perder el control, y si esto sucede, los que lo llevan a cabo, no podrán detectar si el libro que buscan está mal colocado en estantería, o prestado, o en proceso menor, o si está en restauración, etc., o si definitivamente está perdido y entonces hay que darlo de baja.

En esta etapa, los bibliotecarios que están efectuando la búsqueda hacen la revisión volumen por volumen y al detectar un error (como por ejemplo libros mal clasificados, libros con el núm. de adquisición duplicado, etc.) deciden la estrategia a seguir para su corrección. Siempre se van a basar en la consecución del número de adquisición que corresponda en cada caso particular; es decir, su objetivo va a ser el obtener los datos básicos del libro (número de adquisición y número de matriz) para su corrección posterior en el banco de datos (alta, baja o cambio.)

Aquí se hace destacar que el número de adquisición dentro de la base de datos está definido como único para cada dependencia, por lo que no se permiten duplicados en una biblioteca dada, y al presentarse uno, IDM lo elimina o no permite su alta en el banco, ya que por medio de éste se lleva el control de cada libro además de ser la llave de acceso a la información de la biblioteca.

La información (número de adquisición y número de matriz) que corresponde a libros con error, o a libros que falten o que estén de sobra, deberá ser vaciada en archivos tanto para las altas como para las bajas, según sea el caso. Los cambios se manejan como una baja en la base de datos de la información incorrecta y un alta de la correcta.

Cuando no se cuenta con el número de adquisición (por no contenerlo el libro o por estar duplicado en otro volumen), se procede a capturar la información en un archivo donde se registra el número de matriz y el total de volúmenes que para cada título (matriz) se encuentren en esta situación; y al ser procesados, el sistema otorgará en forma automática un número consecutivo a cada volumen de cada matriz a partir del último que se tiene guardado en el banco de datos de esa dependencia.

Cuando a partir de los listados no se puede conocer el número de matriz que corresponde a un título, entonces se separa este libro para que al término del cotejo físico de listados contra libros se pueda obtener su información a través de una búsqueda en línea en la base de datos LIBRUNAM

o a través de los discos compactos que contienen el banco de datos LIBRUNAM.

5. En esta etapa ya se tiene toda la información lista para dar de alta y para dar de baja; aunque las bajas solamente corresponden a los casos en que se trate de cambios.

Una vez que la DGB tiene los archivos (capturados en disquette o directamente) ya transportados o alojados en la máquina anfitriona de IDM, se comienzan los procesos de baja y de alta en la base de datos LIBRUNAM, específicamente en la relación "inventarios" (ver Capítulo 4.)

Primero se procesan las bajas (que en realidad en esta etapa corresponden a un cambio que se va a efectuar posteriormente), ya que si se desea hacer un alta en un número en el que no se le ha efectuado previamente la baja, el sistema lo rechazará como error. Aquí el sistema revisa que efectivamente ese número esté cargado en el inventario de la dependencia, que sea la matriz que le corresponde, y que no esté marcado por un previo proceso de inventario; en cuyo caso lleva a cabo las bajas, guardándolas en una relación temporal como respaldo, y generando listados de esta información, del tipo de los listados que se elaboraron para el cotejo físico. Si se presentan errores, el sistema produce una serie de listados detectando y definiendo estos errores.

Si se presenta información donde efectivamente se desea hacer la baja y ya estaba marcada como inventariada, se pide una verificación y se vuelve a procesar ya en forma definitiva.

Después se procesan las altas, en las que al igual que en las bajas, el sistema verifica la información. Aquí, el sistema revisa que el número que se desea dar de alta no esté ya cargado en el inventario de la biblioteca. De estas altas, se elaboran listados del tipo de los que se generan para el cotejo físico. Si se presentan errores, igualmente el sistema produce una serie de listados indicándolos y definiéndolos. Los más comunes son los de querer cargar un número que ya estaba cargado a otra matriz, y en ese caso se elabora un listado donde se vacía la información de la matriz que tiene asignada dentro del banco ese número de adquisición, y la información de la matriz a la que se le quiere dar de alta el mismo; esto con el fin de que el bibliotecario pueda hacer la investigación en la estantería de estos dos volúmenes para definir si físicamente no está duplicado el número de adquisición a obras diferentes, esto es, que sea un error no detectado durante el cotejo físico, y pueda corregirlo.

De todas estas altas se generan etiquetas con código de barra.

Quando se presentan repeticiones, es decir, se carga al mismo número que ya está cargado en el banco, el sistema sólo genera etiquetas.

6. El siguiente paso es trabajar todos aquellos libros cuyos títulos no se encontraron en el banco de datos. En este caso, lo que se tiene que aplicar es el proceso original que se lleva a cabo con cada título nuevo que ingresa al sistema LIBRUNAM (referirse a la sección 3.4.2 descripción del proceso de registro de los libros.) Aquí, tanto el número de matriz como el número de adquisición de cada libro los asigna automáticamente el sistema.

7. El siguiente paso, consiste en la generación por parte de la DGB de las etiquetas con los códigos de barra de los volúmenes inventariados. Estas etiquetas se entregan en orden topográfico, a fin de facilitar el pegarlas a los libros colocados en la estantería. La información que contiene cada etiqueta es la clasificación, y el número de adquisición con su correspondiente código de barra.

En los casos en que se detecten errores a partir de las etiquetas se procede a efectuar las bajas o los cambios pertinentes; al igual que las altas si llegasen a faltar etiquetas para libros.

Quando sólo se desean repetir etiquetas, debido a pérdidas o mala impresión, se captura la información con el formato antes mencionado (núm. de adq., núm. de matriz) y al procesarse, la máquina revisa que coincida con la información almacenada en el banco para volver a generar estas etiquetas.

Es importante hacer notar que los errores que emanan de la generación de las etiquetas siempre van acompañados de listados donde se indican y describen estos errores ya que las etiquetas son un fuerte apoyo para el préstamo automatizado de los libros a través de los lectores ópticos, que en cada lectura del código de barra permiten ahorrar el tiempo de digitación de números, se evitan errores manuales, y se agiliza el servicio de préstamo.

8. En esta etapa, se siguen revisando las etiquetas generadas y corrigiendo errores, hasta estar seguros de que ya se tiene todo en orden; y finalmente y si la biblioteca lo solicita, se elabora un último listado con todo el acervo que tiene almacenado en el banco, para volver a revisarlo contra los libros en la estantería.

9. En este punto ya se sabe con certeza cuál es la información que está de sobra en el banco de datos, por lo que se procede a darla de baja en igual forma que se hace para la baja que corresponde a cambios, es decir, se captura

la información que se va a eliminar, el sistema la procesa y revisa que efectivamente cada uno de estos números estén cargados en el inventario de la dependencia, y que la matriz que tiene asignada sea la matriz que le corresponde dentro del banco de datos, además, revisa que cada uno de estos números no estén marcados por un previo proceso de inventario; en cuyo caso lleva a cabo la baja. El sistema guarda todas las bajas que va efectuando en una relación temporal como respaldo, y elabora listados de esta información, del tipo de los listados que se generaron para el cotejo físico. Si se presentan errores, el sistema produce una serie de listados indicando y definiendo estos errores.

10. Y el último paso para terminar con este tipo de inventario se presenta cuando ya hechas todas las revisiones y correcciones, se le generan e imprimen (o graban) a la biblioteca sus catálogos. Los catálogos se elaboran en cartón cuando la biblioteca los utiliza; y se entregan por quintuplicado, para ingresar a los catálogos: topográfico, de autores, de títulos, de temas y por último de números de adquisición (los cuatro primeros son para el servicio de los usuarios y el último para uso exclusivo de la administración de la biblioteca para apoyar en el control del acervo.

Si se utiliza catálogo electrónico, la información se graba en disquettes y luego se transporta a la base local alojada en las computadoras personales de la biblioteca.

#### ■ Actual

En este tipo de inventario es donde se lleva a cabo la consignación de las nuevas adquisiciones de cada una de las bibliotecas de la UNAM.

El registro actual del acervo existente en el sistema bibliotecario, es el proceso de actualizar en el banco de datos la información que tiene cada biblioteca con los volúmenes de nueva adquisición de pertenencia de las obras investigadas. Esto se lleva a cabo con los procesos de producción diaria y con los cargos que emiten en línea los localizadores de la DGB al hacer la búsqueda de estas obras en el banco de datos LIBRUNAM; procesos en los cuales la designación del número de adquisición es controlada por el sistema, mismo que confiere un número consecutivo a cada libro comprado por cada biblioteca a partir del último número que tiene almacenado en el banco para cada biblioteca en particular.

Es entonces en estos procesos donde se va registrando el acervo de nueva adquisición dentro del inventario de cada una de las bibliotecas de la UNAM.



Se recuerda que cada número que se da de alta en el inventario del acervo de cada dependencia es único, y que el sistema evita toda duplicidad con un estricto control.

#### **■ Cotejo periódico de inventarios**

En este tipo de inventario es donde se registran periódicamente los cambios que ha sufrido el acervo de la biblioteca a fin de actualizar los inventarios que hayan sido efectuados con anterioridad.

Es el proceso de cotejar físicamente el acervo de la biblioteca con el inventario registrado en el banco, para la actualización de la información almacenada; con el fin de poder actualizar el catálogo electrónico de la biblioteca y/o su catálogo de cartón, y consiguientemente el sistema local de préstamo de la biblioteca departamental, además de llevar un control real sobre el acervo que contiene.

Esta forma de inventario, en realidad representa la conclusión de las dos anteriormente descritas (registros retrospectivo y actual), ya que no se puede llevar un cotejo periódico si no está asentado un inventario retrospectivo, y si no se tiene el control del inventario actual.

En este inventario, son de gran ayuda las etiquetas, ya que ellas facilitan la lectura de la información que se está trabajando.

En realidad, a la fecha se tiene muy poca experiencia en este tipo de inventario, ya que la principal preocupación de las dependencias bibliotecarias es el matener, más o menos, su primer inventario (retrospectivo) y el inventario actual para poder llevar el control sobre su sistema de préstamo de libros; y esto siempre va a ser conforme a los requerimientos de los usuarios, los cuales no siempre van a cubrir todo el acervo.

Esto implica además que la biblioteca no tiene un control real sobre su acervo ya que el inventario es la única forma confiable de registrar los libros que van siendo dados de baja (por robo, mutilación, desuso, etc..)

Por otro lado, si la biblioteca no lleva a cabo periódicamente este tipo de inventario, la DGB pierde también el control del acervo de la biblioteca, y por consecuencia no puede prestar un servicio real a la red de bibliotecas que lo requieran.

Finalmente diremos que, los resultados obtenidos con el levantamiento de cualquier tipo de inventario propicia que se den muchos de los beneficios que pueden otorgar las

bibliotecas de la UNAM a sus usuarios, y con creces justifica el costo, el tiempo y el gran esfuerzo invertidos en él.

Para concluir el capítulo, a continuación se describe el equipo de cómputo utilizado para el desarrollo y manejo del sistema.

### **3.5 HERRAMIENTAS**

La Dirección General de Bibliotecas cuenta con un equipo AM-2000-10, con un equipo AM-1062 de Alpha Microsystems y con un sistema manejador de bases de datos Britton Lee, para desarrollar, administrar y controlar los sistemas automatizados que se generan para apoyar en sus labores a las bibliotecas de la Universidad Nacional Autónoma de México.

A continuación se describen estos equipos:

#### **3.5.1 BRITTON LEE**

##### **Intelligent Database Machine (IDM)**

La (IDM) Máquina Inteligente Manejadora de Bases de Datos Britton-Lee (Britton-Lee Intelligent Database Machine), es un sistema de propósito específico para trabajar con bases de datos relacionales.

Está compuesta por una máquina manejadora de bases de datos a nivel hardware; esto es, el hardware que emplea está especialmente diseñado para ejecutar funciones de bases de datos relacionales.

IDM ofrece un desarrollo más allá del que puede ser llevado a cabo con software convencional de paquetes manejadores de bases de datos, aunque actualmente, ya existen algunos que se están desarrollando en esa línea.

##### **Hardware**

El hardware de la IDM está constituido por lo siguiente:

- Procesador de bases de datos.
- Controlador de memoria y tarjeta de reloj.
- Controlador de disco.
- Tarjetas de entrada y salida seriales.

- Tarjetas de entrada y salida paralelos.
- Tarjeta de memoria.
- Acelerador de discos y de procesos.

Esta máquina no puede ser accesada por sí sola por lo que necesita de una máquina anfitriona (host) para poder trabajar.

Se pueden tener una o varias computadoras anfitrionas (hosts) compartiendo la IDM (soportando hasta 8) o también una computadora anfitriona accediendo varias IDM.

Las máquinas anfitrionas que se utilizan para acceder la IDM en la Dirección General de Bibliotecas son dos y son de la marca Alpha-Micro (de las cuales hablaremos más adelante.)

La gran ventaja de la IDM consiste en que las bases de datos centralizadas están disponibles para toda la variedad de usuarios localizados en muchas máquinas anfitrionas; así, las computadoras anfitrionas son más productivas y confiables, además de ser independientes de las bases de datos.

IDM nos proporciona:

- Administración de bases de datos relacionales.
- Buena capacidad de acceso vía las terminales de la máquina anfitriona.
- Control de seguridad en los datos.
- Procesos de acceso y actualización.
- Manejo de transacciones.
- Seguimientos de auditoría.
- Servidor de archivos para propósito general.
- Facilidad de respaldo y recuperación de la información.
- Control de concurrencia (que la información no se repita; que sea real.)

La siguiente figura (Figura 3.5) muestra un diagrama que ilustra la configuración de una máquina IDM y sus vías de acceso y salida.

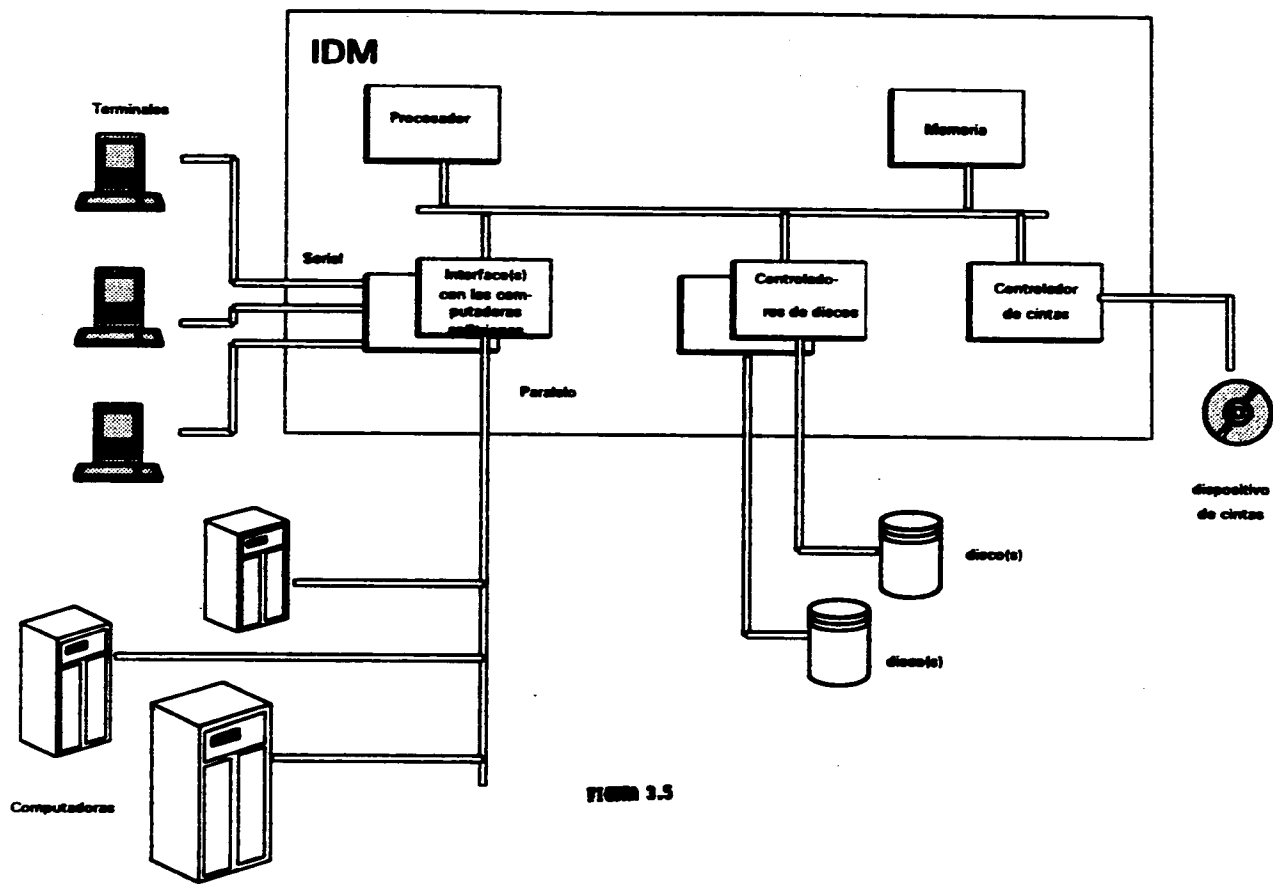


FIGURA 3.5

## **IDL**

El lenguaje que se utiliza para acceder la IDM es el IDL, Lenguaje Inteligente de Bases de Datos (Intelligent Database Language); es un lenguaje interactivo de comandos; es interactivo porque en el momento que se hace una pregunta se recibe la respuesta, no necesita de un compilador. Está basado en QUEL (Query Language) y es muy parecido a SQL.

Es un lenguaje de la cuarta generación (L4G), es decir, no procedimental; lo cual facilita al que lo utiliza la descripción de los resultados que se desean sin que se requiera que se detalle cómo obtenerlos.

### **Las bases de datos en IDM**

Los datos son almacenados en la IDM en bases de datos relacionales. En la IDM, una base de datos es un lugar físico para almacenar datos.

Los datos en un sistema de manejo de bases de datos relacionales está organizado en tablas (archivos) llamadas relaciones, los renglones (registros) son llamadas tuples y las columnas (campos) son los atributos.

Se puede decir que una base de datos en IDM es una colección integrada de relaciones, vistas y comandos almacenados (stored commands.) IDM tiene capacidad para manejar hasta 50 bases de datos en forma independiente.

En la IDM se requiere que las relaciones estén en la 1a. Forma Normal (1FN), esto significa que cada atributo de cada relación es atómico; es decir, en cada intersección de un renglón y una columna de la tabla siempre hay exactamente un valor, nunca un conjunto de valores.

Los tuples en la IDM pueden ser presentados al usuario en cualquier orden y cambiarlo en cualquier momento según las necesidades del mismo.

Un manejo completamente funcional de un sistema de bases de datos debe proveer protección, registro de transacciones, consistencia en los datos y rápido acceso a ellos.

**Seguridad.**- La persona que diseña y crea la base de datos es el administrador de la base de datos (ABD) y tiene ciertos privilegios además de encargarse de su seguridad. Los tipos de protección que controla son: leer, grabar, leer en cinta, grabar en cinta, ejecutar, crear y destruir bases de datos, relaciones, atributos, índices, comandos, etc. .

El ABD asigna a los usuarios los permisos para utilizar las bases de datos o los niega.

**Consistencia.**- Una transacción es un conjunto de uno o más comandos de IDM. Una base de datos consistente significa que si dos o más usuarios están operando simultáneamente los mismos datos, el resultado será como si sólo un usuario estuviera accediendo esos datos. Cualquier tupla que el usuario cambie no será accesible para otros usuarios hasta que la transacción termine.

**Registro de Transacciones.**- La IDM proporciona el Registro de las Transacciones para tres circunstancias: para el manejo de las transacciones, para la recuperación de la información en caso de daño físico y para la consolidación de todos los movimientos.

**Ejecución.**- La IDM provee un rápido acceso a los datos a través de los índices definidos por el usuario. Un índice es un directorio que contiene valores de datos con apuntadores a una localidad física de datos. Un índice es creado para facilitar el acceso a estos datos. IDM guarda los índices en forma de árbol-B. Un árbol-B es una estructura de datos que posee la propiedad de tener un acceso muy rápido a ellos; y este tipo de árboles cuenta con la ventaja de que una vez creados no requieren el mantenimiento periódico por parte del usuario.

Existen dos tipos de índices: ordenados (clustered index) y no ordenados (nonclustered index.) El índice ordenado en un atributo de una relación causa que la relación esté ordenada por ese atributo, y físicamente almacenada en disco en ese orden. Un índice no ordenado es aquél que es creado para un atributo (o grupo de atributos) para el cual los datos no están ordenados. Estas llaves pueden ser llaves concatenadas, compuestas por dos o más atributos manejados por IDM como una sola llave.

En cada relación sólo se permite crear un índice ordenado.

Estos índices pueden ser definidos como únicos, (unique clustered index, o unique nonclustered index), lo que no permite que se duplique la información en más de un tupla en el atributo (o combinación de atributos) del índice.

La filosofía de diseño de la IDM es que el ABD es el único con el conocimiento sobre cuáles atributos se deben de generar los índices, por lo que el administrador es el responsable de ordenar a la IDM la creación de los índices.

IDM también maneja lo que se conoce como comandos almacenados (stored commands), que son una secuencia de instrucciones definidas en IDL, que pueden ser ejecutados simplemente invocando su nombre y que son de mucha ayuda en los procesos que se llevan a cabo en las bases de datos.

### **3.5.2 EL MODELO RELACIONAL DE DATOS EN IDM**

Un modelo de datos es una combinación de tres componentes (Cap. I):

1) una colección de tipos de estructuras de datos (la construcción de cualquier base de datos que conforma el modelo)

2) una colección de operadores o reglas de inferencia, los cuales pueden ser aplicados para cualquier circunstancia válida de los tipos de datos mencionados en 1), para recuperar o derivar datos de cualquier parte de estas estructuras en cualquiera de las combinaciones deseadas

3) una colección de reglas de integridad generales, las cuales implícita o explícitamente definen la consistencia de la base de datos.

Un modelo descrito con la IDM de Britton-Lee trabaja sus modelos relacionales con estos lineamientos, en donde como ya sabemos, la estructura básica en el modelo relacional es la relación.

En las operaciones relacionales que se manejan con IDM, varias clases de operadores son utilizados para generar a partir de relaciones existentes otras nuevas; y estos operadores están basados en el álgebra relacional y en el cálculo relacional.

#### **El Álgebra Relacional en IDM**

Como ya vimos, una expresión en el álgebra exhibe una secuencia específica de operaciones para generar una relación resultado.

IDM soporta tanto las operaciones tradicionales de conjuntos (unión, intersección, diferencia, producto cartesiano), como las operaciones relacionales especiales (proyección, selección, reunión, división.)

## El Cálculo Relacional en IDM

El cálculo relacional, al igual que el álgebra relacional, es una notación que sirve para describir relaciones resultados que se derivan de las relaciones existentes. La diferencia con el álgebra es que en el cálculo no se especifica una secuencia de operaciones, es decir es un lenguaje no-procedimental.

La forma de una expresión de cálculo relacional en IDM se contruye como cualquier forma general de cálculo relacional (Capítulo I):

(<lista> donde <predicado>

Las reglas de integridad, las cuales gobiernan la consistencia de una base de datos, deben ser claramente definidas dentro de cualquier Sistema de Administración de Bases de Datos relacionales.

En IDM el sistema de bloqueo (locking system) permite múltiples lecturas concurrentes en una relación, pero cualquier operación de escritura bloquea todos los otros accesos concurrentes.

La mayoría de estos mecanismos de seguridad son automáticos e invisibles para el usuario.

Otra parte de las características de integridad de IDM es la facilidad para manejar las transacciones. IDM automáticamente asume tres niveles de consistencia para las actualizaciones de los usuarios, lo que no permite que se vean las actualizaciones parciales en las relaciones; sino hasta que éstas hayan concluido; es decir, estos niveles son internos y transparentes para los usuarios.

Los principios de seguridad e integridad en IDM proporcionan controles automática y expresamente sobre el acceso a los datos.

IDM cuenta con una serie de relaciones del sistema en los que lleva el control de todas las bases que se van creando en ella además de los controles ya mencionados.

Algunas relaciones del sistema están en cada base de datos, como son las relaciones: relation, attribute, indices, protect, query, crossref, transact, users, host\_users, blockalloc, disk\_usage, batch y descriptions.

Otras relaciones del sistema están en la base de datos que administra el sistema "system database" que es una base



de datos única y que contiene información acerca de las otras bases de datos alojadas en IDM y la información de la configuración común a todas ellas; estas relaciones son: databases, disks, lock, configure, dbinstat, monitor y dermonitor.

Todas estas relaciones son manejadas exclusivamente por el ABD que corresponde a cada base de datos, que es el propietario o creador de cada base de datos en particular, además de ser el responsable de su diseño y seguridad; y por el Administrador de los Sistemas, que es el administrador de la base de datos del sistema (system database) y responsable del control operacional de todo IDM.

### 3.5.3 ALGEBRA Y CALCULO RELACIONAL CON IDM

A continuación se da una explicación de cómo se basa IDM en el álgebra y el cálculo relacional para llevar a cabo sus operaciones por medio de IDL y posteriormente se dan una serie de ejemplos de operaciones de álgebra relacional con IDL.

Utilizando los conceptos de nombres de atributos para relaciones derivadas, cuando se requiera trabajar con una relación se le asigna una variable para ser utilizada como un alias temporal y por medio de ella hacer la referencia a los atributos de esa relación, no sin antes abrir la base de datos; esto es:

```
open base ;  
  
range of a is rela ;  
range of b is relb ;
```

donde **base** es el nombre de la base de datos, **a** es la variable de tuple que varía sobre la relación con nombre **rela** y **b** es la variable de tuple que varía sobre la relación con nombre **relb**; siendo **a** y **b** definidas en estas relaciones a través de la declaración **range**.

Los operadores "relacionales" que podemos utilizar para efectuar las operaciones que deseemos y que necesitemos para poder formar las condiciones del cálculo relacional son:

```
=      (igual a)  
!=     (no igual a)  
>     (mayor que)  
>=    (mayor o igual que)  
<     (menor que)  
<=    (menor o igual que)
```

y los operadores booleanos:

**and**        ( y )  
**or**         ( o )

Los cuantificadores [para todo] y [existe] se representan dentro de la **condición** (o predicado) precedidos por la declaración **where** como sigue:

**where** <condición>

Con estos elementos y usando una sintaxis adecuada se construyen las expresiones del cálculo de tuples, donde todas las ocurrencias de las variables de tuples son libres dentro de la condición.

Los **comandos** de IDL que podemos utilizar en las expresiones de cálculo con la forma: (<lista>) donde <predicado> donde <lista> controla los atributos que se desean trabajar; (**where**) controla cuáles tuples se quieren trabajar; y en <predicado> se detallan las condiciones para las operaciones; son:

**retrieve**        (ordena recuperar un[os] tuple[s] de una[s] relación[es])  
**append to**        (ordena agregar un[os] tuple[s] a una relación)  
**replace a**        (ordena reemplazar un[os] tuple[s] en una relación con variable de tuple a)  
**delete a**         (ordena borrar un[os] tuple[s] de una relación con variable de tuple a)

Las operaciones de lo que llamamos "funciones agregadas" sobre los atributos de una relación específica siempre bajo una condición dada son:

**sum(---)**        (calcula el total en los valores de un atributo en una relación)  
**avg(---)**        (calcula el promedio en los valores de un atributo en una relación)  
**min(---)**        (encuentra el mínimo en los valores de un atributo en una relación)  
**max(---)**        (encuentra el máximo en los valores de un atributo en una relación)  
**count(---)**      (contabiliza las ocurrencias de un atributo en una relación)  
**any(---)**        (detecta el valor de verdadero en la ocurrencia de un atributo al efectuar una operación condicionada en una relación)

todas estas operaciones guardan su resultado en una variable declarada específicamente para la salida del mismo.

Con IDL también podemos trabajar cálculos aritméticos sobre los atributos de las relaciones, y al igual que los anteriores, guardan su resultado en una variable declarada para la salida del mismo:

+	(suma)
-	(resta)
*	(multiplica)
/	(divide)

En IDL cualquier tipo de salida generada por cualquier ejecución de comando, puede ser obtenida en el orden que se desee sin afectar el que tienen los datos que están almacenados; esto se logra con la siguiente declaración:

**order by** (controla el orden de la salida de los tuples)

También se puede generar una relación a partir de otra(s), con las condiciones que se deseen, y dejarla almacenada con un nombre declarado; esto a través del comando:

**retrieve into** (recupera la información deseada de una(s) relación(es) y la guarda en una relación resultado)

IDL igualmente permite destruir una relación utilizando el comando:

**destroy** (ordena destruir una relación incluyendo comandos e índices inherentes a ella -)

Existen también los comandos de seguridad que maneja el SABD y que permiten o niegan a un usuario (o grupo de usuarios) lo siguiente: leer y/o grabar información en una relación; crear: índices en una relación, bases de datos y relaciones; y por último, permiten o no ejecutar comandos. Estos comandos de seguridad son:

**permit** permite a un usuario (o grupo de usuarios): leer, escribir o los dos en una relación o en una vista; ejecutar comandos; crear índices, crear bases de datos y crear relaciones.

**deny** niega a un usuario (o grupo de usuarios): leer, escribir o los dos en una relación o en una vista; ejecutar comandos; crear índices, crear bases de datos y crear relaciones.

También podemos tener el control del manejo de las transacciones o de un conjunto de instrucciones, que serán abortadas por el sistema si son interrumpidas durante su ejecución en cualquiera de sus pasos por cualquier circunstancia externa; en otras palabras, normalmente cada comando de IDM es una transacción, y utilizando **begin/end/abort**, cualquier cadena de comandos consecutivos puede ser construida dentro de una sola transacción compuesta, definida por el usuario; estas instrucciones son:

<b>begin transaction</b>	pide a IDM que mantenga cualquier bloqueo requerido por el contenido de la cadena de comandos consecutivos, y hace actualizaciones tentativas.
<b>end transaction</b>	finaliza todos los bloqueos requeridos a partir del "begin transaction" y concluye cualquier actualización desde el "begin".
<b>abort transaction</b>	Cesa todos los bloqueos adquiridos desde el "begin" y descarta todas las actualizaciones desde el "begin".

Existen una serie de comandos y utilerías que son exclusivos del control el SABD y que solamente son manejados y/o monitoreados por el ABD encargado de la IDM (Administrador de los Sistemas) para poder detectar errores en el(los) sistema(s), tanto físicos como lógicos. Algunos de estos comandos ayudan en forma automática a la corrección de estas fallas, lo que hacen que la IDM sea considerada como una máquina "inteligente". Algunos de estos procesos de auditoría no corrigen los errores, pero orientan al ABD para la detección y corrección de los mismos.

A continuación se muestran unos ejemplos de operaciones con IDL.

## Ejemplos de operaciones de Algebra Relacional con IDL

En los siguientes ejemplos, se va a trabajar la base de datos denominada *base* y las relaciones *rela* y *relb*, ambas contenidas en dicha base de datos.

Para todos ellos, siempre abrimos primero la base de datos *base* para que IDL nos permita trabajarla y luego declaramos las variables de tuple: *a* para referenciar a la información almacenada en la relación *rela* y *b* para referenciar a la de la relación *relb* (una o ambas, según sea el caso.)

Nota: Cuando se desea(n) obtener solamente uno(s) de los atributos de la relación que se está trabajando, éste(os) se detalla(n) dentro de los paréntesis en el orden en el que se esperan recuperar (a través del nombre con calificación que le corresponde a cada atributo), por ejemplo:

```
retrieve (a.atributo1,a.atributo3,atributo5) ;
```

pero si lo que queremos es obtener toda la información contenida en el(los) tuple(s) que se solicita(n) en el comando, entonces se utiliza la nomenclatura *all* e IDL muestra la lista de los atributos del(los) tuple(s) en el orden en que están definidos en la relación, por ejemplo:

```
retrieve (a.all) ;
```

---

---

```
*** rela UNION relb ***
```

---

---

```
open base ;  
range of a is rela ;  
range of b is relb ;
```

```
[a continuación, recuperamos la información  
contenida en cada una de estas relaciones]
```

```
retrieve (a.all) ;  
retrieve (b.all) ;
```

Nota: No existe un comando en IDL que recupere la unión de dos relaciones; por lo que *b* debe ser agregada a *a*, o generar la unión en una tercera relación.

---

---

\*\*\* rela INTERSECCION relb \*\*\*

---

---

open base ;  
range of a is rela ;  
range of b is relb ;

[a continuación, recuperamos los tuples de la relación rela tal que cumplan con la condición de que la información contenida en el atributo campo de la relación rela sea igual a la contenida en el atributo campo de la relación relb ]

retrieve (a.all) where a.campo=b.campo ;

Nota: campo es el nombre de un atributo; a.campo es el nombre con calificación del atributo campo de la relación rela; y, b.campo es el nombre con calificación del atributo campo de la relación relb.

---

---

\*\*\* rela DIFERENCIA relb \*\*\*

---

---

open base ;  
range of a is rela ;  
range of b is relb ;

[a continuación, recuperamos la información contenida en el atributo campo de los tuples de la relación rela tal que cumplan con la condición de que están en el atributo campo de la relación rela y no en el atributo campo de la relación relb]

retrieve (a.campo) where  
count (b.campo by a.campo where a.campo=b.campo)=0 ;

Nota: La cláusula by ejecuta un producto cruz del atributo campo de rela contra el atributo campo de relb, es decir, la información contenida en el atributo campo de cada tuple de rela se compara contra toda la de la columna campo de relb; y donde no se cumple la igualdad se recupera el tuple de rela (o sea, donde count=0 para a.campo=b.campo)

---

---

\*\*\* rela PRODUCTO relb \*\*\* (Producto cartesiano extendido)

---

---

open base ;  
range of a is rela ;  
range of b is relb ;

[a continuación, recuperamos el producto cruz de la información contenida en cada una de estas relaciones]

retrieve (a.all,b.all) ;

---

---

\*\*\* SELECCION rela DONDE <predicado> \*\*\*

---

---

open base ;  
range of a is rela ;

[a continuación, recuperamos (seleccionamos) los tuples de la relación rela tal que cumplan con la condición de que la información contenida en el atributo campo tenga un valor distinto de 2]

retrieve (a.all) where a.campo>2 or a.campo<2 ;

---

---

\*\*\* PROYECTA rela SOBRE <lista de atributos> \*\*\*

---

---

open base ;  
range of a is rela ;

[a continuación, recuperamos (proyectamos) dos columnas (atributos) de la relación rela]

retrieve (a.campo1,a.campo2) ;

Nota: Donde campo1 y campo2 son atributos de la relación rela.

---

---

\*\*\* REUNE (rela VECES relb) DONDE <predicado> \*\*\*

---

---

open base ;  
range of a is rela ;  
range of b is relb ;

[a continuación, recuperamos el producto cruz de la información contenida en cada una de estas relaciones, pero con la condicionante de que el atributo campo de la relación rela sea igual al atributo campo de la relación relb]

retrieve (a.all,b.all) where a.campo=b.campo ;

---

---

\*\*\* rela DIVIDENTRE relb <SOBRE campo> \*\*\*

---

---

open base ;  
range of a is rela ;  
range of b is relb ;

[primero desplegamos rela y relb]

retrieve (a.all) ;  
retrieve (b.all) ;

[ahora hacemos la división]

retrieve unique (a.campo1) order by a.campo1  
where count unique(b.campo by a.campo1  
where b.campo=a.campo)  
= count unique(b.campo) ;

#### 3.5.4 AM-2000-10 y AM-1062.

Las máquinas anfitrionas que se utilizan para acceder la IDM en la Dirección General de Bibliotecas son la AM-2000-10 y la AM-1062, y a continuación se describen brevemente para terminar este capítulo.

##### 3.5.4.1 Alpha Microsystem 2000-10

El sistema AM-2000-10 de Alpha Microsystems soporta hasta 108 MB de memoria principal y cuenta con una gran capacidad de almacenamiento en disco.



La máquina **AM-2000-10** que se encuentra en la Dirección General de Bibliotecas, cuenta con un sistema operativo propietario denominado **AMOS/32** el cual es multiusuario, multitareas, tiempo compartido, sistema basado en disco; macroensamblador de 3 pasos; encadenador; generador de símbolos; depurador con manejo de símbolos; administración de archivos; incluye los lenguajes **AlphaBASIC**, ensamblador y C.

Su arquitectura de multiprocesadores permite afinar la respuesta del sistema al utilizar procesadores especiales para manejar terminales, realizar respaldos y para los procesos de comunicaciones; asimismo permite agilizar las operaciones matemáticas mediante el uso de un co-procesador aritmético opcional.

La Unidad Central de Proceso de la **AM-2000-10** tiene, para los procesos que lleva a cabo, un microprocesador **Motorola 68020** de 16MHz con una memoria de 12 MB.

Cuenta además con un controlador inteligente de discos que mejora el funcionamiento de este dispositivo y una memoria caché que incrementa la velocidad de procesamiento. Cuenta también con dos controladores inteligentes de puertos seriales.

Su microprocesador tiene direccionamiento de memoria de 32 bits; al igual que la transferencia de datos.

Soporta hasta 240 puertos usando las siguientes opciones: módulos de expansión, controladores inteligentes y chasis de expansión. En la DGB se tienen instalados 54 puertos, con 3 líneas conectadas a modems para realizar enlaces vía red telefónica y a través de la Red **UNAM-Ethernet** (con una limitación actual de 10 usuarios.)

El disco que utiliza para almacenamiento es del tipo **Winchester** (sellado), con una capacidad de 145 MB con 5 1/4", expandible a 3.4 GB; con interfaz **SCSI** (Small Computer System Interface - Sistema de Interfaz para Computadoras Pequeñas.)

Para los respaldos se cuenta con una unidad de cinta magnética de carrete abierto de 1.27 cm. (1/2".)

Se cuenta además con el sistema **AlphaNet** de comunicaciones **VersiCOMM-Plus** para conexión con computadoras personales PC, Ethernet.

#### **3.5.4.2 Alpha Microsystem 1062**

La máquina **AM-1062** cuenta con un sistema operativo propietario llamado **AMOS/L** el cual es multiusuario, multitareas, tiempo compartido, sistema basado en disco;

macroensamblador de 3 pasos; encadenador; generador de símbolos; depurador con manejo de símbolos; administración de archivos; incluye los lenguajes AlphaBASIC, ensamblador y C.

La Unidad Central de Proceso de esta Alpha tiene, para los procesos que lleva a cabo, un microprocesador Motorola 68000 de 10 MHz con una memoria de 4 MB.

Su microprocesador tiene direccionamiento de memoria de 16 bits.

Soporta hasta 60 puertos usando las siguientes opciones: módulos de expansión, controladores inteligentes y chasis de expansión. En esta Alpha se tienen instalados 34 puertos seriales y dos paralelos.

El disco que emplea para el almacenamiento de la información es del tipo Winchester (sellado), con una capacidad de 16 MB con 14".

Y con esto se termina la descripción general de los equipos utilizados en el desarrollo y mantenimiento de los principales sistemas automatizados de la Dirección General de Bibliotecas.

**CAPITULO 4**

---

---

**DESCRIPCION  
DEL  
SISTEMA**

---

---

## **DESCRIPCION DEL SISTEMA**

### **4.1 INTRODUCCION**

El presente capítulo detalla técnicamente el sistema INVENTARIOS, su configuración y los procesos que controla.

Como ya se mencionó anteriormente, este sistema es el que se encarga de controlar automáticamente el acervo bibliográfico que todas las bibliotecas de la UNAM ponen a disposición de los universitarios.

Este fue desarrollado en la Dirección General de Bibliotecas; y forma parte de un conjunto de módulos, que en su totalidad constituyen el sistema integral de información denominado LIBRUNAM.

Es por esto que para describir el módulo de Inventarios, es necesario describir la configuración global del sistema LIBRUNAM junto con los procesos que se aplican al material bibliográfico de las bibliotecas de la UNAM y que afectan a sus inventarios.

### **4.2 ESTRUCTURA DE LA BASE DE DATOS**

#### **4.2.1 Sistema LIBRUNAM: Generalidades**

El Sistema LIBRUNAM, fue diseñado para controlar la información catalográfica de la UNAM, y contiene a la fecha 580,000 títulos diferentes, con un incremento anual promedio de 40,000 fichas bibliográficas que implican un promedio de 65,000 volúmenes. Por otro lado, se tiene un incremento promedio anual de 80,000 volúmenes duplicados que se adquieren de los títulos que ya están registrados en el banco de datos; por lo que fue necesario que el diseño se estructurara sobre una base de datos relacional, donde la consulta por medio del álgebra relacional garantizaran a los usuarios la recuperación rápida y confiable de la información.

El banco de datos LIBRUNAM se construyó con las fichas catalográficas que formaban el catálogo oficial de la UNAM, hasta enero de 1978 y a partir de esa fecha se fue actualizando con la producción diaria del Departamento de Procesos Técnicos de la Subdirección Técnica de la Dirección General de Bibliotecas.

Los procesos automatizados que se llevan a cabo en este banco y que son comunes a todos los procesos que corresponden

a un Sistema de Bases de Datos Relacionales de Información Bibliográfica son:

- Alta de nuevos registros.
- Corrección de registros.
- Acceso directo o indirecto a la información de la base de datos.
- Regeneración del banco de datos.
- Mantenimiento de la base de datos.
- Protección de la base de datos.
- Elaboración de salidas impresas (como son los catálogos de los libros, las remisiones de los cargos, las etiquetas, etc..)
- Elaboración de estadísticas propias del banco de datos.

LIBRUNAM tiene la capacidad, además de funcionar en red y de adaptarse a los diferentes procesos que se realizan en las bibliotecas gracias a su diseño modular. Cuenta también con un sistema de seguridad para garantizar la calidad del banco.

Además de ofrecer información sobre los libros que se buscan, indica en cuáles bibliotecas del sistema universitario podemos obtener el material deseado.

Como ya se mencionó en el Capítulo 3, IDL es el lenguaje que utilizamos en este sistema para la definición, recuperación y manipulación de los datos, así como para el control de acceso y compartimiento de los mismos, lo que permite asegurar su independencia; es decir, se emplea para la interfaz con el DBMS y sus controles, y proporciona la integridad, consistencia y seguridad que necesitamos en la base de datos; además de que faculta la interacción con el Sistema Manejador de la Base de Datos LIBRUNAM.

Este diseño suministra las herramientas necesarias al especialista en los procesos técnicos de los libros, para que pueda investigar, registrar y procesar la información dentro del banco de datos, facilitando al usuario de la biblioteca el acceso a la información.

#### 4.2.2 La información y sus elementos (etiquetas)

Como ya se ha dicho, los datos que permiten a un usuario reconocer un libro y localizarlo tanto en el universo de su información (tema, autor, título, etc.) como físicamente (clasificación, dependencia, etc.), están registrados en lo que conocemos como fichas catalográficas. Estas contienen los datos que permiten identificar del todo una obra y su construcción se hace en base a reglas estandarizadas internacionalmente para facilitar el intercambio de información y/o procesamiento de los libros.

La información de una ficha catalográfica es sumamente variable, tanto en contenido como en frecuencia; por lo que la aplicación de reglas para normalizarla y facilitar su proceso es muy importante.

Los elementos que componen una ficha pueden variar en número desde tres hasta más de treinta; incluso dentro de los mismos no siempre existe uniformidad.

A continuación se nombran cada uno de los elementos que pueden estar contenidos dentro de una ficha: [61]

- *Clasificación* - combinación de letras y números que identifican el tema de la obra.
- *Autor* - responsable de la obra; puede ser una persona (o un grupo), instituciones, etc.
- *Título de la obra*
- *Pie de Imprenta* - fecha, ciudad y casa editorial.
- *Edición* - número de edición y observaciones.
- *Descripción física del libro o colocación* - describe la paginación, si viene en volúmenes o tomos, características especiales, etc. .
- *Serie del libro* - indica si el libro pertenece a una serie dada y el número que ocupa en ella.
- *ISBN* (International Standard Book Number.)
- *Notas bibliográficas* - descripción de ciertas características de las obras.
- *Encabezamiento de materia (temas)* - ramas del conocimiento que están relacionadas con las obras, así como personajes sobre los que tratan, fechas, lugares, etcétera.

- Coautores
- Número de matriz - número que identifica a las fichas en la UNAM.
- Clave de la dependencia - serie de letras que identifican a una biblioteca dentro del sistema bibliotecario de la UNAM.
- Número de adquisición - número que identifica a cada volumen dentro de una biblioteca.

La codificación bibliográfica es el proceso a través del cual los datos que constituyen una ficha bibliográfica son convertidos en elementos que pueda leer y manejar la computadora.

Para poder codificar las fichas bibliográficas, es necesario que cada uno de estos elementos (o campos) sean acompañados de una etiqueta o identificador en base al formato MARC/DGB.

Es tarea del especialista en biblioteconomía el codificar esta información dentro de los campos correspondientes para su procesamiento en el sistema LIBRUNAM con ayuda del formato MARC/DGB.

El formato MARC/DGB asocia un prefijo denominado etiqueta a cada elemento (o campo) de la ficha, y separa los subcomponentes de cada elemento con un símbolo preestablecido.

Cada componente (o campo) es separado por un signo de "\$", para fácil identificación de la etiqueta por la máquina.

Los datos que contiene una ficha bibliográfica pueden ser de dos tipos: datos de longitud fija, donde siempre se tiene el mismo número de caracteres; y datos de longitud variable, en donde no se puede previamente determinar el número de caracteres.

Los primeros son englobados en el campo de longitud fija llamada "parte fija" (que se localiza dentro de la etiqueta 008), y los segundos son asignados a los campos de longitud variable (etiquetas restantes.)

A continuación se señalan las etiquetas que se trabajan en el formato MARC/DGB para la codificación de los libros en el sistema bibliotecario de la UNAM: [62]

\$008	Datos generales de la obra (campos fijos)
\$050	Clasificación
	<b>Asiento Principal:</b>
\$100	Autor personal
\$110	Autor corporativo
\$111	Congreso o reunión (asamblea)
\$130	Título uniforme
	<b>Cuerpo de la ficha:</b>
	Título
\$240	Título uniforme
\$245	Título propiamente dicho
\$250	Edición
\$260	Pie de imprenta
\$300	Descripción física
\$430	Serie
\$020	ISBN
	<b>Notas</b>
\$500	Generales
\$502	Tesis
\$505	Contenido
	<b>Encabezamientos de materia:</b>
\$600	Autor como materia
\$610	Organismos corporativos como materia
\$611	Congreso como materia
\$630	Título uniforme como materia
\$650	Encabezamientos de materia generales
	<b>Asientos secundarios:</b>
\$700	Autores personales
\$710	Autores corporativos
\$711	Congresos
\$730	Títulos uniformes
\$740	Títulos abreviados u otros títulos
	<b>Controles internos:</b>
\$900	Clave de la dependencia
\$035	Número de matriz
\$901	Número(s) de adquisición
\$902	Clasificador, supervisor y capturista

Los datos de longitud fija que se consignan en la parte fija (etiqueta 008) y que son obligatorios son:

1. Fechas
2. Origen de catalogación



3. Clase de ilustraciones que presenta el material
4. Presentación física de los materiales (Forma de reproducción)
5. Tipo de contenido de material (Forma de contenido)
6. Tipo de publicación oficial
7. Idioma
8. Asiento principal como parte del título
9. Código de catalogación
10. Número de identificadores

Cada etiqueta se acompaña de dos caracteres llamados indicadores o identificadores de campo (idc) colocados a continuación de la etiqueta y al inicio de la información del campo. Estos indicadores nos señalan instrucciones sobre los datos del campo; esto es: el primer caracter en el idc de un título nos indica que se debe efectuar una tarjeta secundaria de título, y el segundo caracter nos indica cuántos lugares (caracteres) al inicio de los datos del campo se deben ignorar para la alfabetización. Un ejemplo de esto es si el título comienza con un pronombre, aquí se indica el número de caracteres del pronombre a ser suprimidos.

Cuando el tipo de campo no requiere de este indicador, entonces se deja(n) el(los) espacio(s) en blanco.

Cuando un título es de nuevo ingreso al banco, toda la información correspondiente a sus elementos se codifica y se detalla por los especialistas del Departamento de Procesos Técnicos de la Dirección General de Bibliotecas en las hojas de codificación específicamente diseñadas para este fin.

EL formato de estas hojas facilita y resume los datos del libro para que posteriormente éstos sean capturados para su proceso en la computadora.

En ellas, la mayoría de las etiquetas vienen impresas, total o parcialmente a fin de agilizarle la tarea al personal a la hora de codificar la información.

Se utiliza una hoja de codificación por cada obra.

A continuación se muestra el diseño general de la base de datos de LIBRUNAM y la estructura de cada una de las relaciones que la componen.

#### 4.2.3 Estructura de la Base de Datos "libros"

El nombre "libros" fue el que se le asignó a la base de datos diseñada para el sistema LIBRUNAM con la máquina IDM.

La base de datos "libros" del sistema LIBRUNAM está compuesta por 5 relaciones básicas fijas y un número indeterminado de relaciones temporales las que son, o de apoyo para los procesos que se llevan a cabo con el sistema, o producto de los procesos efectuados en estas 5 relaciones (relaciones resultado.) También existe un número fijo de relaciones que utiliza el sistema DBMS para controlar la base, mismas que lógicamente el DBMS permite acceder, pero no modificar.

Las 5 relaciones que contienen el banco de datos LIBRUNAM son las siguientes:

- fichas
- dispersion
- lista3
- numerounico
- tablainv
- inventario

Estas relaciones se detallan cada una más adelante, pero para ello antes se da una descripción general de los comandos que se utilizaron para crear sus índices, y se muestran los tipos de campos que se emplearon para definir sus atributos.

En términos genéricos, el tamaño de los tuples de cada una de estas relaciones se determinó en la definición de cada una de ellas al asignar el tamaño a sus atributos; y estos atributos se especificaron utilizando los siguientes tipos de datos:

- ii dato tipo entero de 1 byte (es decir, almacena un entero menor a  $2^7$  que equivale a un número menor a 128)

- i2** dato tipo entero de 2 bytes (es decir, almacena un entero menor a  $2^{15}$  que equivale a número menor a 32768)
- i4** dato tipo entero de 4 bytes (es decir, almacena un entero menor a  $2^{31}$  que equivale a un número menor a 2147483648)
- c** dato tipo caracter (cadena) comprimido; es decir, se almacena la información quitando espacios en blanco (con un tamaño máximo de 255 caracteres.)
- uc** dato tipo caracter (cadena) sin comprimir; es decir, se almacena la información respetando los espacios en blanco (con un tamaño máximo de 255 caracteres.)

Por otro lado también tenemos que las declaraciones de los índices en IDL también son comunes en su construcción para todas las relaciones de la base de datos; y la sintaxis es la siguiente:

**create tipo\_indice on nombre\_relación (atributo) <opciones>**

Donde **tipo\_indice** puede ser:

- **clustered index** - ordena la relación físicamente por este índice
- **nonclustered index** - índice que no afecta el orden físico de la información
- **unique clustered index** - igual que clustered pero determina la unicidad de la información en este campo
- **unique nonclustered index** - igual que nonclustered pero determina la unicidad de la información en este campo

Donde **nombre\_relación** puede ser:

- cualquiera de las relaciones ya mencionadas (fichas, dispersion, lista3, tablainv e inventario.)

Donde **atributo** puede ser cualquiera de los atributos que componen la relación y siempre dentro de paréntesis.

El índice se puede definir sobre un atributo o sobre una lista de atributos cuando se trate de una llave compuesta.

Y **opciones**, como su nombre lo indica, se refiere a los atributos opcionales que apoyan la creación del índice.

En la generación de los índices de algunas de las relaciones de la base **libros** sólo se utilizó la opción:

■ **with delete\_dups**

Esta opción sólo se maneja cuando se define la unicidad del índice.

Si no se hace uso de ella cuando se genera un índice único sobre una relación, la máquina al encontrar un duplicado en un proceso de actualización (reemplazar, copiar o agregar) aborta la operación y manda un mensaje de error; pero por el contrario si se utiliza, entonces no marca error, se hacen las actualizaciones y sólo borra los repetidos, es decir, no lleva a cabo las ejecuciones que implican una duplicidad en una cierta información.

Lo mismo sucede si se trabaja la información en lote, en donde, al encontrar repeticiones lo aborta completo sin el uso de esta opción.

De igual forma se comporta en los procesos de regeneración de índices, o cuando se crea uno nuevo sobre una relación que ya tiene almacenada información duplicada, por lo que hay que tener cuidado en su manipulación, ya que al aplicarla en relaciones con información en esta situación, puede efectuar eliminaciones que no se deseen.

A continuación se detallan cada una de las relaciones mencionadas:

**FICHAS**

En esta relación se almacena toda la información contenida en las fichas catalográficas, es decir, la que se refiere a la descripción de los libros, o lo que podríamos llamar el cuerpo de la ficha.

El almacenamiento del contenido de cada ficha catalográfica se hace verticalmente desglosada por sus campos, donde los tuples que corresponden a una ficha están vinculados por medio del número de *matriz*, por lo que es a

través de este número que el sistema accesa la información (los registros que se liga(n) a cada libro.)

La estructura de esta relación para cada tuple es la siguiente:

CAMPO	NOMBRE DEL ATRIBUTO	TIPO	LONGITUD
1	matriz	i4	4
2	etiqueta	uc	3
3	tarjeta	i1	1
4	idc	c	2
5	texto	c	248

**matriz** - es el número único consecutivo asignado a cada ficha catalogáfica nueva que ingresa al banco de datos del sistema LIBRUNAM, y cuyo valor, a la fecha, no puede ser mayor al número 999,999.

**etiqueta** - es el número (etiqueta - sección 4.2.2 de este capítulo) asignado según el formato Marc/DGB a la información contenida en el texto.

**tarjeta** - indica el número de registros para cada etiqueta y su orden.

**idc** - es el identificador de campo

**texto** - es donde se almacena la información de la etiqueta correspondiente (contenido del campo.) Cuando el tamaño de la cadena de caracteres rebasa de los 248, entonces se corta y se continúa en el siguiente registro con un número consecutivo de tarjeta.

Esta relación tiene una llave primaria que permite duplicidad sobre el atributo **matriz** (clustered index sobre el atributo **matriz**), ya que cada tuple corresponde a la información de cada campo (etiqueta) de la ficha y el conjunto de tuples que forman la ficha se accesan a través de esta llave.

Esta relación además está físicamente ordenada por este índice.

### DISPERSION

En esta relación se almacenan todas las palabras recuperables de las fichas catalográficas; es decir aquí se almacenan las llaves de acceso a información de las fichas.

La estructura de la relación para cada tuple es la siguiente:

CAMPO	NOMBRE DEL ATRIBUTO	TIPO	LONGITUD
1	llave	c	20
2	numuni	i4	4
3	existen	i4	4

**llave** - son las palabras recuperables en los procesos de búsqueda.

Se deja un tamaño para la palabra no mayor a 20 caracteres aún cuando la información que se trabaja se refiera a una clasificación, ya que se guarda sin espacios ni puntuación; y el único problema que se puede llegar a presentar es en los casos raros de clasificaciones de mayor tamaño las cuales quedan almacenadas truncas (como llave de recuperación.)

**numuni** - es el número único asignado consecutivamente a cada nueva palabra que ingresa a las palabras recuperables.

**existen** - es el número de veces que aparece la llave en el banco de datos (a excepción de cuando se trate de una clasificación que debe de ser única y en cuyo caso se guarda aquí el número de matriz que le corresponde.) Se utiliza para dar información al usuario cuando hace una recuperación en línea del número de fichas que contienen la palabra llave.

Esta relación tiene una llave primaria en el atributo **llave** que no permite duplicidad, y por la que está físicamente ordenada (unique clustered index sobre el campo llave); y dos llaves alternas (nonclustered index), una sobre el atributo **numuni** y otra en el atributo **existen**.

### LISTAS

En esta relación se almacenan las listas de la palabras que entran al sistema; es decir, es la relación que liga los

datos de la relación  *fichas* con los de la relación  *dispersion*.

La estructura de la relación para cada tuple es la siguiente:

CAMPO	NOMBRE DEL ATRIBUTO	TIPO	LONGITUD
1	numfam	i4	4
2	tipo	i1	1
3	matriz	i4	4

**numfam** - es el número único para cada palabra con familia (se hacen familias por autor, títulos, temas, etc. según sea el tipo de información), para agilizar los procesos de recuperación. Se obtiene aplicando la fórmula:

$$\text{numfam} = (\text{número único de palabra} * 100) + \text{tipo}$$

**tipo** - es el tipo de etiqueta; es decir, es donde se engloba el tipo de información, como podría ser: autor, título, tema, editorial, etc.

**matriz** - es el número único consecutivo, asignado a cada ficha catalogáfica nueva que ingresa al banco de datos del sistema LIBRUNAM, y cuyo valor (a la fecha) no puede ser mayor al número 999,999.

Esta relación tiene una llave primaria compuesta en los atributos **numfam** y **matriz** (nonclustered index sobre (numfam,matriz) ), y una llave foránea en el atributo **matriz** (nonclustered index); y ambos índices admiten duplicados.

#### NUMEROUNICO

Es la relación donde se almacenan el total de títulos en el banco (último número de matriz), y el último número único de palabras recuperables del sistema LIBRUNAM.

Es una relación pequeña de 2 tuples, que ya no crece y donde sólo se accesa y actualiza su información durante los procesos de alta de fichas bibliográficas.

La estructura de la relación para cada tuple es la siguiente:

CAMPO	NOMBRE DEL ATRIBUTO	TIPO	LONGITUD
-------	---------------------------	------	----------

1	llave	c	6
2	numuni	i4	4

**llave** - aquí se define de qué se trata la información, es decir de la *matriz* o del número único *numuni*.

**numuni** - es el total de títulos (último número de *matriz*) o el total de palabras distintas (último número único de palabra distinta que ingresó al banco de datos.)

#### TABLAINV

En esta relación se almacenan los datos generales de las dependencias bibliotecarias de la UNAM.

Cuando una biblioteca ingresa por primera vez al sistema bibliotecario de la UNAM, se le asigna una clave única, que va a ser la que la identifique dentro del ámbito universitario, y por otro lado, se le asigna un número de dependencia (consecutivo conforme al último asignado que se dio de alta en el sistema LIBRUNAM) que debe ser único y con el que se le identifica dentro de la base de datos; es decir dentro del sistema LIBRUNAM.

Se utiliza también para ir controlando los últimos números de adquisición de los libros que van ingresando a las bibliotecas de la UNAM, y generalmente se toma como un parámetro para dar una idea del tamaño de la colección de cada biblioteca.

La estructura de la relación para cada tupla es la siguiente:

CAMPO	NOMBRE DEL ATRIBUTO	TIPO	LONGITUD
-------	---------------------------	------	----------

1	dep	i2	2
2	clave	c	6
3	nombre	c	120
4	numadq	i4	4



**dep** - es el número de la dependencia, asignado consecutivamente a su ingreso al sistema y que se utiliza para detectar la biblioteca dentro de la base de datos; es decir, para control interno dentro de la DGB. A la fecha es un número menor a 350.

**clave** - es la clave de la biblioteca dentro del sistema bibliotecario asignada por la DGB; está compuesta por una cadena de caracteres alfanuméricos; y también es única y es por medio de la cual se identifica a la biblioteca dentro del ámbito universitario.

**nombre** - es el nombre completo de la biblioteca

**numadq** - es el último número de adquisición asignado al libro de más reciente ingreso al acervo de la biblioteca. En este número se basa el sistema para asignar automáticamente y en forma consecutiva uno nuevo a cada uno de los volúmenes que va obteniendo la biblioteca; a la vez que lo va actualizando.

Esta relación tiene una llave primaria en el atributo **clave** que no permite duplicidad, y por la que está físicamente ordenada (unique clustered index sobre el campo clave); y una llave alterna sobre el atributo **dep** que tampoco permite duplicidad (unique nonclustered index.)

La última relación del sistema, se refiere a la que almacena los controles de los inventarios, por lo que se decidió poner en una sección aparte.

#### 4.2.4 Estructura de la relación "inventarios"

##### INVENTARIO

En esta relación se almacena toda la información referente a los inventarios de los libros, es decir, toda la información que detecta y controla cada ejemplar adquirido por las bibliotecas de la UNAM; y es por medio de ella que podemos obtener todo tipo de informes y estadísticas acerca de los libros comprados y/o contenidos en una biblioteca específica.

Para poder registrar en la relación inventario la información, se presentan varios tipos de situaciones, las cuales veremos con detalle en la sección 4.4 de este capítulo.

Aquí solamente daremos una explicación general de los procesos para dar de alta en esta relación la información que se refiere a una adquisición nueva (compra o donación) hecha por una biblioteca ya que está involucrada con el procesamiento general de un libro en el sistema LIBRUNAM.

El registro de este tipo de información se divide en dos formas de llevarlo a cabo dependiendo de la obra de que se trate, a saber: 1) alta de un título de nuevo ingreso al banco de datos, y 2) cargo que se hace de uno o más volúmenes adquiridos por una biblioteca a un libro que ya con anterioridad está registrado en el banco de datos.

En ambos casos, la asignación del número de adquisición que corresponde a cada volumen para su registro en esta relación se hace en forma automática, y el número lo obtiene el sistema de la relación *tablainv*, donde guarda en el atributo *numadq* el último número de adquisición asignado al último ejemplar adquirido por la biblioteca en actualización.

Cuando es un proceso de alta de una obra nueva para el banco de datos, en el paso previo al ingreso de la información a las otras relaciones del sistema, se separan los datos generales de la ficha que se refieren a su inventario, como son el número de matriz asignado a ella, el número de volúmenes a afectar, y el número de la dependencia que hace la adquisición de las obras.

Si se trata de un título que ya está registrado en el banco, entonces sólo se efectúa el proceso de cargo de los datos mencionados anteriormente, con el programa de investigación en línea de LIBRUNAM.

La estructura de la relación *inventario* para cada tupla es la siguiente:

CAMPO	NOMBRE DEL ATRIBUTO	TIPO	LONGITUD
1	dep	i2	2
2	numadq	i4	4
3	matriz	i4	4
4	fecha	uc	4
5	tipo	i1	1
6	colec	uc	1

**dep** - es el número de la dependencia, asignado consecutivamente a una biblioteca a su ingreso al sistema y que se utiliza para detectarla dentro de la base de datos; es decir, para control interno dentro de la DGB. A la fecha es un número menor a 350.

**numadq** - es el número de adquisición asignado automáticamente y en forma consecutiva a cada uno de los volúmenes que va adquiriendo la biblioteca; y es único para cada ejemplar dentro de la biblioteca.

**matriz** - es el número único consecutivo, asignado a cada ficha catalográfica nueva que ingresa al banco de datos del sistema LIBRUNAM, y cuyo valor a la fecha, no puede ser mayor al número 999,999.

**fecha** - es la fecha en que se graba la información dentro de esta relación. Está asentada en una cadena de 4 caracteres con un formato mensual como sigue: AAMM donde AA corresponde al año y MM al mes en el que se hace el registro.

**tipo** - es el tipo de proceso por medio del cual se dio de alta en esta relación la información (ver sección 4.4 de este capítulo). Este atributo sirve para detectar si el libro corresponde a un cargo, a un alta de un título nuevo, a un alta por medio de los procesos de inventario, o si es un volumen en espera de que se le procesen etiquetas, etc. .

**colec** - es un campo auxiliar que nos sirve para detectar el estado de la información, por ejemplo, en el caso de la Biblioteca Central es donde se registra en qué piso se localiza físicamente el libro; en la Facultad de Ciencias, se registra en cuál de sus bibliotecas se encuentra el libro. Si está en alguno de los procesos de un inventario se registran ciertas características de la información, etc. .

Esta relación tiene una llave primaria en el atributo **matriz** que sí permite duplicidad, ya que un título puede amparar varios volúmenes), y es por la que está físicamente ordenada (clustered index sobre el campo matriz.)

Esta llave funciona principalmente como llave foránea, ya que que nos relaciona con la relación *fichas* para obtener los datos completos de cada libro; la razón de hacer de esta llave la primaria es por su funcionalidad, pues el procedimiento de acceso se hace por medio de la ficha catalográfica cuando se está trabajando con las altas y los cargos.

Sin embargo, para los procesos específicos del control del inventario de la biblioteca, la llave más utilizada y pudiera decirse que la principal para este tipo de procesos, es la llave combinada sobre los campos **dep** y **numadq**, cuya combinación no permite duplicidad en el banco de datos (unique nonclustered index sobre los atributos (dep,numadq) ) asegurando con ello la integridad de la información.

Existen otras dos llaves (terciaria y cuaternaria) que son menos utilizadas y que permiten duplicidad, mismas que se emplean para procesos de apoyo a los inventarios y que sí son importantes para este tipo de trabajos.

Los procesos que involucran el manejo de estas llaves nos proporcionan productos tan fundamentales como son la generación de: etiquetas, de listados de estadísticas, de relaciones resultado para todo tipo de apoyo a las distintas etapas de los procesos de inventario, etc..

Estas dos llaves son también combinadas y están definidas una sobre los atributos (**tipo,colec**) y otra sobre los atributos (**dep,colec**).

Existe otra llave combinada sobre los atributos (**tipo,fecha**), creada periódicamente en forma temporal (una vez por año o cada dos) para apoyo de los procesos estadísticos que se efectúan en base a los libros adquiridos dentro del sistema bibliotecario.

#### **4.2.5 Esquema general del Sistema LIBRUNAM**

A continuación se muestra un esquema de la estructura general de la base de datos **libros** y las ligas entre las relaciones que lo componen. (Figura 4.1)

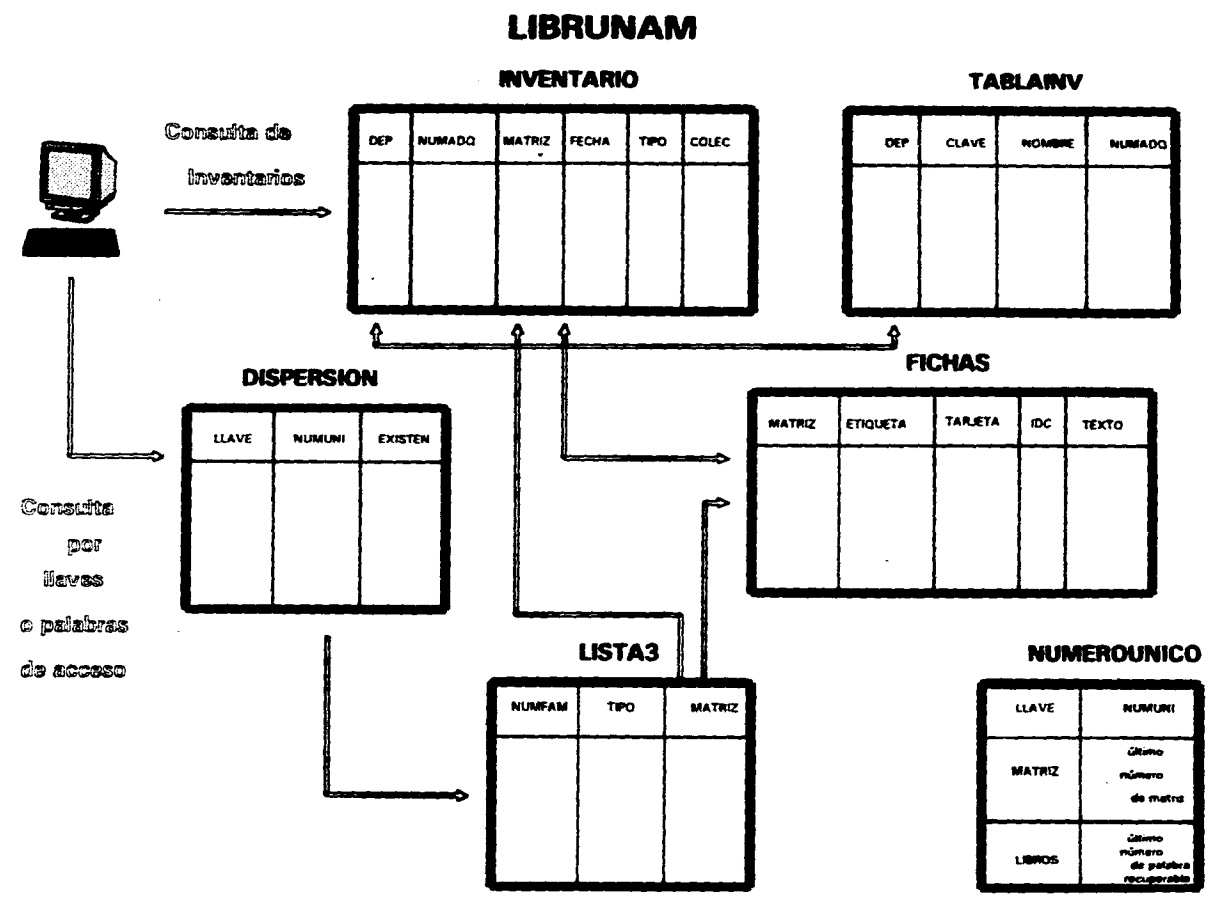


FIGURA 4.1

### 4.3 PROCEDIMIENTOS GENERALES EN LIBRUNAM

#### 4.3.1 Captura y procesos de validación de la información

Después de codificar la información, ésta es capturada en el Departamento de Producción de la Subdirección de Informática de la DGB, donde se vierten los datos en archivos que se graban en una de las computadoras anfitrionas (Alpha-Micro) de la IDM y posteriormente se procesan para su alta en la base de datos "libros" de la Britton-Lee.

El programa que se utiliza para la captura de esta información es muy amigable y de fácil uso, de tal forma que va guiando al capturista durante este proceso, además de efectuar las validaciones sencillas de corrección inmediata y que no requieren de un criterio técnico para ello, ya que este tipo de errores deberán arreglarlos los especialistas en la codificación de las fichas.

El formato que sigue el registro de cada libro en los archivos de captura para su proceso es el que sigue:

```
... información ... $ETIQUETA ... información ... -  
$ETIQUETA ... información ... ●
```

donde la 1a. información pertenece a la etiqueta 008 con una posición definida para cada campo fijo, y las posteriores etiquetas, como son de longitud variable, finalizan hasta que se presenta el signo de pesos "\$" seguido de otra etiqueta o el signo de arroba "@" que indica a la computadora que finalizan los datos correspondientes a una ficha.

Antes de ejecutar el alta, se realiza un proceso que filtra la información contenida en estos archivos de captura a fin de revisar y validar los datos en su construcción y sintaxis, y de eliminar duplicados; de tal forma que se puedan suprimir los errores que no corresponden a los de digitación; ya que en este tipo de fallas la máquina no puede detectar si existe una equivocación o no; como por ejemplo, un título o un autor mal escritos, o cualquier otro tipo de error similar, garantizando con esto que la información sólo presente errores ortográficos o de concepto cuya corrección se efectúa posteriormente por medio del Sistema de Depuración de LIBRUNAM.

Parte de este filtro como ya se mencionó, se realiza a la hora de la captura en las fallas de verificación inmediata; y parte con un programa especialmente diseñado para ello que revisa etiqueta por etiqueta, y que además empieza a preparar la información para los procesos de alta en el banco.

Si se presentan erratas, se generan archivos con los errores detectados y detallados claramente; los cuales se imprimen para que posteriormente sean corregidos por personal del Departamento Técnico de la DGB y se vuelvan a procesar las veces que sea necesario hasta que ya no haya más errores.

En el procedimiento que sigue la información capturada, la dependencia (etiqueta 900) y los números de adquisición (etiqueta 901) llevan un proceso especial que los separa de la información propia de la ficha bibliográfica, y se adecúan para ser dados de alta en relación inventario de la base libros (ver sección 4.4 .)

#### **4.3.2 Procesamiento y almacenamiento de la información bibliográfica en la base de datos "libros"**

Como ya se dijo, el diseño se llevó a cabo en una base de datos relacional, donde se establece la independencia de los datos eliminando las redundancias innecesarias y donde se asegura la consistencia de la base de datos.

Una vez capturada la información de los títulos nuevos que deben ingresar al banco de datos se preparan los procesos para validarla y liberarla de errores; y posteriormente se ingresa en el banco de datos, separando y procesando la información dependiendo del tipo de dato para su alta en las distintas relaciones que conforman la base de datos.

Como una anotación final, se aclara que en las formas de codificación no se dan, ni el número de matriz que le corresponde a la ficha, ni los números de adquisición de los volúmenes que ampara ese título cuando son de asignación automática; y esto es debido a que en los procesos de alta de la información de cada nueva matriz, el sistema obtiene de la base de datos, el último número de matriz almacenado y lo incrementa automáticamente para asignárselo a la ficha nueva que se está procesando y posteriormente almacenarlo.

Lo mismo sucede con los números de adquisición de asignación automática, en cuyo caso sólo se codifica el número de volúmenes a procesar.

#### **4.3.3 Procesos para el uso en línea del banco de datos LIBRUNAM (generalidades)**

La recuperación en línea es el proceso más utilizado dentro del sistema LIBRUNAM.

En este documento sólo nos interesa hacer una descripción en forma muy general del mismo, ya que sirve de apoyo para la investigación de los libros en inventario y

para una de las formas que utiliza el usuario para dar de alta la información en los inventarios de las bibliotecas de la UNAM (explicación en la sección siguiente.)

En el diseño de este proceso se tuvo la finalidad de hacerlo muy flexible, accesible y fácil de manejar para el usuario en general, logrando con esto el aligerarle la tarea de la investigación del material bibliográfico universitario.

Este proceso permite la búsqueda de las fichas catalográficas almacenadas en el banco de datos, desde una terminal remota, la que está conectada a la computadora Alpha-Micro para su interfaz con Britton-Lee.

Funciona de la siguiente manera:

i) El usuario emite las instrucciones de búsqueda desde una terminal.

ii) El sistema filtra y valida estas instrucciones y en caso de error regresa un mensaje para que el usuario pueda rectificarlas.

iii) Cuando las instrucciones están correctas, ejecuta las rutinas de búsqueda haciendo la interfaz con la base de datos.

iv) Regresa al usuario los resultados de su búsqueda.

El procedimiento que debe seguir el usuario para hacer la recuperación de una obra deseada es el que sigue:

1) El sistema solicita la clave del usuario, la cual verifica sea válida. Cuando una clave de usuario ya fue aceptada por el sistema como válida, y el usuario decide cambiar de clave sin terminar la sesión, puede hacerlo ingresando la nueva clave al teclear **USU** .

2) El sistema coloca el cursor con el mensaje **BUSCO>** lo cual indica al usuario que está esperando sus instrucciones para ejecutar la búsqueda de libros.

3) El usuario teclea los datos para hacer la recuperación de las fichas catalográficas a través de cualquiera de las llaves de acceso.

4) Si el sistema detecta algún error en sus instrucciones, responde con un mensaje descriptivo para que el usuario lo pueda corregir.

En caso contrario responde con el número de fichas que cumplen con su solicitud aunado a la lista de las fichas



recuperadas, a fin de que el usuario las enliste en el orden que desee, ya sea en forma secuencial (oprimiendo la tecla return o enter) o aleatoriamente (solicitando el número que tiene la ficha escogida dentro de la lista que el sistema está mostrando.)

Cuando el usuario, al responderle el sistema, se da cuenta de que es muy larga la lista de fichas que cumplieron con los requisitos de su búsqueda, puede agregar más instrucciones para afinar más la recuperación. Estas instrucciones que se añaden a la lista deben de ir precedidas de un signo positivo "+" y adyacente a la llave de búsqueda. Concretamente, esto se ejecuta cuando el sistema responde: EXISTEN n FICHAS, LISTAS O AGREGAS, y entonces el usuario deberá teclear +\$LLAVE datos para afinar más la lista de las fichas encontradas.

5) El usuario puede decidir salirse de la lista tecleando el signo "=" o la palabra "otra" .

6) Como una información extra de ayuda, cuando el usuario encuentra la ficha deseada, puede solicitar la lista de las claves de las dependencias donde físicamente se localiza dicha obra, tecleando el comando DEP y con la ficha catalográfica desplegada en pantalla, y enseguida el sistema mostrará esta lista.

Si el usuario desconoce las claves de las bibliotecas, puede utilizar el comando UDE clave-de-la-dependencia para averiguar a qué bibliotecas se refieren dichas claves.

7) Para terminar la sesión de trabajo, el usuario deberá teclear FIN .

La forma de ingresar los datos de búsqueda es digitando el signo de pesos e inmediatamente después la llave de búsqueda, dejando enseguida un espacio en blanco y luego los datos de la información que se desea obtener dentro de esa llave, ya sea una palabra o una cadena de palabras, y finalmente oprimiendo la tecla de emisión a la computadora (enter o return o intro.)

Si el usuario desea obtener la ficha catalográfica de un libro que incluye indagar en llaves distintas, podrá en una sola emisión meter estos datos en forma secuencial, es decir, en una sola ejecución todas las llaves de búsqueda cada una seguida de sus datos, siempre tecleando el signo de pesos previo a cada cambio de etiqueta.

Para hacer la búsqueda más eficiente y rápida, el usuario deberá tratar de ingresar los datos de forma que no generalice mucho en su investigación, es decir, intentando reducir el universo de búsqueda a la máquina, para que internamente no maneje una gran lista de fichas

catalográficas a la hora de hacer las operaciones y cálculos algebraicos.

De igual forma, resulta también inconveniente manejar mucha información en línea ya que esto puede acaparar recursos de la máquina, haciendo que los procesos de los demás usuarios además del propio se vuelvan lentos.

Siempre se eliminan en estas rutinas de búsqueda los artículos, pronombres, conjunciones además de intentar no meter palabras muy generales.

Por ejemplo si se desea una obra donde el tema sea México, el usuario deberá tratar de introducir más elementos de búsqueda, ya que de lo contrario dejará a la máquina que maneje una lista muy larga de información; y si ya se mandó ejecutar el proceso pesado, entonces tendrá que utilizar el signo "+" para reducir la lista de fichas recuperadas y no hacer también pesado el despliegue de éstas en pantalla, además del tiempo y del espacio que se desperdicia en ello.

El usuario podrá introducir en sus rutinas, tantas llaves como requiera para hacer la búsqueda, sólo deberá tener cuidado de teclearlas siempre precedidas por el signo de pesos.

Las llaves o palabras recuperables en el proceso de búsqueda en línea se obtienen de los siguientes campos:

Número de matriz (etiqueta 035)

Clasificación (etiqueta 050)

Autor (cualquiera que sus tipos - asiento principal -  
etiquetas 100s)

Título (etiquetas 240 y 245)

Pie de imprenta (etiqueta 260)

Serie (etiqueta 430)

ISBN (etiqueta 020)

Tema (encabezamiento de materia, etiquetas 600s)

Asientos secundarios (etiquetas 700s)

A continuación se describen los nombres mnemotécnicos que tiene el sistema LIBRUNAM para facilitar el uso de las llaves de acceso; o en su defecto las etiquetas que también se pueden emplear:

Para la clasificación: \$CLA o \$050

Para el autor: \$AUT o \$100 o \$110 o \$111  
o \$130 o \$700 o \$710  
o \$711 o \$730

Para el título: \$TIT o \$240 o \$245 o \$740

Para el pie de imprenta: \$EDI o \$260

Para la serie: \$SER o \$430

Para el ISBN: \$ISB o \$020

Para los encabezamientos de materia, es decir, para los temas: \$STEM o \$600 o \$610 o \$611  
o \$630 o \$650

Para el número de matriz: \$MAT o \$035

Para obtener en línea los número(s) de adquisición que tiene una obra se mandan ejecutar procesos especiales por el encargado autorizado, ya que no se puede dejar como consulta libre porque hay libros que comprenden muchos volúmenes y el accederlos puede hacer más lento el sistema, además de que su consulta sólo puede ser solicitada para los procesos de inventario.

En la siguiente página se da un ejemplo de una recuperación donde una de las etiquetas tiene una lista de más de 170 fichas y al hacer la intersección con la lista de la otra etiqueta se recuperan sólo dos fichas:

DATOS : \$aut knuth \$tem computacion

Z253  
.4T47  
K58

Knuth, Donald Ervin, 1938-  
The texbook / Donald E. Knuth ; il. by Duane Bibby.  
-- Reading, Massachusetts ; Mexico : Addison Wesley, c1986.  
483 p. -- (Computers & typesetting ; A)  
ISBN 0-201-13447-0

1. Matematicas - Impresion 2. TeX (Sistema de computacion)  
3. Tipos, Composicion de (Calculadoras electricas) I. Bibby,  
Duane, il. II.T.

NUC 390528-4

DATOS : \$aut knuth \$tem computacion

QA39  
.2  
G734

Graham, Ronald Lewis, 1935-  
Concrete mathematics / Ronald L. Graham, Donald E. Knuth, Oren Patashnik. -- Reading, Massachusetts : Addison-Wesley, c1989.  
xiii, 625 p  
ISBN 0-201-14236-8

1. Matematicas - 1961- 2. Computacion electronica de datos -  
Matematicas I. Patashnik, Oren, coaut. II. Knuth, Donald  
Ervin, 1938- , coaut. III.T.

NUC 464956-7

#### **4.4 ACTUALIZACION DE LOS INVENTARIOS**

**\*\* Daniel J. Boorstin de la Biblioteca del Congreso de Estados Unidos en Washington, en relación con el inventario de los libros afirma que se necesita determinar la cantidad de material desaparecido, el que ha sido colocado incorrectamente, el que necesita reclasificarse, y el que está en tratamiento de preservación necesaria. Los resultados mejorarán en gran medida el servicio que se ofrece a los usuarios, ayudarán a descartar el material obsoleto y permitirán el control continuo del mismo, esto es válido para toda biblioteca y cuando se trabaja para un sistema de bibliotecas, además se debe considerar la existencia y ubicación del acervo. \*\* 1**

Con este preámbulo se describe muy acertadamente cuán importante resulta el efectuar un inventario en una biblioteca, sobre todo si se habla de una que pertenezca a un sistema bibliotecario.

Como consecuencia, los procedimientos a seguir para llevar a cabo un inventario en una biblioteca, y para además, resolver todos los problemas que se puedan presentar en su ejecución, adquieren una importancia vital; luego entonces estos procedimientos son los que nos van a facilitar la tarea para lograr el objetivo final, que es controlar el acervo de la biblioteca y así poder otorgar un mejor y más presto servicio, tanto a nivel particular (para cada biblioteca) como a nivel general (para todo el conjunto bibliotecario.)

##### **4.4.1 Descripción del proceso de registro de Inventarios**

En esta sección se va describir el proceso de registro de los inventarios y se va a detallar cómo se van controlando con el sistema INVENTARIOS dentro del sistema bibliotecario de la UNAM, y entre éstos, cuáles son las distintas opciones que se pueden ir presentando.

1. Esta cita fue encontrada entre los documentos del archivo del Departamento de Sistemas de la DGB., y aunque carece de referencia bibliográfica considero pertinente la inclusión de ésta con el propósito de ilustrar la importancia de llevar a cabo los inventarios de los libros.

En el control de los inventarios del sistema LIBRUNAM, existen dos procedimientos generales para que el sistema trabaje la información:

- 1) Altas de la información
- 2) Actualización de la información

Dentro de los procesos de altas de la información, se presentan dos formas de hacer el almacenamiento:

- Altas en los inventarios a partir de los procesos normales de los libros.
- Altas en los inventarios a partir de los procesos de inventario llevados a cabo en una biblioteca.

Dentro de los procesos de inventario para la actualización de la información, además de las altas, se trabajan:

- Cambios en la información de los inventarios
- Bajas de la información de los inventarios

A continuación se muestra un diagrama del control de los inventarios:

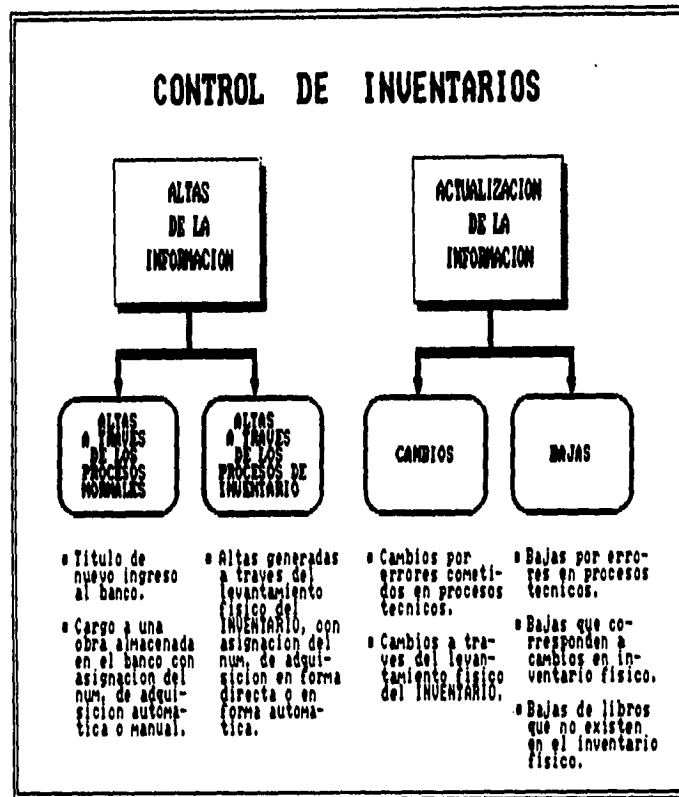


FIGURA 4.2

En las subsecuentes secciones se describe detalladamente cómo el sistema trabaja cada uno de estos procesos.

#### 4.4.1.1 Altas de la información

Como ya se dijo, el sistema tiene principalmente dos formas para dar de alta los nuevos registros que corresponden a los inventarios de las bibliotecas; los que finalmente se ingresan directamente en la relación *inventarios* de la base de datos *libros*.

Con la declaración del índice (dep,numadq) sobre esta relación, se tiene perfectamente controlada la unicidad de la información, y cualquier duplicado es inmediatamente rechazado por el sistema.

Esto permite llevar un estricto control sobre los números de adquisición que corresponden a cada biblioteca y que en definitiva son el indicador principal de su inventario.

**Altas en los inventarios a partir de los procesos normales de los libros.**

Este tipo de altas se derivan de los procesos técnicos que se llevan a cabo con todos los libros adquiridos por las bibliotecas de la UNAM.

Cuando se trata de una obra que ingresa por primera vez al banco de datos, al codificarse sólo se especifica el número de volúmenes que ampara esa ficha catalográfica, y posteriormente al ser procesada, el sistema le asigna el número de matriz (según el último que se tiene guardado en la relación *numerounico*), y el número de adquisición que le corresponde, que el sistema designa automática y consecutivamente a cada volumen que entra al acervo de la biblioteca que hace la adquisición (a partir del último que tiene almacenado en la relación *tablainv*.)

A este procedimiento se le conoce como *proceso original de una obra*.

Se hace un registro en la relación *inventario* por cada volumen, con la fecha en que se lleva a cabo esto, y al campo *tipo* se le asigna el número 1, lo que nos va a indicar cuando accedemos el tuple, que su ingreso correspondió al de un alta de un proceso original.

Por otro lado, cuando el localizador hace la búsqueda en línea y encuentra que se trata de una obra cuya ficha ya está almacenada en el banco, sólo tiene que hacer un cargo de los volúmenes que se le están procesando a la biblioteca en turno.

Aquí, el sistema tiene mucho control sobre los usuarios y sus permisos para autorizarles el acceso a la información de la base de datos, ya que está trabajando en línea directa con ella.

El procedimiento es como sigue:

El localizador hace la búsqueda en línea según los datos contenidos en la portada del libro que se está investigando, determina cuál ficha de la lista recuperada es la que corresponde a la portada que está trabajando, y ejecuta el cargo, para lo que necesariamente se tiene presente en la pantalla de la terminal la ficha del libro.



El sistema por medio de las llaves de búsqueda recupera cada ficha a través de acceder primero la relación que contiene las llaves del banco **dispersión**, y posteriormente hacer la liga con las listas en la relación **lista3**, todo esto con ayuda de las operaciones del álgebra relacional correspondientes al tipo de búsqueda que se hace Unión, Intersección, Diferencia, etc. mismas que se ejecutan con IDL.

Con lo anterior, se obtienen los números de matriz que permiten obtener de la relación **fichas**, las fichas catalográficas que cumplen con las condiciones de búsqueda; para finalmente mostrar en la terminal, el contenido completo de cada una de ellas.

El localizador revisa en pantalla cada una de las fichas recuperadas, y cuando determina cuál es la que se ajusta más a sus necesidades, ejecuta el proceso de cargo.

El localizador debe decidir si de este cargo quiere generar un juego de tarjetas de cartón o no, dependiendo de la biblioteca que esté trabajando; aunque generalmente sí se solicitan, para lo que utiliza el siguiente formato de comando:

**Comando Clave-de-la-dependencia Número-de-volúmenes**

■ Donde **Comando** puede ser:

JUE - Se utiliza cuando se desea hacer el cargo y generar juegos de tarjetas (por autor, título, tema, clasificación.)

CAR - Se utiliza cuando se desea hacer solamente el cargo sin generar juegos de tarjetas.

Nota: existen dentro del sistema otros tipos de comandos, que no se tratan aquí porque no afectan al sistema de inventarios.

■ Donde **Clave-de-la-dependencia** es la clave de la biblioteca asignada por la DGB, que la identifica dentro del sistema bibliotecario de la UNAM.

■ Y donde **Número-de-volúmenes** es el total de volúmenes que está adquiriendo la biblioteca sobre ese título.

Nota: el sistema no permite hacer cargos de un número de volúmenes mayor a 14 en cada emisión del comando para evitar equivocaciones por parte del usuario, ya que la experiencia demuestra que se cometen muchos errores al teclear números, y una falla en agregar uno o más dígitos puede afectar gravemente el control del inventario de la biblioteca al aumentar a su base decenas o centenas de

registros basura; además de que el hacer luego la corrección representa un trabajo extra innecesario tanto para el Administrador de la Base de Datos como para el encargado de estos procesos y aún hasta para el encargado del control del inventario de la biblioteca; por lo que si es necesario, se tendrá que mandar el cargo tantas veces como se requiera.

Este proceso, al igual que el alta de un proceso original, hace un registro en la relación **inventario** por cada volumen con la **fecha** en que se lleva a cabo; aunque con la diferencia de que toma el número de matriz a partir de la ficha en pantalla sobre la que se hace el cargo, y al campo **tipo** se le asigna el número 2 ó 5 (dependiendo de la persona que lo ejecuta), o el número 7 ó 9 (utilizados para controles dentro de la sección de inventarios exclusivamente para Biblioteca Central), lo que nos va a indicar cuando accedemos el tuple, que su ingreso correspondió al de un alta por medio de un cargo.

En este caso, el campo **coleccion** solamente se utiliza si el cargo se efectúa por la sección de inventarios para Biblioteca Central, esto es, porque sólo en esta biblioteca se trabajan en línea los procesos de los inventarios.

Durante este procedimiento de alta, el sistema ejecuta varios controles sobre la información para consolidarla y asegurar su integridad; esto es, cuando el usuario emite el cargo, el sistema bloquea el acceso a la información **exclusivamente de la dependencia**, tanto en la relación **tblainv** como en la relación **inventario**, y no deja que otros usuarios la accedan hasta que haya efectuado el cambio en el contador de la relación de las dependencias y la haya dado de alta en la relación de los inventarios.

Esto lo logra valiéndose de la opción **Begin transaction** y **end transaction** de IDL, que se utiliza con comandos almacenados y/o también con lenguaje de programación de interfaz, de tal suerte que no hace los cambios en la base de datos de los comandos incluidos dentro del begin-end, hasta que no los haya concluido satisfactoriamente; y en el caso de que se presente una interrupción, aborta los que llevaba hasta el momento y vuelve a dejar intacta la información. También ofrece la ventaja de que si otro usuario intenta acceder esos mismos tuples, le emite un mensaje de aborto y le informa a este segundo usuario que debe repetir la operación.

Lo mismo sucede si dos usuarios hacen el mismo intento a la vez, en cuyo caso el sistema solicita a ambos que reiteren la operación.

Al término de la ejecución de estos procesos, el sistema como respuesta a la persona que transmitió el comando, muestra en la pantalla los números de adquisición

que automáticamente fueron asignados a los volúmenes que cargó, para que pueda tomar nota de ellos para su control.

Finalmente diremos que en las altas a partir de procesos normales, se puede presentar un caso especial tanto en las que corresponden a una obra de nuevo ingreso al banco como las que son cargos; y éste es que, por una situación no controlada se hayan perdido números de adquisición dentro del banco; por ejemplo, cuando se transportó la información de un máquina a otra; o por equivocación, la persona que controla los cargos los haya borrado de la base al hacer una corrección, etc.; por lo que la asignación de los números de adquisición en el alta se tienen que hacer en forma manual. Si este caso se presenta y la biblioteca que manifiesta el problema no está efectuando procesos de inventario (que sería un medio para corregir estos errores), entonces, la persona que va a ingresar la información debe especificar los números de adquisición que quiere dar de alta; ya sea asentándolos en las hojas de codificación si se trata de una ficha que no esté almacenada en el banco de datos o, haciendo el cargo en línea con el número de adquisición definido, si el libro ya se encuentra registrado en el banco. En este punto, el que efectúa el cargo utiliza otro programa distinto al que usa para los cargos de asignación automática, donde la sintaxis del comando es idéntica a la que ya se vio (JUE o CAR), sólo que en lugar del número de volúmenes se escribe exactamente número de adquisición que se desea cargar; por lo que lógicamente se tiene que emitir el comando por cada número que se quiera dar de alta.

Durante este proceso, el sistema revisa que el número de adquisición que se intenta dar de alta no esté ya almacenado en el banco de datos, y si esto sucede, entonces el sistema regresa un mensaje de error y no lleva a cabo el registro.

#### **Altas en los inventarios a partir de los procesos de inventario llevados a cabo en una biblioteca.**

Los procesos que se describen a partir de esta etapa, están ya involucrados específicamente con el levantamiento de un inventario llevado a cabo por cualquier biblioteca de la UNAM.

En esta parte, dentro de las altas de información, vamos a tratar lo que respecta a las altas por medio de los procesos que involucran el levantamiento de un inventario.

Como ya se vio en el Capítulo III, la primera intervención del sistema dentro del inventario de una biblioteca es la elaboración de los listados con la información que tiene almacenada en el banco de datos.

Aquí, se corre un proceso que hace un vaciado de lo que tiene guardado dicha dependencia bibliotecaria en la relación inventarios de la base libros a un archivo en Alpha-Micro; compacta la información por el número de matriz para facilitar y agilizar su procesamiento, ya que el archivo donde quedó grabada la información tiene desglosado un registro por cada número de adquisición, y la trabaja obteniendo los datos generales de la ficha que corresponden a cada matriz (clasificación, autor, título), usando ésta como llave para acceder la información en la relación fichas; y con esto, finalmente se logran los listados con los datos que identifican a cada libro dentro de su acervo.

Estos listados, se generan en 4 ordenamientos diferentes: topográfico, por número de adquisición, por autor y por título; los cuales se vuelven a reformatear para mostrarse en forma desglosada (una línea por cada número de adquisición), con el fin de apoyar y facilitar la tarea al personal que va a realizar el cotejo físico.

A partir del trabajo que se lleve a cabo con estos listados y con la ayuda de búsqueda en disco compacto o en línea para cuando no tenga toda la información (como se vio en el Capítulo III), la dependencia finalmente obtiene los datos necesarios para dar de alta la información faltante en el inventario de la biblioteca.

Esta información debe ser vaciada en archivos o disquettes que puedan ser transportados en la DGB a la máquina anfitriona Alpha-Micro para su posterior proceso dentro de la IDM.

El formato que debe de tener esta información por cada registro es la siguiente:

CAMPO	CONTENIDO DEL CAMPO	TIPO	LONGITUD
1	núm. adquisición	Caracter	6
2	núm. matriz	Caracter	6
3	clasificación	Caracter	20

Donde deben ser obligatorios: el campo del número de adquisición; y, o el campo del número de matriz, o el campo de la clasificación (obligatoriamente cualquiera de los dos para poder identificar la ficha catalográfica.)

De preferencia se desea que se dé el número de la matriz, porque con el sistema de Depuración de LIBRUNAM

continuamente se están corrigiendo errores en las clasificaciones asentadas en el banco; lo que no siempre permite acceder una ficha si la llave de búsqueda ya no corresponde a la que está almacenada en LIBRUNAM; mas sin embargo, el número de matriz no cambia, además de que es la llave de acceso principal a la información de los libros; por lo que sólo se utiliza el campo de la clasificación en la eventualidad de que no se tenga el medio de obtener el número de la matriz correspondiente a la información que se quiere dar de alta.

Lo mismo sucede para el caso de que se tengan ambos datos, donde sólo se debe de capturar la matriz, ya que si se captura además la clasificación, el sistema realizará más verificaciones (revisa que exista la clasificación en el banco, y revisa que efectivamente corresponda a dicho número de matriz), y consecuentemente rechazará más información con errores; además de que teclear clasificaciones se presta a que se cometan más equivocaciones por ser combinaciones de números, letras y puntuación.

Ya teniendo en Alpha-Micro la información que se desea dar de alta, el coordinador de estos procesos deberá, como primer paso, investigar la clave y el número de dependencia que tiene asignados dentro del sistema LIBRUNAM la biblioteca que está realizando estos procesos. Esto se logra si se cuenta con un listado que contenga la información referente a las dependencias bibliotecarias o en su defecto, entrando directamente a la Britton-Lee a acceder la relación tablainv por medio de IDL.

Por otro lado, y para poder llevar un control de la evolución de los bloques de información que se le van trabajando a la biblioteca durante los procesos de su inventario, se hace la identificación de cada uno de ellos por medio del campo *coleccion*, asignando el encargado de estos procesos, una letra en orden progresivo para cada alta masiva que ejecute; es decir que, al primer archivo de datos (o bloque de información formado por varios archivos) que vaya(n) a ser procesado(s) a un mismo tiempo, se le(s) deberá asignar la colección "A"; de cuya información se generarán, al finalizar el alta, listados y etiquetas. Para una segunda corrida con otro bloque, se tendrá que asignar la subsiguiente letra (la letra "C" ya que la "B" es la que se utiliza para marcar los registros a dar de baja) y a partir de ella, obtener los listados y las etiquetas de la información procesada, y así sucesivamente; por lo que, es importante llevar un control del rango de colección en el que van los procesos de cada dependencia.

En caso de duda el responsable del sistema podrá investigar en la relación inventario de la base *libros* la última letra procesada a una biblioteca.

Ya teniendo esta información, el responsable podrá hacer el alta utilizando el comando **ALTDEP**; pero antes deberá asegurarse de registrarse en Alpha-Micro como el propietario de la base *libros* en Britton-Lee para que la máquina le permita hacer las modificaciones en la relación *inventario*; además de autorizarlo para poder cargar los archivos que contienen los comandos de ejecución de IDL, comandos almacenados en la base "*libros*" creados específicamente para los procesos de inventario.

Este comando en realidad es un archivo de trabajo, ya que Alpha-Micro permite mandar ejecutar una serie de acciones contenidas en un archivo de este tipo con solo teclear su nombre.

A este archivo, se le pasan los parámetros antes mencionados en la forma que se describe a continuación.

Uso del comando:

**ALTDEP Archivo-a-procesar Num-dep. Colección**

donde:

- *Archivo-a-procesar* es el archivo de datos que se transportó del diskette a la Alpha-Micro. El nombre no debe tener una longitud mayor de 6 caracteres ni una extensión mayor de 3 (es decir, no más de 3 caracteres después del punto.)
- *Num-dep* es el número de la dependencia.
- *Colección* es la letra progresiva que le corresponde al bloque según la explicación anteriormente dada.

Este archivo de comandos en la Alpha anfitriona carga automáticamente los comandos almacenados definidos en el archivo **BAJBC.IDL** en Alpha-Micro, comandos de ejecución dentro de IDL específicamente creados para las rutinas de inventario que serán utilizados durante los procesos que manda ejecutar el comando **ALTDEP**.

Posteriormente, dentro del mismo archivo de comandos y en forma automática, se corre el programa **ALTASU**, que es el que se encarga de leer los datos del *Archivo-a-procesar*, filtrándolos y detectando errores.

Durante el proceso de alta, el sistema verifica que el número de adquisición no esté ya almacenado en la relación *inventario* en el banco de esa dependencia; y en el caso de que esto suceda lo rechaza y lo graba en un archivo de

errores; lo mismo si el valor es nulo, o contiene basura, o excede al número máximo que se tiene guardado en la relación *tablainv*.

En el caso de que el número de adquisición esté asignado dentro del banco a otro número de matriz, en el listado de errores se especifica este otro número.

Con las matrices, como ya se mencionó, verifica que exista asignada a una ficha dentro de la relación *fichas*; o en su defecto que exista la clasificación, y que ésta le corresponda a una ficha bibliográfica dentro del banco.

En el caso de que la información traiga el número de matriz y la clasificación, el sistema además verifica que la clasificación corresponda a ese número de matriz dentro del banco.

El programa genera dos listados; uno con la información que se dio de alta en la base libros de Britton-Lee, llamado: **ARCH.ALT** y otro con los errores que no pasaron el filtro, llamado: **ARCH.ERA**, definiéndose en este último, los diferentes tipos de error que se suscitaron durante el proceso.

En ambos casos, **ARCH** es el nombre del Archivo-a-procesar sin su extensión.

De los tipos de error ya mencionados, se revela uno en particular que no representa realmente un equívoco, y éste es cuando se intenta dar de alta un número de adquisición que ya está referenciado en el banco a la misma matriz que se le está asignando en el proceso, pero igualmente es rechazado ya que para el sistema representa un número de adquisición duplicado.

Por esto, y a fin de apoyar al usuario, al archivo **ARCH.ERA** se le aplica otro proceso donde se separan los errores que ya no proporcionan más información como son: los que presentan problemas en la clasificación, los que presentan problemas en los números de matrices (de fichas no existentes), los que presentan falta de información, etc., mismos que se quedan grabados en un archivo con nombre **ARCH.ER2** donde **ARCH** es el nombre del archivo a procesar ya mencionado; y por otro lado se separan y procesan de nuevo los errores que pueden aportar más datos que ayuden al usuario a corregirlos; uno de éstos es el caso que ya se vió del cargo a la misma matriz, en el que el sistema sólo marca dentro de la base la colección que le corresponde (misma que se pasa como parámetro a la hora de mandar ejecutar el proceso) y que sirve para detectar la información dentro de la base a fin de que se tenga el control del inventario en esa información. Otro caso es el de los números de

adquisición que ya están asignados a otra matriz en el banco de datos, para los cuales el sistema genera un nuevo listado con nombre **ERANum-dep.DAT** (donde Num-dep es el número de la dependencia) y en el que se detallan el número de adquisición, y la clasificación junto con el título tanto de la matriz que se intenta dar de alta como de la que se tiene asignada dentro del banco de datos a ese número.

Este proceso lo manda ejecutar el responsable del sistema utilizando el comando **ALTERR**; asegurándose para ello de registrarse como propietario de la base *libros* para que la máquina le permita hacer las modificaciones en la relación inventario.

Al igual que **ALTDEP**, este comando es un archivo de trabajo al cual se le pasan los parámetros como sigue:

Uso del comando:

**ALTERR Archivo-a-procesar Num-dep. Colección**

donde:

- *Archivo-a-procesar* es el archivo de errores **ARCH.ERA** generado en la Alpha-Micro con el comando **ALTDEP**.
- *Num-dep* es el número de la dependencia.
- *Colección* es la letra progresiva que le corresponde al bloque según el control del encargado.

Por último diremos que al igual que un alta de un proceso normal, se hace un registro en la relación *inventario* por cada volumen con la fecha en que se lleva a cabo y se asigna al campo *tipo* el número 3, lo que nos va a indicar cuando accedemos el tuple, que su ingreso correspondió al de un alta a partir de un proceso de inventario efectuado en una biblioteca. Como ya se dijo, el campo *colección* contendrá la letra consecutiva que le corresponde al proceso dentro de los que se han llevado a cabo por medio del sistema a la biblioteca en cuestión.

Todo lo que se ha hablado hasta este punto sobre las altas derivadas de los procesos de levantamiento de un inventario dentro de una biblioteca, ha dado por hecho que a partir de las investigaciones se obtienen el número de adquisición y el número de matriz (o la clasificación) que le corresponde al volumen a dar de alta en la base de datos. Existe otro caso, como ya se habló en el Capítulo III, en donde se desconoce el número de adquisición, en cuyo caso hay que asignarle uno en forma automática.



**Altas en los inventarios con asignación automática de números de adquisición a partir de los procesos de inventario llevados a cabo en una biblioteca.**

Como un módulo dentro de las altas en los inventarios derivadas de los procesos de inventario, se tienen las altas de volúmenes con asignación automática de su número de adquisición.

Estos casos se presentan cuando:

1) Dos libros dentro del acervo tienen asignado el mismo número de adquisición, por lo que al que no esté en la lista se le debe de asignar uno nuevo.

2) Aparecen dentro del acervo libros sin números de adquisición, y no existe dentro de la lista ninguna indicación para deducir su número, por lo que también hay que hacerles esta asignación.

Cuando el usuario se halla en esta situación debe entonces investigar en línea o por medio de los discos compactos de LIBRUNAM, el número de matriz que le corresponde al título de la obra en cuestión; y en el caso de que no encuentre la ficha almacenada en el banco de datos, deberá pasar los datos a la Dirección General de Bibliotecas para su proceso original y la asignación del número de adquisición será efectuada por este medio (como se menciona en la primera parte de esta sección.)

Cuando sí obtiene del banco el número de matriz, deberá vaciar esta información en archivos o disquettes que puedan ser transportados en la DGB a la máquina anfitriona que tiene conectada Britton-Lee.

El formato que debe tener esta información por cada registro es la siguiente:

CAMPO	CONTENIDO DEL CAMPO	TIPO	LONGITUD
1	núm. matriz	Character	6
2	núm. volúms.	Character	3

donde deben ser obligatorios los dos campos, y donde núm. vols. se refiere al número de volúmenes que pertenecen a la misma obra y a los cuales se les quiere asignar automática y consecutivamente su número de adquisición.

Se hace notar que no debe pasar de 999 ya que es casi imposible que existan más de 999 volúmenes de un mismo título en cualquier biblioteca.

Ya teniendo en Alpha-Micro esta información el coordinador de estos procesos, ejecuta un programa que valida los datos y los desdobra, dejándo un número consecutivo para cada registro, cada uno con un número de adquisición y su correspondiente número de matriz. Al igual que en las altas de libros de los que se conoce el número de adquisición, el encargado tiene que investigar la clave y el número de dependencia que tiene asignada la biblioteca para ejecutar los procesos de alta de esta información. De igual forma, debe de obtener de la base de datos el último número de adquisición que se le designó a la biblioteca para que a partir de él haga, por medio del programa, la asignación de la numeración consecutiva. Se hace notar que en este punto, el encargado debe de tener cuidado de que otros usuarios no intenten acceder esa dependencia mientras está llevando a cabo estos procesos, ya que va a cambiar el contador del último número de adquisición que sirve de base para controlar el acervo de la biblioteca.

Después de esto, da directamente de alta la información en la relación inventario de la base de datos libros, utilizando programas de utiliería de la Alpha-Micro.

Por último diremos que al igual que las altas de los procesos normales de inventario, se hace un registro en la relación inventario por cada volumen con la fecha en que se lleva a cabo y se asigna al campo tipo el número 3, lo que nos va a indicar cuando accedemos el tuple, que su ingreso correspondió al de un alta a partir de un proceso de inventario efectuado en una biblioteca. Como ya se dijo, el campo colección contendrá la letra consecutiva que le corresponde al proceso dentro de los que se han llevado a cabo por medio del sistema a la biblioteca en cuestión.

#### **4.4.1.2 Actualización de la información**

En esta parte se va a hablar de la actualización de la información; es decir, de los procesos que implican efectuar cambios exclusivamente en la información que controla los inventarios de las bibliotecas de la UNAM.

La información almacenada en el banco LIBRUNAM que corresponde a los inventarios puede sufrir reformas por dos situaciones: cuando se realizan correcciones en la base de datos por errores cometidos durante los procesos técnicos o, cuando se hacen cambios por procesos derivados del levantamiento de un inventario en una biblioteca universitaria.

En el primer caso, cuando el encargado de las búsquedas y cargos en línea detecta un error derivado de los procesos técnicos, hace uso de utilería especialmente diseñada para corregir estos errores, donde sólo efectúa bajas en cargos mal ejecutados; por lo que teóricamente no hablaríamos de actualización de la información que ya está almacenada, sino de corregir errores humanos de creación reciente.

En el segundo caso, sí se trata de actualizaciones propiamente derivadas de un levantamiento de inventario físico; las que se presentan cuando se llevan a cabo los siguientes procedimientos:

- Altas en la información de los inventarios
- Cambios en la información de los inventarios
- Bajas de la información de los inventarios

Estos procedimientos se derivan, como ya vimos en el Capítulo III, de trabajar los listados generados a partir de la información almacenada en LIBRUNAM que corresponde a la biblioteca que se está inventariando contra el acervo que físicamente se tiene en estantería; donde se detecta la información faltante en los listados, la que requiere de cambios, y la que está registrada en el banco mas no así en el acervo físico.

Se hace hincapié en que la información que se trabaja dentro de estos procedimientos, está almacenada en la relación inventario de la base de datos libros; donde la llave de acceso para llevarlos a cabo, es la llave combinada (dep,numadq) que define la unicidad de la información, y donde se almacena un registro por número de adquisición.

La información faltante en los listados de los libros que están en estantería se trabaja y se da de alta como ya se reseñó en la sección anterior (4.1.1); por lo que nos vamos a concretar a detallar a continuación cómo el sistema trabaja los procesos de cambios y bajas.

#### **Cambios en la información de los inventarios**

Los cambios en la información se presentan cuando la persona que está efectuando el inventario detecta que los datos de un libro dentro de la lista no corresponden con los del que se tiene en estantería; esto es, cuando la información almacenada en el banco de un número de adquisición está en desacuerdo con los datos del libro en el anaquel por estar asignado a una obra distinta con otro número de matriz.

Los cambios se manejan siempre como una baja de la información errónea registrada en el banco de datos y un alta de la información correcta; esto es debido a que no podemos dar de alta el número de adquisición con los datos modificados mientras en el banco exista registrado el incorrecto ya que el sistema no permite números de adquisición duplicados.

Sólo hay que tener cuidado de que el número que se desee dar de baja realmente no exista dentro de la colección en los anaqueles, ya que es muy frecuente que entre el acervo se repitan números de adquisición en libros distintos.

Cuando se presenta este problema, la persona encargada debe además hacer la búsqueda en los anaqueles del libro que está marcado en los listados con el mismo número de adquisición, a fin de detectar si ambos existen.

Cuando ambas obras existen, para no generar más trabajo se mantiene inalterable en la base de datos la información que se tiene en la lista (que es la que refleja lo que está almacenado en el banco), por lo que se capturan los datos del otro libro para darlo de alta asignándole en forma automática un nuevo número de adquisición (ver sección anterior.)

Resumiendo diremos que el cambio del número de adquisición que se requiere efectuar en un libro dado, llevan al localizador a capturar los datos erróneos en un archivo para su baja de la base de datos (como veremos en la sección siguiente), y capturar los correctos en otro archivo para su alta (como vimos en la sección anterior 4.1.1); ya sea que se le asigne específicamente un número de adquisición o que esto se lleve a cabo en forma automática por el sistema.

El encargado del sistema de Inventarios entonces siempre procesará primero los archivos con la información para dar de baja para poder ejecutar posteriormente las altas sin problemas.

El último caso que se presenta es cuando el volumen marcado en el listado no existe físicamente en el acervo. Este es el caso que vamos a ver a continuación.

#### **Bajas de la información de los inventarios**

El módulo de bajas en inventario, tiene como objetivo principal auxiliar en forma automatizada en la actualización que se hace dentro del banco de datos LIBRUNAM de la información que corresponde a cualquiera de las bibliotecas que pertenecen al sistema bibliotecario de la UNAM, proporcionando un apoyo en todos los procesos que se generan

durante el levantamiento físico del mismo; y refiriéndose concretamente, al de dar de baja toda aquella información de los libros que se encuentra detectada dentro del banco de datos y no así físicamente en estantería.

La información almacenada en el banco de datos LIBRUNAM que corresponde a los inventarios de las bibliotecas puede requerir el ser eliminada por tres circunstancias:

Para corregir errores cometidos durante los procesos técnicos; o, cuando se realizan cambios a partir de los procesos de inventarios; o finalmente, cuando se necesitan efectuar bajas de números de adquisición cuyos datos del libro están almacenados en el banco pero no así físicamente en la biblioteca.

En el primer caso, cuando el encargado de los procesos técnicos detecta un error cometido al cargar en el banco un número de adquisición equivocadamente, hace uso de programación especialmente diseñada para su corrección dando de baja el o los números erróneos; esto no se considera como una actualización de la información pero se menciona aquí porque de alguna forma afecta a la relación inventario.

El segundo y tercer casos, se derivan de llevar a cabo procesos de inventario en una biblioteca; y directamente de trabajar contra la estantería los listados que se generan de todo el acervo que tiene registrado en la base de datos.

Las obras que tienen que sufrir una baja por no encontrarse en estantería es debido a diversas causas, como por ejemplo: por haber sido robadas, por estar tan deterioradas que ya no pueden ser utilizadas, por ser obsoletas, etc. .

Cuando las personas que están llevando a cabo el inventario de una biblioteca deciden dar de baja un volumen, deben de estar ciertos de que efectivamente no exista el libro y deben agotar todas las investigaciones para asegurarse de ello. Por esto, a menos que se trate de un cambio, las bajas se podrán trabajar hasta que ya se haya hecho una verificación de todo el acervo, es decir hasta que ya se haya hecho la primera revisión completa del inventario físico, ya que puede darse el caso de que la obra haya sido mal colocada en estantería y por lo tanto esté perdida en los anaqueles de la biblioteca.

También debe el encargado de cerciorarse de que el libro no se halle en un proceso menor (encuadernado, etc.), en préstamo, o en cualquier otra situación similar.

Cuando los bibliotecarios ejecutores detectan libros a dar de baja, deben vaciar los datos de los libros en archivos o disquettes que puedan ser transportados en la DGB a la

máquina anfitriona Alpha-Micro para su posterior proceso dentro de la IDM.

Aquí, necesariamente se obtiene esta información de los listados, ya que ahí está registrado un reflejo de la información almacenada en el banco de datos; por lo que el encargado no tendrá ningún problema en obtener dicha información.

El bibliotecario debe tener cuidado de separar este archivo de los que está trabajando para dar de alta.

El formato que debe de tener la información grabada en el archivo por cada registro es la siguiente:

CAMPO	CONTENIDO DEL CAMPO	TIPO	LONGITUD
1	núm. adquisición	Character	6
2	núm. matriz	Character	6

donde deben ser obligatorios los dos campos.

La razón por la que se piden obligatoriamente estos dos datos, es porque si sólo se pidiera el número de adquisición, se prestaría a que, si el bibliotecario comete una equivocación al capturarlo y este número erróneo coincide con uno almacenado en el banco, entonces el sistema da de baja un número que sí es válido en el inventario de la biblioteca. Entonces, en estos procesos el sistema revisa:

i) Que el número de adquisición capturado para dar de baja, efectivamente esté almacenado en el banco en los inventarios de la biblioteca en cuestión

ii) Que el número de matriz que tiene asignado en la captura el número de adquisición a dar de baja, efectivamente corresponda al que está almacenado en la base de datos con ese número de adquisición.

iii) Que el campo colección no tenga ninguna letra asignada, ya que eso querría decir que ese registro ya pasó por un proceso de inventario y ya quedó marcado.

En el caso de que efectivamente se quisiera dar de baja un registro ya marcado por un proceso de inventario, entonces se pide verificación para poder efectuar la baja.

Cuando el Administrador de la Base de Datos del sistema ya tiene el archivo a procesar grabado en la Alpha-Micro, debe registrarse como propietario de la base libros en Britton-Lee para que la máquina le permita hacer las eliminaciones en la relación inventario; además de autorizarlo para poder cargar los archivos que contienen los comandos de ejecución de IDL (comandos almacenados específicamente creados para los procesos de baja de inventario.)

El encargado utiliza el comando **BAJDEP** para ejecutar estos procesos.

Este comando, al igual que **ALTDEP** y **ALTERR** en realidad es un archivo de trabajo que manda ejecutar una serie de acciones contenidas en él con solo teclear su nombre; y al cual se le pasan parámetros para poder efectuar los procesos de baja en la forma que se describe a continuación.

Uso del comando:

**BAJDEP Archivo-a-procesar Num-dep.**

donde:

- **Archivo-a-procesar** es el archivo de datos que se transportó del diskette a la Alpha-Micro. El nombre no debe de tener una longitud mayor de 6 caracteres ni una extensión mayor de 3 (es decir, no más de 3 caracteres después del punto.)
- **Num-dep** es el número de la dependencia.

Este comando carga automáticamente los comandos almacenados definidos en el archivo **BAJBC.IDL** en Alpha-Micro, comandos de ejecución dentro de IDL específicamente creados para las rutinas de inventario que serán utilizados durante los procesos que manda llamar el comando **BAJDEP**.

Posteriormente, dentro del mismo archivo de comandos y en forma automática, se corre el programa **BAJAUL**, que se encarga de leer los datos del **Archivo-a-procesar**, filtrándolos y detectando sus errores.

Durante el proceso de baja, el sistema verifica que el número de adquisición esté ya almacenado en la relación inventario en el banco de esa dependencia; y en el caso de que esto no suceda lo rechaza y lo graba en un archivo de errores; lo mismo si el valor es nulo, contiene basura, o excede al número máximo que se tiene guardado en la relación **tablainv**.

Con las matrices, como ya se mencionó, verifica que exista asignada a ese número de adquisición dentro de la relación inventario.

En el caso de que la información ya tenga una marca de proceso de inventario en el campo colección, se rechaza y se marca como error en un archivo de salida.

Si la información pasa todos los filtros, el campo colección queda marcado con la letra "B" que va a indicar que es un registro marcado para dar de baja.

El programa genera dos listados; uno con la información que se dio de baja en la base libros de Britton-Lee, llamado ARCH.BAJ y otro con los errores que no pasaron el filtro, llamado ARCH.ERB, definiéndose en este último, los diferentes tipos de error que se suscitaron durante el proceso.

En ambos casos, ARCH es el nombre del Archivo-a-procesar sin su extensión.

De los tipos de error ya mencionados, y a fin de apoyar al usuario, al archivo ARCH.ERB se le aplica otro proceso donde se separan los errores que ya no proporcionan más información como: cuando no existen los números de matrices, cuando falta información, cuando no existe el número de adquisición, etc.. El sistema deja estos errores grabados en un archivo con nombre ARCH.ERF (donde ARCH es el nombre del archivo arriba mencionado); y por otro lado separa y procesa de nuevo los errores que pueden aportar más datos que ayuden al usuario a corregirlos; uno de éstos es el caso en que se encuentran los que ya tienen marcada la colección de un proceso de inventario; otro caso es cuando la información del número de matriz no coincide con la que tiene en el banco. Para todos estos casos, el sistema genera un nuevo listado con nombre ERBNum-dep.DAT (donde Num-dep es el número de la dependencia), en el que se detallan el número de adquisición, y las clasificaciones con el título correspondiente tanto de la matriz que se intenta dar de baja como la que se tiene asignada dentro del banco de datos a ese número.

Este procedimiento lo manda ejecutar el responsable del sistema utilizando el comando BAJERR; asegurándose para ello de registrarse como propietario de la base libros, para que la máquina le permita hacer las modificaciones en la relación inventario. Al igual que BAJDEP, este comando es un archivo de trabajo al cual se le pasan los parámetros como sigue:

Uso del comando:

**BAJERR Archivo-a-procesar Num-dep.**

donde:



- *Archivo-a-procesar* es el archivo de errores ARCH.ERA generado en la Alpha-Micro con el comando ALTDEP.
- *Num-dep* es el número de la dependencia.

Como ya se dijo, el campo colección contendrá la letra "B" que define al registro para dar de baja.

Después de esto, el encargado crea en la Britton-Lee una relación temporal llamada *bajasdep* en la que se respaldan los registros que se van a dar de baja en la relación *inventario*, esto con el comando CREBAJ.DO, y de igual forma debe estar registrado el que ejecuta estos procesos como propietario de la base de datos *libros*.

Todo lo que se ha hablado hasta este punto sobre las bajas derivadas de los procesos de levantamiento de un inventario dentro de una biblioteca, ha dado por hecho que a partir de las investigaciones se obtienen el número de adquisición y el número de matriz (o la clasificación) que le corresponde al volumen a dar de baja en la base de datos.

Por último, se ejecuta el comando DEPBAJ, que efectúa la baja de los registros en la relación de los inventarios, y manda ejecutar el comando INVBAJ, el cual genera listados detallados del acervo dado de baja en el inventario similares a los generados a las altas (ver sección 4.4.2.)

#### **4.4.2 Generación de listados detallados del acervo bibliográfico dado de alta y de baja en el inventario.**

Cuando el encargado del sistema de Inventarios efectúa un proceso (o bloque de procesos) de alta o de baja, entonces posteriormente debe generar listados detallados de la información que procesó ya sean las altas que ejecutó, o las bajas que se hicieron en la base (y que se marcaron antes de definitivamente eliminarlas), a fin de que el(los) encargado(s) del inventario en la biblioteca verifique(n) si los resultados están correctos y tenga(n) un control de ello en un documento donde se pueda consultar cuando sea necesario.

Estos listados son similares a los que se entregan a la biblioteca para que los bibliotecarios comiencen su inventario contra el acervo en estantería, la diferencia estriba en que se generan a partir del campo colección, o mejor dicho, a partir del número de proceso que está marcado en el campo; por ejemplo si se dio de alta un bloque de información con la letra A, entonces se generan listados de sólo la información que esté marcada con esa letra.

La información que se muestra en estos listados por cada volumen es: la clasificación, su autor, su título, el número de matriz y el número de adquisición que le corresponden.

Para ello, el responsable del sistema deberá hacer uso del comando **INVEN**.

Uso del comando:

**INVEN** Clave-dep. Identif-proceso Num-dep. Colec.

donde:

- Clave-dep es la clave con la que se identifica la dependencia dentro del sistema LIBRUNAM.
- Identif-proceso es el nombre del proceso que le quiera dar el responsable a fin de que se identifique dentro de la bitácora de tiempos de corrida; esto es con el objeto de que, si falla la máquina o el sistema durante la ejecución de alguno de los pasos, se pueda identificar en cuál se quedó y retomar la continuación para su término; además de que, también le sirve para detectar tiempos consumidos y sacar estadísticas en cruce con el volumen de información procesada. El archivo que automáticamente el programa va grabando para bitácora, se denomina BOOTAL.LOG y su acceso se hace ya sea por medio de un type a este archivo (lectura secuencial) o por medio del editor de Alpha-Micro (VUE BOOTAL.LOG).
- Num-dep es el número de la dependencia con la que se identifica la biblioteca dentro del sistema LIBRUNAM.
- Colección es la letra progresiva que se le asignó al bloque de información en los procesos.

NOTA: Si no se está seguro o no se cuenta con los datos de Clave de la dependencia, Número de la dependencia y Colección para hacer uso de este comando, se pueden obtener según la explicación dada a detalle en el comando ALTDEP, (leer con cuidado porque en el punto que estamos tratando el encargado debe ya forzosamente llevar este control.)

El archivo de comandos **INVEN.DO** en realidad es un archivo de trabajo que manda ejecutar una serie de acciones contenidas en él con solo teclear su nombre; y al cual se le pasan parámetros para poder llevar a cabo los procesos de generación de listados. Como primer paso, manda ejecutar el archivo de comandos **SACTIT.DO** el cual carga automáticamente

el comando almacenado **ejecin** definido en el archivo **INVDEP.IDL**, que es un comando de ejecución dentro de IDL específicamente creado para la rutina de vaciar de la relación inventario ubicado en la base *libros* la información de la biblioteca marcada con la colección que se le está pasando como parámetro.

Posteriormente, dentro del mismo archivo **INVEN.DO** y en forma automática también, corre el comando **EX**, programa de utilería de Alpha-Micro que se interrelaciona directamente con Britton-Lee y por medio del cual se manda ejecutar el comando almacenado arriba mencionado, para vaciar en el archivo de datos **TITSOR.DAT** la información requerida.

En seguida, dentro de las rutinas incluidas en el archivo de comandos **INVEN**, se manda ejecutar el archivo de trabajo **SACSOB.DO**, el cual corre el programa **SACSOR** que se encarga de compactar por el número de matriz la información recién obtenida para facilitar y agilizar su procesamiento, ya que el archivo **TITSOR.DAT** tiene desglosado un registro por cada número de adquisición; esto es porque el sistema trabaja usando la matriz como llave para acceder la información en la relación *fichas*, para finalmente obtener los datos generales que corresponden a cada una de ellas (clasificación, autor, título). Por último, el sistema deja la información ya procesada en el archivo **SALSOR.DAT**.

Después, el comando **INVEN** manda procesar el archivo de trabajo **CORSOR.DO**, el cual carga utilería de Alpha-Micro para en sus procesos poder ordenar los archivos de datos; luego de esto, dentro de este comando se corre el programa **SORTEO** que se encarga de ligar la información de la relación inventario con la información propiamente de las fichas bibliográficas guardada en la relación *fichas*, y finalmente genera los listados con los datos de número de adquisición, número de matriz, clasificación, autor y título; ordenado topográficamente (i.e. por clasificación). Hecho esto, el sistema deja la información grabada en el archivo con nombre **SALNum-dep.LST**; mismo que, al término de su corrida se renombra automáticamente como **SALTOP.Num-dep**.

Este programa **SORTEO** también genera un archivo de datos imagen llamado **SAL.LST** que posteriormente será utilizado como base para generar los demás listados con diferentes ordenaciones. Esto es, al término de **CORSOR** y de renombrar **SALTOP**, en el mismo archivo de comandos **INVEN** se mandan llamar los archivos de comandos (con extensión **.CMD**) **COSOR2**, **COSOR3** y **COSOR4**, que ordenan el archivo **SAL.LST** por número de adquisición, por título y por autor respectivamente, quedándose renombrados finalmente en los archivos **SALADQ.Num-dep**, **SALTIT.Num-dep** y **SALAUT.Num-dep**.

A estos cuatro listados se les vuelve a aplicar un proceso para desglosar la información a un registro por cada número de adquisición, a fin de que los bibliotecarios puedan hacer en ellos las anotaciones pertinentes al comparar cada número contra el acervo físico.

(Nota: en todos estos casos Num-dep es el número de la dependencia.)

Por último, el encargado debe mandar imprimir estos listados, para luego remitirlos al(los) responsable(s) de la biblioteca.

Como un proceso especial, se ejecuta el comando **INVBAJ** especialmente diseñado para obtener los registros dados de baja en la relación inventario. Los listados que genera son exactamente igual a los generados a partir de las altas, la única diferencia es que el comando, en lugar de generarlos a partir de la relación inventario, obtiene la información a partir de la relación temporal creada de las bajas que se hicieron, es decir a partir de la relación *bajasdep*.

Para ello, el responsable del sistema deberá hacer uso del comando **INBAJ**.

Uso del comando:

**INVBAJ**    **Clave-dep.**    **Identif-proceso**    **Num-dep.**

donde:

- *Clave-dep* es la clave con la que se identifica la dependencia dentro del sistema LIBRUNAM.
- *Identif-proceso* es el nombre del proceso que le quiera dar el responsable a fin de que se identifique dentro de la bitácora de tiempos de corrida; esto es con el objeto de que, si falla la máquina o el sistema durante la ejecución de alguno de los pasos, se pueda identificar en cuál se quedó y retomar la continuación para su término; además de que, también le sirve para detectar tiempos consumidos y sacar estadísticas en cruce con el volumen de información procesada. El archivo que automáticamente el programa va grabando para bitácora, se denomina **BOOTAL.LOG** y su acceso se hace ya sea por medio de un type a este archivo (lectura secuencial) o por medio del editor de Alpha-Micro (**VUE BOOTAL.LOG**).
- *Num-dep* es el número de la dependencia con la que se identifica la biblioteca dentro del sistema LIBRUNAM.

#### 4.4.3 Generación de etiquetas del acervo bibliográfico dado de alta en el inventario.

Cuando el coordinador de estos procesos haya verificado que todas las etapas por las que ha pasado la información se han efectuado correctamente, entonces podrá pasar a la siguiente, que es la de la generación de las etiquetas para los libros que corresponden a esta información; en donde se especifican el número de adquisición con su código de barra y la clasificación del libro perteneciente a ese volumen.

Estas etiquetas vienen a finalizar el proceso de inventario de cada libro trabajado; y sirven como parámetro para que el encargado verifique si el inventario se hizo correctamente o no. Esto es, que toda etiqueta forzosamente deberá corresponder a un volumen específico; sirviendo además como un indicador de la información a corregir, ya que todas las etiquetas sobrantes (agotados todos los recursos de investigación), representan información que deberá ser dada de baja en el banco de datos, así como todos aquellos libros a los les falte de pegar su etiqueta de código de barra, será información a dar de alta en el banco de datos.

Se hace notar que este proceso se manda ejecutar una vez que ya se hayan hecho todas las correcciones necesarias a partir de los listados de errores generados durante los procesos de alta y/o de cambio.

La finalidad de utilizar etiquetas de código de barra es la facilidad y rapidez que proporciona al servicio de préstamo de libros, ya que ahorra al bibliotecario el teclear los números además de que se eliminan los errores de digitación; y todo esto con una sola pasada de la etiqueta por el lector óptico de código de barras.

Otra ventaja es que facilita los subsecuentes levantamientos de inventario al pasar estas etiquetas por este lector y dejar grabado en un archivo la información para su cotejo con el banco de datos.

Para lograr esto, el responsable del sistema deberá hacer uso del comando **ETITOD**.

Uso del comando:

**ETITOD Num-dep. Colección**

donde:

- Num-dep es el número de la dependencia con la que se identifica la biblioteca dentro del sistema LIBRUNAM.

- Colección es la letra progresiva que se le asignó al bloque de información en los procesos de alta.

NOTA: Si no se está seguro o no se cuenta con los datos de Número de la dependencia y Colección para hacer uso de este comando, se pueden obtener según la explicación dada a detalle en el comando ALTDEP, (leer con cuidado porque en el punto que estamos tratando el encargado debe ya forzosamente llevar este control.)

El archivo de trabajo (comandos) ETITOD.DO, como primer paso, manda cargar automáticamente el comando almacenado `sacetitod` definido en el archivo `ETITOD.IDL`, comando de ejecución dentro de IDL específicamente creado para las rutinas de vaciar de la relación inventario en la base libros, la información de la biblioteca marcada con la colección que se le está pasando como parámetro y de hacer la liga con la relación `fichas` para obtener la clasificación que le corresponde a cada volumen.

Posteriormente, dentro del mismo archivo ETITOD y en forma automática también, corre el comando `EX`, programa de utilería de Alpha-Micro que se interrelaciona directamente con Britton-Lee y por medio del cual se manda ejecutar el comando almacenado arriba mencionado para vaciar, en el archivo de datos `ETITOD.DAT`, la información requerida.

Después, el comando ETITOD manda procesar el programa `SORTOD` que se encarga de ordenar en forma topográfica el archivo de datos `ETITOD.DAT`, y deja una copia de este archivo ya organizado en uno con nombre `ETINum-dep.Colección`.

Finalmente el comando manda procesar el programa `BARILE` que se encarga de transformar la información ya procesada y grabada en el archivo `ETITOD.DAT` en las etiquetas de códigos de barras.

Estas quedan grabadas en un archivo llamado `BARNum-dep.n` donde `n` es un número progresivo a partir de 1, esto es porque el programa corta un archivo cada 5000 etiquetas por conveniencias de impresión.

Por último, el encargado debe mandar imprimir las etiquetas para luego remitirlas al(los) responsable(s) de la biblioteca.

Un proceso especial de generación de etiquetas corresponde a la reposición de etiquetas dañadas o mal impresas; para lo cual el bibliotecario deberá capturar la información de sólo los números de adquisición de las etiquetas que quiere reponer en un archivo secuencial (archivo con un solo campo numérico de 6 caracteres); para

ser trabajado en forma similar al de la captura de las altas, transportándolo en la DGB a la Alpha-Micro, y procesándolo posteriormente en la IDM.

Se le generará e imprimirá un archivo de errores de todos aquellos números que esté solicitando y no existan en su inventario dentro de la base de datos *libros*.

A continuación y para terminar el capítulo, se muestran los diagramas de flujo de los procesos descritos involucrados tanto en las altas como en las bajas.

DIAGRAMA DE FLUJO DE LOS PROCESOS DE ALTA

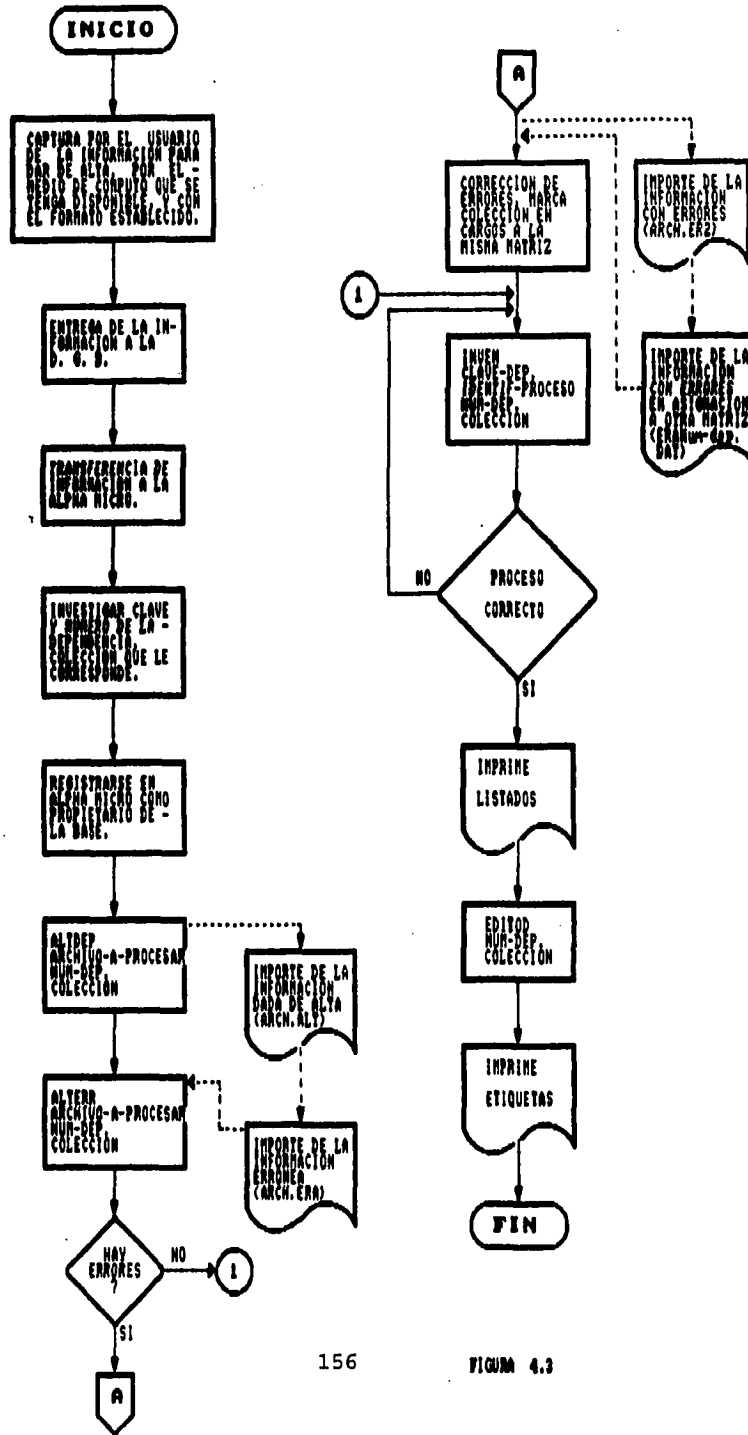


FIGURA 4.3



DIAGRAMA DE FLUJO DE LOS PROCESOS DE BAJA

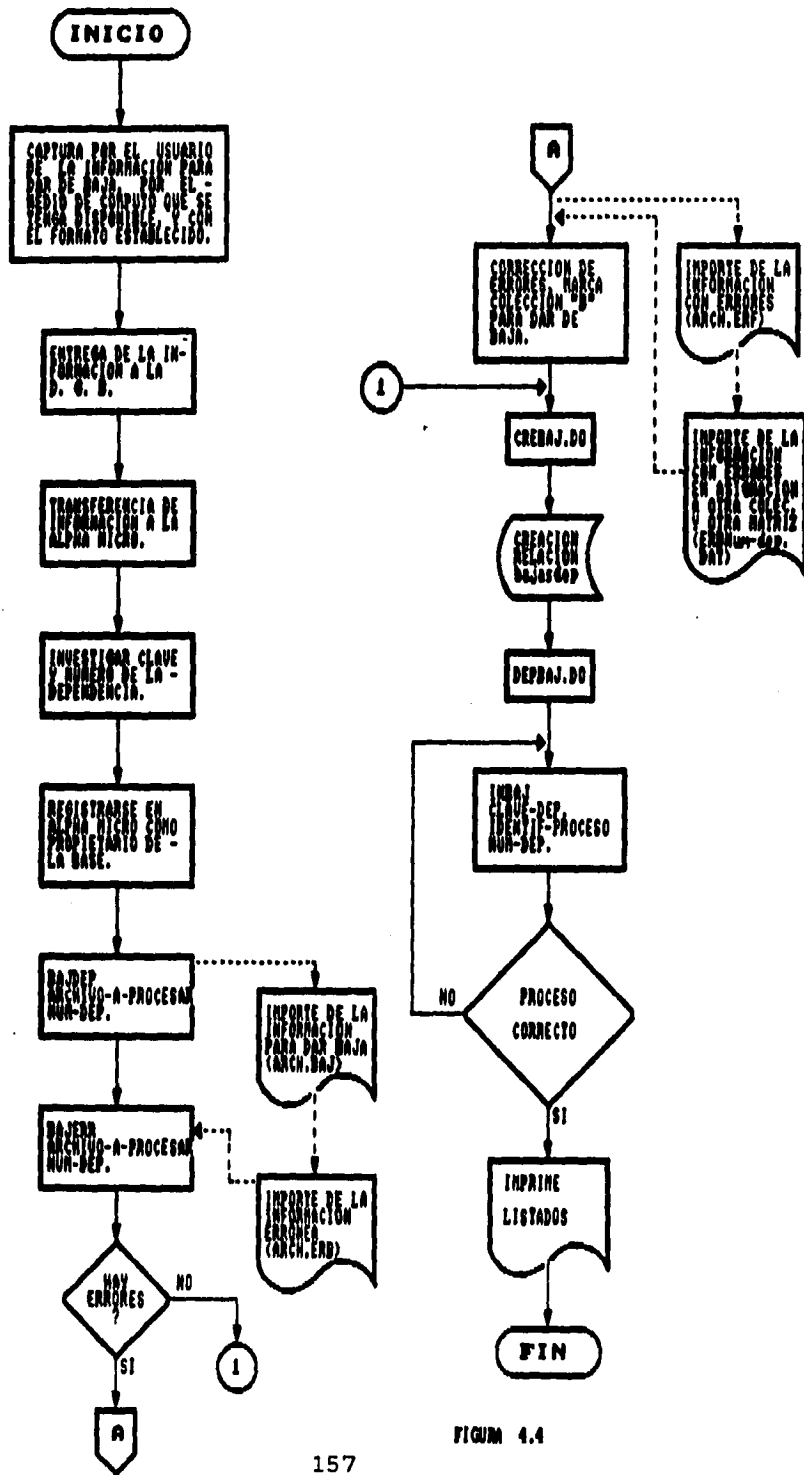


FIGURA 4.4

---

---

**CONCLUSIONES**

---

---

## CONCLUSIONES

Durante la exposición del presente trabajo se demostró que llevar a cabo un desarrollo automatizado profesional de un sistema, donde el análisis y la planificación tienen sus bases en estudios e investigaciones bien establecidos en el área a nivel mundial garantiza el éxito del sistema, como lo podemos observar en la creación, el desarrollo y la implementación del sistema INVENTARIOS.

El presente trabajo representa también el esfuerzo, tanto personal como colectivo, que se está llevando a cabo en la Subdirección de Informática de la Dirección General de Bibliotecas para apoyar, por medio del sistema INVENTARIOS, a las bibliotecas de la UNAM a controlar los libros que ingresan al sistema bibliotecario y en particular al banco de datos LIBRUNAM; donde se ha trabajado interactuando con otros sistemas y con otras áreas.

El sistema INVENTARIOS es el que se encarga del control del acervo bibliográfico de cada una de las bibliotecas de la Universidad Nacional Autónoma de México, mismo que se vuelve imprescindible para poder llevar a cabo un servicio de préstamo de libros eficiente, rápido, confiable, con estantería abierta; que por un lado ofrezca realmente una herramienta de ayuda al desarrollo académico de la comunidad universitaria, y por otro que auxilie en el control del patrimonio bibliográfico universitario.

El diseño y manejo de la base de datos "libros" nos comprueba que las bases de datos relacionales son las más adecuadas para el almacenamiento y la explotación de la información debido a que el lenguaje utilizado está totalmente basado en el álgebra y el cálculo relacionales, donde la estructura básica es un conjunto de relaciones normalizadas; lo que asegura la independencia, integridad y consistencia de los datos almacenados, y por ende de la base de datos.

Para el desarrollo del sistema se analizaron todas las necesidades y experiencias que se obtuvieron a partir del control manual del acervo en las bibliotecas de la UNAM, y se mostró durante la presentación del trabajo, que la aplicación de la metodología establecida en la ingeniería del software nos permitió desarrollar un software confiable y de calidad.

El cuidado en la planeación en el medio ambiente relacional de LIBRUNAM y el uso del lenguaje IDL empleado en este sistema para la definición, recuperación y manipulación de los datos, así como para el control de acceso y compartimiento de los mismos, nos ha permitido cumplir con las condiciones que se deben de considerar en el manejo de

bases de datos grandes, a la vez que el acceso a la información así como su procesamiento a través de eliminar, agregar y modificar información en cualquiera de las relaciones no ha representado un problema en la obtención de los resultados y en la interacción con los otros sistemas que se trabajan en la DGB, a pesar del gran volumen de información de que se trata.

Los objetivos establecidos en cuanto al sistema de INVENTARIOS como un sistema integral y con el propósito de dar un mejor y más eficiente servicio a los usuarios de la bibliotecas de la UNAM fueron, como quedó resaltado en el Capítulo 3, los siguientes:

- Conocer el material bibliográfico con que cuenta cada dependencia y en cuál de las bibliotecas se encuentra disponible.
- Tener herramientas para hacer inventarios periódicos en las bibliotecas y así controlar el patrimonio universitario y apoyar las rutinas de selección y adquisición de materiales.
- Promover la cooperación entre bibliotecas y fomentar el préstamo interbibliotecario.
- Ofrecer catálogos locales o UNION<sup>1</sup> a todas las bibliotecas del sistema.
- Ofrecer información sobre las colecciones de cada biblioteca para que se normen criterios de adquisición y descarte y/o convenios de cooperación entre bibliotecas que tienen colecciones similares y/o complementarias.
- Utilizar los recursos modernos para agilizar cualesquiera de las fases del proceso de inventarios o las rutinas de mantenimiento.
- Crear la infraestructura de cómputo para implantar:
  - Sistemas locales de circulación automatizada.
  - Sistemas globales automatizados de préstamo interbibliotecario.
  - Bibliotecas con estantería abierta al público.

1. Catálogos UNION se refiere a un catálogo que contenga todo el acervo localizado en las bibliotecas de la UNAM.

La estructura de la base de datos "libros" descrita en el Capítulo 4 permite almacenar en forma confiable la información, y consecuentemente la utilización de la base se convierte en una labor que simplifica el acceso a la información.

Por otro lado, el diseño de la sección de INVENTARIOS, está elaborado de tal forma que admite la fácil recuperación de la información para los fines que requiera el usuario, independientemente de los programas de aplicación que utilicen la base de datos y específicamente la relación "inventario".

Asimismo, su estructura es relativamente sencilla de entender por cualquier persona especializada en el área de cómputo, lo que facilita su transportabilidad a otras máquinas o plataformas más adecuadas a los cambiantes requerimientos de la dinámica universitaria, al mismo tiempo que continúa asegurando la consistencia de la base de datos.

Esta misma estructura permite que la actualización de la información y todos los productos que de ella se generan sea rápida y confiable, a fin de poder otorgar en forma oportuna la información y ayuda que requiera cada biblioteca del sistema universitario.

Todo lo anterior lleva finalmente a determinar con exactitud en qué biblioteca se encuentra un título buscado y cuántos volúmenes del mismo título están contenidos en ella.

Los objetivos que se establecieron en la creación de este sistema han sido logrados en un alto porcentaje con el equipo de cómputo que se tiene en la Dirección General de Bibliotecas; sin embargo, la problemática actual del sistema bibliotecario de la UNAM demanda nuevas y más adecuadas alternativas de soluciones, que requieren de una nueva plataforma de cómputo para poder desarrollarse.

Como una anotación final, y para completar las conclusiones con un comentario acerca de los lineamientos futuros en el desarrollo de los sistemas en la Dirección General de Bibliotecas, diremos que el equipo en el que actualmente se trabaja el sistema LIBRUNAM (la máquina IDM), ha otorgado a la fecha todos los beneficios que se esperaban de él, y se puede decir que actualmente es explotado en su máxima capacidad; por lo que resulta ya insuficiente para satisfacer las necesidades que se han generado a partir de manejar el volumen de información que se ha ido incrementado en forma vertiginosa.

Por otro lado, los requerimientos de los usuarios, y consecuentemente los cambios que se derivan de ellos obligan a buscar nuevas estrategias de trabajo, principalmente la descentralización de funciones, y esto aunado a los objetivos anteriormente mencionados ha llevado a la necesidad de investigar nuevas alternativas en equipo de cómputo, donde se puedan seguir desarrollando los sistemas que controlan en todos sentidos el acervo universitario.

Después del trabajo de análisis y tomando en cuenta todos los beneficios que se pueden obtener al utilizar esta nueva tecnología, en la Dirección General de Bibliotecas se tomó la decisión de adquirir dos equipos de cómputo, un **SUN SPARC SERVER 10** y un **SUN SPARC SERVER 1000**; los cuales trabajan con un sistema que combina el rendimiento de bases de datos con la habilidad de integrar las redes locales existentes con la red de la UNAM; ya que uno de los objetivos que se persiguen en este cambio es el de tener un equipo por cada base de datos.

El diseño de este equipo, ofrece la capacidad de multiprocesamiento, que permite que varios programas se ejecuten simultáneamente, utilizando cada uno su propio procesador.

También puede dividir una misma aplicación entre varios procesadores, para ejecutarse con mayor rapidez que en un procesador único.

El sistema SPARCserver opera en el ambiente de computación distribuida Solaris, por lo que trabaja con el sistema operativo SunOS UNIX, además que el entorno operativo Solaris incluye la capacidad de la función disponible en el ambiente de computación distribuida de este sistema operativo, que acelera las aplicaciones individuales dividiéndolas para ejecutarse paralelamente entre varios procesadores y consecuentemente obteniendo resultados con mayor rapidez que en un procesador único.

Por último, se hace notar que las estructuras y las soluciones propuestas durante el análisis para el desarrollo del sistema actual, no deben variar en mucho en el desarrollo del nuevo sistema, solamente se ajustarán a las nuevas perspectivas, ya que el manejador de la base de datos de la nueva máquina (SQL de Informix) también está basado en el uso del álgebra y el cálculo relacionales.

---

---

**REFERENCIAS**

---

---

## REFERENCIAS

1. Rivero, E. C. Bases de datos relacionales. Madrid : PARANINFO S.A., 1988. -- p. 15-17.
2. Martin, James. Organización de las bases de datos. México : PrenticeHall Hispanoamericana, S.A., 1977. -- p. 4-6.
3. Date, C. J. Introducción a los Sistemas de Bases de Datos. México : Addison-Wesley Iberoamericana, S.A., 1986. -- p. 5-7.
4. Martin, James. op. cit. p. 13.
5. Pressman, Roger S. INGENIERIA DEL SOFTWARE. Un enfoque práctico. -- 2a. ed. -- México : McGraw-Hill /Interamericana de España, S.A., 1986. -- p. 37.
6. Date, C. J. op. cit. p. 7-8.
7. Pressman, Roger S. op. cit. p. 38.
8. Date, C. J. op. cit. p. 16-19.
9. Martin, James. op. cit. p. 19.
10. Date, C. J. op. cit. p. 19-27.
11. Ibid. p. 20.
12. Martin, James. op. cit. p. 53.
13. Date, C. J. op. cit. p. 39-40.
14. Ibid. p. 31.
15. Ibid. p. 93.
16. Ibid. p. 95-97.
17. Ibid. p. 97-98.
18. Ibid. p. 99-100.
19. Ibid. p. 227-228.
20. Ibid. p. 229.
21. Ibid. p. 230-233.
22. Ibid. p. 85.



23. Ibid. p. 83.
24. Martin, James. op. cit. p. 156.
25. Date, C. J. op. cit. p. 238.
26. Ibid. p. 265-266.
27. Ibid. p. 271.
28. Ibid. p. 268.
29. Ibid. p. 274-278.
30. Ibid. p. 293-294.
31. Fleming, Candace C. Handbook of relational database design. Reading, Massachusetts : Addison-Wesley, 1989. p 532.
32. Pressman, Roger S. INGENIERIA DEL SOFTWARE. Un enfoque práctico. -- 2a. ed. -- México : McGraw-Hill /Interamericana de España, S.A., 1988. -- p. 33.
33. Pressman, Roger S. INGENIERIA DEL SOFTWARE. Un enfoque práctico. -- 3a. ed. -- México : McGraw-Hill /Interamericana de España, S.A., 1992. -- p. 7.
34. Pressman, Roger S. 2a. ed. op. cit. p. 15-16.
35. Ibid. p. 6-7.
36. Naur, P., y B. Randell (eds.), Software Engineering: A Report on a Conference sponsored by the NATO Science Comittee, NATO, 1969.
37. Pressman, Roger S. 2a. ed. op. cit. p. 22.
38. Ibid. p. 30-32.
39. Pressman, Roger S. 3a. ed. op. cit. p. 35-38.
40. Pressman, Roger S. 2a. ed. op. cit. p. 52.
41. Ibid. p. 57-60.
42. Pressman, Roger S. 3a. ed. op. cit. p. 154.
43. Ibid. p. 37.
44. Ibid. p. 155.

45. Ibid. p. 192-198.
46. Pressman, Roger S. 2a. ed. op. cit. p. 159.
47. Ibid. p. 223-231.
48. Pressman, Roger S. 3a. ed. op. cit. p. 329-333.
49. Ibid. p. 333-346.
50. Ibid. p. 346-354.
51. Ibid. p. 349.
52. Ibid. p. 351.
53. Ibid. p. 693-702 ; 717-718.
54. La Dirección General de Bibliotecas y la Biblioteca Central de la UNAM : historia, organización y servicios / Universidad Nacional Autónoma de México, Secretaría de Servicios Académicos, Dirección General de Bibliotecas; comp. César Augusto Ramírez Velázquez...[et al.]. México : UNAM, Dirección General de Bibliotecas, 1993. p. 1-2.
55. Ibid. p. 3-5.
56. Dirección General de Bibliotecas / Universidad Nacional Autónoma de México, Dirección General de Orientación Vocacional, Departamento de Información Académica; comp. Teresa Anaya Carpio ...[et al.]. México : UNAM, Dirección General de Orientación Vocacional, 1980. p. 9.
57. LIBRUNAM : sistema automatizado para bibliotecas. / Charlotte Bronsoiler...[et al.]. México : UNAM, Dirección General de Bibliotecas, 1982. p. 15.
58. Ibid. p. 23-24.
59. Ibid. p. 27-29.
60. Ibid. p. 29.
61. Ibid. p. 37-39
62. Martínez, Filiberto F. A. Manual de codificación para catalogadores. / García, Carlos L. México : UNAM, 1989. p. 1-2

---

---

**BIBLIOGRAFIA  
CONSULTADA**

---

---

## BIBLIOGRAFIA CONSULTADA

1. Alpha BASIC. User's manual. Santa Ana, California : Alpha Microsystems, 1986. 1v. (hojas sustituibles.)
2. AMOS Software documentation library. Santa Ana, California : Alpha Microsystems, 1986. 1v. (hojas sustituibles.)
3. Date, C. J. Introducción a los sistemas de bases de datos. México : Addison-Wesley Iberoamericana, S.A., 1986. 684 p.
4. Dirección General de Bibliotecas / Universidad Nacional Autónoma de México, Dirección General de Orientación Vocacional, Departamento de Información Académica; Teresa Anaya Carpio ...[et al.]. México : UNAM, Dirección General de Orientación Vocacional, 1980. 36 p.
5. Fleming, Candace C. Handbook of relational database design. Reading, Massachusetts : Addison-Wesley, 1989. 605 p.
6. González Trápaga, Ileana Inventarios / Ileana González Trápaga y Rocío Haza Remus. Trabajo presentado para el IV Encuentro de Bibliotecarios de la U.N.A.M. México : UNAM, 1985, Antiguo Colegio de Medicina.
7. ---- Inventarios / Ileana González Trápaga y Luz Ma. Mosqueda Martínez. Trabajo publicado en el Boletín informativo de la Dirección General de Bibliotecas de la U.N.A.M. México : UNAM, Vol I # 2, 1986.
8. IDL Intelligent database language. Winchester Boulevard, Los Gatos, CA : Britton Lee Inc., 1983. 1v. (hojas sustituibles.)
9. IDM Software reference manual. Winchester Boulevard, Los Gatos, CA : Britton Lee Inc., 1984. 1v. (hojas sustituibles.)
10. La Dirección General de Bibliotecas y la Biblioteca Central de la UNAM : historia, organización y servicios / Universidad Nacional Autónoma de México, Secretaría de Servicios Académicos, Dirección General de Bibliotecas; comp. César Augusto Ramírez...[et al.]. México : UNAM, Dirección General de Bibliotecas, 1993. 53 p.

11. LIBRUNAM : sistema automatizado para bibliotecas. / Charlotte Bronsoiler...[et al.]. México : UNAM, Dirección General de Bibliotecas, 1982. 174 p.
12. Loomis, Mary E. S. Estructura de datos y organización de archivos. México : PrenticeHall Hispanoamericana, S.A., 1991. 516 p.
13. Martin, James. Organización de las bases de datos. México : PrenticeHall Hispanoamericana, S.A., 1977. 544 p.
14. Martínez, Filiberto F. A. Manual de codificación para catalogadores. / García, Carlos L. México : UNAM, 1989. 150 p.
15. Pfleeger, Shari L. Software engineering. The production of quality software. 2a ed. E.U.A. : Macmillan Publishing Company, 1991. 517 p.
16. Pressman, Roger S. Ingeniería del software. Un enfoque práctico. 2a ed. México : McGraw-Hill /Interamericana de España, S.A., 1988. 628 p.
17. Pressman, Roger S. Ingeniería del software. Un enfoque práctico. 3a ed. México : McGraw-Hill /Interamericana de España, S.A., 1992. 807 p.
18. Rivero, E. C. Bases de datos relacionales. Madrid : PARANINFO S.A., 1988. 309 p.