

UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA



*SISTEMA DE CONTROL DE INVENTARIOS  
PARA FARMACIAS (SCIF)*

---

Tesis que Para Obtener el Título de  
INGENIERO EN COMPUTACIÓN

PRESENTA

JOSÉ MANUEL RAMÍREZ SOSA

---

Acatlima, Huajuapán de León, Oaxaca.

Agosto de 1999

---

## *AGRADECIMIENTOS*

A mis padres: Ignacio y Ana María,  
y a mis hermanos: Marita, Gaby,  
Nachito, Vero y Rosy por cuidar  
siempre de mí.

Al Ing. Gerardo García, al Lic. Carlos  
Santibañez y a mi asesor M.C. Luis  
Zarza por todo su apoyo.

A todos mis amigos, que  
afortunadamente son muchos

Y a Dios por haberme dado una  
familia tan maravillosa y haber puesto  
en mi camino excelentes amigos.

U. T. M. 10050

## ÍNDICE

	Página
INTRODUCCIÓN.....	1
Antecedentes.....	1
Planteamiento del problema.....	1
Objetivo de la tesis.....	1
CAPÍTULO I. Tecnología de información.....	2
1.1. Bases de datos.....	3
1.1.1. Definición.....	4
1.1.2. ¿Por qué utilizar una base de datos?.....	4
1.1.3. Ventajas del enfoque de base de datos.....	4
1.1.4. Arquitectura de las bases de datos.....	6
1.1.5. Sistema de gestión de bases de datos.....	7
1.1.6. Administrador de la base de datos.....	8
1.1.7. Restricciones de integridad y seguridad.....	9
1.1.7.1. Violaciones de la integridad y seguridad.....	9
1.1.7.2. Restricciones de dominio.....	9
1.1.7.3. Valores nulos.....	10
1.1.7.4. Integridad referencial.....	10
1.1.7.5. Aspectos generales de seguridad.....	10
1.1.8. Recuperación y atomicidad.....	11
1.1.8.1. Clasificación de los fallos.....	11
1.1.8.2. Tipos de fallos.....	12
1.1.8.3. Estados de transacción.....	12
1.1.9. Diseño.....	13
1.1.10. Modelo relacional.....	14
1.1.10.1. Tablas.....	14
1.1.10.2. Estructura de datos relacional.....	14
1.1.10.3. Claves.....	16
1.1.10.4. Propiedades de las relaciones.....	17
1.1.11. Normalización.....	18
1.1.12. Dependencia funcional.....	18
1.1.12.1. Determinante.....	19
1.1.12.2. Primera forma normal: 1FN.....	19
1.1.12.3. Dependencia funcional total: 2FN.....	20
1.1.12.4. Dependencia transitiva: 3FN.....	20
1.1.12.5. Forma normal de Boyce Cood: BCFN.....	21

1.1.12.6. Dependencias de valores múltiples: 4FN.....	21
1.1.13. Modelo entidad-relación.....	21
1.1.13.1. Grado de las relaciones.....	23
1.1.13.2. Tipos de participación de una entidad.....	23
1.1.13.3. Casos de participación de las entidades.....	24
1.1.13.4. Obtención de las relaciones a partir del diagrama.....	25
1.1.13.5. Relaciones preliminares para las correspondencias binarias de grado 1:1 .....	25
1.1.13.6. Relaciones preliminares para las correspondencias binarias de grado 1:n .....	28
1.1.13.7. Relaciones preliminares para las correspondencias binarias de grado m:n .....	29
1.1.13.8. Relaciones preliminares para las correspondencias ternarias.....	30
1.2. Software.....	31
1.2.1. Definición.....	31
1.2.2. Características del software.....	31
1.2.3. Aplicaciones del software .....	32
1.2.4. Problemas del desarrollo del software.....	35
1.3. Ingeniería del software.....	36
1.3.1. Definición.....	36
1.3.2. Elementos de la ingeniería del software .....	36
1.3.3. Paradigmas de la ingeniería del software.....	37
1.3.3.1. El ciclo de vida clásico.....	37
1.3.3.2. Construcción de prototipos.....	39
1.3.3.3. El modelo en espiral.....	40
1.3.4. Fases del desarrollo del software.....	42
CAPÍTULO II. El proceso administrativo en los negocios.....	43
2.1. Información gerencial.....	44
2.1.1. Significado de la información.....	44
2.1.2. Atributos de la información.....	44
2.1.3. Necesidades de la información por parte de los gerentes.....	45
2.1.4. Sistemas de información gerencial.....	46
2.1.4.1. Sistemas de procesamiento de las transacciones.....	46
2.1.4.2. Sistemas de informes gerenciales.....	48

---

2.1.4.3. Sistemas de apoyo a las decisiones.....	48
2.1.5. Alcance de los sistemas de información.....	49
2.1.6. Componentes de un sistema de información.....	49
2.1.7. Efecto de los sistemas de información .....	49
2.2. Inventarios.....	51
2.2.1. Variables controlables.....	51
2.2.2. Variables no controlables.....	52
2.2.3. Componentes básicos del inventario.....	53
2.2.4. Sistema de inventarios.....	55
2.2.4.1. Sistema automatizado de información.....	55
2.2.4.2. Requerimientos del sistema.....	56
2.2.4.3. Requisitos.....	57
2.2.4.4. Desarrollo del sistema.....	58
2.3. Historia de la industria farmacéutica en México.....	60
CAPÍTULO III. Análisis y diseño del SCIF.....	62
3.1. Análisis del SCIF.....	63
3.2. Diseño del SCIF.....	72
3.3. Diccionario de datos.....	86
3.4. Diagramas de flujo.....	88
3.5. Características del SCIF.....	95
CONCLUSIONES.....	100
BIBLIOGRAFÍA Y REFERENCIAS .....	101
APÉNDICE A.....	103
APÉNDICE B.....	104

## INTRODUCCIÓN

### Antecedentes

El procesamiento de la información es fundamental para planificar cualquier actividad, ya sea, desde conocer lo que sucede alrededor del mundo o el disponer en todo momento de una información actualizada de los datos de una empresa.

Una consecuencia de todo lo anterior es la gran cantidad de datos que es necesario manejar en cualquier actividad de nuestra vida, sobre todo en el mundo empresarial. Al aumentar el volumen de datos a manejar, también debe aumentar la rapidez con que se procesan dichos datos. Cualquier empresa mediana, en la actualidad, maneja miles de millones de elementos individuales de datos.

En toda empresa comercial, resulta de vital importancia el manejo adecuado de los datos referentes a los inventarios, ya que estos tienen relación directa con las ventas, que son la parte medular de las empresas.

Se han desarrollado paquetes de software para llevar el control de los inventarios en las distintas ramas de la industria, y el ramo farmacéutico no es la excepción. Sin embargo para el nivel de vida que existe en Huajuapán resulta un poco costoso para los comerciantes adquirir uno de estos paquetes, por lo que en esta población son pocas las personas que cuentan con un sistema de inventarios, tomando en cuenta que es una población totalmente comercial.

### Planteamiento del problema

Por las características de las farmacias, se manejan una gran cantidad de productos. Varios son los problemas que se presentan. En primer lugar los productos farmacéuticos sufren una variación de precios casi diario y debido a esto se tiene que estar actualizando el inventario periódicamente. Aunado a lo anterior, se requiere de una revisión constante en cuanto a la fecha de caducidad de todos y cada uno de los productos, para promocionar los que estén próximos a caducar o eliminar los que ya caducaron. Parte importante es también, llevar una relación de las ventas realizadas durante el día, ya que a partir de esta información saldrá la lista de lo que será el pedido a los proveedores. Al final del año, y por cuestiones contables, es necesario realizar un conteo físico de todos los productos para saber el valor que tenemos en inventario. Todas las operaciones anteriores y más, son llevadas a cabo manualmente, lo que se traduce en un trabajo tedioso y cansado.

### Objetivo de la tesis

Diseñar, desarrollar e implantar un sistema de control, para manejar de manera automatizada, los inventarios de una farmacia.

---

# CAPÍTULO I

*Tecnología de información*

---

---

## 1.1. BASES DE DATOS

En la actualidad, las computadoras han venido a revolucionar la forma de almacenar y tratar la información. A principios de los años sesenta, varias empresas empiezan a guardar sus datos no en papel, como se venía haciendo, sino en soportes magnéticos para ser tratados directamente por la computadora.

El costo de los equipos para el almacenamiento de los datos decrece más rápidamente que cualquier otra forma, y hoy es más barato almacenar los datos en los archivos de la computadora que sobre el papel. Por otro lado, al aumentar la capacidad de los soportes para almacenar los datos, disminuye el costo por bit almacenado, lo que hace aún más atractivos los soportes magnéticos en detrimento del papel.

En las primeras épocas del tratamiento automático de la información, las aplicaciones y los programas se desarrollaban independientemente y cada uno de ellos procesaba sus propios archivos. Como resultado, algunos datos se repetían y además eran utilizados en diferentes operaciones.

Al final de los años sesenta, para combatir los defectos de los sistemas de información, surgen las bases de datos. En una única base de datos se integran archivos individuales para ser compartida por todos los usuarios de una empresa.

El aumento del volumen de datos a tratar lleva implícito el diseño de un soporte físico con la suficiente capacidad para poder almacenar dichos datos, pero, además, el adecuado aprovechamiento de estas grandes cantidades de datos lleva consigo el desarrollo de:

- *Transmisión de datos.* La capacidad de acceso a la base de datos desde lugares remotos.
- *Diálogo entre el hombre y la computadora.* Permite al usuario realizar las siguientes operaciones:
  - a) Agregar nueva información a la base de datos.
  - b) Clasificar la base de datos en algún orden significativo.
  - c) Buscar algún dato en la base de datos, de acuerdo a algún criterio de información que le interese.
  - d) Imprimir la información deseada de la base de datos para incluirla en informes preformateados.
  - e) Modificar la información que contiene la base de datos.
  - f) Borrar datos que han dejado de ser útiles en la base de datos.
- *Diseño de la base de datos.* Su principal objetivo es que los datos se almacenen de tal forma que puedan ser utilizados por una gran variedad de aplicaciones, incluso al mismo tiempo, y se pueda modificar fácilmente la forma de utilizarlos.



### 1.1.1. Definición

“Una base de datos es un conjunto de datos almacenados de tal forma que permitan guardar y extraer información útil por medio de la ejecución de programas especiales” [1].

### 1.1.2. ¿Por qué utilizar una base de datos?

¿Por qué ha de utilizarse un sistema de base de datos? ¿Qué ventajas representa? Hasta cierto punto la respuesta a estas preguntas depende de si el sistema es de uno o varios usuarios.

Las ventajas de un sistema de base de datos sobre los métodos tradicionales de mantener registros en papel por mencionar algunas serían:

- *Es compacto.* No hacen falta archivos de papeles que pudieran ocupar mucho espacio.
- *Es rápido.* La máquina puede obtener y modificar datos con mucha mayor velocidad que un ser humano. Así es posible satisfacer con rapidez consultas de casos particulares, sin necesidad de búsquedas visuales o manuales que requieren mucho tiempo.
- *Es menos laborioso.* Se elimina gran parte del tedio de mantener archivos a mano. Las tareas mecánicas siempre serán mejor realizadas por las máquinas.
- *Es actual.* Se dispone en cualquier momento de información precisa y al día.

Desde luego, las ventajas anteriores tienen aún más importancia en un ambiente multiusuario, donde la base de datos es con toda probabilidad mucho más grande y compleja que en el caso de un solo usuario. Con todo, existe una ventaja adicional en un ambiente así, el sistema de base de datos ofrece a la empresa un control centralizado de su información. Una situación así presenta un fuerte contraste con respecto a la de una empresa sin un sistema de bases de datos, donde cada aplicación tiene por lo regular sus propios archivos privados -y con frecuencia también sus propias cintas y discos- de manera que los datos están muy dispersos y con seguridad son difíciles de controlar en cualquier forma sistemática.

### 1.1.3. Ventajas del enfoque de base de datos

A continuación veremos algunas de las ventajas específicas que resultan del concepto de control centralizado.

- *Es posible disminuir la redundancia.* En aplicaciones en las que no se cuenta con sistemas sin bases de datos cada una de ellas tiene sus propios archivos privados. Esta situación puede provocar redundancia en los datos almacenados, y por ende, desperdicio en el espacio de almacenamiento.

Con lo anterior no se quiere decir que es posible o deseable eliminar toda la redundancia. En ocasiones existen motivos válidos de negocios o técnicos para mantener varias copias de los mismos datos almacenados. Pero es importante controlar con mucho cuidado esa redundancia; es decir, el DBMS<sup>1</sup> debe tenerla presente, si existe, y asumir la responsabilidad de “propagar las actualizaciones”.

- *Es posible evitar la inconsistencia (hasta cierto punto).* Si por ejemplo, el hecho de que el producto P1 es surtido por el proveedor PR3 está representado por dos entradas distintas en la base de datos almacenada y DBMS no está sabe de esta duplicación (es decir, la redundancia no está controlada), habrá ocasiones en que las dos entradas no coincidan, cuando se haya puesto al día una y la otra no. Quedando la base en un estado inconsistente. Lo que puede producir información errónea.
- *Es posible compartir los datos.* No sólo las aplicaciones ya existentes pueden compartir la información de la base de datos, sino también que se pueden desarrollar aplicaciones nuevas para trabajar con los mismos datos almacenados.
- *Es posible hacer cumplir las normas.* Si se tiene un control centralizado de la base de datos, el Administrador de la Base de Datos (DBA<sup>2</sup>) puede garantizar la aplicación de las normas que tiene que ver con los datos. Estas normas pueden derivarse de varias partes: de la empresa, de la instalación, del departamento, de la industria, nacionales o internacionales, o de todos estos tipos. La normalización de formatos de los datos almacenados es deseable sobre todo como apoyo para el intercambio de información, o migración de datos entre sistemas. El aplicar normas para nombrar y documentar los datos son muy convenientes como ayuda para el compartimiento y comprensibilidad de la información.
- *Es posible aplicar restricciones de seguridad.* Para ello el DBA se encargará de
  - a) asegurar que el acceso a la base de datos sea sólo a través de los canales apropiados y, por tanto,
  - b) definir las verificaciones de seguridad por realizar cuando se intente acceder a la información delicada.
- *Es posible mantener la integridad.* El problema de la integridad radica en asegurar que la información de la base de datos sea correcta. La inconsistencia entre dos entradas que supuestamente representan el mismo “hecho” es un ejemplo de falta de integridad; por supuesto, ese problema en particular sólo puede presentarse si existe redundancia en la base de datos almacenada. Sin embargo, aun cuando no haya redundancia, es evidente que la base de datos puede contener información errónea. Por ejemplo, una consulta podría revelar que un producto cuesta \$ -50.00 en vez de \$ 50.00, o que pertenece al laboratorio L2, el cual no existe. Un control centralizado de la base de datos puede contribuir a evitar este tipo de problemas -hasta donde pueden evitarse- si el DBA define verificaciones de integridad que deben realizarse en toda operación de actualización de los datos.

<sup>1</sup> En el apartado 1.1.5. se hablará acerca del DBMS.

<sup>2</sup> El DBA se analiza en el apartado 1.1.6.

La integridad de la información es todavía más importante en un sistema multiusuario de bases de datos que en un ambiente de archivos privados, porque se comparte la base de datos. Sin los controles adecuados, un usuario puede modificar en forma incorrecta la base de datos, generando información errónea que afectaría a muchos usuarios.

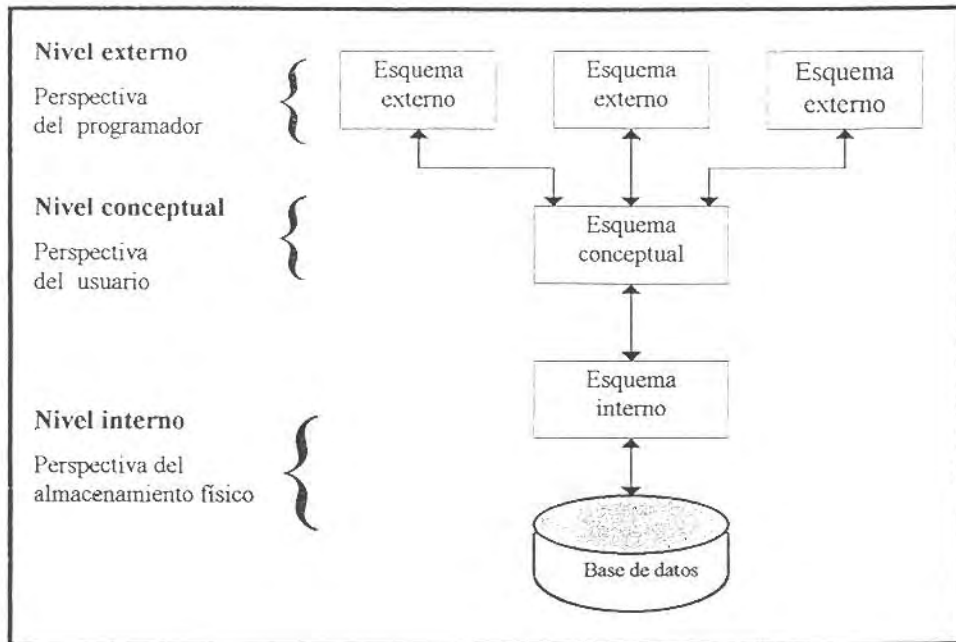
- *Es posible equilibrar requerimientos opuestos.* Al conocer cuáles son los requerimientos de la empresa -y los de cualquier usuario individual- el DBA puede estructurar el sistema con miras a proporcionar un servicio general óptimo para la empresa.

#### 1.1.4. Arquitectura de las bases de datos

Uno de los objetivos de un DBMS es proporcionar a los usuarios una visión abstracta de la información, es decir, ocultar ciertos detalles referentes a la forma en que los datos se almacenan y mantienen, pero siempre permitiendo una recuperación eficaz.

La arquitectura de una base de datos debe poseer tres niveles (Figura 1.1): *interno o físico, conceptual y externo o de visión.*

- *Nivel interno o físico.* Es el nivel más bajo de abstracción y describe *cómo* se almacenan realmente los datos. En este nivel, se describen en detalle las estructuras de datos complejas del nivel bajo.
- *Nivel conceptual.* Es el nivel más alto de abstracción, aquí se describe *qué* datos son realmente almacenados en la base de datos y las relaciones entre ellos. Además se describe la base de datos completa en términos de un número pequeño de estructuras relativamente sencillas. La implementación de las estructuras sencillas en el nivel conceptual puede implicar estructuras complejas del nivel físico, sin embargo, el usuario del nivel conceptual no necesita darse cuenta de esto. Este nivel es usado por los administradores de las bases de datos, quienes deben decidir qué información se va a guardar en la base de datos.
- *Nivel externo o de visión.* El nivel conceptual describe sólo parte de la base de datos completa. A pesar del uso de estructuras más sencillas en él, queda algo de complejidad debido al gran tamaño de la base de datos. Esta información no resultará de interés para muchos usuarios. En cambio, dichos usuarios sólo necesitan una parte de la base de datos. Para simplificar su interacción con el sistema, se define el nivel de abstracción de visión. El sistema puede proporcionar muchas visiones para la misma base de datos.



**Figura 1.1** Los tres niveles de la arquitectura de una base de datos.

### 1.1.5. Sistema de gestión de bases de datos

Un sistema de gestión de bases de datos (DBMS Database Management System) consiste en una colección de datos (base de datos) interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un DBMS es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. Sus principales funciones son:

1. Crear y organizar la base de datos.
2. Establecer y mantener las rutas de acceso a cualquier dato de la base de la forma más rápida posible.
3. Manejar los datos según las necesidades de los usuarios.
4. Mantener la integridad y seguridad de los datos, tanto contra las caídas del sistema como contra los intentos de acceso no autorizados.
5. Llevar un control de los usuarios que acceden a la base de datos.

La Figura 1.2 muestra cómo un DBMS sirve de intermediario entre una base de datos y las peticiones de los usuarios para recobrar la información de ésta.

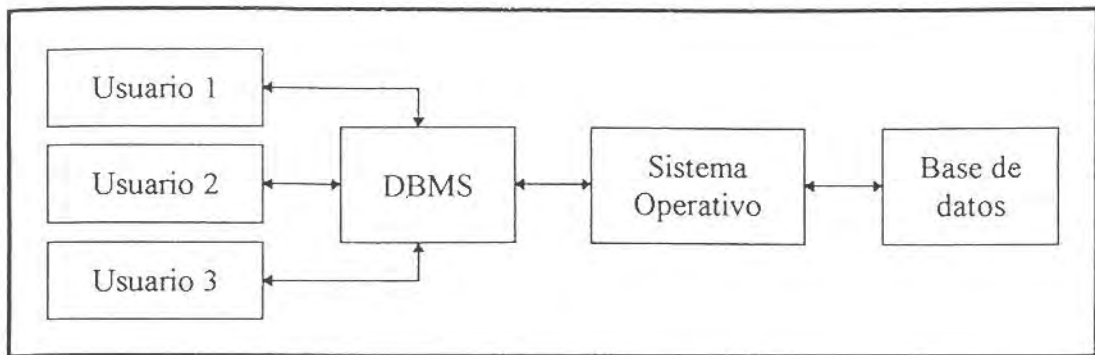


Figura 1.2 El DBMS como intermediario.

Ante cualquier pregunta de los usuarios, el DBMS responde utilizando los subprogramas apropiados para interpretar la consulta o localizar los datos deseados y presentarlos en el orden requerido. El DBMS permite el acceso a los datos en cualquier parte de la base de datos sin necesidad de conocer su organización dentro del dispositivo de almacenamiento.

### 1.1.6. Administrador de la base de datos

En una empresa que utilice un sistema de base de datos debe existir una persona o grupo de personas llamado *Administrador de la Base de Datos* (DBA por sus siglas en inglés, *Database Administrator*) cuya responsabilidad sea de controlar todos los datos.

Entre las responsabilidades del DBA se incluyen:

- *Definición del esquema.* Se deberá escribir una serie de definiciones que el compilador del lenguaje de definición de datos traduce a un conjunto de tablas que se almacenan de forma permanente en el diccionario de datos.
- *Definición de la estructura de almacenamiento y del método de acceso.* Se lleva a cabo al escribir la definición de la estructura de almacenamiento, mediante el lenguaje de definición de datos, que posteriormente son traducidas por el compilador del lenguaje de almacenamiento y definición de datos.
- *Concesión de autorización para el acceso a los datos.* Mediante este acceso se regulan las partes de la base de datos a las que van a tener acceso los diversos usuarios. Además se debe especificar la correspondencia entre cualquier esquema externo y el esquema conceptual.
- *Definición de una estrategia de respaldo y recuperación.* En el caso de que se produzca un fallo, el DBA debe definir y poner en marcha una estrategia de recuperación adecuada para reparar los datos pertinentes, con la mayor brevedad posible y con la menor repercusión para el resto del sistema.
- *Modificación del esquema y de la organización física.* Aunque estos cambios son poco frecuentes, el DBA necesitará varios programas de utilidad para realizar estas modificaciones, entre las que se encuentran:

- a) Rutinas de carga para crear inicialmente la base de datos.
- b) Rutinas de reorganización, por ejemplo, reordenar la base de datos para la recuperación física del espacio ocupado por los datos obsoletos.
- c) Rutinas de registro de sucesos diarios, que consisten en anotar cada operación sufrida por la base de datos junto a la identificación del usuario y los valores anterior y posterior a la misma.
- d) Rutinas de recuperación, que consisten en restaurar la base de datos al estado anterior al fallo producido en el hardware o en la programación.
- e) Rutinas de análisis estadístico, que ayudan a controlar el desempeño.

### 1.1.7. Restricciones de integridad y seguridad

Por medio de las restricciones de integridad se va a asegurar que los cambios que se lleven a cabo en la base de datos por los usuarios no van a ocasionar una inconsistencia de los datos. Estas restricciones protegen la base de datos contra daños accidentales. En tanto que la seguridad se refiere a la protección de los datos contra una alteración o destrucción no autorizada.

#### 1.1.7.1. Violaciones de la integridad y seguridad

El mal uso que se haga de la base de datos puede ser intencionado (con fines indebidos) o accidental. La pérdida accidental de la consistencia de los datos puede deberse a:

- Caídas durante el procesamiento de las transacciones.
- Anomalías por acceso concurrente a la base de datos.
- Anomalías que resultan de la distribución de los datos entre varias computadoras.
- Un error lógico que viola la suposición de que las transacciones respetan las protecciones de consistencia de la base de datos.

Es más fácil prevenir la pérdida accidental de la consistencia de los datos que prevenir el acceso mal intencionado a la base de datos. Algunas de las formas de acceso indebido son las siguientes:

- Lectura de datos sin autorización (robo de información).
- Modificación de datos sin autorización.
- Destrucción no autorizada de los datos.

#### 1.1.7.2. Restricciones de dominio

Un dominio es un conjunto de valores escalares todos del mismo tipo. Así, los dominios son fondos de valores, de los cuales se extraen los valores reales que aparecen en los

atributos<sup>3</sup>. Cada atributo debe estar “definido sobre” un dominio subyacente, y sólo uno, lo cual significa que los valores de ese atributo deben proceder de ese dominio. De ahí que la importancia de los dominios radica en que restringen las comparaciones.

Por tanto, si dos atributos toman sus valores del mismo dominio, es probable que tengan sentido las comparaciones -y por ende las reuniones, uniones y muchas otras operaciones- donde estén implicados esos dos atributos porque se comparan cosas similares. En cambio, si los dos atributos obtienen sus valores de diferentes dominios, las comparaciones donde estén implicados esos dos atributos sin duda carecerán de sentido. Así, una ventaja del manejo de dominios por parte del sistema es que lo hace capaz de impedir errores tontos de los usuarios. Si el usuario tratara de ejecutar una operación que incluyera una comparación entre dominios distintos, el sistema podría interrumpirla e indicar al usuario el probable error.

Así pues, los límites de dominios son la forma más elemental de restricciones de integridad. Son fáciles de probar por el sistema siempre que se introduce un nuevo dato en la base de datos.

### 1.1.7.3. Valores nulos

La inserción de tuplas<sup>4</sup> incompletas puede introducir valores vacíos en la base de datos. Existen atributos, que al recibir valores nulos puede ser inapropiados. Supongamos que tenemos una relación Inventario con los siguientes atributos {nombre\_producto, precio\_producto, laboratorio\_producto} y que para nombre\_producto tenemos un valor vacío, esto nos lleva a tener una tupla con un precio y laboratorio para un producto anónimo y, por tanto, no contiene información útil.

La clave primaria<sup>5</sup> de una relación nunca podrá recibir valores nulos.

### 1.1.7.4. Integridad referencial

Debemos asegurar que cuando se lleve a cabo una actualización sobre un valor que aparece en una relación para un conjunto de atributos dado se realice la misma si éste también aparece para un cierto conjunto de atributos en otra relación, es decir, se deberán llevar en ambos casos sus correspondientes actualizaciones.

### 1.1.7.5. Aspectos generales de seguridad

Para proteger a la base de datos es necesario adoptar medidas de seguridad en varios niveles [8]:

<sup>3</sup> Los atributos son las características propias de los objetos que los hacen distinguibles unos de otros. También se les conoce como campos.

<sup>4</sup> Conjunto de atributos de datos relacionados.

<sup>5</sup> Más adelante en el apartado 1.1.10.3. se analizará el tema de las claves primarias.

- *Físico.* Los lugares donde se encuentran las computadoras deben protegerse físicamente contra la penetración armada o clandestina.
- *Humano.* Se debe ser muy cuidadoso cuando se conceda autorización a los usuarios, por que estos podrían permitir el acceso a personas ajenas a lugares restringidos.
- *Sistema operativo.* Si bien, aunque el sistema de base de datos esté bien protegido, pero el sistema operativo no, éste puede servir para obtener acceso sin autorización a la base de datos. La mayoría de los sistemas de bases de datos permiten acceso remoto a través de terminales o redes, la seguridad a nivel software dentro del sistema operativo es tan importante como la seguridad física.
- *Sistemas de base de datos.* Algunos usuarios pueden estar autorizados sólo para tener acceso a una porción limitada de la base de datos. Es posible también que a algunos usuarios se les permita hacer consultas, pero se les prohíba modificar la base de datos. El sistema de base de datos tiene la responsabilidad de garantizar que no se violen estas restricciones.

La seguridad en todos los niveles anteriores debe mantenerse para asegurar la seguridad de la base de datos. Un punto débil en un nivel bajo de seguridad (física o humana) permiten que se burlen medidas estrictas de seguridad a alto nivel (base de datos).

En muchas aplicaciones vale la pena dedicar un esfuerzo considerable a la conservación de la seguridad e integridad de la base de datos.

### 1.1.8. Recuperación y atomicidad

Las computadoras, como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallos. Existen varias causas que pueden ocasionar fallos, incluyendo roturas del disco, interrupción en el suministro de energía, errores de software, un incendio en la habitación de la máquina, etc. En cada uno de estos casos se pierde información referente al sistema de base de datos. Una parte integral de un sistema de una base de datos es un esquema de recuperación que es responsable de la eliminación de fallos y de la restauración de la base de datos a un estado consistente que existía antes de que ocurriera el fallo.

A menudo, varias operaciones de la base de datos forman una unidad de trabajo lógica simple. Un ejemplo es la transferencia de fondos, en la que se carga una suma en cuenta y se da un crédito a otra.

Es fundamental para la consistencia de la base de datos que el débito y el crédito se lleven a cabo o que ninguno de los dos ocurra, es decir, la transferencia de fondos debe ocurrir por completo o no ocurrir; este requisito de "todo o nada" se denomina *atomicidad*.

#### 1.1.8.1. Clasificación de los fallos

Existen varios tipos de fallos que pueden presentarse en un sistema, cada uno de los cuales necesita ser tratado de una manera diferente. El tipo de fallo más sencillo de tratar es



el que no resulta en pérdida de información en el sistema. Los fallos más difíciles de tratar son los que resultan en pérdida de información. Para determinar como debe recuperarse el sistema de los fallos necesitamos identificar los modos del fallo de los dispositivos utilizados para almacenar los datos. A continuación debemos considerar cómo afectan estos modos de fallo a los contenidos de la base de datos. Podemos así proponer algoritmos para asegurar la consistencia de la base de datos y la atomicidad de las transacciones a pesar de fallos. Estos algoritmos tienen dos partes:

1. Acciones tomadas durante el procesamiento normal de transacción para asegurar que existe suficiente información para permitir la recuperación de fallos.
2. Acciones tomadas a continuación de un fallo para asegurar la consistencia de la base de datos y la atomicidad de las transacciones.

### 1.1.8.2. Tipos de fallos

Un esquema de recuperación debe invocarse como resultado de varios tipos de fallos, entre los cuales podemos encontrar :

- *Errores lógicos.* Existe alguna condición externa que impide continuar con la transacción, como puede ser una entrada inválida, información no localizada, desbordamiento (overflow) o que se exceda el límite de los recursos.
- *Errores de sistema.* El sistema ha entrado en un estado no deseable, por lo cual la transacción no puede continuar con su ejecución normal. Sin embargo, la transacción puede volverse a ejecutar más tarde.
- *Caída del sistema.* El hardware funciona mal, causando la pérdida del contenido de la memoria principal. El contenido de la información en el disco permanece intacto.
- *Fallo de disco.* Un bloque del disco pierde su contenido debido a la rotura de la cabeza o a fallos durante una operación de transacción de información.

### 1.1.8.3. Estados de transacción

El programador de una aplicación, que codifica la transacción, es responsable de asegurar la ejecución correcta. Esta tarea puede facilitarse mediante la prueba automática de las restricciones de integridad. Asegurar la atomicidad es responsabilidad de la base de datos, específicamente del componente gestor de las transacciones. En ausencia de fallos, todas las transacciones terminan con éxito y la atomicidad se logra fácilmente. Sin embargo, como se explicó anteriormente, una transacción puede que no siempre se termine con éxito. En este caso se dice que *la transacción se abortó*. Para garantizar la propiedad de atomicidad, una transacción abortada no debe afectar al estado de la base de datos. Así, el estado de la base de datos debe restaurarse al estado en que estaba justo antes de que se comenzara a ejecutar la transacción en cuestión. Decimos que *la transacción retrocede*. Es parte de la responsabilidad del esquema de recuperación la gestión de los abortos de transacciones.

Una transacción que termina con éxito su ejecución se denomina *cometida*. Una transacción cometida que realiza actualizaciones transforma la base de datos en un nuevo estado consistente.

### 1.1.9. Diseño

El diseño de una base de datos significa pensar sobre la forma en que deberán almacenarse los datos y en cómo se solicitará información a partir de los datos existentes.

La tarea de diseñar es difícil, sobre todo cuando se pretende alcanzar soluciones óptimas. Hay que tener en cuenta que hay muchas formas de estructurar los datos, cada una de ellas con sus ventajas e inconvenientes, además, cada tipo de dato requiere una organización específica, todo ello unido a las exigencias de los usuarios.

El diseño de una base de datos implica tres pasos principales:

1. Definición de los datos (un análisis de los datos existentes).
2. Refinación de los datos (depuración de los datos necesarios).
3. Establecer relaciones entre los atributos.

- *Definición de los datos.* Durante la primera fase del diseño de datos, se debe realizar una lista, de todos los atributos (campos) importantes implicados en la aplicación. Para hacer esto, se debe examinar la aplicación en detalle para determinar exactamente la clase de información que debe almacenarse en la base de datos.

Un punto importante que no se debe olvidar, es que durante la fase de diseño se deberá considerar todos los atributos que pueda requerir la base de datos. No será problema considerar más atributos de los que realmente son necesarios para la aplicación, ya que durante la fase de refinamiento de datos se eliminarán los que no sean necesarios.

- *Refinamiento de datos.* Durante esta fase, se refinará la lista inicial de atributos, de modo que éstos constituyan una descripción precisa de los tipos de datos necesarios en la base de datos. Esta etapa es vital para incluir las sugerencias de tantos usuarios de la base de datos como sea posible; las personas que utilizan la base de datos son quienes mejor conocen la clase de información que necesitan obtener de la misma.
- *Consideraciones de las relaciones.* Durante la tercera fase del diseño de una base de datos, se debe pensar sobre las relaciones futuras entre los atributos. Esto puede ayudar a determinar si se necesitarán utilizar varias bases de datos.

Durante las tres fases del diseño, los usuarios potenciales de la base de datos deben ser consultados para determinar la clase de información que esperan le suministre la misma. Pero, ¿qué clase de informes deseamos de la base de datos? ¿Qué clase de consultas harán los empleados de la base de datos? Haciendo continuamente este tipo de preguntas, servirá de ayuda para determinar lo que es importante o no.

Hay que tener en cuenta que incluso después de las fases de diseño de la base de datos, el diseño del archivo no es un conjunto rígido. Si es necesario, pueden hacerse cambios en el diseño posteriormente. Sin embargo, si en el diseño de la base de datos para su aplicación se sigue una aproximación sistemática, hay mayor probabilidad de crear una base de datos que proporcione la información que necesita y de que no sea necesario volverla a diseñar de nuevo.

### 1.1.10. Modelo relacional

Para describir la estructura de una base de datos es necesario definir el concepto de modelo de datos, el cual es una colección de herramientas conceptuales para describir datos, relaciones entre ellos, semántica asociada a los datos y restricciones de consistencia.

El modelo relacional representa los datos y las relaciones entre los datos mediante una *colección de tablas bidimensionales*, cada una de las cuales tiene un número de columnas (atributos) con nombres únicos.

#### 1.1.10.1 Tablas

Una tabla en un sistema relacional se compone de una fila de cabeceras de columnas junto con cero o más filas de valores de datos (diferente número de filas de datos en diferentes momentos). Para una tabla dada:

- a) La fila de cabeceras de columna especifica una o más columnas (dando, entre otras cosas, un tipo de datos para cada una).
- b) Cada fila de datos contiene un solo valor escalar para cada una de las columnas especificadas en la fila de cabeceras de columna. Además, todos los valores de una columna dada tienen el mismo tipo de datos, a saber, el tipo especificado en la fila de cabeceras de columna para esa columna.

#### 1.1.10.2. Estructura de datos relacional

Los términos especiales más importantes en relación con la estructura de los datos en el modelo relacional se muestran en la Figura 1.3 y son [12]:

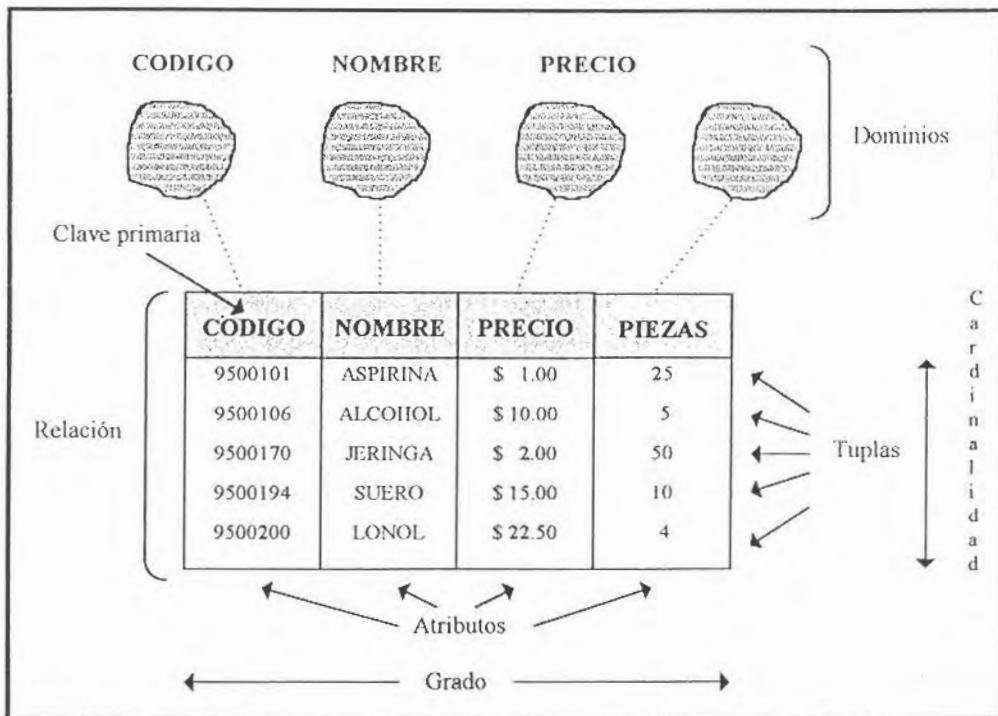


Figura 1.3 Términos de la estructura de los datos

- **Relación.** Corresponde a lo que hasta ahora hemos llamado tabla.
- **Tupla.** Corresponde a una fila de esa tabla y cuyos atributos están relacionados. El número de tuplas se denomina *cardinalidad* y el número de atributos se llama *grado*.
- **Clave primaria.** Es un identificador único para la tabla; es decir, una columna o combinación de columnas con la siguiente propiedad: nunca existen dos filas de la tabla con el mismo valor en esa columna o combinación de columnas.
- **Dominio.** Es una colección de valores, de los cuales uno o más atributos (columnas) obtienen sus valores reales.

Terminología de la estructura de datos:

Término relacional formal

- Relación
- Tupla
- Cardinalidad
- Atributo
- Grado
- Clave primaria
- Dominio

Equivalentes informales

- Tabla
- Fila o registro
- Número de filas
- Columna o campo
- Número de columnas
- Identificador único
- Fondos de valores legales

### 1.1.10.3. Claves

En términos informales, la *clave primaria* de una relación es sólo un identificador único para esa relación. La clave primaria puede ser compuesta, esto quiere decir que en una relación el único identificador (osea, la clave primaria) es el atributo compuesto formado por la combinación de todos los atributos de la relación; es decir, la relación es "toda clave".

También es posible, aunque poco usual, tener una relación con más de un identificador único. En una situación así, se dice que la relación tiene varias *claves candidatas*; en este caso se escogería una de esas claves candidatas como clave primaria, y las demás quedarían como *claves alternativas*.

La clave candidata se define de la siguiente manera. El atributo K (posiblemente compuesto) de la relación R es una clave candidata de R si y sólo si satisface las siguientes dos propiedades, independientes del tiempo:

- a) *Unicidad*. En cualquier momento dado, no deben existir dos tuplas en R con el mismo valor de K.
- b) *Minimalidad*. Si K es compuesto, no podremos eliminar ningún componente de K sin destruir la propiedad de unicidad.

Toda relación tiene por lo menos una clave candidata, porque como ya se mencionó anteriormente, las relaciones no contienen tuplas repetidas. En la práctica, las relaciones tienden a tener una y sólo una clave candidata, pero sin duda es posible que tengan más. Vamos a suponer, por ejemplo, que tenemos una relación llamada ELEMENTOS, la cual representa la tabla de elementos químicos. Cada elemento tiene un nombre único, un símbolo único y un número atómico único. Es evidente que la relación tiene tres claves candidatas distintas.

Ahora bien, del conjunto de claves candidatas de una relación dada, se elige una y sólo una como clave primaria de esa relación; las demás, si existen, se llamarán claves alternativas. Así, una clave alterna es una clave candidata que no es la clave primaria. En el caso de ELEMENTOS, por ejemplo, podríamos elegir "símbolo del elemento" como clave primaria; por lo tanto "nombre del elemento" y "número atómico" serían entonces claves alternativas.

La importancia de las claves primarias radica en que constituyen el mecanismo de direccionamiento a nivel de tuplas básico en un sistema relacional. Es decir, el único modo garantizado por el sistema, de localizar alguna tupla específica es por el valor de su clave primaria.

Podemos definir una clave ajena así: una *clave ajena* es un atributo (quizá compuesto) de una relación R2 cuyos valores deben concordar con los de la clave primaria de alguna relación R1.

Un valor de clave ajena representa una referencia a la tupla donde se encuentra el valor correspondiente de la clave primaria (la tupla referida o tupla objetivo). Por tanto, el problema de garantizar que la base de datos no incluya valores no válidos de una clave ajena se conoce como el problema de la *integridad referencial* (vista anteriormente). La

restricción según la cual los valores de una clave ajena determinada deben concordar con los valores e la clave primaria correspondiente se conoce como *restricción referencial*. La relación que contiene a la clave ajena se conoce como *relación referencial* y la relación que contiene a la clave primaria correspondiente se denomina *relación referida* o *relación objetivo*.

#### 1.1.10.4. Propiedades de las relaciones

Las relaciones poseen ciertas propiedades, todas ellas consecuencia inmediata de la definición de "relación" y todas ellas muy importantes. Las propiedades son cuatro. Dentro de una relación dada:

1. No hay tuplas repetidas.
  2. Las tuplas no están en ningún orden.
  3. Los atributos tampoco están ordenados.
  4. Todos los valores de los atributos son atómicos.
- *No existen tuplas repetidas*. Esta propiedad es consecuencia del hecho del que el cuerpo de la relación es un conjunto matemático (es decir, un conjunto de tuplas), y en matemáticas los conjuntos por definición no incluyen elementos repetidos.

Un corolario importante de la ausencia de tuplas repetidas es que siempre existe una clave primaria. Como las tuplas son únicas, es evidente que por lo menos la combinación de todos los atributos de la relación tiene la propiedad de unicidad, de modo que al menos la combinación de todos los atributos puede servir (en caso necesario) como clave primaria. Desde luego, en la práctica casi nunca resulta necesario implicar a todos los atributos; suele ser suficiente alguna combinación más limitada.

- *Las tuplas no están ordenadas* (de arriba hacia abajo). Esta propiedad también se desprende del hecho de que el cuerpo de una relación es un conjunto matemático. Los conjuntos en matemáticas no son ordenados. Por tanto, no puede hablarse de "la quinta tupla" o la "tupla 97" o "la primera tupla" de una relación, y tampoco existe "la siguiente tupla"; en otras palabras, no existe el concepto de direccionamiento por posición, y tampoco el de "adyacencia".
- *Los atributos no están ordenados* (de izquierda a derecha). Esta propiedad se desprende del hecho de que la cabecera de una relación se define también como conjunto (es decir, un conjunto de atributos o, dicho en forma más precisa, de pares atributo-dominio). Así pues, no puede hablarse de "el primer atributo" o "el segundo atributo", etc., ni de "el siguiente atributo" (una vez más, no existe el concepto de "adyacencia").
- *Todos los valores de los atributos son atómicos*. Una forma más precisa de expresar esta propiedad es: "todos los valores de los atributos simples son atómicos". Se trata, desde luego, de una consecuencia del hecho de que todos los dominios subyacentes son a su vez simples; es decir, contienen sólo valores atómicos. Esta propiedad también se puede expresar de la siguiente manera: en cada posición de una tupla y un atributo dentro de la

relación, siempre existe un solo valor, nunca una lista de valores. O en otra forma equivalente: las relaciones no contienen grupos repetitivos. Si una relación satisface esta condición, se dice que está normalizada.

La razón para insistir en la normalización de todas las relaciones es la siguiente. En esencia, una relación normalizada es una estructura más simple desde el punto de vista matemático, que una no normalizada. En consecuencia, los operadores correspondientes son también más sencillos, y su número es menor.

### 1.1.11. Normalización

Se entiende por *normalización* la descomposición o subdivisión de una relación en dos o más relaciones para evitar la redundancia. Tales relaciones satisfacen la condición de que, en cada posición de fila y columna dentro de la relación, siempre hay un, y sólo uno, valor de datos, nunca un conjunto de múltiples valores.

Los argumentos a favor de las relaciones normalizadas, pueden resumirse en los siguientes puntos :

- a) Las relaciones normalizadas se pueden visualizar como tablas y por medio de esta representación se facilita el empleo y la comprensión de los sistemas relacionales, y la que ayuda a razonar sobre el comportamiento de esos sistemas.
- b) La normalización hace posible un mecanismo más simple para nombrar y especificar direcciones de los datos que el que se necesitaría en caso de no haberla. Se puede especificar la dirección de un valor individual en una base de datos relacional si se especifica el nombre de la tabla en la cual está contenido más el nombre de la columna en la cual está contenido más el valor de la clave primaria de la fila en la cual está contenido. Dicho de otro modo, la estructura de filas y columnas de una relación normalizada constituye una especie de sistemas de coordenadas con el cual es posible identificar con facilidad y sin ambigüedad cada uno de los valores de la base de datos.
- c) Una relación normalizada es una estructura más simple, que una no normalizada; todos los datos están representados de una y sólo una manera, a saber, como valores explícitos dentro de filas dentro de columnas dentro de tablas. Los operadores necesarios para manejar relaciones normalizadas son también más simples y menos numerosos.
- d) Por último, existe toda una teoría sólida (la teoría de dependencias, llamada también teoría de normalización adicional) disponible para utilizarse en el diseño de bases de datos fundamentadas en relaciones normalizadas.

### 1.1.12. Dependencia funcional

La normalización se basa en la dependencia funcional. El concepto de dependencia funcional se tomó de las matemáticas.  $[Y = f(X)]$ , Y es función de X si el valor de Y está siempre determinado por el valor de X .

La **dependencia funcional (DF)** se define: Dados dos atributos A y B de una relación R se dice que B es funcionalmente dependiente del atributo A si para cada valor de A existe un valor de B, y sólo uno, asociado con él. En otros términos: si en cualquier instante, conocido el valor de A, podemos conocer el valor de B. Se simboliza por:

$$A \longrightarrow B$$

Tanto A como B pueden ser un conjunto de atributos en lugar de atributos simples. La dependencia funcional establece condiciones entre atributos pertenecientes a la misma relación. No permite establecer condiciones entre atributos de diferentes relaciones.

Las dependencias funcionales se determinan al estudiar las propiedades de todos los atributos de la relación y deducir cómo están relacionados los atributos entre sí.

La dependencia funcional está íntimamente ligada con el concepto de clave. Se debe encontrar primero la clave candidata, para ello preciso estudiar las dependencias funcionales y, a partir de ellas, obtener el mínimo conjunto posible de atributos tales que, una vez conocidos sus valores en la tupla, los demás queden definidos.

#### 1.1.12.1 Determinante

**Determinante:** Si  $A \longrightarrow B$  es una DF y A no es funcionalmente dependiente de B se dice que A es el determinante de B.

Un determinante son todos los atributos situados en el lado izquierdo de una DF.

#### 1.1.12.2 Primera forma normal: 1FN

Una relación está en primera forma normal si todo atributo contiene un valor indivisible. Por lo general una relación en 1FN tiene problemas a la hora de hacer actualizaciones por la información redundante que posee.

Las anomalías de almacenamiento, que se deben a la presencia de campos no clave en la relación, se pueden subsanar de la siguiente forma:

- Dividiendo la relación universal en nuevas relaciones.
- Cada relación tiene la propiedad de que su clave, en su totalidad, es necesaria para definir cada uno de los campos no clave.

Al proceso de dividir cualquier relación en dos o más relaciones se llama *proceso de normalización*. Consiste en reemplazar las relaciones por proyecciones adecuadas, de tal forma que la reunión natural de las proyecciones genere la relación original, es decir, que no se produzca pérdida de la información. Incluso las nuevas relaciones pueden contener



información que no se podía representar originalmente (una nueva tupla en alguna de las nuevas relaciones), pero siempre conservando las dependencias funcionales.

*Descomposición sin pérdida*, es la descomposición de una relación R en R1, R2, ... RN, tal que:

$$R = R1 * R2 * \dots * RN$$

Cuando se actualiza la base de datos, el sistema debe poder comprobar que la actualización no va a generar una relación ilegal, es decir, una que no satisfaga todas las dependencias funcionales establecidas.

Para llevar a cabo el proceso de normalización es aconsejable dar los siguientes pasos:

1. Elegir una clave primaria que puede representar de forma única a cada tupla de la relación.
2. Construir un diagrama de dependencia en función de esas claves.
3. Construir las nuevas relaciones basándose en dichas claves.

### 1.1.12.3. Dependencia funcional total: 2FN

**Dependencia funcional total:** En una relación R, un atributo o colección de atributos B tiene una dependencia funcional total de otra colección de atributos A de la relación R, si B es funcionalmente dependiente de todos los atributos de A pero no de un subconjunto de A.

### 1.1.12.4. Dependencia transitiva: 3FN

Supongamos la relación R(A, B, C). Si  $A \longrightarrow B$ ,  $B \longrightarrow C$  y  $B \not\longrightarrow A$ ; entonces se dice que C depende transitivamente de A y se puede formar la cadena  $A \longrightarrow B \longrightarrow C$ .

En un diagrama de dependencia funcional, C es transitivamente dependiente de A si tiene la siguiente situación:



Se puede descomponer en dos relaciones de la forma



### 1.1.12.5. Forma Normal de Boyce Codd: BCFN

Una relación R se encontrará en la forma normal Boyce Codd si cada determinante de dicha relación es una clave candidata.

Si una relación no está en BCFN deberá descomponerse en otras dos de la siguiente manera:

Supongamos que la siguiente relación  $R(A,B,C,D)$  no está en BCFN, y sea  $C \longrightarrow D$  una de las dependencias funcionales que ocasionan lo anterior, siendo C un determinante pero no una clave candidata. Se formarán entonces dos nuevas relaciones:  $R_1(A,B,C)$  y  $R_2(C,D)$ . Después se comprueba si  $R_1$  y  $R_2$  están en BCFN, en caso de no ser así, se repite el mismo proceso.

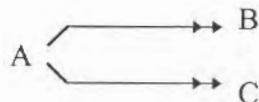
### 1.1.12.6. Dependencias de valores múltiples: 4FN

A partir de las dependencias funcionales se ha realizado la descomposición de las relaciones en 1FN, 2FN, 3FN y BCFN, con lo que se puede decir concluye el proceso de normalización. Sin embargo, en ocasiones, estas formas son insuficientes para eliminar la redundancia y las anomalías de actualización si alguna relación contiene dependencias de valores múltiples, con lo que se tendrá que realizar una normalización adicional.

En la **dependencia de valores múltiples** un atributo A, determina un conjunto de valores múltiples B. Se representa de la siguiente manera:



Supongamos la relación  $R(\underline{A}, B, C)$  con las siguientes dependencias multivaluadas:



se podrá llevar a cabo su descomposición sin pérdida de información en dos relaciones  $R_1(\underline{A}, B)$  y  $R_2(\underline{A}, C)$ .

### 1.1.13. Modelo entidad-relación

Otra forma de llevar a cabo la normalización es a través del Modelo entidad-relación, el cual sólo tiene en cuenta las DF en la etapa final del diseño.

La terminología utilizada por este modelo es la siguiente:

- *Entidad*. Se define como un tipo concreto de información que tiene interés para la empresa. Generalmente son nombres. Ejemplos: coche, cliente, proveedor, producto. Una

entidad está compuesta por *atributos* (campos). El atributo o conjunto de atributos que identifican a una tupla se llama *clave de la entidad*.

- *Relación*. Se define como una correspondencia, asociación o conexión entre dos o más entidades. Generalmente son verbos. Ejemplo: proveedor VENDE productos. Cuando la correspondencia se establece entre dos entidades se llama relación binaria. Toda relación tiene una clave que identifica a una ocurrencia de la entidad.

Este modelo se puede representar gráficamente mediante los diagramas de *ocurrencias ER* o por los diagramas de *tipo ER*.

Consideremos las entidades PROVEEDOR y PRODUCTO ligadas entre sí por la correspondencia VENDE. Ambos tipos de diagramas se muestran en la Figura 1.4

En el diagrama de tipo ER aparecen:

- *Rectángulos* que representan las entidades. Debajo aparece el atributo clave subrayado. Los puntos suspensivos representan a los demás atributos que no son clave.
- *Rombos* que representan las relaciones.
- *Líneas* que conectan las entidades con sus relaciones.
- *Números* encima de cada línea, que indica el grado de relación entre ambas entidades.

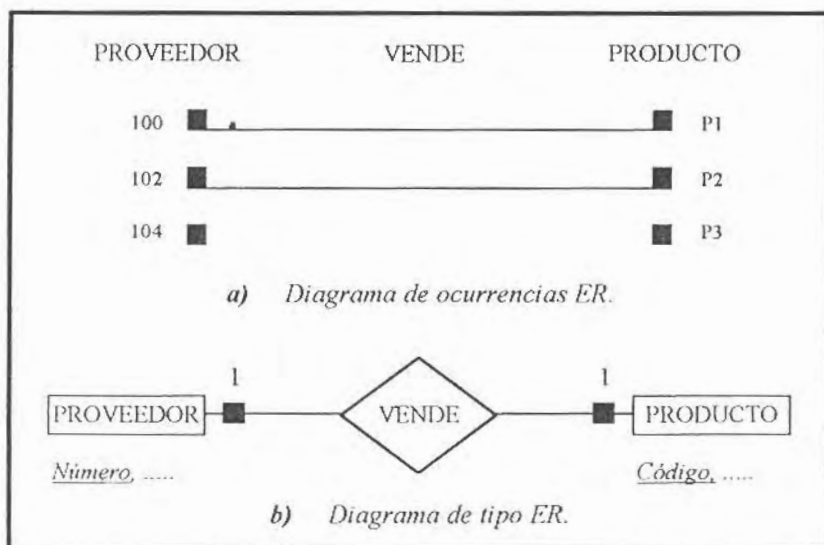


Figura 1.4 Tipos de diagramas entidad-relación.

### 1.1.13.1. Grado de las relaciones

La Figura 1.5 muestra los grados de las relaciones asociadas a los diagramas de tipo ER.





Relación	Diagrama tipo ER
Una a una	
Una a muchas	
Muchas a una	
Muchas a muchas	

Figura 1.5 Tipo de relaciones.

### 1.1.13.2. Tipos de participación en una entidad

Según la participación de las ocurrencias de una entidad en la relación se pueden clasificar en:

- *Opcionales.* No todas las ocurrencias de una entidad tienen que participar en la relación. Se representan con un rectángulo.
- *Obligatorias.* Todas las ocurrencias en una entidad deben participar en la relación. Se representan con un pequeño rectángulo con un punto dentro del rectángulo de la relación.

En la Figura 1.6 la entidad PROVEEDOR es obligatoria, mientras que la entidad PRODUCTO es opcional. Todos los proveedores deben vender un producto, pero puede haber algún producto que no sea vendido por ningún proveedor.

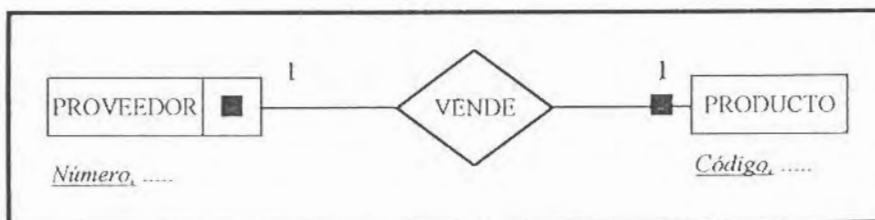


Figura 1.6 Diagrama entidad-relación.

### 1.1.13.3. Casos de participación de las entidades

Todos los grados de las relaciones presentan los mismos casos de participación de las entidades. Por simplicidad se considera el caso de las relaciones binarias cuando el grado de correspondencia es uno a uno (1:1) sobre las entidades PROVEEDOR y PRODUCTO.

Los casos de participación pueden ser:

- *Opcional-opcional.* Cada proveedor vende sólo un producto y cada producto es vendido por un único proveedor. Implica que puede haber proveedores que no vendan ningún producto y algún producto puede no ser vendido.
- *Obligatorio-opcional.* Cada proveedor obligatoriamente vende sólo un producto y cada producto sólo es vendido por un único proveedor. Algún producto puede que no sea vendido por ningún proveedor.
- *Opcional-obligatorio.* Cada producto es vendido obligatoriamente por un único proveedor. Puede haber proveedores que no vendan ningún producto.
- *Obligatorio-obligatorio.* Todos los proveedores venden sólo un producto y cada producto es vendido por un único proveedor.

Estos casos de participación son mutuamente excluyentes, es decir, una vez elegido uno, los demás serán ilegales.

La Figura 1.7 muestra estos casos utilizando el diagrama de ocurrencias ER. La Figura 1.8 muestra los mismos casos utilizando el diagrama de tipo ER.

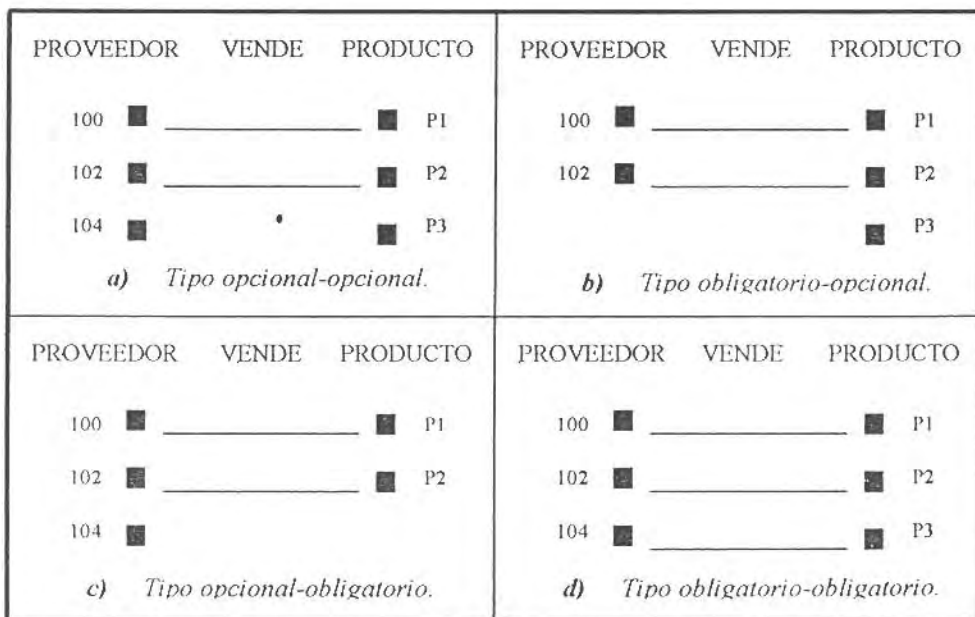


Figura 1.7 Diagramas de ocurrencias ER con diferentes participaciones.

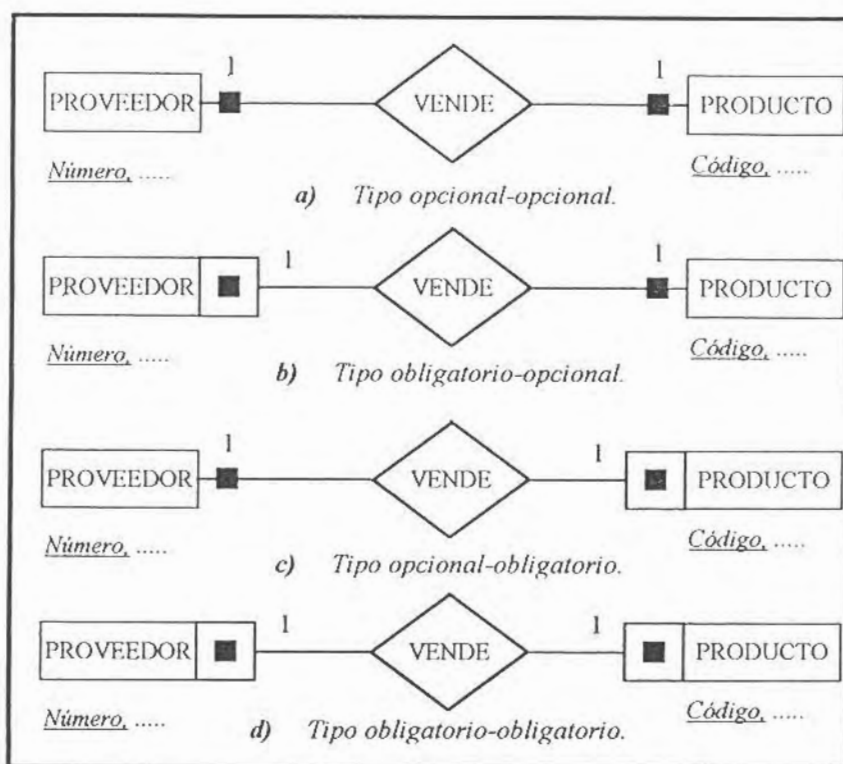


Figura 1.8 Diagrama de tipo ER con diferentes participaciones.

#### 1.1.13.4. Obtención de las relaciones a partir del diagrama

El diseño de una base de datos por el modelo ER requiere los siguientes pasos:

1. Dibujar un diagrama de tipo ER con todas las entidades y relaciones que son de interés para la empresa.
2. Deducir un conjunto de relaciones preliminares, cada una con su clave primaria.
3. Incluir los atributos de interés que aún no estén asignados a las relaciones, de forma que las relaciones estén normalizadas.

#### 1.1.13.5. Relaciones preliminares para las correspondencias binarias de grado 1:1

Para obtener las relaciones hay que examinar las alternativas lógicas y escoger la mejor de ellas a partir de las siguientes reglas:

**Regla 1.** Si el grado de la correspondencia binaria es 1:1 y es obligado el tipo de participación de ambas entidades, sólo es necesaria una relación. Como clave primaria de la relación se puede tomar cualquiera de las claves de la entidad.

La Figura 1.9 muestra el contenido de la relación PROVEEDOR (NUMERO, NOMBRE, CODIGO, CONCEPTO); se han añadido los atributos nombre del proveedor y el concepto del producto.

Relación PROVEEDOR

<u>NUMERO</u>	<u>NOMBRE</u>	<u>CODIGO</u>	<u>CONCEPTO</u>
100	AUTREY	9500101	ASPIRINA
110	NADRO	9500106	ALCOHOL
120	DROGUEROS	9500200	LONOL

Figura 1.9 Contenido de una relación simple.

Esta relación, al ser las dos entidades obligatorias, no presenta ningún tipo de anomalía.

Sin embargo, cuando algunas de las relaciones deja de ser obligatoria para ser opcional, aparecen atributos en blanco (Figura 1.10). Para subsanar estas anomalías es necesario definir una nueva regla.

<u>NUMERO</u>	<u>NOMBRE</u>	<u>CODIGO</u>	<u>CONCEPTO</u>
100	AUTREY	9500101	ASPIRINA
110	NADRO	9500106	ALCOHOL
120	DROGUEROS	9500200	LONOL
130	BENAVIDES		
		9500225	AFRINEX

Figura 1.10 Relación simple con anomalías

**Regla 2.** Si el grado de la correspondencia binaria es 1:1 y el tipo de participación de una entidad es obligatorio y el de la otra opcional, son necesarias dos relaciones. Cada una contendrá la información concerniente a una entidad y su clave primaria será la clave de la entidad correspondiente. La clave de la entidad opcional se añadirá como un atributo más en la relación cuyo tipo de participación sea obligatorio.

Si se considera a la relación PROVEEDORES como obligatoria y a la relación PRODUCTOS como opcional se obtienen dos relaciones (Figura 1.11) con los siguientes atributos:

PROVEEDORES (NUMERO, NOMBRE, CODIGO)

PRODUCTOS (CODIGO, CONCEPTO)

Relación PROVEEDORES

<u>NUMERO</u>	<u>NOMBRE</u>	<u>CODIGO</u>
100	AUTREY	9500101
110	NADRO	9500106
120	DROGUEROS	9500200
130	BENAVIDES	9500225

Relación PRODUCTOS

<u>CODIGO</u>	<u>CONCEPTO</u>
9500101	ASPIRINA
9500106	ALCOHOL
9500200	LONOL
9500225	AFRINEX
9500229	LOTRIMIN

**Figura 1.11** Contenido de las relaciones cuando una entidad es obligatoria y la otra opcional.

Sin embargo, cuando las dos relaciones son opcionales, vuelven a aparecer atributos en blanco. Es el caso del atributo CODIGO en la relación PROVEEDORES. Para subsanar esta anomalía es necesario definir una tercera regla.

**Regla 3.** Si el grado de la correspondencia es 1:1 y el tipo de participación en ambas entidades es opcional, son necesarias tres relaciones, una para cada entidad y otra para la correspondencia. La clave de las relaciones de las entidades es la clave primaria de la entidad correspondiente. La relación con la correspondencia deberá contener las claves de las entidades.

Según esta regla debe aparecer una nueva relación VENDE, siendo sus atributos los campos clave de las otras relaciones, VENDE (NUMERO, CODIGO) (Figura 1.12). Cualquier valor de los atributos aparece sólo una vez por ser la relación 1:1.

Las tres relaciones son:

**PROVEEDORES** (NUMERO, NOMBRE)

**PRODUCTOS** (CODIGO, CONCEPTO)

**VENDE** (NUMERO, CODIGO)

Relación PROVEEDORES

<u>NUMERO</u>	<u>NOMBRE</u>
100	AUTREY
110	NADRO
120	DROGUEROS
130	BENAVIDES

Relación PRODUCTOS

<u>CODIGO</u>	<u>CONCEPTO</u>
9500101	ASPIRINA
9500106	ALCOHOL
9500200	LONOL
9500225	AFRINEX
9500229	LOTRIMIN

Relación VENDE

<u>NUMERO</u>	<u>CODIGO</u>
100	9500101
110	9500106
120	9500200

**Figura 1.12** Contenido de las tres relaciones aplicando la regla 3.



Con tres relaciones es la única manera de que no se produzca ningún tipo de anomalías. Puede haber cualquier número de proveedores y de productos, sin que se produzca redundancia en los datos almacenados, debido a que la relación VENDE es uno a uno y contiene exactamente el producto que vende cada proveedor.

#### 1.1.13.6. Relaciones preliminares para las correspondencias binarias de grado 1:n

Las relaciones uno a uno no consideran la posibilidad de que un proveedor pueda vender varios productos o que un producto pueda ser suministrado por varios proveedores; es decir, no prevén la relación uno a muchos (1:n).

Las relaciones binarias de grado 1:n necesitan dos reglas. La regla a aplicar depende de la entidad que ocupe el lado "muchas" de la relación, según sea obligatoria u opcional. En el resultado final no influye para nada la entidad que ocupe el lado 1.

**Regla 4.** Si el grado de la correspondencia binaria es 1:n y la entidad del lado "n" es obligatoria, se necesitan dos relaciones, una para cada entidad. La clave de las relaciones de las entidades es la clave primaria de la entidad correspondiente. La relación de la entidad "n" contiene la clave de la entidad "1".

Consideremos el caso de que un proveedor puede vender varios productos. Se tiene la siguiente situación: todos los productos tienen un proveedor y cada proveedor puede servir varios productos. La Figura 1.13 muestra este caso.

Las relaciones que se forman son:

**PROVEEDORES** (NUMERO, NOMBRE)

**PRODUCTOS** (CODIGO, CONCEPTO, NUMERO)

Relación PROVEEDORES

NUMERO	NOMBRE
100	AUTREY
110	NADRO *
120	DROGUEROS
130	BENAVIDES

Relación PRODUCTOS

CODIGO	CONCEPTO	NUMERO
9500101	ASPIRINA	100
9500106	ALCOHOL	110
9500200	LONOL	120
9500225	AFRINEX	100
9500229	LOTRIMIN	100

**Figura 1.13** Contenido de las relaciones cuando se aplica la regla 4.

Aparentemente parece que las relaciones son correctas, pero si se dan altas de productos, el atributo NUMERO aparece en blanco, surge entonces una anomalía.

**Regla 5.** Si el grado de la correspondencia binaria es 1:n y la entidad del lado "n" es opcional se necesitan tres relaciones, una para cada entidad y otra para la correspondencia. La clave de las relaciones de las entidades es la clave primaria de la entidad correspondiente; la relación con la correspondencia contendrá las claves de las entidades.

Las tres relaciones son:

**PROVEEDORES** (NUMERO, NOMBRE, CODIGO)

**PRODUCTOS** (CODIGO, CONCEPTO)

**VENDE** (NUMERO, CODIGO)

Una muestra de las tuplas de las relaciones se detalla en la Figura 1.14

Relación PROVEEDORES		Relación PRODUCTOS		Relación VENDE	
<u>NUMERO</u>	NOMBRE	<u>CODIGO</u>	CONCEPTO	<u>NUMERO</u>	<u>CODIGO</u>
100	AUTREY	9500101	ASPIRINA	100	9500101
110	NADRO	9500106	ALCOHOL	100	9500106
120	DROGUEROS	9500200	LONOL	110	9500200
130	BENAVIDES	9500225	AFRINEX	110	9500225
		9500229	LOTTRIMIN		

Figura 1.14 Contenido de las relaciones cuando se aplica la regla 5.

### 1.1.13.7. Relaciones preliminares para las correspondencias binarias de grado $m:n$

Cuando el grado de la correspondencia es muchos a muchos son necesarias tres relaciones, independientemente del tipo de participación de las entidades. Es necesario aplicar una nueva regla.

**Regla 6.** Si el grado de la correspondencia binaria es  $m:n$  se necesitan tres relaciones, una para cada entidad y la otra para la correspondencia. La clave de las relaciones de las entidades es la clave primaria de la entidad correspondiente. La relación con la correspondencia deberá contener las claves de la entidades.

Si no existe ninguna DF entre los atributos de una relación, la clave primaria está formada por todos los atributos de la misma. Estas relaciones se denominan de *clave total* y están siempre normalizadas. Es lo que sucede a la relación VENDE, cuya clave primaria será (NUMERO, CODIGO).

La Figura 1.15 muestra el contenido de las relaciones.

Relación PROVEEDORES		Relación PRODUCTOS		Relación VENDE	
<u>NUMERO</u>	NOMBRE	<u>CODIGO</u>	CONCEPTO	<u>NUMERO</u>	<u>CODIGO</u>
100	AUTREY	9500101	ASPIRINA	100	9500101
110	NADRO	9500106	ALCOHOL	100	9500106
120	DROGUEROS	9500200	LONOL	110	9500101
130	BENAVIDES	9500225	AFRINEX	110	9500106
		9500229	LOTTRIMIN	120	9500106

Figura 1.15 Contenido de las relaciones cuando se aplica la regla 6.

### 1.1.13.8. Relaciones preliminares para las correspondencias ternarias

Si mediante una correspondencia binaria no se puede representar alguna situación de la vida real es necesario recurrir a las correspondencias de orden superior, por ejemplo a las de orden 3.

Para este tipo de correspondencia se aplica la siguiente regla:

**Regla 7.** Si existe una correspondencia ternaria se necesitan cuatro relaciones, una para cada entidad, y una más para la correspondencia. La clave de las relaciones de las entidades es la clave primaria de la entidad correspondiente. La relación que contenga los datos de la correspondencia contendrá entre sus atributos las claves de las entidades.

En general, para una correspondencia de grado  $n$ , serán necesarias  $n + 1$  relaciones preliminares.

## 1.2. SOFTWARE

El software ha superado al hardware como la clave del éxito de muchos sistemas basados en computadoras. Tanto si se utiliza la computadora para llevar un negocio, controlar un producto o capacitar un sistema, el software es el factor que marca la diferencia. Lo que diferencia a una compañía de su competidora es la suficiencia y oportunidad de la información dada por el software. Así mismo, el diseño de un producto de software "amigable a los humanos" va a estar por encima de los productos con características funciones similares.

Durante las tres primeras décadas de la informática, el principal desafío era el desarrollo del hardware de las computadoras, de forma que se redujera el costo de procesamiento y almacenamiento de datos. A lo largo de la década de los ochenta, los avances en microelectrónica han dado como resultado una mayor potencia de cálculo a la vez que una reducción del costo. Hoy, el problema es diferente. El principal desafío es mejorar la calidad (y reducir el costo) de las soluciones basadas en computadoras, soluciones que se implementan con el software.

La potencia de las grandes computadoras de la era de los ochenta está hoy disponible en una computadora personal. Las enormes capacidades de procesamiento y almacenamiento del hardware moderno representan un gran potencial de cálculo. El software es el mecanismo que nos facilita utilizar y explotar este potencial.

### 1.2.1. Definición

"Software, es el nombre genérico que se da a los programas de una computadora, pero que implica una responsabilidad adicional: asegurar que el programa o sistema cumple por completo con sus objetivos, opera con eficiencia, está adecuadamente documentado y es sencillo de operar" [18].

### 1.2.2. Características del software

Para poder comprender lo que es el software, es importante examinar las características del software que lo diferencian de otras cosas que los hombres pueden construir. Cuando se construye hardware, el proceso creativo humano (análisis, diseño, construcción, prueba) se traduce finalmente en una forma física. Si construimos una nueva computadora, nuestro boceto inicial, diagramas formales de diseño y prototipo de prueba, evolucionan hacia un producto físico.

Debido a que el software es un elemento del sistema que es lógico, en lugar de físico, va a tener características distintas a las del hardware:

- *El software se desarrolla, no se fabrica en un sentido clásico.* El desarrollo de software y del hardware se buena calidad, se puede realizar mediante un un buen diseño, pero la construcción del hardware puede introducir problemas de calidad que no existen (o son

fácilmente corregibles) en el software. Ambas actividades requieren la construcción de un producto, pero los métodos son diferentes.

- *El software no se estropea.* Los fallos en el hardware se van presentando con el paso del tiempo debido al desgaste que sufren sus componentes (o circuitos), debido a factores climatológicos y del manejo de los mismos.
- Sin embargo, el software no es susceptible a los males del entorno que hacen que el hardware se deteriore. Un componente del hardware se puede sustituir por una pieza de repuesto. Pero para el software no hay piezas de repuesto. Cada fallo en el software indica un error en el diseño.
- *La mayoría del software se construye a medida, en vez de ensamblar componentes existentes.* Consideremos la forma en que se diseña y se construye el hardware de control para un producto basado en microprocesador. El ingeniero de diseño construye un sencillo esquema de la circuitería digital, hace algún análisis fundamental para asegurar que se realiza la función adecuada y va al catálogo de ventas de componentes digitales existentes. Cada circuito integrado tiene un número de pieza, una función definida y válida, una interfaz bien definida y un conjunto estándar de criterios de integración. Después de seleccionar cada componente, puede solicitarse la compra.

Por desgracia, los diseñadores del software no disponen de esa comodidad que se acaba de describir. Con unas pocas excepciones, no existen catálogos de componentes de software. Se puede comprar software ya desarrollado, pero sólo como una unidad completa, no como componentes que puedan reensamblarse en nuevos programas<sup>6</sup>.

### 1.2.3. Aplicaciones del software

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales (es decir, un algoritmo). Para determinar la naturaleza de una aplicación de software, hay dos factores importantes que se deben considerar: el *contenido* y el *determinismo* de la información. El *contenido* se refiere al significado y a la forma de la información de entrada y de salida. Por ejemplo, muchas aplicaciones bancarias usan unos datos de entrada muy estructurados (una base de datos) y producen informes con determinados formatos. El software que controla una máquina automática (por ejemplo: un control numérico) actúa sobre elementos de datos discretos con una estructura limitada y produce órdenes concretas para las máquinas en rápida sucesión.

El *determinismo* de la información se refiere a la predecibilidad del orden y del tiempo de llegada de los datos. Un programa de ingeniería acepta datos que están en un orden predefinido, ejecuta el algoritmo sin interrupción y produce los datos resultantes en un informe o formato gráfico. Se dice que tales aplicaciones son *determinadas*. Un sistema operativo multiusuario, por otra parte, acepta entradas que tienen un contenido variado y que se producen en instantes arbitrarios, ejecuta algoritmos que pueden ser interrumpidos

<sup>6</sup> Pero esta situación está cambiando rápidamente con el uso de la Programación Orientada a Objetos, la cual permite crear este tipo de situaciones.

por condiciones externas y produce una salida que depende de una función del entorno y del tiempo. Las aplicaciones con estas características se dice que son *indeterminadas*.

Algunas veces es difícil establecer categorías genéricas para las aplicaciones del software que sean significativas. Conforme aumenta la complejidad del software, es más difícil establecer compartimentos nítidamente separados. Las siguientes áreas del software indican la amplitud de las posibilidades de aplicación.

- *Software de sistemas.* El software de sistemas son programas desarrollados para servir a otros programas. Algunos programas de sistemas como los compiladores, procesan estructuras de información complejas pero determinadas. Otras aplicaciones de sistemas, como ciertos componentes del sistema operativo, procesan datos en gran medida indeterminados. Este tipo de software se caracteriza por una fuerte interacción con el hardware de la computadora; una gran utilización por múltiples usuarios; una operación concurrente que requiere una planificación, una compartición de recursos y una sofisticada gestión de procesos; unas estructuras de datos complejas y múltiples interfaces externas.
- *Software de tiempo real.* El software que mide/analiza/controla sucesos del mundo real conforme ocurren, se denomina de tiempo real. Los elementos que forman parte de este tipo de software se encuentran: un componente de adquisición de datos que recolecta y da formato a la información recibida del entorno externo, un componente de análisis que transforma la información según lo requiera la aplicación, un componente de control/salida que responda al entorno externo y un componente de monitorización que coordina todos los demás componentes, de forma que pueda mantenerse la respuesta en tiempo real (típicamente en el rango de 1 milisegundo a 1 minuto). Un sistema de tiempo real debe responder dentro de unas ligaduras estrictas de tiempo.
- *Software de aplicación.* Es el conjunto de programas escritos para resolver problemas específicos. Estos problemas no derivan de disfunciones internas de la computadora, sino que es el usuario quien los plantea y pretende su resolución mediante el uso de la computadora.

Supongamos un empresario que cada fin de año tiene problemas para dejar su contabilidad bien cerrada, sin ningún error y en el tiempo justo; y para resolver este asunto decide valerse de la computadora. Este empresario no solucionará su problema si no dispone de toda una serie de programas, que la computadora pueda ejecutar, hechos a medida para resolver su problema de contabilidad.

Al conjunto de programas escritos para resolver un problema determinado se le llama *aplicación*. Así podemos hablar de aplicación de contabilidad, aplicación de nóminas, aplicación de almacenes, etc.

Cada programa que compone una aplicación sirve para la resolución de una de sus partes o de un problema determinado. Así por ejemplo, la aplicación de contabilidad contendrá un programa para construir el plan contable, otro para efectuar el diario contable y así sucesivamente.

Al diseñar todos los programas de una aplicación, puede ocurrir que estos sirvan para resolver un problema determinado de un usuario o que se utilicen para resolver el problema de muchos usuarios. Así podemos hablar de dos tipos de aplicaciones:

- a) Aplicaciones estandarizadas
- a) Aplicaciones a medida.

Pensemos en un colectivo sujeto a las mismas reglamentaciones, como es el caso de los farmacéuticos. Se puede construir una aplicación que resuelva sus problemas de almacenamiento de medicamentos: que controle todos los específicos, que informe de cuantos frascos se tienen en el almacén, o de su consumición, para efectuar los pedidos necesarios al laboratorio proveedor; que de a conocer en el momento requerido el valor total de los productos almacenados, etc. La aplicación que se construya para tal fin servirá a todos los farmacéuticos por igual; de ahí su nombre, estandarizada. Si un farmacéutico quiere solucionar un problema específico de su negocio mediante la computadora, se le diseñará una aplicación exclusivamente para él, o sea, se le hará a la medida de su problema. Cada vez se construyen menos aplicaciones a medida, ya que se acostumbran a retocar los programas de la aplicaciones estandarizadas para que contemplen las facetas que el usuario necesite.

- *Software de ingeniería y científico.* El software de ingeniería y científico está caracterizado por los algoritmos de “manejo de números”. Las aplicaciones van desde la astronomía a la vulcanología y desde la biología molecular a la fabricación automática. Sin embargo, las nuevas aplicaciones del área de ingeniería/ciencia se han alejado de los algoritmos convencionales numéricos. El diseño asistido por computadora (del inglés CAD), la simulación de sistemas y otras aplicaciones interactivas, han comenzado a tomar características del software del tiempo real e incluso del software de sistemas.
- *Software empotrado.* Los productos inteligentes se han convertido en algo común en casi todos los mercados de consumo e industriales. El software empotrado reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo. El software empotrado puede ejecutar funciones muy limitadas y curiosas (por ejemplo: el control de las teclas de un horno de microondas) o suministrar una función significativa y con capacidad de control (por ejemplo: funciones digitales en una automóvil, tales como control de la gasolina, de aceite, sistema de frenado, etc.).
- *Software de computadoras personales.* El mercado del software de las computadoras personales ha germinado en la pasada década. El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, entretenimientos, gestión de bases de datos, aplicaciones financieras, de negocios y personales, y redes o acceso a bases de datos externas son algunas de las cientos de aplicaciones. De hecho, el software de las computadoras personales continúa representando uno de los diseños del software más innovadores en el campo del software.
- *Software de inteligencia artificial.* El software de inteligencia artificial (IA) hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo. Actualmente, el área más activa de la IA es la

de los sistemas expertos, también llamados sistemas basados en el conocimiento. Otras áreas de aplicación para el software de IA es el reconocimiento de patrones (imágenes y voz), la prueba de teoremas y los juegos. Otra rama importante dentro del software de IA es la llamada redes neuronales artificiales. Una red neuronal simula la estructura de proceso del cerebro (las funciones de la neurona biológica).

#### 1.2.4. Problemas del desarrollo del software

Los problemas no se limitan al software que “no funciona correctamente”. Es más, el mal abarca los problemas asociados a cómo desarrollar software, cómo mantener el volumen cada vez mayor de software existente y cómo poder esperar mantenernos al corriente de la demanda creciente del software.

Los problemas que afligen al desarrollo del software se pueden caracterizar bajo muchas perspectivas diferentes, pero los responsables de los desarrollos del software se centran sobre los aspectos de fondo:

1. La planificación y estimación de costos son frecuentemente muy imprecisas.
2. La productividad de la comunidad del software no se corresponde con la demanda de sus servicios.
3. La calidad del software no llega a ser a veces ni aceptable.

Se han experimentado desajustes en los costos de hasta un orden de magnitud. Se ha errado en la planificación en meses o años. Se ha hecho muy poco para mejorar la productividad de los trabajadores del software. Los errores en los nuevos programas producen en los clientes insatisfacción y falta de confianza. Tales problemas son sólo las manifestaciones más visibles de otras dificultades del software:

- No tenemos tiempo de recoger datos sobre el proceso de desarrollo del software. Sin datos históricos como guía, la estimación no ha sido buena y los resultados previstos muy pobres. Sin una indicación sólida de la productividad, no podemos evaluar con precisión la eficacia de las nuevas herramientas, técnicas o estándares.
- La insatisfacción del cliente con el sistema terminado se produce demasiado frecuentemente. Los proyectos de desarrollo de software se realizan frecuentemente con sólo una vaga indicación de los requisitos del cliente. Normalmente, la comunicación entre el cliente y el que desarrolla el software es muy escasa.
- La calidad del software es normalmente cuestionable. Hemos empezado a comprender recientemente la importancia de la prueba sistemática y técnicamente completa del software. Están comenzando a emerger conceptos cuantitativos sólidos sobre la fiabilidad del software y las garantías de calidad.
- El software existente puede ser muy difícil de mantener. La tarea de mantenimiento del software se lleva la mayor parte de todo el dinero invertido en el software. El mantenimiento no se ha considerado un criterio importante en la aceptación del software.



## 1.3 INGENIERIA DEL SOFTWARE

A lo largo de las dos últimas décadas, la ingeniería del software se ha hecho mayor. Hoy en día, está reconocida como una verdadera disciplina, derivada de una investigación seria, un estudio minucioso y un debate multitudinario. Los métodos, procedimientos y herramientas de la ingeniería del software han sido adoptados con éxito en una gran variedad de aplicaciones industriales. Tanto los gestores como los desarrolladores reconocen la necesidad de un enfoque más disciplinado del desarrollo de software.

### 1.3.1. Definición

Una de las primeras definiciones de Ingeniería del Software fue la propuesta por Fritz Bauer en la primera conferencia importante dedicada al tema:

“El establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales”.

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

### 1.3.2. Elementos de la ingeniería del software

La ingeniería de software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de tres elementos clave -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. A continuación se describen cada uno de estos elementos:

- *Los métodos de la ingeniería del software* indican “cómo” se va a construir técnicamente el software. Estos métodos abarcan una gran cantidad de tareas entre las que se encuentran: planificación y estimación de proyectos, análisis de los requisitos del sistema y del software, diseño de estructuras de datos, arquitectura de programas y procedimientos algorítmicos, codificación, prueba y mantenimiento. Dichos métodos utilizan una notación especial orientada a un lenguaje o gráfica y un conjunto de criterios para la calidad del software.
- *Las herramientas de la ingeniería del software* suministran un soporte automático o semiautomático para los métodos. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software, llamado ingeniería del software asistida por computadora (del inglés, CASE). CASE combina software, hardware y bases de datos sobre ingeniería del software (una estructura de datos que contenga la información relevante sobre el análisis, diseño, codificación y prueba) para crear un entorno de ingeniería del software.

- *Los procedimientos de la ingeniería del software* son el pegamento que junta los métodos y las herramientas y facilita un desarrollo racional y oportuno del software de computadora. Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas (documentos, informes, formas, etc.) que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las directrices que ayudan a los gestores del software a evaluar el progreso.

### 1.3.3. Paradigmas de la ingeniería del software

La ingeniería del software está integrada por varios pasos que abarcan los métodos, las herramientas y los procedimientos antes mencionados. Estos pasos se denominan paradigmas de la ingeniería del software. Dependiendo de la naturaleza del proyecto, Existen tres paradigmas y se mencionan a continuación:

#### 1.3.3.1. El ciclo de vida clásico

La Figura 1.16 ilustra las actividades que abarca el paradigma del ciclo de vida clásico. También llamado modelo en cascada. Este ciclo exige un enfoque sistemático y secuencial del desarrollo del software que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento [22]:

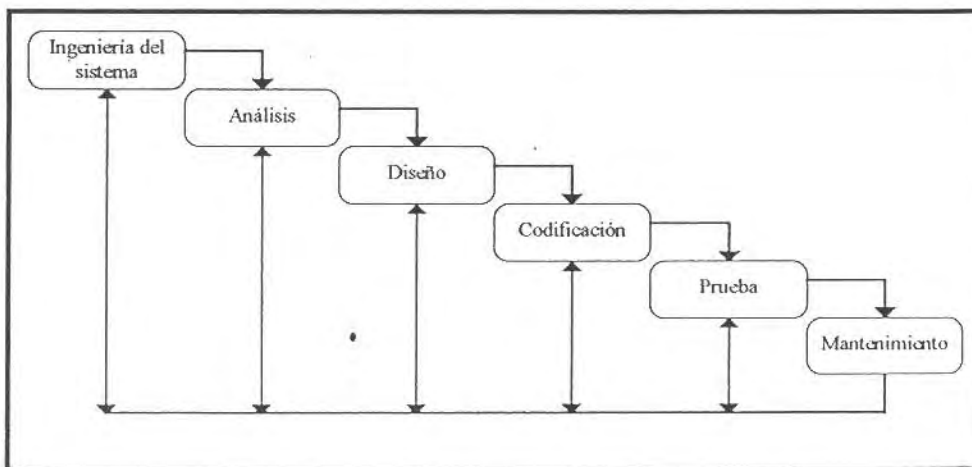


Figura 1.16 El ciclo de vida clásico.

- *Ingeniería y análisis del sistema.* El software es siempre parte de un sistema mayor. Se parte del establecimiento de requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software. Este planteamiento del sistema es esencial cuando el software debe interrelacionarse con otros elementos, tales como hardware, personas y bases de datos.

- *Análisis del los requisitos del software.* El proceso de recopilación de los requisitos se centra e intensifica especialmente para el software. Para comprender la naturaleza de los programas que hay que construir, el ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas. Los requisitos, tanto del sistema como del software, se documentan y se revisan con el cliente.
- *Diseño.* El diseño del software es realmente un proceso multipaso que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.
- *Codificación.* El diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una forma detallada, la codificación puede realizarse mecánicamente.
- *Prueba.* Una vez que se ha generado el código, se comienza con la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, y realizado pruebas que aseguren que los procesos realizados dentro del mismo produzcan los resultados esperados.
- *Mantenimiento.* El software, por lo general, sufrirá cambios después de que se le entregue al cliente. Esto se puede deber a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo o debido a que el cliente requiera ampliaciones funcionales o del rendimiento. Para realizar el mantenimiento del software se deben aplicar los pasos anteriores, pero tomando en cuenta que se está trabajando sobre su sistema ya hecho.

El ciclo de vida clásico es el paradigma más antiguo y más usado en la ingeniería del software. Sin embargo, con el paso de los años, se han producido críticas al paradigma, las cuales cuestionan su aplicabilidad a todas las situaciones. Entre los problemas que se presentan algunas veces, cuando se aplica el paradigma del ciclo de vida clásico, se encuentran:

1. Los proyectos reales raramente siguen el flujo secuencial que propone el modelo. Siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
2. Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos proyectos.
3. El cliente debe tener paciencia. Hasta llegar a las etapas finales del desarrollo del proyecto, no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso.

### 1.3.3.2. Construcción de prototipos

Normalmente un cliente define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida. En otros casos, el programador puede no estar seguro de la eficiencia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma en que debe realizarse la interacción hombre-máquina. En éstas y muchas otras situaciones, puede ser mejor método de ingeniería de software la construcción de un prototipo.

La construcción de prototipos es un proceso que facilita al programador la creación de un modelo de software a construir. El modelo tomará una de las tres formas siguientes [23]:

1. Un prototipo en papel o un modelo basado en PC que describa la interacción hombre-máquina, de forma que facilite al usuario la comprensión de cómo se producirá tal interacción;
2. un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado, o
3. un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

La Figura 1.17 muestra la secuencia de sucesos del paradigma de construcción de prototipos.

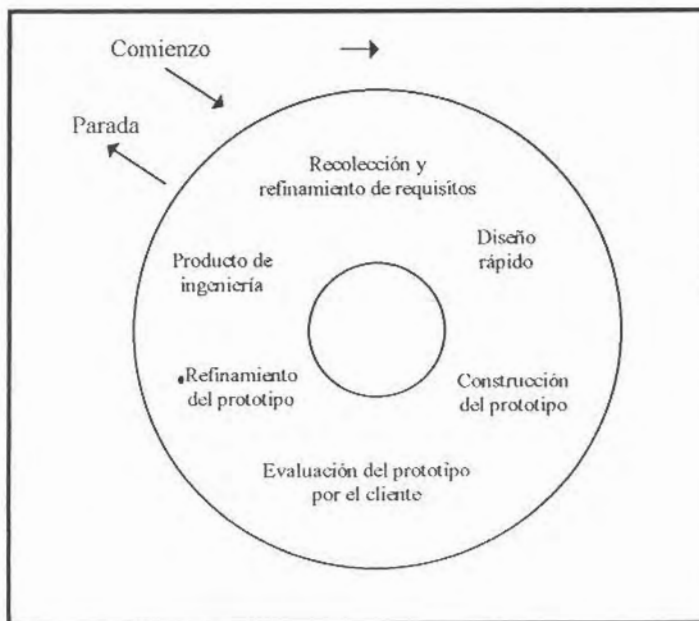


Figura 1.17 Creación de prototipos.

Como en todos los métodos de desarrollo de software, la construcción de prototipos comienza con la recolección de los requisitos. El programador y el cliente se reúnen y definen los objetivos globales para el software, identifican todos los requisitos conocidos y perfilan las áreas en donde será necesario una mayor definición. Luego se produce un

“diseño rápido”. El diseño rápido se enfoca sobre la representación de los aspectos del software visibles al usuario (por ejemplo, métodos de entrada y formatos de salida). El diseño rápido conduce a la construcción de un prototipo. El prototipo es evaluado por el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. Se produce un proceso interactivo en el que el prototipo es “afinado” para que satisfaga las necesidades del cliente, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer.

Idealmente, el prototipo sirve como mecanismo para identificar los requisitos del software. Si se va a construir un prototipo que funcione, el programador intenta hacer uso de fragmentos de programas existentes o aplica herramientas (por ejemplo, generadores de informes, gestores de ventanas, etc.) que faciliten la rápida generación de programas que funcionen.

Al igual que en ciclo de vida clásico, la construcción de prototipos como paradigmas para la ingeniería del software, puede ser problemática por las siguientes razones:

1. El cliente ve funcionando lo que parece ser una primera versión del software, ignorando que el prototipo se ha hecho con “plastilina y alambres”, ignorando que, por las prisas en hacer que funcione, no hemos considerado los aspectos de calidad o del mantenimiento del software a largo plazo. Cuando se le informa de que el producto debe ser reconstruido, el cliente se vuelve loco y solicita que se apliquen “cuantas mejoras” sean necesarias para hacer del prototipo un producto final que funcione. El gestor del desarrollo del software cede demasiado a menudo.
2. Quien desarrolla el software, frecuentemente, impone ciertos compromisos de implementación con el fin de obtener un prototipo que funcione rápidamente. Puede que utilice un sistema operativo o un lenguaje de programación inapropiados, simplemente porque ya está disponible y es conocido; puede que implemente ineficientemente un algoritmo, sencillamente para demostrar su capacidad. Después de algún tiempo puede haberse familiarizado con esas elecciones y haber olvidado las razones por las que eran inapropiadas. La elección menos ideal forma ahora parte integral del sistema.

Aunque pueden aparecer problemas, la construcción de prototipos es un paradigma efectivo para la ingeniería del software. La clave está en definir al comienzo las reglas del juego; esto es, el cliente y el desarrollador deben estar de acuerdo en que el prototipo se construya para servir sólo como un mecanismo de definición de los requisitos. Posteriormente, ha de ser descartado (al menos en parte) y debe construirse el software real, con los ojos puestos en la calidad y en el mantenimiento.

### 1.3.3.3. El modelo en espiral

El modelo en espiral para la ingeniería del software ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo, que falta en esos paradigmas. El modelo, representado mediante la espiral de la Figura 1.18 define cuatro actividades principales, representada por los cuatro cuadrantes de la figura [24]:

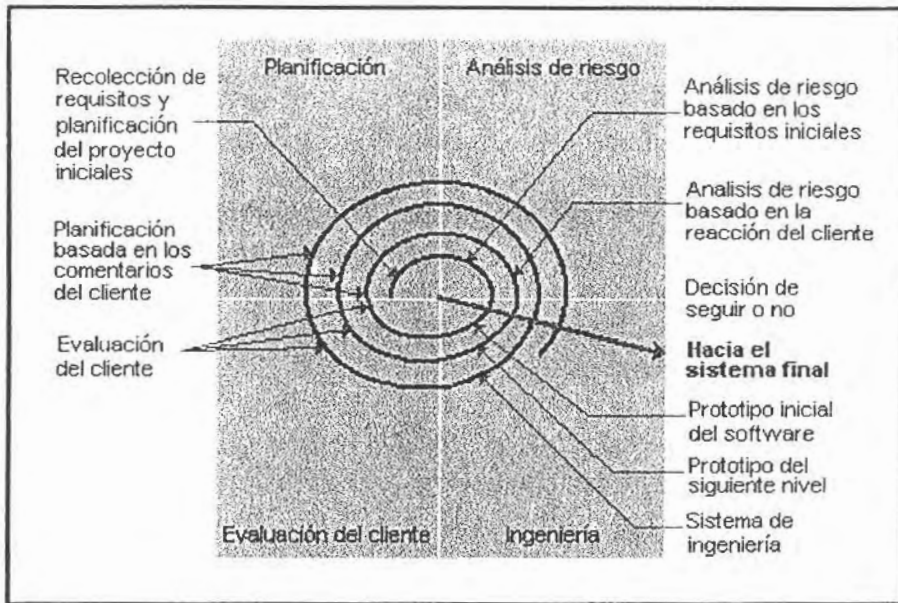


Figura 1.18 El modelo en espiral.

1. *Planificación*: determinación de objetivos, alternativas y restricciones.
2. *Análisis de riesgo*: análisis de alternativas e identificación/resolución de riesgos.
3. *Ingeniería*: desarrollo del producto de "siguiente nivel".
4. *Evaluación del cliente*: valoración de los resultados de la ingeniería.

Un aspecto intrigante del modelo en espiral se hace evidente cuando consideramos la dimensión radial representada en la Figura 1.18. Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas. Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado del desarrollo como al cliente. Se pueden usar simulaciones y otros modelos para definir más el problema y refinar los requisitos.

El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación del cliente) y sugiere modificaciones. En base a los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir". Si los riesgos son demasiados grandes, se puede dar por terminado el proyecto.

Sin embargo, en la mayoría de los casos, se sigue avanzando alrededor del camino de la espiral, y ese camino lleva a los desarrolladores hacia afuera, hacia un modelo más completo del sistema, y, al final, al propio sistema operacional. Cada vuelta alrededor de la espiral requiere ingeniería (cuadrante inferior derecho), que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o de la creación de prototipos. Debe tenerse en cuenta que

el número de actividades de desarrollo que ocurren en el cuadrante inferior derecho aumenta al alejarse del centro de la espiral.

El paradigma del modelo en espiral para la ingeniería del software es actualmente el enfoque más realista para el desarrollo del software y de sistemas a gran escala. Utiliza un enfoque "evolutivo" para la ingeniería del software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción del riesgo, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporándola dentro de un marco de trabajo interactivo que refleja de forma más realista el mundo real. El modelo en espiral demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

#### 1.3.4. Fases del desarrollo del software

El proceso de desarrollo del software contiene tres fases genéricas, independientemente del paradigma de ingeniería elegido. Las tres fases: *definición*, *desarrollo* y *mantenimiento*, se encuentran en todos los desarrollos de software, independientemente del área de aplicación, del tamaño el proyecto o de la complejidad.

- *La fase de definición* se centra 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 requisitos clave del sistema y del software.
- *La fase de desarrollo* se centra en 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 y cómo ha de realizarse la prueba.
- *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 requisitos del cliente dirigidos a reforzar o a ampliar el sistema. La fase de mantenimiento vuelve a aplicar las fases de definición y de desarrollo, pero en el contexto del software ya existente.

---

# CAPÍTULO II

*El proceso administrativo  
en los negocios*

---

---



## 2.1. INFORMACION GERENCIAL

Uno de los recursos más importante de los gerentes es la información. En la administración, la información enriquece el conocimiento que una persona tiene respecto a una entidad de interés (un individuo, lugar, cosa o acontecimiento). Con el conocimiento se reduce la incertidumbre del gerente ante un hecho posible o real.

### 2.1.1. Significado de la información

La base de toda información son los datos, los hechos que describen una entidad. Toda la información se funda en datos, pero no todos éstos son la base de una información útil, como veremos más adelante. Sólo lo son los datos concernientes a las necesidades de la dirección en un momento determinado.

Los datos se convierten en información al procesarlos y darles una forma significativa. El procesamiento puede requerir combinar los hechos o desechar los detalles irrelevantes. En otras palabras, la información es, en realidad, datos transformados para comunicar un significado o conocimiento. La información acrecienta, pues, el conocimiento que las personas tienen respecto a una entidad. Les revela cosas que ignoraban o no podían predecir.

### 2.1.2. Atributos de la información

Para que la información sea útil para los gerentes, ha de reunir las siguientes cualidades:

- *Exactitud.* Debe ser verdadera y correcta, y describir con fidelidad el objeto o hecho.
- *Oportunidad.* Debe estar disponible cuando se necesite y sin demasiado retraso.
- *Pertinencia.* Debe relacionarse con la situación en cuestión. La información pertinente en un momento tal vez no lo sea en otro, si no acrecienta el conocimiento que necesita quien va a decidir.
- *Integridad.* Proporciona al usuario todos los detalles que necesita para entender la situación. Rara vez se encuentra la información completa (esto es, la certidumbre).
- *Frecuencia.* Se prepara o suministra a los usuarios con bastante frecuencia para que esté actualizada.
- *Horizonte de tiempo.* Se orienta a actividades y hechos pasados, presentes o futuros.
- *Alcance.* Abarca en forma amplia o reducida un área de interés.
- *Origen.* Puede originarse en fuentes dentro de la organización o en fuentes externas.
- *Forma de presentación.* Las tablas de números o la presentación gráfica de información son las formas escritas o impresas más comunes. Puede incluir además una presentación verbal.

### **2.1.3. Necesidades de la información por parte de los gerentes**

La información que necesitan los directivos varía con la naturaleza del trabajo que efectúan y las metas que buscan, es decir, las necesidades varían según el nivel de actividad. Un gerente de nivel superior necesita por lo regular mucho menos detalles que otros de nivel más bajo o un empleado operativo. En efecto, aquél tiene una perspectiva más amplia de la organización y de su misión. La información de que se vale complementa esa perspectiva.

Las necesidades de información también dependen de su naturaleza interna y externa. La información interna, que se origina en el seno de la empresa, es indispensable para las operaciones diarias. A continuación se dan ejemplos ordinarios de esta clase de datos:

1. Cantidad de un producto disponible en el inventario.
2. Costo y precio de venta del producto.
3. Ingresos y gastos diarios.
4. Cuotas de los vendedores.

La información externa se origina fuera de la organización. Y a menudo se requiere en los niveles superiores de la dirección para planear y, en general, orientar la empresa en una dirección rentable. He aquí algunos ejemplos comunes de información externa:

1. Descripciones de la satisfacción del cliente con los productos y servicios.
2. Demanda de productos o servicios.
3. Conocimiento de promociones, cambios de precios o productos planeados por firmas de la competencia.
4. Detalles o cambios en las normas del gobierno.
5. Información que describe los cambios en proveedores o políticas aplicadas a ellos.

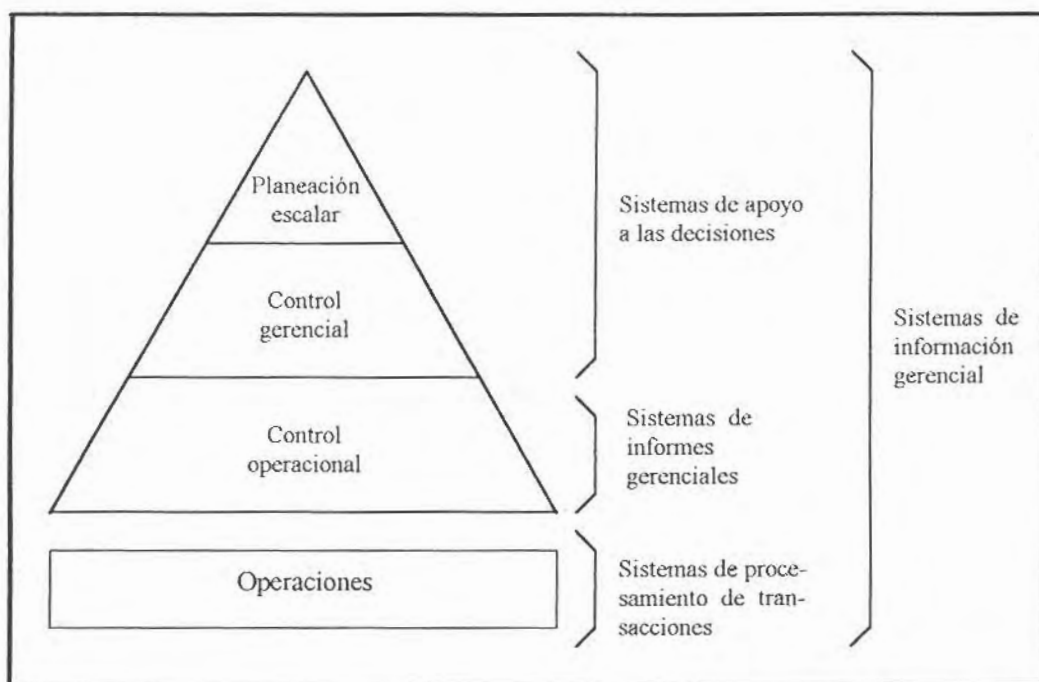
La mayor parte de las firmas también suministran información a usuarios externos. Algunos son los siguientes:

1. Ingresos y utilidades por concepto de ventas declarados a la oficina de impuestos.
2. Cantidad de productos en órdenes de bienes dada a los proveedores.
3. Precio de los productos y servicios cotizados a los clientes.

Un sistema eficaz de información tendrá la capacidad de ofrecer información interna y externa a la vez. Más aún, lo hará en una forma que produzca información oportuna y precisa, dos de los atributos mencionados con anterioridad. En la siguiente sección, explicaremos las características de los sistemas de información gerencial que hacen alcanzables esos atributos.

## 2.1.4. Sistemas de información gerencial

El sistema de información gerencial (MIS) es un sistema integrado de personas, procedimientos, datos y equipo destinados a proporcionar información para apoyar las operaciones, el control y las actividades de planeación de una organización. La información puede orientarse al pasado o al presente, puede relacionarse con actividades y hechos futuros. El sistema es, en realidad, una familia de sistemas de información, como veremos luego. La ayuda de la computadora es cada vez más una parte integral del sistema de información gerencial.



**Figura 2.1** Jerarquía de los sistemas de información en un sistema de información gerencial.

Según la Figura 2.1, este sistema es más bien una jerarquía de sistemas de información. Cada aspecto está ligado a los tipos de actividades que se realizan en diversos niveles de administración de la organización. Los sistemas de la jerarquía se centran en el procesamiento de transacciones, jerarquía de los gerentes y apoyo a las decisiones.

### 2.1.4.1. Sistemas de procesamiento de las transacciones

El sistema más fundamental basado en la computadora en una organización pertenece al procesamiento de las transacciones de negocios. Una transacción (u operación) es cualquier hecho o actividad que afecte a la organización. Las transacciones comunes son la colocación

de órdenes, la facturación a clientes, la contratación de empleados, el depósito de cheques y muchas otras. Los tipos de las que se realizan varían de una organización a otra pero prácticamente todas las empresas procesan las transacciones como una parte esencial de sus actividades diarias de negocios. Las más prósperas efectúan este trabajo de manera ordenada y eficiente.

El procesamiento de transacciones es el conjunto de procedimientos para el manejo de las operaciones. Entre las actividades comunes están las siguientes:

1. Cálculo.
2. Clasificación.
3. Ordenación.
4. Almacenamiento.
5. Resumen.

Las actividades anteriores pueden llevarse a cabo en el nivel operacional de cualquier organización.

Si el lector estudia una muestra de las organizaciones, se dará cuenta de que existen características similares en cada una de ellas:

1. Hay un alto volumen de transacciones.
2. Cada transacción es semejante.
3. Los procedimientos para procesar las transacciones se conocen bien y pueden describirse con detalle.
4. Se dan pocas excepciones en los procedimientos normales.

Por ello, pueden establecerse rutinas para la administración de las transacciones. Las rutinas describen lo que debe buscarse en cada operación, qué medidas tomar, qué procedimientos aplicar y lo que ha de hacerse en caso de excepciones. Con frecuencia a los procedimientos con que se procesan las transacciones se les llama procedimientos estándar de operación.

El alto volumen de transacciones bien entendidas relacionado con el nivel operativo de una organización, lo mismo que la capacidad de los gerentes para desarrollar procedimientos específicos con los cuales manejarlas, a menudo favorecen la implantación de la computadora. La mayor parte de las firmas empieza a usarla porque necesitan desarrollar formas más eficaces y eficientes de procesar los datos concernientes a las operaciones. Los procedimientos están integrados en los programas de computadora que controlan la captura de datos, el procesamiento de detalles y el almacenamiento y presentación de datos e información. Esto se aplica por igual a las organizaciones pequeñas y grandes.

Al procesar los datos referentes a las transacciones, las computadoras ayudan al hombre porque ofrecen rapidez y velocidad, además de que pueden programarse para que sigan rutinas sin variación alguna. Una idea errónea muy generalizada es que las computadoras

desplazan a las personas; rara vez sucede eso: muchas organizaciones descubren que al instalarlas también es preciso contratar a más personal. Y además a las computadoras pueden asignárseles las tareas que los empleados no quieren o no hacen bien, de modo que pueden centrar su atención en otras actividades.

#### **2.1.4.2. Sistemas de informes gerenciales**

Los sistemas de transacción están orientados a las operaciones. En cambio, los sistemas de informes gerenciales tienen por objeto ayudar a los directivos en la toma de decisiones y en caso de problemas. Se sirven de los datos almacenados como resultado del procesamiento de transacciones, pero también utilizan otra clase de información.

En toda organización hay muchas decisiones que muestran una recurrencia periódica (semanal, mensual, trimestral, etc.). Cada vez que se realizan, se requiere cierta información para llegar a la decisión. Puesto que el proceso de decisión está bien estudiado, es posible identificar la información que se necesitará para adoptar la decisión. Y a su vez el sistema de información puede elaborarse de modo que los informes se preparen regularmente para apoyar las decisiones recurrentes. Cada vez que haga falta información, se preparará en una forma establecida previamente y que contenga datos concretos presentado en cierto formato.

Los especialistas en sistemas de información a menudo dan el nombre de decisiones estructuradas a las que se basan en estos sistemas. Están estructuradas porque los gerentes saben qué factores tener en cuenta al momento de llegar a una decisión y cuáles variables determinarán en gran medida si la decisión será buena o mala. Por su parte, los analistas de sistemas elaboran informes bien estructurados que contienen la información que necesitan para la decisión o que indica el estado de las variables importantes.

#### **2.1.4.3. Sistemas de apoyo a las decisiones**

No todas las decisiones son de índole recurrente. Algunas se presentan una sola vez. Otros se efectúan rara vez. Los sistemas de apoyo (soporte) a las decisiones ayudan a los gerentes que deben tomar decisiones poco estructuradas. Con frecuencia reciben el nombre de decisiones estructuradas o semiestructuradas. Se considera que pertenecen al primer tipo cuando no se dispone de procedimientos claros para tomarlas y cuando todos los factores que han de tenerse en cuenta no se pueden identificar fácilmente por anticipado.

Un factor de suma importancia en la utilización de los sistemas de apoyo a las decisiones consiste en averiguar qué información se necesita. En situaciones bien estructuradas es posible precisar de antemano las necesidades de información. Pero en un ambiente no estructurado, resulta difícil hacerlo. Contando ya con toda la información disponible, quizá el directivo se percate de que requiere más datos. Es decir, el hecho de tener información hará advertir la necesidad de otros datos.

Un sistema de apoyo a las decisiones debe tener mayor flexibilidad que los sistemas de informes gerenciales. El usuario podrá solicitar informes al definir sus contenidos e incluso la manera de producirlos. De modo análogo, los datos necesarios para generar la información

pueden provenir de muchos archivos o bases de datos, no de un solo archivo maestro como suele suceder con los sistemas de transacciones y con algunos sistemas de informes.

El juicio de los gerentes influye de manera muy importante en la toma de decisiones cuando el problema no está estructurado. El sistema de apoyo a las decisiones respalda el juicio de los directivos, pero no lo sustituye.

### **2.1.5. Alcance de los sistemas de información**

Es importante señalar que en las organizaciones no hay uno sino varios sistemas de información gerencial. Desde el punto de vista estructural, ese tipo de sistemas es un conjunto de sistemas de información de negocios. Podemos pensar en un sistema para la mercadotecnia, otro para la producción, otro para el personal y otro más para las compras, por nombrar unos cuantos. Dentro de cada una de esas funciones básicas de la empresa se advierten actividades de nivel de transacciones, toma sistemática de decisiones y la presencia de necesidades especiales de decisión.

Es natural que las diversas funciones de negocios en una organización requieran el apoyo o soporte de los sistemas de información; de ahí la noción de sistemas de información para áreas funcionales. Es así como los sistemas de información evolucionan en el seno de las empresas.

### **2.1.6. Componentes de un sistema de información**

Un sistema de información es un conjunto organizado de datos, equipo, procedimientos y personas que intervienen en la captura, almacenamiento y procesamiento de datos para generar la información que se requiere para dirigir una organización.

El elemento en que se fundan todas las actividades de un sistema de información son los datos. En efecto, todos los sistemas giran en torno a ellos y no al equipo, pero a condición de que los datos sean exactos y accesibles.

La administración de los datos en los sistemas de información es una tarea tan importante que los especialistas, llamados gerentes o administradores de datos, tienen la obligación de cerciorarse de que los datos del sistema sean verídicos. Se aseguran de que se sigan los procedimientos con que se verifica la corrección al momento de capturarse y almacenarse los datos. De modo análogo, dan pautas o instrucciones para guardarlos y recuperarlos en archivos y en bases de datos. Sin datos exactos y accesibles, los sistemas de cómputo y de información carecerían de utilidad.

### **2.1.7. Efecto de los sistemas de información**

Los sistemas de información afectan a las organizaciones de diversas maneras. La tendencia consiste en concentrarse en los aspectos en que la tecnología de dichos sistemas pueden ejercer un efecto positivo: reducir los costos de operación, aumentar la rapidez del

procesamiento y mejorar la comunicación de la información. Pero también debe recordarse, que toda organización es una combinación de tecnología, estructura, tareas y, desde luego, de personas. Cada uno de esos elementos interactúa con todos los demás. Cuando se cambia uno de ellos, los otros colaboran para reducir el impacto del cambio. Por ejemplo, si uno de los empleados más importantes abandona la organización, la empresa hace lo posible por reducir al mínimo el efecto de su salida (otros se encargan de parte del trabajo y, acaso, se redefine la naturaleza de la vacante).

De manera análoga, cuando se introducen los sistemas de información, la organización muestra una tendencia a amortiguar su impacto.

Si los usuarios no participan de una manera significativa en el desarrollo de un sistema de información, es posible que aparezca la resistencia y por ende fracase la posible utilidad del sistema. En general se distinguen tres tipos de resistencia.

- *Agresión.* Cualquier forma de ataque al sistema con la intención de hacerlo físicamente inoperante o ineficaz. Ejemplo: sabotaje del equipo al dañar las pastillas o circuitos electrónicos, vaciar líquidos en los componentes o borrar los medios magnéticos de almacenamiento. También puede incluir la captura de datos con errores de los archivos.
- *Proyección.* Atribuir al sistema las dificultades que se encuentran al utilizarlo o al interactuar con él. Ejemplo: atribuir a la computadora un error en el manejo de una operación con el cliente "fue un error de la computadora", aunque de hecho el individuo introdujo datos equivocados o emprendió una acción equivocada (la causa real puede ser un programa inadecuado de capacitación).
- *Evitación.* Rehuir o evitar la interacción con el sistema. Por ejemplo: negarse a servirse de los informes de inventario preparados por el sistema y, en cambio, telefonar para determinar la cantidad disponible.

La explicación precedente puede sintetizarse en los siguientes principios esenciales de los sistemas de información:

1. La información gerencial es en realidad un conjunto, o sea una federación de sistemas más que un sistema individual.
2. Los datos, no el hardware ni el software, constituyen el elemento central en un sistema de información gerencial.
3. Los sistemas de niveles de transacciones que comprenden la parte esencial de la organización han de estar instalados y funcionando mucho antes que otros sistemas sean confiables.

Los usuarios necesitan participar de manera importante en el proceso de desarrollo de sistemas, a fin de asegurarse de que se identifiquen correctamente las necesidades y de que el sistema las satisfaga. Su participación puede reducir al mínimo la resistencia al sistema. El respaldo de la dirección a los sistemas de información es indispensable para garantizar su integración a las actividades de la empresa.

## 2.2. INVENTARIOS

Un inventario consiste de recursos utilizables, pero que están ociosos. Dichos recursos pueden ser de cualquier tipo: hombres, materiales, máquinas o dinero, etc. Si el recurso es material o artículos en cualquier etapa de su acabado, el inventario generalmente se conoce como existencias en almacén.

El objetivo primordial del control de los inventarios es buscar que los costos que se generen del manejo de estos sean los más bajos posible.

Una de las funciones básicas del control de inventarios es conocer en todo momento la cantidad, sitio y condiciones en que se encuentran todos los materiales que tiene una empresa en un determinado momento. El controlar las existencias tiene por misión vigilar y controlar el movimiento de los materiales, ya que de esta manera se podrán fijar las cantidades requeridas para cada uno de estos.

“Si tienes dos unidades por cada cosa, tienes demasiado”, esta frase de uno de los presidentes de la General Motor’s define muy bien lo anterior. En la práctica lo que se persigue es conseguir tener el mínimo y dar al mismo tiempo un buen servicio a los talleres y a los clientes.

### 2.2.1. Variables controlables

Dentro de los inventarios podemos encontrar variables que se pueden controlar, ya sea, separadamente o en combinación, las cuales son [30]:

- *La cantidad adquirida* (por compra, producción o algún otro medio), o sea *cuánto*. Esto se puede fijar para cada tipo de recursos en forma separada o para todos colectivamente; por ejemplo, la compra o el nivel de producción, o ambos. Las decisiones acerca del número de puntos de almacenamiento también afectan el volumen de existencias.
- *La frecuencia o tiempo de abastecimiento*, es decir, *cuándo*. Quien toma las decisiones puede tener control sobre ambos o solamente uno de estos tipos de variables controlables. Por ejemplo, un ama de casa no controla la frecuencia de las visitas del lechero, pero si puede controlar qué cantidad le va a dejar de cada producto.
- *La etapa de acabado de los artículos almacenados*. Cuanto más avanzado sea el grado de acabado de los artículos que se tienen en el inventario, menor será el tiempo que se tardará en suministrar dichos artículos a los clientes, sin embargo, el costo de almacenamiento de los mismos será mayor. Y viceversa, entre menos terminados estén los artículos, más tiempo tomará cumplir con los pedidos, pero el costo de mantener las existencias será menor. El número de los artículos diferentes que se deben almacenar, normalmente crece muy rápido con los avances en el grado de acabado de los artículos almacenados.



### 2.2.2. Variables no controlables

Así como en el manejo de los inventarios encontramos variables que se pueden controlar, de igual manera vamos a encontrar variables no controlables. Dichas variables se dividen en variables de costo y otras. Entre estas podemos encontrar [31]:

- *Costos de mantenimiento de inventario.* Son los costos que se incrementan en proporción directa al crecimiento del inventario y al tiempo que van a permanecer almacenados los artículos. El componente más común y que es estrictamente proporcional al nivel de existencias y al tiempo es el costo del capital invertido, debido a que mientras tengamos almacenados productos por mucho tiempo no podremos ver fructificar la inversión que sobre ellos se hizo. La proporción adecuada del capital, depende principalmente de otra utilización que se le puede dar al dinero. La selección de esta proporción generalmente es un asunto que decidirá la administración financiera.

Además de los costos de capital, debemos considerar los de mantenimiento del archivo y los cargos administrativos. Otros componentes importantes del costo del mantenimiento del inventario son:

- a) *Costos de manejo.* Estos incluyen los costos que conlleva el mover las existencias, como son: grúas, armazones para soportar barriles, montacargas y otro equipo usado para este propósito.
- b) *Costos de almacenamiento.* Entre estos podemos encontrar la renta del local o el interés y depreciación del local si éste pertenece a la empresa.
- c) *Seguros e impuestos.*
- d) *Costos de depreciación, deterioro y obsolescencia.* Estos son particularmente importantes para artículos que cambian químicamente durante el almacenamiento, tales como los alimentos o productos volátiles.

Existen algunos artículos para los cuales el costo de almacenamiento puede depender del volumen de espacio existente, entre los cuales están:

- *Costos de déficit.* Son los costos que surgen cuando algún artículo que se demanda no se tiene en existencia. Un déficit se puede evitar si se toman medidas de emergencia, de tal manera que las entregas se sigan haciendo, o se hagan antes de la fecha que lo requieren los clientes. Hay que tomar en cuenta que al llevar a cabo tales medidas pueden verse involucrados otros costos, como pueden ser: costos de transporte más elevados (transportar la carga por avión en vez de camión, aumento en los costos por tiempo extra, costos administrativos o el costo de romper el programa de producción planeado). También puede haber el costo del equipo de producción de relevo que se usa solamente en una emergencia. Tal equipo puede ser obsoleto o estar fuera de uso y tener costos de producción más altos que el equipo regular.
- *Costos debidos a los cambios en la tasa de producción.* Los cuales incluyen costos de arranque que resultan de cambiar la tasa de producción desde cero a un volumen positivo. En el caso de una compra, implican los costos fijos administrativos de colocar un pedido.

Otros costos que dependen de los cambios que se hagan a las tasas de producción son aquellos de contratación y adiestramiento y mano de obra adicional y los correspondientes al despido de personal. Este último puede ser grande, especialmente cuando hay un sindicato de por medio, ya que estos obligan a que se tenga en cuenta la antigüedad para hacer los movimientos de personal. Así mismo un gran número de bajas puede dar como resultado nuevas asignaciones para una gran parte de la fuerza laboral, lo que implica, un proceso de aprendizaje mientras cada hombre se adapta en su nueva actividad.

- *Precio de compra o costos de producción directos.* El costo unitario por cada artículo comprado puede depender de la cantidad que se compre. Del mismo modo, el costo unitario por artículo producido también puede reducirse bastante debido a una gran eficiencia de hombres y máquinas en corridas largas de producción continua.
- *Demanda.* Es el número de artículos que se requieren por periodo. Cabe hacer notar que esto no es necesariamente la cantidad vendida, debido a que parte de la demanda puede no satisfacerse por déficit o demoras. Pero esto sería realmente la cantidad que se vendería, si todo lo que se necesitara estuviera disponible.
- *Tiempo de reorden.* Es el tiempo que transcurre entre la colocación del pedido y la llegada del artículo al almacén.
- *Cantidad entregada.* Es la cantidad que realmente se surte por concepto de una compra. En este punto, lo ideal es que la cantidad solicitada (que se realiza en el pedido de la compra) menos la cantidad entregada sea cero.

### 2.2.3. Componentes básicos del inventario

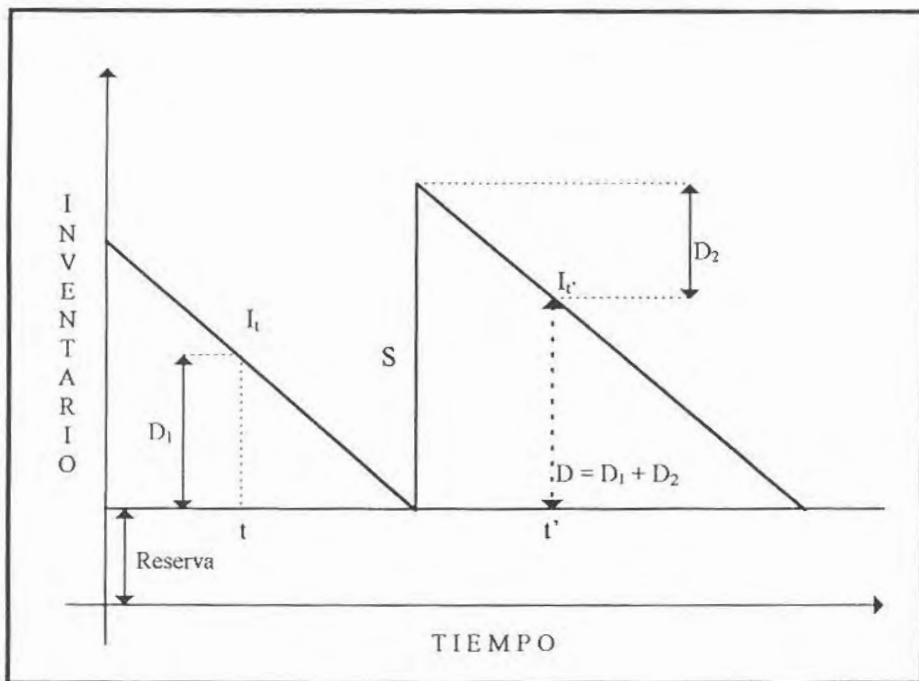
Entre los componentes básicos del inventario están los siguientes [32]:

1. *Componente de rotación.* Es el material que existe para su utilización rutinaria y tiene una relación definida con el lote de entrega. Generalmente es igual o aproximadamente igual a la mitad de dicho lote de entrega.
2. *Componente de reserva.* Es el material que se mantiene para hacer frente a las demandas inesperadas o entregas retrasadas. Normalmente se trata éste como un ciclo de reserva, que se da como los días entre la fecha de entrega programada y la fecha de utilización esperada. Estos días, multiplicados por el uso diario promedio, nos da la cantidad de material que constituye la reserva.
3. *Componente en tránsito.* Es la parte cargada al inventario que se encuentra en tránsito hacia algún almacén.
4. *Componente en exceso.* Es el material por encima del volumen justificado por la demanda actual. Este exceso es material útil, pero está en exceso, bien porque así fue planeado o bien por errores en las solicitudes o entregas de mercancías.
5. *Componentes de excedentes.* Es el material en inventario, que por cambios en la demanda o en el diseño, se ha quedado rezagado u obsoleto y para el cual no hay demanda futura conocida.

La Figura 2.2, muestra un sistema típico de inventario. Dichos sistemas tienen una ecuación de balance que relaciona las existencias en el tiempo  $t$  con aquellas en un tiempo posterior  $t'$ . Hagamos que  $I_t$  sea la existencia en el tiempo  $t$ ,  $S$  la cantidad adicionada al inventario en el intervalo de tiempo de  $t$  a  $t'$ , y  $D$  la demanda. La existencia física en el tiempo  $t'$  está dada por:

$$I_{t'} = I_t + S - D$$

Esto se da cuando la cantidad es positiva en virtud de que  $S$  entrará a las existencias de manera que la demanda nunca las redujera a cero durante el período o intervalo. Sin embargo, si la demanda excede a la oferta, la existencia física será cero.



**Figura 1** Sistema típico de inventario

En resumen, un sistema de control de inventarios debe lograr:

- Tener cantidades adecuadas de los materiales o productos para poder hacer frente a las necesidades de la empresa.
- Evitar pérdidas considerables en las ventas.
- Evitar pérdidas innecesarias por el deterioro u obsolescencia de los recursos, o por tener un exceso de material almacenado.
- Reducir las posibles interrupciones de la producción.
- Reducir los costos referentes a: mantenimiento de inventario, de déficit, debido a los cambios en las tasa de producción, etc.

El control de inventarios se encarga de regular en forma óptima las existencias en los almacenes, tanto de refacciones, herramientas, materias primas, así como de productos terminados. Este tipo de control protege a la empresa de costos innecesarios por acumulación o falta de existencias en el almacén. Por ello, las empresas deben contar con un inventario suficiente para satisfacer sus necesidades. La escasez o retraso de un producto por falta de material, puede ser la causa de la pérdida de un cliente lo que se traduce en pérdidas financieras

#### **2.2.4. Sistema de inventarios**

Este sistema se enfoca primordialmente al establecimiento de las operaciones administrativas para el control de los bienes adquiridos, adscritos y asignados. Constituye una fuente de información cuantitativa y cualitativa que aporta elementos de juicio para la toma de decisiones en referencia al uso, aprovechamiento y control de los bienes, así como para la formulación de programas específicos, tendientes a la utilización racional de los recursos materiales y el destino final de los mismos.

Por otra parte, si se entiende por manejo de artículos el hecho de recibir los diferentes bienes de activo fijo, tener conciencia de su procedencia así como de su fecha de llegada al almacén, de su tipo y uso, de su destino así como de la cantidad asignada a cada área específica, resulta evidente la necesidad de tener un sistema eficiente que permita tener información actualizada de todo el activo fijo y de consumo, con lo cual se agiliza la toma de decisiones.

El desarrollo y la aplicación de inventarios se fundamenta a través de la formulación de los siguientes módulos:

- Control y conservación de bienes en el almacén, así como la organización de éste, aplicando normas y procedimientos correspondientes.
- Suministro oportuno de bienes en el almacén, en el lugar, cantidad y con la seguridad requerida.
- Aprovechamiento racional de los bienes adquiridos de acuerdo con la naturaleza y finalidad del servicio, con los requerimientos internos y con las medidas de mantenimiento preventivo y correctivo, si éste es el caso.

##### **2.2.4.1 Sistema automatizado de información**

Todo sistema que se desee automatizar debe enfocarse principalmente a evaluar las funciones y procedimientos que actualmente se encuentran en operación. Así, para establecer un Sistema Automatizado de Inventarios, se debe definir claramente el tipo de artículo que se maneja, si es activo fijo o de consumo, tipo de documento o documentos que respalda a cada artículo; y sobre todo, establecer rigurosamente el tipo de datos por documento. Esto es con el fin de consolidar la información de que se dispone y proyectarla hacia el Sistema Automatizado de acuerdo con las siguientes etapas [33]:

1. *Integración.* Levantamiento de un censo de información a fin de conformar el catálogo general de bienes muebles, inmuebles, artículos de activo fijo, de consumo, etc., clasificando la información en niveles de grupo y subgrupo.
2. *Actualización.* Formulación de los tipos de movimientos que modifican el contenido de la información, considerándose las altas, bajas, transferencias internas, de depósito, entradas y salidas.
3. *Explotación.* Generación de reportes estadísticos, utilizando la información almacenada.

#### 2.2.4.2. Requerimientos del sistema

En todo Sistema de Control de Inventarios el objetivo fundamental que se persigue es tener un control absoluto y actualizado de las existencias en materia de activo fijo y de consumo. Surge la necesidad de tener registrados los siguientes conceptos:

- Procedencia del artículo.
- Fecha de ingreso.
- Tipo de artículo.
- Descripción del artículo.
- Costo del artículo.
- Adscripción específica.

1. *Procedencia del artículo.* Este concepto permite conocer en cualquier momento bajo qué partida presupuestal fue adquirido el artículo; si es donación, préstamo o tiene cualquier otra procedencia.
2. *Fecha de ingreso.* Permite saber cuándo llegó cada artículo al almacén o área de control; determinar el tiempo que lleva en existencia, deduciéndose con esto si se justifica su reemplazo, mantenimiento, etc.
3. *Tipo de artículo.* Permite hacer separaciones sobre los diferentes rubros, si se refiere a un artículo eléctrico, mecánico, de oficina, de salón de clases, de laboratorio, deportivo, médico, etc.
4. *Descripción del artículo.* Este concepto determina la descripción detallada del artículo para el conocimiento específico del mismo.
5. *Costo del artículo.* El costo de los artículos ayuda a saber el monto de activo fijo que se maneja, ya sea desde un punto de vista general o individual. Permite decidir si es justificable el reemplazo, mantenimiento, etc., de los mismos.
6. *Adscripción específica.* Permite saber con exactitud en qué lugar se encuentra un artículo dado así como la cantidad de artículos de activo fijo con que cuenta.

Del manejo de la información anterior se deben obtener informes como:

- Inventario total de existencias.
- Inventario por tipo de artículo.
- Inventario por adscripción del artículo.
- Calendarios de mantenimiento para artículos de activo fijo.
- Avisos de reemplazo de artículos.
- Resúmenes de totales generales.
- Resúmenes de totales por especie.
- Resúmenes de totales por adscripción.
- Resúmenes de totales por procedencia.
- Aspectos contables.
- Amortizaciones de las inversiones, etc.

Lo único que resta definir son los requisitos más sobresalientes que apoyan al Sistema Automatizado de Inventarios a fin de darle fluidez, dinamismo y confiabilidad al sistema. Estos requisitos deben satisfacer una a una las necesidades administrativas que integran como actividades y funciones el procedimiento del Sistema de Inventarios no Automatizado.

#### **2.2.4.3. Requisitos**

Entre los requisitos indispensables para un Sistema de Inventarios Automatizado se encuentran los siguientes:

1. Es necesario que el sistema sea lo suficientemente flexible para que se ajuste hasta donde sea posible, a los procedimientos administrativos que se siguen en este renglón; o bien, una vez definidos nuevos procedimientos, debe adaptarse a las nuevas condiciones de operación bajo una explotación óptima de sus recursos.
2. Debe emplearse en forma satisfactoria presentando ventajas en lo que se refiere a confiabilidad, versatilidad y oportunidad en el manejo de información.
3. Su desarrollo e implementación debe representar un costo adicional que no sea gravoso para el presupuesto utilizado para tal fin, ya que la automatización del sistema implica la adquisición de equipo electrónico especializado.
4. La creación y actualización de los archivos deberá de ser de fácil aplicación.
5. La explotación de la información responderá a las necesidades reales de información de los altos niveles ejecutivos y de administración.
6. Se utilizará única y exclusivamente la información que se considere de utilidad real.

#### 2.2.4.4. Desarrollo del sistema

La mejor manera de manejar inventarios se logra mediante una orientación sistémica. Una vez que se han analizado y desarrollado varias soluciones alternas, se han hecho estudios y comparaciones de tipo técnico, administrativo y económico, se indica cuál de esas opciones es la más justificable.

La solución se podría encaminar a la utilización de la informática a fin de desarrollar un sistema que apoye con agilidad y dinamismo el manejo de la información, desde su registro hasta su actualización y explotación.

En términos generales, el sistema de inventarios consiste en integrar un inventario lógico para lo cual es necesario conformarlos de las tres etapas siguientes:

1. Creación de un archivo maestro.
2. Actualización del archivo maestro.
3. Generación y explotación de archivos secundarios.

1. *Creación de un archivo maestro.* Para la creación de este archivo, se parte de la información contenida en la integración del inventario físico. A partir de esta información se estructura el archivo maestro cuyo contenido en cuanto a conceptos puede ser: Clave Genérica del Artículo, Fecha de Ingreso al Almacén (fecha de alta), Tipo de Artículo, Descripción del Artículo, Adscripción Específica del Artículo, etc.

De este archivo maestro se deben generar listados o reportes de movimientos rechazados y movimientos aceptados.

2. *Actualización del archivo maestro.* Partiendo del hecho de que periódicamente se integren (alta) y desechen (baja) artículos de activo fijo y de consumo a los almacenes, resulta imprescindible la existencia de un módulo de actualización del archivo maestro que permita tener la información debidamente actualizada.
3. *Generación y explotación de archivos secundarios.* Si bien la integración y la actualización del archivo maestro es fundamental, la parte que realmente es de uso productivo y rutinario la constituye la explotación del mencionado archivo y será por lo tanto este tercer módulo el encargado de proporcionar los resultados a ser manejados por el usuario del sistema.

Se mencionan a continuación algunos tipos de listados o reportes que deben y pueden ser de utilidad como producto final del sistema:

- *Listado general de inventarios con totales.* Contiene la información del inventario, dando en forma adicional los totales correspondientes al costo del inventario y a la cantidad total de artículos del inventario.
- *Listado de existencias por tipo de artículo.* Contiene una clasificación de existencias por tipo de artículo, proporcionando totales en cantidad por cada tipo de artículo y el costo de esos totales.

- *Listado de existencias por fecha de ingreso.* Proporciona información sobre todo el inventario clasificado por fechas de ingreso al almacén, haciendo cortes por mes, año o como se juzgue conveniente.
- *Listado de artículos para mantenimiento.* Genera un listado de artículos a los cuales es necesario proporcionar mantenimiento preventivo en un determinado parámetro de tiempo.
- *Listado de artículos para reemplazo.* Genera un listado de artículos que por su tiempo de permanencia ya sea en el almacén o en su área de asignación, requieren ser reemplazados por artículos de nueva adquisición.
- *Listado de artículos por su ubicación física.* Proporciona una relación de artículos adscritos físicamente a cada área específica.
- *Listado de artículos por su procedencia.* Contiene información concerniente a artículos de acuerdo con su procedencia específica.
- *Resumen de inventarios.* Este listado presenta cifras totales en forma de resumen; esto es, total de artículos en inventario, totales por tipo, costos totales, totales por adscripción, procedencia, etc.

Este módulo, por su naturaleza, puede verse incrementado en los resultados a obtenerse, incrementando el número de listados de explotación; o bien, cambiando algunos ya en explotación si la experiencia demuestra que no son del todo útiles, pudiéndose eliminar sin afectar al resto del sistema.



### 2.3. HISTORIA DE LA INDUSTRIA FARMACÉUTICA EN MÉXICO

En México, la actividad farmacéutica se remonta a las culturas precolombinas. Antes de la llegada de los españoles, la medicina había alcanzado un notable progreso y ésta se apoyaba en los efectos que tenían las plantas sobre los seres humanos, es decir, la botánica estaba aplicada a la medicina. Este tipo de prácticas se daba entre Mayas y Aztecas. Los indígenas conocían una gran variedad de plantas así como sus aplicaciones. Al llegar los conquistadores españoles al territorio del Imperio Azteca, se sorprendieron al ver los grandes conocimientos que tenían en este campo.

El boticario precolombino era más que un simple herbolario, se acercaba más a lo que era un médico que a un comerciante. Todos los conocimientos se transmitían de padres a hijos, y que mejor que ellos fueran los que supieran todos sus secretos acerca de las plantas. Por lo general la mayoría de los boticarios contaban con jardines botánicos.

El desarrollo de la herbolaria indígena fue frenado por la guerra de conquista, sin embargo los conocimientos que se tenían sobre las plantas pudieron conservarse. En el Real Colegio de la Santa Cruz, en Tlatelolco, se encuentra un libro escrito por el médico indígena Martín de la Cruz, en el cual describe el uso de las plantas medicinales mexicanas. Dicho libro contiene la representación de 185 plantas, las cuales fueron pintadas a mano, y debajo de cada una de ellas se explica sus aplicaciones para curar diversas dolencias. Cabe mencionar que muchas de estas plantas forman parte de la medicina tradicional y aún se les puede encontrar en los mercados populares.

La situación de las farmacias durante el virreinato no es del todo claro; cuando se fundó la Real y Pontificia Universidad de México, primer centro de educación superior en América, se impartieron cursos sobre Filosofía, Teología y Medicina, pero no se hicieron estudios de Farmacia. Toda persona que pretendía ejercer esta profesión, tenía que tener algunos años de experiencia laboral dentro de una farmacia, además de tener nociones de latín. Con estos requisitos el aspirante podía presentar un examen para poder obtener el título correspondiente.

En el año de 1831 se fundó la Facultad Médica y se expidió una ley para que los farmacéuticos fueran examinados por tres individuos de su profesión y dos médicos. En 1833 se creó el Establecimiento de Ciencias Médicas, y en ese mismo año por decreto del presidente Valentín Gómez Farias, quedó establecida en el Colegio de Medicina la cátedra de Farmacia. En 1939 se fundó la Academia Farmacéutica, la cual fue la encargada de editar, en 1846, la primera Farmacopea<sup>7</sup> Mexicana, de 436 páginas. En 1871 aparece la Sociedad Farmacéutica de México, cuyo trabajo se vio reflejado tres años más tarde con la publicación de la Nueva Farmacopea Mexicana.

A través de los años han surgido los Laboratorios, empresas dedicadas a la producción de medicamentos, lo cual ha hecho que la cantidad de estos productos vaya en aumento dando como consecuencia que hoy en día exista en el mercado una gran cantidad de ellos. Lo

<sup>7</sup> Libro que el gobierno de cada país edita y hace revisar en las oportunidades convenientes, con la lista de los medicamentos autorizados, sus principales aplicaciones y efectos, requisitos para su preparación y entrega al público entre otros.

anterior dio a su vez la aparición de casas distribuidoras, las cuales se han encargado de distribuir los medicamentos a todas las farmacias de México y en algunos casos en países de Latinoamérica. Nadro, Autrey, Benavides y Fármacos Nacionales, son algunos ejemplos de ellos. Estas grandes distribuidoras, no sólo se han encargado de distribuir los medicamentos, sino también de comprar, vender, exportar, importar y en algunos casos hasta de producir sus propios productos. Y todas ellas comprometidas con sus clientes para ofrecerles el mejor de sus servicios.

Gracias al surgimiento de una gran cantidad de Laboratorios, han hecho que la competencia en este ramo sea muy dura, lo cual, hace que cada uno de ellos se esfuerce día con día para crear los mejores productos, pero sobre todo, de ofrecer a la población productos de calidad en beneficio de su salud.

---

# CAPÍTULO III

*Análisis y diseño del SCIF*

---

---

### 3.1. ANÁLISIS DEL SCIF

El ciclo de desarrollo del software utilizado en el SCIF, fue el ciclo de vida clásico, se decidió utilizarlo por ser bastante comprensible y además porque se ajustaba al sistema a realizar ya que no es muy grande. Se comenzó por delimitar lo que sería el sistema y las partes que interactuarían con él de modo general. De ahí se determinó que una parte importante sería la persona que nos proveería de la información necesaria para el desarrollo del mismo.

El siguiente paso, fue realizar un análisis de los requisitos del software. Para lo anterior se sostuvieron una serie de entrevistas con el dueño de una farmacia para determinar toda la información que maneja y que pudiera ser de utilidad.

Una vez obtenida toda la información, se procedió a depurarla para obtener la que era realmente importante. A partir de aquí se comenzó a diseñar lo que sería el software. Como las bases de datos eran la parte medular del sistema, se determinó el modelo con el cual se iba a trabajar (*modelo relacional*) para posteriormente diseñar las relaciones y parte de las operaciones que realizaría el sistema.

Ya con el diseño de la base de datos y de lo que serían los procesos implicados en el sistema se pasó a la codificación del sistema. Realizándose la codificación en Visual Basic y creándose la base de datos en Access. Ya terminado el sistema, se realizaron una serie de pruebas en varios establecimientos farmacéuticos para conocer varias opiniones respecto al sistema. Más que el funcionamiento, se mostró las operaciones que realiza el sistema.

Aparte de las pruebas anteriores, se realizaron otras para comprobar su funcionamiento. Estas consistieron en hacer pruebas de escritorio de todas las operaciones que se llevan a cabo y comprobar que realmente lo que se obtenía en papel también se obtenía al manipular el sistema con los datos respectivos. En todas las pruebas el sistema arrojó los mismos valores obtenidos en papel. Se hicieron varias pruebas de este tipo manejando más datos y de esta manera se pudo comprobar la efectividad del sistema.

Las pruebas de funcionamiento del sistema se realizaron en una computadora 486 a 25 Mhz. con 4 Mb en RAM y en una computadora personal 386 a 66 Mhz con 4 Mb en RAM.

En ambas computadoras el sistema tardaba 2 segundos en presentar la primera ventana, que corresponde a la clave de seguridad y otros dos segundos para entrar al menú principal. En cuanto a las actualizaciones que se llevan a cabo, se efectuaban en períodos de aproximadamente 3 segundos para las consultas y 4 segundos para grabar los datos si intervenían varias relaciones y hasta 2 cuando eran pocas las relaciones afectadas. Estos tiempos se mantenían para un promedio de 100 productos aproximadamente.

La última fase de mantenimiento se llevó a cabo, después de haber realizado las pruebas con los dueños de otras farmacias ya que hubo uno que hizo la observación de un dato que faltaba por agregar y que no se había considerado. De aquí se volvió a realizar el análisis de los requisitos del software y los puntos sucesivos, pero estos fueron mucho más rápidos de realizar debido a que ya se tenía todo y sólo faltaba añadir el nuevo dato. Por las características de Visual Basic, el agregar nuevos módulos, resulta realmente muy sencillo, por lo que el mantenimiento del sistema será más fácil.

Para llevar a cabo el análisis de los requisitos necesarios para el desarrollo del sistema, se realizaron una serie de entrevistas con la encargada de una farmacia (con la cual se estuvo trabajando durante todo el desarrollo del sistema) y con ello determinar además de las características deseables del sistema, la información que contendría nuestra *base de datos*.

Los puntos más importantes se resumen a continuación:

- Todos los productos que se reciben vienen respaldados por una factura, la cual contiene información de estos. Entre los datos que vienen en la factura, los cuales se encuentran representados en forma de columnas, están:

⇒ Nombre de la empresa de la que provienen los productos.

⇒ Nombre y domicilio del cliente.

⇒ *Descripción*: en esta columna vienen descritos los nombres de los productos, especificando si se trata de gotas, cápsulas y en que cantidad; mililitros, gramos, etc.

⇒ *Cantidad de piezas*: aquí se especifican la cantidad de piezas que se están surtiendo de cada uno de los productos.

⇒ *Precio Público Sugerido*: este es el precio que viene marcado en los productos, es decir, es el *Precio Máximo al Público*. Este precio para un producto es el mismo en cualquier parte del país, ya que se trata de un precio controlado. Ninguna farmacia puede vender un producto a un precio mayor que el Precio Público Sugerido que viene marcado en el mismo. El precio de venta de un producto en una farmacia estará determinado por el porcentaje de descuento que ofrezcan los proveedores a sus clientes. Ya que en la mayoría de los casos, los clientes traspasan el descuento que les hacen sus proveedores a los consumidores finales.

⇒ *Precio Farmacia*: este precio es el que se toma como base para determinar a cómo venderá el proveedor un producto al cliente. Este precio también es igual para un mismo producto en cualquier parte del país.

⇒ *El costo final* de un producto resultará de aplicar a su Precio Farmacia, el descuento que ofrece el proveedor y al resultado agregarle el IVA correspondiente, en caso de que lo haya.

⇒ Existe una columna que especifica si el producto está o no exento del IVA y otra que indica si al producto se le hará o no el descuento que ofrece el proveedor. Aunque es muy raro que a un producto no le aplique el proveedor su descuento correspondiente.

⇒ *Importe*: esta columna contiene el resultado de multiplicar el Precio Farmacia de un producto por el número de piezas surtidas.

⇒ *Productos sin existencia al surtir*: aquí aparece el nombre de los productos que se pidieron pero que ya no había para surtirnos.

⇒ *Total*: es la cantidad monetaria que se tendrá que pagar al proveedor por concepto de los productos que se están recibiendo.

- Todo producto trae impreso en sus etiquetas y en los empaques que los contienen, características bien definidas que los diferencian unos de otros, entre los cuales podemos encontrar: *Nombre del Producto*, *Precio Máximo al Público*, *Fecha de Caducidad* (se puede dar el caso que existan productos que no tengan fecha de caducidad), *Número de Lote*, *Nombre del Laboratorio* que lo produce y su *Código de Barras*. Tanto el nombre como el código de barras son únicos para cada producto. El código de barras<sup>8</sup> esta compuesto por una secuencia de valores numéricos.
- Existen varios productos con el mismo nombre, lo que los hace diferentes es la *presentación* en la que vienen. Por ejemplo, tenemos el Pentrexil, el cual lo podemos encontrar en las siguientes presentaciones:
  - ⇒ Pentrexil cápsulas de 250 mg.
  - ⇒ Pentrexil cápsulas de 500 mg.
  - ⇒ Pentrexil tabletas de 1 gr.
  - ⇒ Pentrexil suspensión de 125 mg.
  - ⇒ Pentrexil suspensión de 250 mg.
  - ⇒ Pentrexil suspensión de 500 mg.
  - ⇒ Pentrexil frasco ampula de 125 mg.
  - ⇒ Pentrexil frasco ampula de 250 mg.
  - ⇒ Pentrexil frasco ampula de 500 mg.
  - ⇒ Pentrexil frasco ampula de 1 gr.
- Algunos productos se producen en lotes, si llega a haber algún problema con los productos que pertenecen a este lote y ya se repartieron en los distintos establecimientos, será fácil reconocerlos porque ellos contendrán este número marcado en su caja, de ahí su importancia.
- La mayoría de los medicamentos están exentos del IVA, es decir, están gravados con tasa 0%, y se les conoce como productos farmacéuticos y a los que se les carga el IVA, en la cual están incluidos todos los productos de perfumería, son llamados productos no farmacéuticos. Cabe mencionar que se puede dar el caso de que un medicamento con tasa 0% deje de serlo y se le cargue el IVA y viceversa. Así mismo puede haber un cambio general en la tasa de aplicación del IVA.
- La dueña de la farmacia trabaja con dos proveedores, uno de ellos en un 100%, sin embargo los faltantes de éste serán los pedidos para el otro.
- Por lo general, en todos los productos, ambos proveedores ofrecen un descuento. A la hora de vender medicamentos se les aplica este descuento sobre el precio que trae marcado el producto. Sin embargo, cuando son cantidades muy pequeñas casi no hace el

<sup>8</sup> Para más detalles consultar el APÉNDICE A.

descuento. A los productos no farmacéuticos no se les aplica ningún descuento sobre su precio. El IVA ya va incluido en su precio.

- Cuando algún proveedor no hace el descuento sobre algún producto, éste se traslada al consumidor final. Ya que no hay forma de saber a la hora de vender, si un producto tiene o no descuento, porque toda la mercancía se marca con el Precio Público Sugerido y sin ninguna anotación o seña que indique que ese producto no tiene descuento, por tanto se les aplicará el descuento a todos los productos.
- Los dos proveedores ofrecen descuentos diferentes, con el que se trabaja en un 100% ofrece un mejor descuento. Pero a la hora de vender algún producto se aplica descuento más alto en general, porque no se tiene una forma de identificar que el producto corresponde a un pedido del otro proveedor, salvo que se recuerde. Si por ejemplo, existen dos piezas de un determinado producto y cada una corresponde a diferentes proveedores, el descuento a aplicarle para su venta, sería el promedio de los descuentos que cada uno ofrece. Esto sería lo idóneo, pero como lo mencionamos anteriormente, no hay forma de hacer tal distinción.
- Se lleva diario una relación de ventas, en esta lista se anotan los productos vendidos en el día y su precio correspondiente. Al final de mes se deben hacer notas separando lo que tiene IVA y lo que no tiene.
- Para realizar pedidos al proveedor, se lleva diario una relación de ventas. De los productos que se vendieron durante el día saldrá el pedido.
- Aparte de la lista de venta, se lleva otra en la que se anotan los productos que se pedirán al proveedor. Los datos se toman de la lista de venta y se le agregarán los nombre de los productos que no se tengan y los hayan solicitado algún cliente de la farmacia. A la llegada del pedido se saca la lista de los productos sin existencia al surtir y este será un nuevo pedido pero para el otro proveedor.
- Diario (por lo general en la noche) llega un agente del proveedor con el que se trabaja en un 100% a recoger los pedidos que se tengan. Al otro día en la mañana son surtidos los productos. Con el otro proveedor, su agente pasa a revisar si hay algún pedido cada tercer día. Además de levantar el pedido, los agentes entregarán a sus clientes sus notas de crédito por la compra que están efectuando y se les pagará.
- Cuando en el pedido llegan más productos o menos de los que se solicitaron o alguno no corresponde con lo que se pidió o ya caducaron o están próximos a caducar, lo que se hace es separar esta mercancía y anotar lo que no se surtió, y en la noche que pase el agente se encargará de llenar una forma especial para este tipo de situaciones, anotando los nombres y cantidad de los productos en los que hubo un error. Todo lo anterior se llevará a cabo siempre y cuando aparezcan cargados en la factura. En caso contrario, se entregarán al agente o se hará un nuevo pedido. Cabe mencionar, que los proveedores no devolverán a su cliente el dinero en efectivo por cualquiera de las situaciones anteriores, sino que se les bonificará en el próximo pedido.
- Para ambos proveedores, cuando se hace una devolución, una copia del formato para este fin es para la farmacia (misma que se tendrá que entregar al agente cuando salga su nota

de crédito). En dicho formato se anota el Precio Farmacia de los productos que se están devolviendo menos el descuento que ellos ofrecen, con la finalidad de saber el total de la bonificación que le tendrán que hacer al cliente. Si los productos devueltos tienen IVA, este también se agrega. En la copia de devolución el Precio Farmacia que se asienta es el que corresponde a la factura de su fecha de adquisición. Dicha cantidad sólo la asienta en su copia el cliente con el fin de llevar un control y saber por cuánto le tiene que llegar su próxima nota de crédito.

- Se deben anotar los precios de los productos que se están devolviendo en el formato que entrega el agente, aunque no siempre el día que se hace la devolución el agente proporciona la copia del formato, ya sea porque no lleva formatos o no le da tiempo, etc., por lo que se va juntando mercancía (ya que como diario llegan pedidos, pueden traer algunos errores). Cuando levantan las devoluciones y se rellena el formato se tiene que recurrir a las facturas pasadas, encontrar los productos que se están devolviendo, revisar los precios y asentarlos en su copia, todo esto con el fin de saber por cuánto le tiene que llegar su nota de crédito ya que se descontará la bonificación por concepto de devoluciones.
- Todos los martes, jueves y sábado de cada semana, llegan las listas de cambios de precios. Esta lista contiene los nombres de los productos que subieron, el Laboratorio al cual pertenecen, su Precio Público Sugerido, su Precio Farmacia y las fechas en que subieron los productos. También contiene información de cambios de precios para productos no farmacéuticos.
- Cuando llega la lista de cambios de precios hay que revisar los productos afectados y reetiquetarlos.
- A principio de cada año se hace una lista con los nombres de los productos y su fecha de caducidad. Pero de aquellos que van a expirar ese año. Se anotan separándolos por mes. Para obtener estos datos hay que revisar todo el inventario físicamente. Los productos que llegan en el transcurso del año no se anotan. Esto representará pérdidas para el negocio en caso de que un producto llegue a caducar en ese mismo año y no se hayan sacado para entregar a los agentes.
- *Caducidades:* a principio de cada mes (10 primeros días) se recogen los productos que están próximos a caducar. Las fechas de devoluciones varían. El agente indicará que meses se le tienen que tener preparado para que se les entregue. Todos los productos tienen mes y año, algunos el día y otros no traen fecha de caducidad. Los que no caducan los elimina según si su aspecto ya no es muy confiable. En caducidades no se aplica el IVA, ya que los productos que recoge el agente son los que tienen tasa 0%. Aunque se lleven algún producto con IVA no lo cargan. Se sacan todos los productos próximos a caducar, tanto farmacéuticos como no farmacéuticos, y los que no se lleva el agente se tiran.
- Los proveedores dan un 50% de bonificación sobre los productos que se hayan entregado como caducados.



- El proveedor expide una nota de crédito por los productos correspondientes a caducidades. Sobre el Precio Farmacia actual total de todos los productos se aplica el 50% y sobre esa cantidad sale la nota de crédito.
- Para comprobar que la nota de crédito por concepto de caducidades sea el correcto, lo que se hace, es poner en una hoja Cantidad, Nombre y Precio Público Sugerido para cada uno de los productos que se están entregando y al final se suman todos los Precios Públicos Sugeridos y se le aplica un 23% de descuento y sobre lo que queda se aplica el 50%. Esto sólo es una aproximación ya que es lo más fácil, porque el PPS se obtiene directamente del producto. La forma más exacta de encontrar ésta cantidad sería si se tuviera el Precio Farmacia actual de cada producto, pero realizar esto sería muy tardado, ya que se tendría que buscar en todas las facturas el Precio Farmacia correspondiente a cada uno de ellos. Es por esta razón por la cual se basa en el método de tomar el PPS aunque no sea muy exacto.
- Aún utilizando el Precio Farmacia la cantidad obtenida no concuerda con la cantidad que aparece en la nota de crédito por caducidades (aunque es la mejor), debido a que hay ocasiones en que los que reciben las mercancías desechan algunas que según ellos ya no pasa o porque está en malas condiciones, y sobre lo que queda hacen la nota de crédito correspondiente.
- *Medicamentos sueltos:* se tiene suelto lo más comercial. Se manejan igual que con cajas cerradas. Por lo general se vacía una caja en los anaqueles y una queda de reserva. Si se acaban las pastillas de los anaqueles o se vende una caja completa es cuando se solicita en el pedido. Los medicamentos sueltos no se anotan en la lista de ventas. Antigripales, antidearreicos y analgésicos son los únicos que se venden sueltos.
- Es necesario conocer cada 6 meses el total del inventario, para saber el valor de lo que se tiene y de cómo va el negocio. Además cada fin de año se necesita conocerlo por cuestiones contables. Las características que contiene el reporte de inventario son: *Cantidad, Descripción, Precio Farmacia, Precio Público Sugerido* y su *IVA* correspondiente. Al final del reporte se deben especificar las siguientes cantidades:

<b>suma total PF</b>	<b>suma total PPS</b>
+ <u>          IVA</u>	+ <u>          IVA</u>
<b>suma total</b>	<b>suma total</b>

- El de fin de año debe contener lo mismo que el de 6 meses. Este documento es obligatorio que lo haga, el de 6 meses no, ya que sólo le sirve a ella para saber como anda el negocio. En ambos documentos se requiere saber el inventario tanto de los productos farmacéuticos como los no farmacéuticos.
- El reporte de inventario de fin de año lo realiza uno de los proveedores sólo se le tiene que enviar Cantidad, Descripción y Precio Público Sugerido de cada producto, y ellos se encargan de poner el Precio Farmacia, en caso contrario la obtención de este dato sería muy tardado, porque se tendría que buscar en todas sus facturas los PF correspondientes a cada producto.

- *Productos controlados:* Existen productos que están controlados y requieren un tratamiento especial a la hora de venderse. Están clasificados en tres grupos A, B y C, con las siguientes leyendas:
  - ⇒ *Grupo A o Grupo 1:* Estos medicamentos sólo podrán adquirirse exclusivamente con Receta Especial, cuando el enfermo lo requiera por lapsos no mayores de cinco días y con Permiso Especial para el mismo tipo de medicamentos, cuando el paciente lo requiera por lapsos mayores de cinco días, y que los recetarios y permisos son suministrados, editados y autorizados por la propia Secretaría de Salubridad y Asistencia.
  - ⇒ *Grupo B o Grupo 2:* Para el surtido de estos medicamentos se requiere receta médica vigente por 30 días, ésta será surtida por única vez y el farmacéutico deberá retenerla haciendo las anotaciones pertinentes en su libro de control.
  - ⇒ *Grupo C o Grupo 3:* Estos medicamentos requieren para su surtido receta médica, la cual se deberá marcar con sello fechador y devolverse al cliente, esta podrá surtirse en tres ocasiones en seis meses.
- Se cuenta con un catálogo en el que vienen especificados los productos controlados así como el grupo al que pertenecen.
- Se aceptan devoluciones sobre ventas, siempre y cuando sean porque el producto vendido presente algunas irregularidades.

Con los datos anteriores, se determinaron las acciones que debería llevar a cabo el sistema.

#### ***Sobre los productos:***

- 1) Darlos de alta, de baja, consultarlos y modificarlos.
- 2) En cuanto a las consultas poder realizar las siguientes:
  - ⇒ Por su código.
  - ⇒ De productos controlados.
  - ⇒ De productos por su fecha de caducidad.
  - ⇒ De productos por número de lote.
  - ⇒ De las bajas realizadas por la fecha de caducidad vencida, para conocer la bonificación que hará el proveedor.
  - ⇒ De todo el inventario.

#### ***Sobre las ventas:***

- 1) Manejar el punto de venta.
- 2) Consultar las ventas efectuadas en un determinado lapso de tiempo.
- 3) Llevar a cabo devoluciones de las ventas realizadas.

***Sobre las compras:***

- 1) Realizar las actualizaciones en cuanto a las cantidades de los productos que se están comprando.
- 2) Dar de alta, baja, consultar y modificar tanto de las devoluciones que se realicen a los proveedores por concepto de compras, así como de los faltantes.

***Sobre los proveedores:***

- 1) Dar de alta, baja, consultar y modificar a los proveedores.

***Sobre el IVA:***

- 1) Dar de alta, baja, consultar y modificar los valores correspondientes al IVA.

Aunado a lo anterior, será necesario poder imprimir reportes tales como:

1. De todo el inventario.
2. De productos que están próximos a caducar.
3. De las ventas efectuadas en cierto período.
4. Del punto de venta.

Con los datos anteriores podemos ya diseñar la base de datos, para lo cual utilizaremos el Modelo Relacional, representando los datos y las relaciones entre estos por medio de una colección de tablas, además de las ventajas que a continuación se presentan:

1. Toda la información de una base de datos relacional se representa por medio de tablas, por lo que son fáciles de usar.
2. Es posible crear y cargar unas cuantas tablas y comenzar de inmediato a utilizarlas.
3. No se requiere un diseño completo y correcto de la base de datos antes de comenzar a capturar y utilizar los datos, ya que es posible añadir tablas, vistas, índices o campos nuevos en cualquier instante. De igual manera se pueden eliminar en el momento que ya no los necesitamos.
4. Una de las ventajas más importantes es que se pueden reunir dos o más tablas en una sola, es decir, se puede obtener información de más de una tabla.

***Herramientas usadas***

La programación de este sistema se realizó en Visual Basic por las características que se enlistan a continuación:

1. Maneja bases de datos relacionales.

2. Se maneja en un ambiente Windows, lo que lo hace que el desarrollo de sistemas tengan una mejor presentación y los usuarios puedan interactuar mejor con ellos.
3. Contiene una serie de herramientas con las que se pueden crear varios objetos dentro de la aplicación. Cada objeto tiene asociado: eventos, propiedades y métodos programables que requieren código para realizar las acciones que nos interesen.
4. Puede manipular datos creados en diferentes formatos para bases de datos tales como Access, FoxPro, dBase y Paradox por mencionar algunas.
5. Permite utilizar el estándar oficial del lenguaje relacional SQL, logrando una mejor manipulación de los datos. Se pueden crear instrucciones para seleccionar registros de una o varias relaciones a la vez, crear filtros para obtener determinados datos u ordenarlos según nuestras necesidades. Así mismo, podremos borrar, insertar nuevos datos o modificar los ya existentes.
6. Se pueden crear índices a los atributos de las relaciones, con lo cual se acelera el acceso a los datos.
7. En un solo archivo físico se almacenan las relaciones de una aplicación en general, a diferencia de otros paquetes, que utilizan un archivo físico por cada relación.
8. Maneja objetos para acceso a bases de datos. Entre los más importantes podemos mencionar: OpenDatabase, Dynaset y Snapshot.
9. Para poder trabajar con una base de datos es necesario establecer la conexión con la misma y el OpenDatabase es quien se encarga de hacerlo. Además contiene una serie de parámetros que dan seguridad al abrir la base de datos y que no permiten la modificación de la información por usuarios no autorizados, esto es muy importante cuando a la base pueden acceder varios usuarios de terminales remotas.
10. Los Dynasets y los Snapshots son objetos que, una vez creada una consulta para cualquiera de ellos mediante una instrucción SELECT, guardan los registros resultantes. Con lo anterior se puede disponer en todo momento de la información que cada uno tiene. Los Dynaset tiene la propiedad de lectura y escritura, es decir, podremos realizar modificaciones a los registros obtenidos. Comprueban el mecanismo de bloqueos de escritura para un ambiente multiusuario.
11. Los SnapShots son objetos de sólo lectura. Tienen mayor rapidez de acceso a los datos, ya que no tienen que comprobar el mecanismo de bloqueo de escritura como lo hacen los Dynaset. Por lo anterior, los SnapShots son los mejores indicados a utilizar cuando se trata de operaciones en que sólo se realicen lecturas.
12. Permite trabajar a nivel multiusuario. Varios usuarios pueden acceder a la base de datos desde distintas terminales, con la seguridad que esto requiere.
13. Visual Basic, incorpora dentro de sus instrucciones el manejo de las transacciones (acciones que modifiquen la base de datos). A través de estas instrucciones se podrán llevar a cabo por completo dichas transacciones o no ejecutarse para evitar inconsistencias en los datos.

### 3.2. DISEÑO DEL SCIF

En el diseño de todas las relaciones utilizadas en el SCIF, en cuanto a la normalización de las mismas, sólo se llegó hasta la 3FN, es decir, donde se tiene ya una dependencia funcional total y la no existencia de dependencias transitivas. Hasta aquí fue suficiente, ya que la representación de las relaciones no contienen redundancia de información que pudiera ocasionarnos problemas. Además, en ningún caso se presentaron dependencias multivaluadas. Las distintas relaciones que aparecen en el SCIF se fueron creando a través de las dependencias funcionales que se iban formando entre los distintos atributos que se encontraron durante el análisis.

Para realizar el diseño del sistema, el siguiente paso fue determinar los atributos que van a formar parte de la base de datos y reunirlos en una *Relación Universal* (Figura 3.1). A continuación se listan los atributos que formarán parte de la base de datos, así como su significado.

<i>Atributo</i>	<i>Significado</i>
• COD	Código del producto.
• NOM	Nombre del producto.
• LAB	Nombre del laboratorio al que pertenece el producto.
• PPS	Precio público sugerido.
• PFA	Precio farmacia.
• NTP	Número total de piezas.
• TPR	Tipo de producto (farmacéutico o no farmacéutico).
• IVA	Impuesto al Valor Agregado.
• TPC	Tipo de producto controlado.
• CAD	Fecha de caducidad del producto.
• LOT	Número de lote del producto.
• PRO	Nombre del proveedor.
• DES	Descuento que ofrecen los proveedores.
• DCT	Descuento a aplicar a cada producto.
• FVT	Fecha de venta.
• NPV	Número de piezas vendidas.
• PRU	Precio unitario.

Podemos apreciar en la lista anterior que existen dos clases de descuento, el que nos ofrece el proveedor por comprarle y el que se aplica a cada producto, este descuento sería el mismo si un sólo proveedor surtiera el producto, pero como puede haber el caso de que un mismo producto puede ser surtido por uno o más proveedores, entonces el descuento a aplicar a cada producto será el promedio de los descuentos que ofrezcan los proveedores que surtan esos productos.

Relación Universal								
COD	NOM	LAB	PPS	PFA	NTP	TPR	IVA	TPC
9500101	Aspirina	Alcon	15.00	10.00	5	Farmacéutico	0	---
9500101	Aspirina	Alcon	15.00	10.00	5	Farmacéutico	0	---
9500101	Aspirina	Alcon	15.00	10.00	2	Farmacéutico	0	---

CAD	LOT	PRO	DES	DCT	EVT	NPV	PRU
25/05/1999	8JKII	Nadro	15	14	15/01/1998	2	12.75
25/06/1999	9JKL	Nadro	15	14	20/12/1998	3	12.75
25/06/1999	9JKL	Autrey	13	14	20/12/1998	3	13.05

Figura 3.1 Relación Universal.

Como se puede apreciar en la tabla anterior, tenemos mucha información redundante, la cual dará como consecuencia muchos problemas cuando se deseen llevar a cabo actualizaciones sobre dicha base de datos, por lo que procedemos a obtener las dependencias funcionales correspondientes y de esta forma normalizarla, es decir, descomponer la Relación Universal en dos o más relaciones, de tal forma que solucionemos el problema de la redundancia de información, pero que al mismo tiempo nos permitan generar por medio de la unión la relación original sin pérdida de información.

Para determinar las dependencias funcionales, encontraremos la *clave primaria* de cada relación y los atributos que corresponden a cada clave. Vamos a nombrar a la primera relación INVENTARIO. En esta relación tenemos 2 claves candidatas, que son: COD y NOM, cualquiera de las dos las podemos utilizar como la clave primaria porque son identificadores únicos para la relación, ya que cada producto tiene un código y nombre único, es decir, no existen dos o más productos diferentes con el mismo código ni con el mismo nombre. Como queda a consideración del diseñador elegir la clave primaria cuando se presenta el caso anterior, se elegirá como clave primaria a COD.

De la Relación Universal tomamos los atributos que dependan de COD, entre los cuales encontramos: NOM, LAB, PPS, PFA, NTP, TPR, IVA, TPC y DCT, todos estos atributos dependen funcionalmente de COD, ya que para cada valor de COD, existe un sólo valor de los demás atributos asociados a él. Lo representamos por medio de un diagrama de dependencia funcional (Figura 3.2). La relación está representada en la Figura 3.3

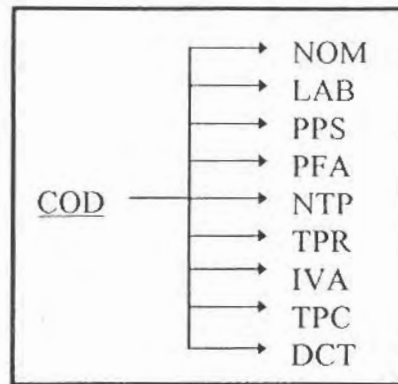


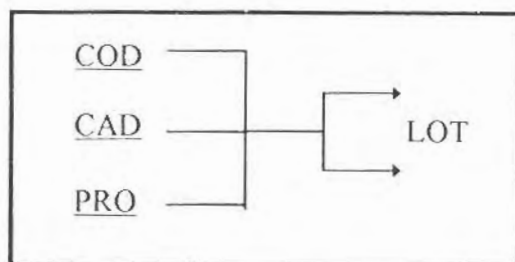
Figura 3.2 Diagrama de dependencia funcional de la relación INVENTARIO.

Relación INVENTARIO									
COD	NOM	LAB	PPS	PFA	NTP	TPR	IVA	TPC	DCT
9500101	Aspirina	Alcon	15.00	10.00	12	Farmacéutico	0	—	14

Figura 3.3 Relación INVENTARIO.

Esta relación se encuentra ya normalizada, porque existe una dependencia funcional total de los atributos no clave con respecto a la clave y no hay dependencias transitivas que ocasionen problemas, ya que los atributos que no pertenecen a la clave no dependen de ningún atributo no clave. Aunque se puede apreciar que NOM, un atributo no clave, puede definir a otros atributos no clave, va existir una excepción a la regla anterior siempre y cuando el atributo no clave que defina a otros atributos no clave, sea una clave candidata, y para este caso NOM lo es, ya que este atributo pudo haberse tomado también como la clave primaria en lugar de COD.

La siguiente relación la llamaremos CADUCIDAD. En esta relación estarán los atributos CAD, LOT, PRO, DES y COD, ya que por medio de éste último podremos hacer la referencia a los demás datos de la relación INVENTARIO y conocer de que producto son las fechas de caducidad. La clave primaria estará formada por los atributos COD, CAD, PRO ya que estos en su conjunto permiten identificar de forma única a una tupla dentro de la relación. La Figura 3.4 muestra el diagrama de dependencia funcional y la relación tal como quedaría se aprecia en la Figura 3.5.



**Figura 3.4** Diagrama de dependencia funcional de la relación CADUCIDAD.

Relación CADUCIDAD				
COD	CAD	PRO	LOT	DES
9500101	25/05/1999	Nadro	8JKH	15
9500101	25/06/1999	Nadro	9JKL	15
9500101	25/06/1999	Autrey	9JKL	13

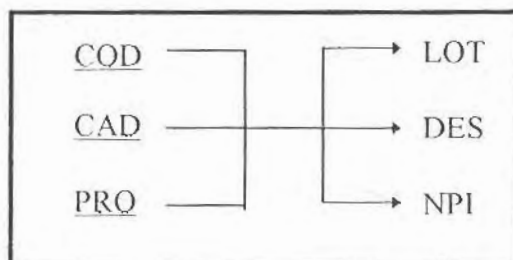
**Figura 3.5** Relación CADUCIDAD.

En esta relación no está representada el número de piezas de cada artículo por proveedor, la cual nos ayudará a determinar y aplicar el descuento correspondiente. Para esto agregaremos un nuevo atributo.

*Atributo*      *Significado*

- NPI      Número de piezas de cada producto por proveedor.

Con lo que tendremos un nuevo diagrama de dependencia funcional (Figura 3.6), así como una nueva relación CADUCIDAD (Figura 3.7).



**Figura 3.6** Diagrama de dependencia funcional agregando el nuevo atributo a la relación CADUCIDAD.



<u>COD</u>	<u>CAD</u>	<u>PRO</u>	<u>LOT</u>	<u>DES</u>	<u>NPI</u>
9500101	25/05/1999	Nadro	8JKH	15	5
9500101	25/06/1999	Nadro	9JKL	15	5
9500101	25/06/1999	Autrey	9JKL	13	2

Figura 3.7 La relación CADUCIDAD con el nuevo atributo.

Para esta relación se pueden observar dos situaciones que harán que haya un redundancia de información, porque tenemos dos dependencias transitivas y están son: el atributo DES no es totalmente dependiente de los atributos COD y CAD sino sólo de PRO, ya que DES es el descuento que ofrece cada proveedor a su cliente, más no es el descuento que se le aplica a cada producto; y el atributo LOT no es totalmente dependiente de los atributos COD y PRO sino sólo de CAD, ya que a cada fecha de caducidad le corresponde un número de lote determinado. Para solucionar este problema crearemos dos relaciones más, una llamada PROVEEDOR y la otra llamada CADLOT. El nuevo diagrama de dependencia funcional de CADUCIDAD, así como su relación se pueden apreciar en las Figuras 3.8 y 3.9.

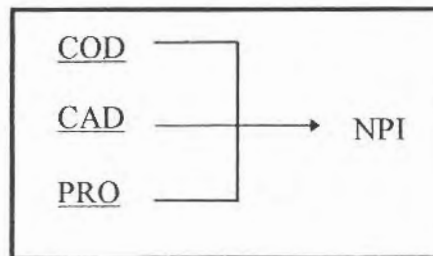


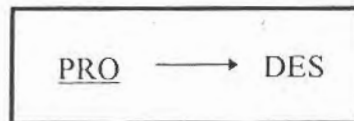
Figura 3.8 Diagrama de dependencia funcional para la relación CADUCIDAD.

<u>COD</u>	<u>CAD</u>	<u>PRO</u>	<u>NPI</u>
9500101	25/05/1999	Nadro	5
9500101	25/06/1999	Nadro	5
9500101	25/06/1999	Autrey	2

Figura 3.9 Relación CADUCIDAD.

Ahora si la relación CADUCIDAD se encuentra normalizada, porque existe una dependencia funcional total de los atributos no clave con respecto a la clave y ya no hay dependencias transitivas que puedan ocasionar problemas de actualización.

Como vimos anteriormente es necesario crear otras dos relaciones para evitar redundancia de información en la relación CADUCIDAD. Para la relación PROVEEDOR intervendrán los atributos PRO y DES, siendo la clave primaria PRO. Su diagrama de dependencia funcional así como la relación se muestran en la Figura 3.10 y Figura 3.11 respectivamente.



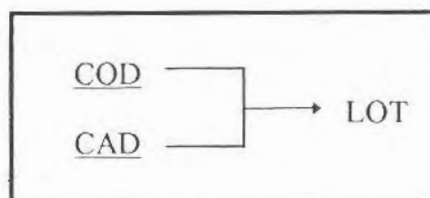
**Figura 3.10** Diagrama de dependencia funcional de la relación PROVEEDOR.

Relación PROVEEDOR	
<u>PRO</u>	DES
Nadro	15
Autrey	13

**Figura 3.11** Relación PROVEEDOR.

Con esta relación no hay ningún problema, se encuentra ya normalizada, existe una dependencia funcional total de los atributos. Tampoco hay dependencias transitivas.

La otra relación formada, CADLOT, contendrá los atributos COD, CAD Y LOT. El atributo COD lo necesitamos para conocer el nombre del producto que tiene asociados determinadas fechas de caducidad, haciendo referencia para ello a la relación INVENTARIO. En esta relación tendremos como clave primaria los atributos COD y CAD. El diagrama de dependencia funcional y la relación se pueden apreciar en la Figura 3.12 y Figura 3.13.



**Figura 3.12** Diagrama de dependencia funcional de la relación CADLOT.

COD	CAD	LOT
9500101	25/05/1999	8JKH
9500101	25/06/1999	9JKL

Figura 3.13 Relación CADLOT.

En la relación CADLOT existe una dependencia funcional total del atributo no clave con los atributos clave, además se encuentra sin dependencias transitivas, por tanto se encuentra normalizada.

Crearemos otra relación con los datos que nos faltan por representar, esta relación se llamará VENTAS y tendrá los atributos COD, FVT, NPV y PRU. A través del atributo COD podremos saber de qué producto se trata haciendo la referencia con la relación INVENTARIO. En esta relación la clave primaria estará conformada por los atributos COD, FVT y PRU, donde NPV depende funcionalmente de cada uno de los atributos de la clave primaria, es decir, tenemos ya una dependencia funcional total, y como sólo hay un atributo no clave se anula la posibilidad de dependencias transitivas, es por ello que esta relación ya se encuentra normalizada.

Su diagrama de dependencia funcional y la relación están representadas en las Figuras 3.14 y 3.15.

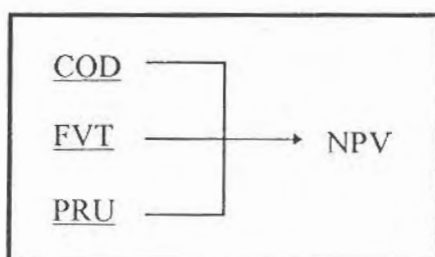
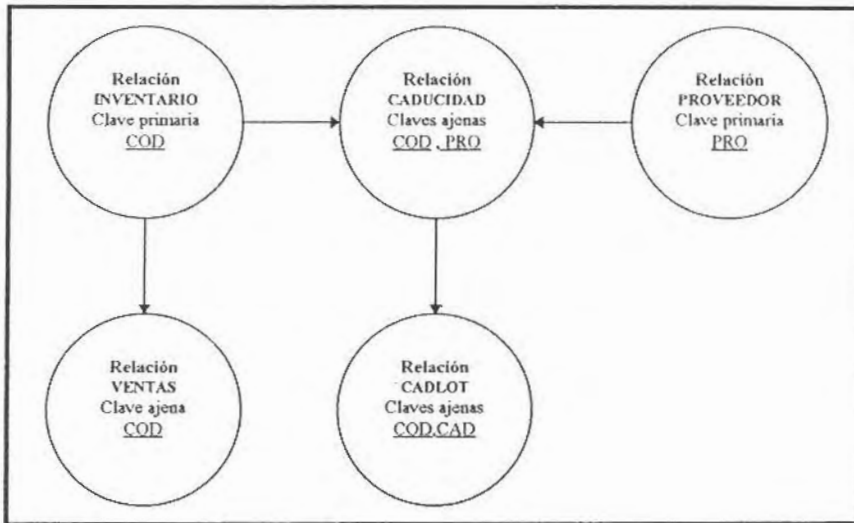


Figura 3.14 Diagrama de dependencia funcional de la relación VENTAS.

COD	FVT	PRU	NPV
9500101	15/01/1998	12.75	2
9500101	20/12/1998	12.75	3
9500101	20/12/1998	13.05	3

Figura 3.15 Relación VENTAS.

Si realizamos la reunión natural de cada una de las relaciones obtenidas por medio de sus claves ajenas, generaremos la Relación Universal sin pérdida de información. Esta situación la podemos apreciar en la (Figura 3.16 y Figura 3.17).



**Figura 3.16** Obtención de la Relación Universal a través de las claves ajenas de todas las relaciones.

$$\text{Relación Universal} = \text{Relación INVENTARIO} * \text{Relación CADUCIDAD} * \text{Relación PROVEEDOR} * \text{Relación VENTAS} * \text{Relación CADLOT}$$

Relación Universal								
COD	NOM	LAB	PPS	PFA	NTP	TPR	IVA	TPC
9500101	Aspirina	Alcon	15.00	10.00	12	Farmacéutico	0	---
9500101	Aspirina	Alcon	15.00	10.00	12	Farmacéutico	0	---
9500101	Aspirina	Alcon	15.00	10.00	12	Farmacéutico	0	---

DCT	CAD	LOT	PRO	NPI	DES	FVT	PRU	NPV
14	25/05/1999	8JKH	Nadro	5	15	15/01/1998	12.75	2
14	25/06/1999	9JKL	Nadro	5	15	20/12/1998	12.75	3
14	25/06/1999	9JKL	Autrey	2	13	20/12/1998	13.05	3

**Figura 3.17** Obtención de la Relación Universal a partir de las demás relaciones.

Hasta aquí tenemos representados los atributos de la relación inicial, sin embargo debido a las características del sistema, las cuales se mencionan al principio de este capítulo, es necesario la creación de otras relaciones que nos permitan representar la demás información solicitada por el usuario.

Representando los datos anteriores por medio de un diagrama de entidad-relación, tendríamos lo siguiente:

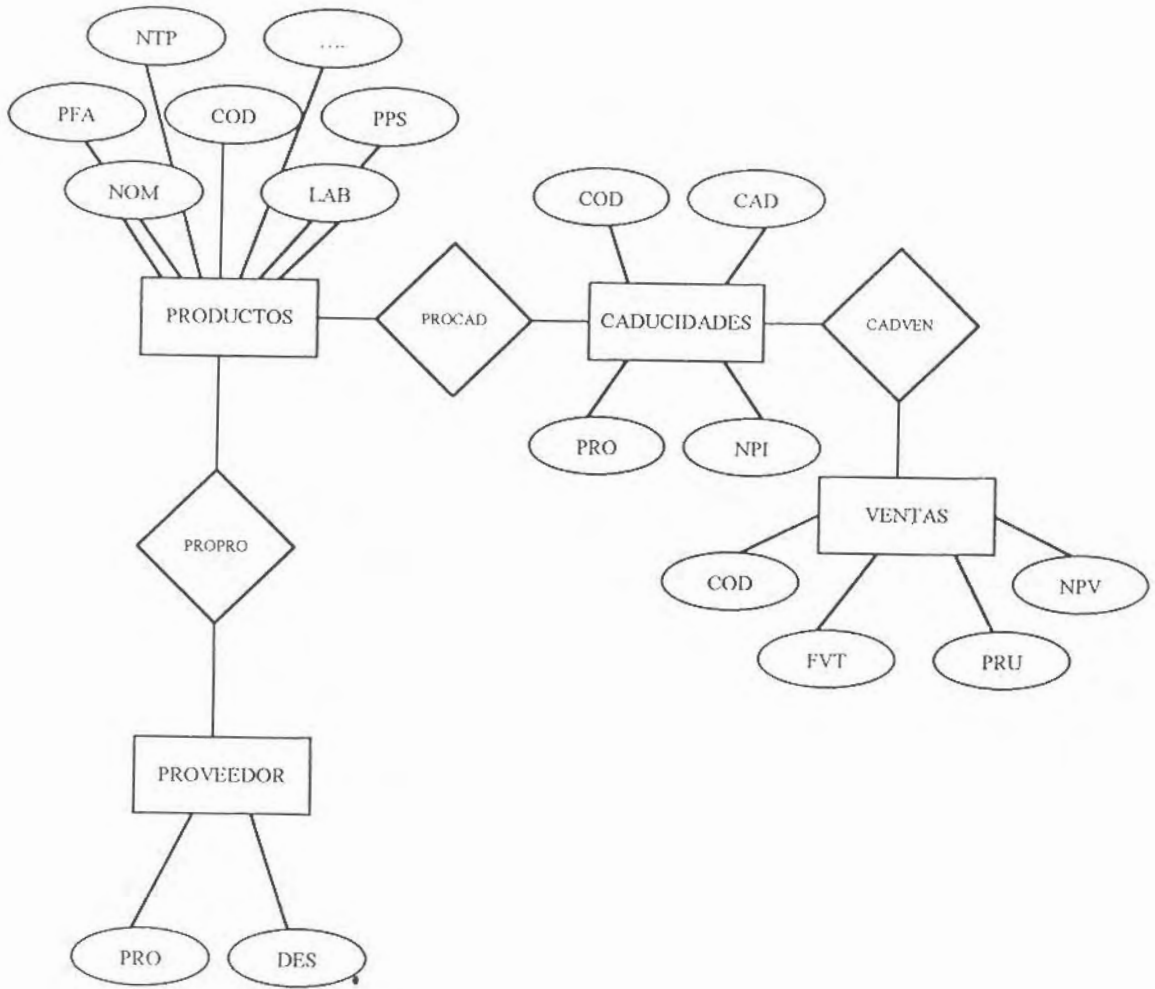


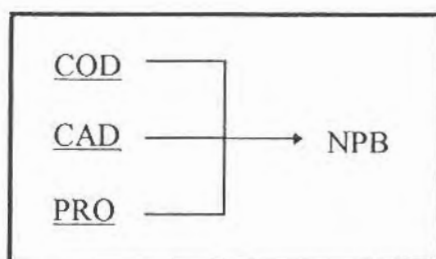
Diagrama entidad-relación

Dentro de la información manejada por el sistema está la de saber los productos que se dieron de baja por causa de su caducidad, así como de la empresa que proviene, ya que la empresa ofrecerá una bonificación por esos productos, de ahí su importancia. Por esta razón necesitaremos crear una relación que nos permita almacenar la información anterior. Esta relación la llamaremos BAJASCADU e intervendrán los atributos COD, CAD, PRO y NPB.

**Atributo**      **Significado**

- NPB      Número de piezas dadas de baja.

La clave primaria de esta relación estará dada por los atributos COD, CAD y PRO. Su diagrama de dependencia funcional y la relación, se presentan en las Figuras 3.18 y 3.19.



**Figura 3.18** Diagrama de dependencia funcional de la relación BAJASCADU.

Relación BAJASCADU			
<u>COD</u>	<u>CAD</u>	<u>PRO</u>	NPB
9500101	25/05/1999	Nadro	5
9500101	26/06/1999	Nadro	5
9500101	26/06/1999	Autrey	2

**Figura 3.19** Relación BAJASCADU.

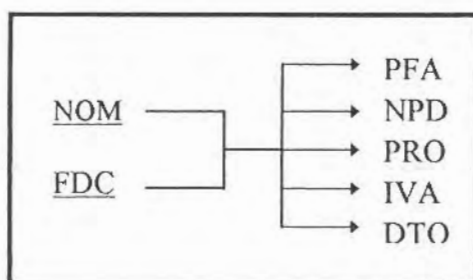
El atributo no clave que en este caso es NPB, tiene una dependencia funcional total con cada uno de los atributos que forman la clave primaria, y por ser sólo un atributo no clave no puede existir dependencia transitiva alguna. Encontrándose la relación normalizada.

Como se menciona en el análisis, parte de las operaciones que se desean realice el sistema es llevar un control sobre las devoluciones que sobre las compras se tengan. Esta acción se va a llevar a cabo, como ya vimos, cuando nos lleguen productos que no se pidieron y que sin embargo se están cargando a la cuenta del cliente. Para ello, vamos a crear una relación en la que se pueda representar esta situación. A esta relación la llamaremos DEVCOMPRAS. Esta relación contendrá los atributos NOM, FDC, PFA, NPD, PRO, IVA y DTO.

<i>Atributo</i>	<i>Significado</i>
• FDC	Fecha de devolución sobre compras.
• NPD	Número de piezas devueltas.
• DTO	Descuento para cada producto según proveedor.

Intervendrán los atributos anteriores y no haremos referencias hacia otras relaciones como por ejemplo INVENTARIO, porque las devoluciones se pueden tratar de productos que no estén dentro de la base de datos. Además de que el conocimiento de estos datos es importante para conocer la bonificación que tendrá que realizar un determinado proveedor al cliente.

En esta relación, la clave primaria estará formada por la combinación de dos atributos NOM y FDC, ya que con ambos es suficiente para identificar dentro de una relación una tupla de otra. Su diagrama de dependencia funcional se muestra en la Figura 3.20 y la relación en la Figura 3.21.



**Figura 3.20** Diagrama de dependencia funcional de la relación DEVCOMPRAS.

Relación DEVCOMPRAS						
<u>NOM</u>	<u>FDC</u>	PFA	NPD	PRO	IVA	DTO
Miconazol	01/03/1999	30.00	10	NADRO	0	15

**Figura 3.21** Relación DEVCOMPRAS.

Para esta relación DTO no tiene una dependencia transitiva con PRO, ya que es el descuento que hay que aplicar directamente a ese producto para obtener la bonificación. Por lo tanto la relación está normalizada, además existe una dependencia funcional total.

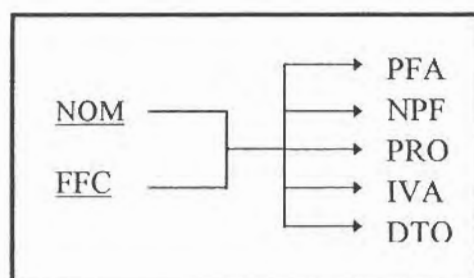
Otra relación que requerimos para representar información solicitada por el cliente, es una que nos permita saber cuánto deberá bonificar un proveedor a su cliente por concepto de productos que pidió, los cuales no le fueron surtidos y sin embargo se le están cobrando.

Para ello requeriremos una relación que llamaremos FALCOMPRAS conteniendo los atributos NOM, FFC, PFA, NPF, PRO, IVA y DTO.

<i>Atributo</i>	<i>Significado</i>
• FFC	Fecha del faltante sobre compras.
• NPF	Número de piezas faltantes.

Al igual que en la de devoluciones sobre compra, se utilizan varios datos y no se hace referencia a otras relaciones para conocer ciertos datos de un producto, porque aquí también se puede dar el caso de que el o los productos faltantes no se encuentren en la base de datos.

En esta relación, la clave primaria estará compuesta por la combinación de dos atributos, NOM y FFC, ya que con ambos es suficiente para identificar una tupla de otra. Su diagrama de dependencia funcional y la relación se representan por medio de las Figuras 3.22 y 3.23.



**Figura 3.22** Diagrama de dependencia funcional de la relación FALCOMPRAS.

Relación FALCOMPRAS						
NOM	FFC	PFA	NPF	PRO	IVA	DTO
Miconazol	01/03/1999	30.00	10	NADRO	0	15

**Figura 3.23** Relación FALCOMPRAS.

Esta relación también se encuentra ya normalizada. Con una dependencia funcional total por parte de sus atributos no claves, y sin dependencias transitivas.

Como se vio en el análisis, el IVA de los productos puede presentar variaciones globales, es decir, un cambio para todos los productos con un mismo valor del IVA. Esta relación (Figura 3.24) la nombraremos IVA y contendrá sólo un atributo, llamado IVA.



Relación IVA	
	IVA
	0
	15

Figura 3.24 Relación IVA.

Para terminar con el diseño de nuestras relaciones que formarán nuestra base de datos, nos falta representar la situación en la cual le devuelvan al cliente productos por concepto de una venta. El problema aquí radica en que cuando nos hacen una devolución no sabemos a que proveedor pertenece dicho producto. Si nos quedaran piezas de esos productos, sólo tendríamos que hacer una consulta y ver que proveedor tiene asociado y el problema está solucionado, pero puede darse el caso de que ya no tengamos de esos productos en existencia. También deberemos saber el descuento que le hace cada proveedor a sus productos, para que a partir de este, si es más de uno el proveedor que surte un mismo producto, se pueda realizar el promedio y obtener el descuento real a aplicar a la hora de vender, que vendría siendo el atributo DCT de la relación INVENTARIO. Estas dos situaciones se pueden resolver a través de una relación que llamaremos PRODPRO, en la que intervendrán los atributos: COD, PRO, DTO y NPP.

<i>Atributo</i>	<i>Significado</i>
-----------------	--------------------

- |       |                                 |
|-------|---------------------------------|
| • NPP | Número de piezas por proveedor. |
|-------|---------------------------------|

Para esta relación tenemos como clave primaria a los atributos COD y PRO. Las Figuras 3.25 y 3.26 muestran el diagrama de dependencia funcional y la relación.

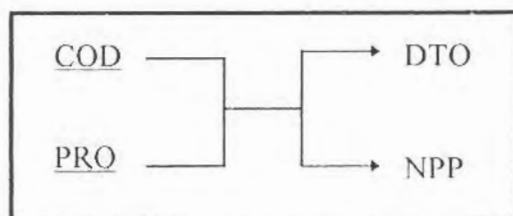


Figura 3.25 Diagrama de dependencia funcional de la relación PRODPRO.

COD	PRO	DTO	NPP
9500101	Nadro	15	10
9500101	Autrey	13	2

Figura 3.26 Relación PRODPRO.

Para esta relación contamos ya con una dependencia funcional total sin dependencias transitivas que pudieran ocasionarnos problemas, por ello también esta normalizada.

Con esta relación sabremos quienes pueden ser los posibles proveedores de un determinado producto, porque en ella tendremos como un historial de quién o quiénes surten los productos. Así mismo, cuando tengamos más de un proveedor que sea el que surta un mismo producto, bastará sacar el promedio de los descuentos, los cuales aparecen dentro de la relación.

#### Reglas de integridad para el SCIF:

- 1) No puede haber dos productos diferentes con el mismo código.
- 2) El código de los productos estará formado por *números enteros positivos*.
- 3) No puede haber dos productos diferentes con el mismo nombre.
- 4) Los nombres de los productos se compondrán de *datos alfanuméricos*. Con un espacio en blanco entre palabra y palabra.
- 5) El Precio Público Sugerido podrá tomar sus valores de los *números reales positivos mayores de cero*, con dos dígitos después del punto decimal.
- 6) El Precio Farmacia tomará sus valores de los *números reales positivos mayores de cero*, con dos dígitos después del punto decimal.
- 7) El Precio Público Sugerido siempre es mayor al Precio Farmacia.
- 8) Los valores para el número de piezas estarán formados por *números enteros positivos*, incluyendo al cero.
- 9) El número total de piezas de cada producto deberá ser igual a la suma de las piezas que existen por proveedor de ese mismo producto.
- 10) No se podrá dar de baja más piezas de un producto de los que se tiene en inventario.
- 11) No se podrán vender más piezas por producto de los que se tiene en inventario.
- 12) Para el tipo de cada producto se considerarán *datos lógicos*, pudiendo tomar sólo dos valores: F(farmacéutico) y N(no farmacéutico).
- 13) El IVA tomará su valores de los *números enteros positivos*, incluyendo al cero. Estos valores no podrán ser mayores a 100.
- 14) No se podrá dar de baja valores para el IVA si dicho valor está asociado a productos.

- 15) El tipo de producto controlado únicamente tendrá tres valores, y estos pueden ser cualquiera de las tres primeras letras del alfabeto: A (grupo 1), B (grupo 2) y C (grupo 3). En su defecto no contendrá ninguno de ellos.
- 16) Los descuentos que se aplican a los productos, así como los que hace el proveedor a los mismos, serán *números enteros positivos*, incluyendo al cero y no pueden ser valores mayores a 100.
- 17) La notación de fechas manejadas será día/mes/año tomando los siguientes valores: para el día serán *números enteros positivos no mayores a 31*; para el mes *números enteros positivos no mayores a 12* y para el año *números enteros positivos* entre 100 y 9999 inclusive.
- 18) El nombre del proveedor tomará sus valores a partir de *datos alfanuméricos*. Si está compuesto de dos o más palabras, solo habrá un espacio en blanco entre ellas.
- 19) No se podrá dar de baja a un proveedor mientras éste tenga productos asignados.
- 20) El número de lote de los productos estará formado por *datos alfanuméricos*.
- 21) El precio unitario de cada producto se formará con *números reales positivos mayores a cero*. Teniendo dos dígitos después del punto decimal.
- 22) Al modificar algún valor dentro de la base de datos, se deberá hacer las actualizaciones en las relaciones correspondientes donde aparezca dicho dato. Esta actualización se llevará a cabo completamente o en su defecto no se realizará.

### 3.3. DICCIONARIO DE DATOS.

<i>NOMBRE</i>	<i>TIPO</i>	<i>DESCRIPCIÓN</i>	<i>DOMINIO</i>
COD	Cadena	Código del producto.	De 1 a 20 dígitos
NOM	Cadena	Nombre del producto.	Cadena de hasta 50 caracteres
LAB	Cadena	Nombre del laboratorio al que pertenece el producto.	Cadena de hasta 15 caracteres
PPS	Flotante	Precio público sugerido.	Valores > 0
PFA	Flotante	Precio farmacia.	Valores >0
NTP	Entero	Número total de piezas.	Valores >=0
TPR	Carácter	Tipo de producto.	-F (Farmacéutico) -N (No farmacéutico)
IVA	Entero	IVA	valores >=0
TPC	Carácter	Tipo de producto controlado.	-A (grupo 1) -B (grupo 2) -C (grupo 3)
DCT	Flotante	Descuento a aplicar a cada producto.	Valores >=0

<i>NOMBRE</i>	<i>TIPO</i>	<i>DESCRIPCIÓN</i>	<i>DOMINIO</i>
CAD	Date	Fecha de caducidad.	Tipo fecha
NPB	Entero	Número de piezas dadas de baja.	Valores $\geq 1$
PRO	Cadena	Nombre del proveedor.	Cadenas de hasta 15 caracteres
NPI	Entero	Número de piezas de cada producto que existen por proveedor (para fechas de caducidad).	Valores $\geq 1$
LOT	Cadena	Número de lote.	Cadenas de hasta 15 caracteres
NPD	Entero	Número de piezas devueltas.	Valores $\geq 1$
FDC	Date	Fecha de devolución sobre compras.	Tipo fecha
DTO	Entero	Descuento para cada producto según proveedor.	Valores $\geq 0$ y $\leq 100$
NPF	Entero	Número de piezas faltantes.	Valores $\geq 1$
FFC	Date	Fecha del faltante sobre compras.	Tipo fecha
NPP	Entero	Número de piezas por proveedor.	Valores $\geq 1$
DES	Entero	Descuento que ofrecen los proveedores.	Valores $\geq 0$ y $\leq 100$
FVT	Date	Fecha de venta.	Tipo fecha
NPV	Entero	Número de piezas vendidas.	Valores $\geq 1$
PRU	Flotante	Precio Unitario.	Valores $\geq 1$

De acuerdo a las características vistas en este capítulo, a continuación se enlistan las operaciones que serán llevadas a cabo por el Sistema de Control de Inventarios para Farmacias (SCIF).

### *PRODUCTOS*

- Altas
- Bajas
- Consultar
- Modificar

### *VENTAS*

- Punto de venta
- Consultar ventas
- Devoluciones sobre ventas

### COMPRAS

- Compras
- Devoluciones sobre compras
- Faltantes sobre compras

### PROVEEDORES

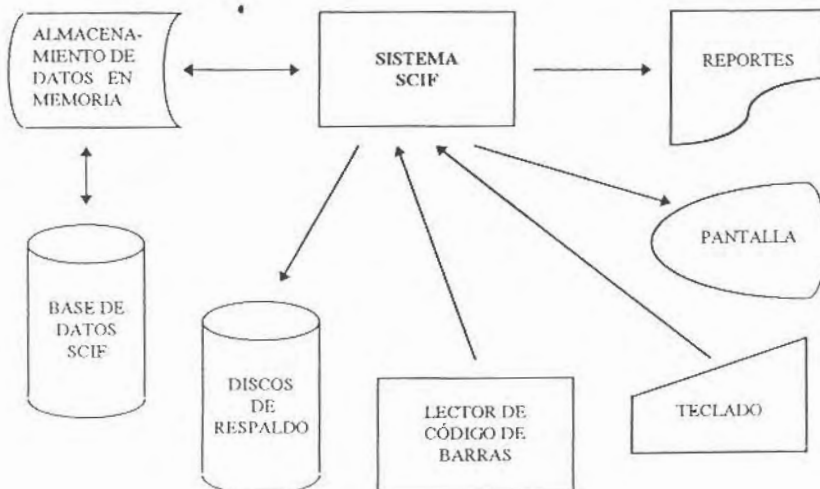
- Altas
- Bajas
- Consultar
- Modificar

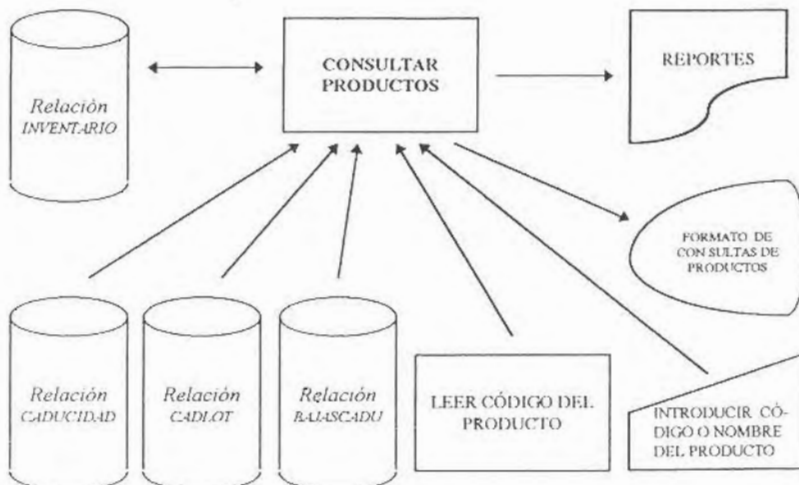
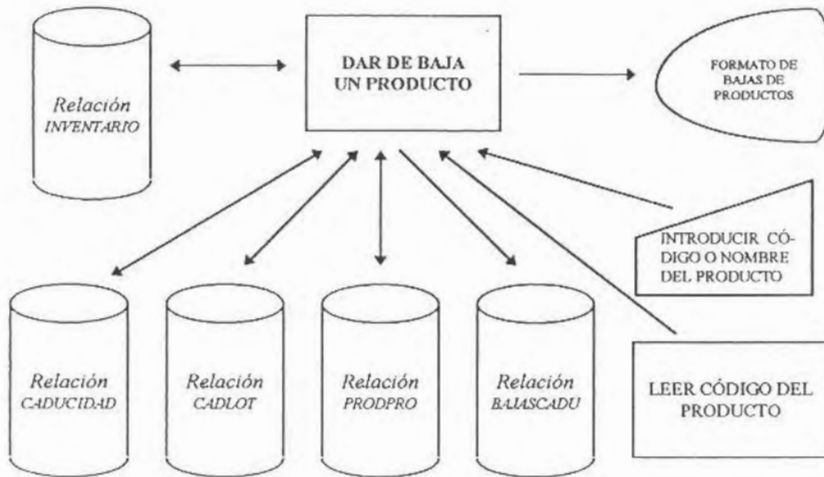
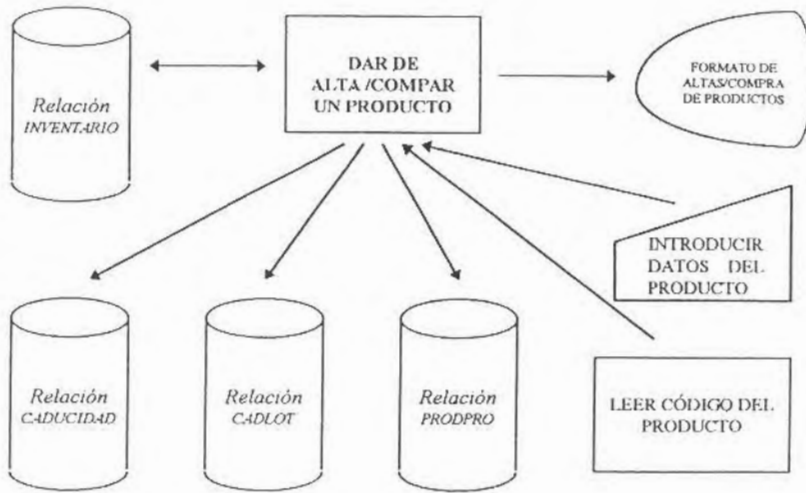
### IVA

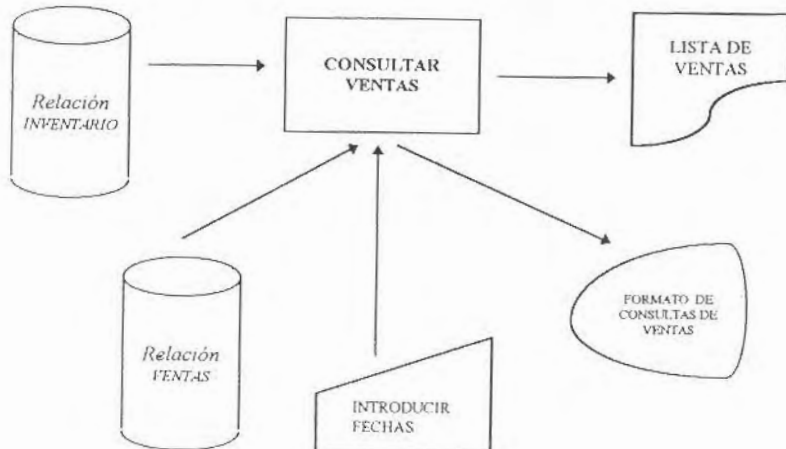
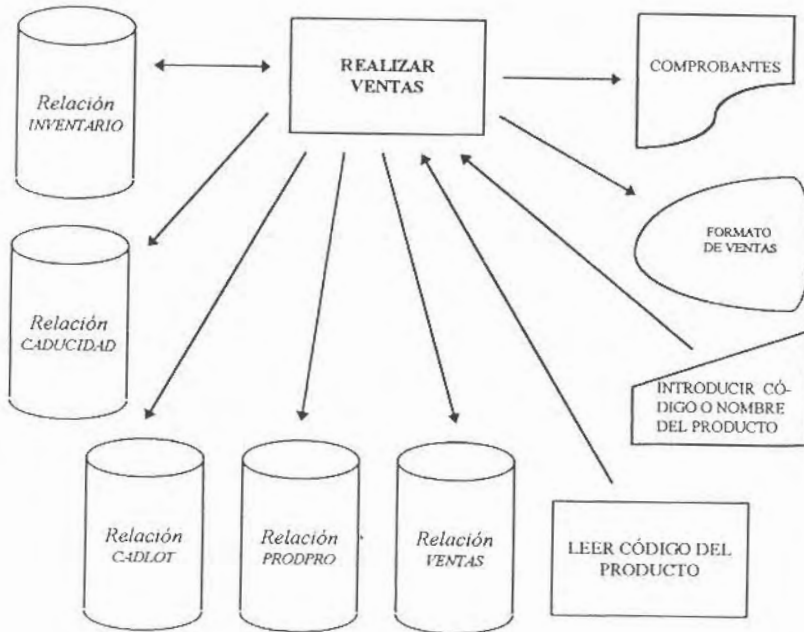
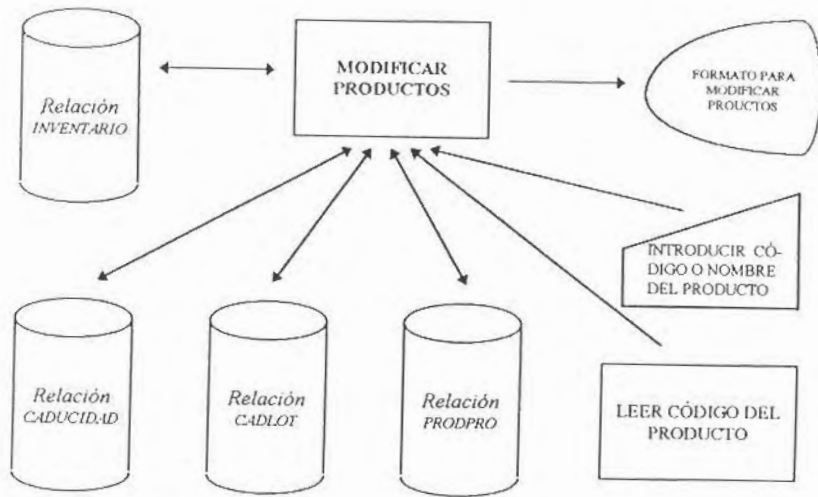
- Altas
- Bajas
- Consultar
- Modificar

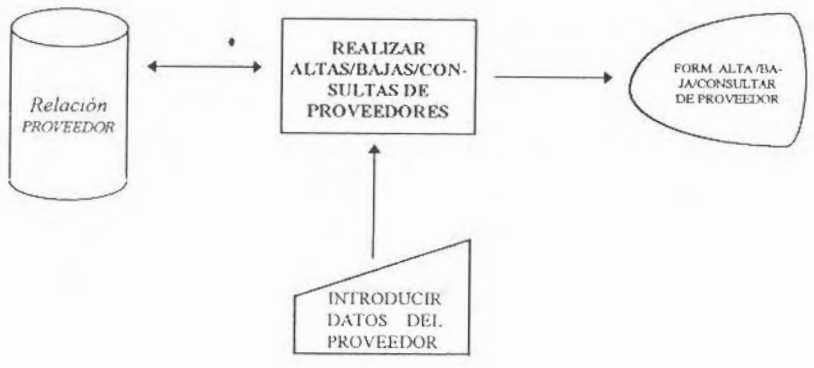
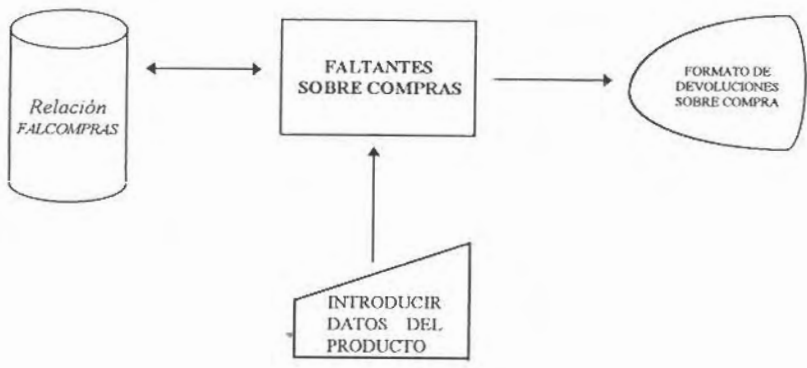
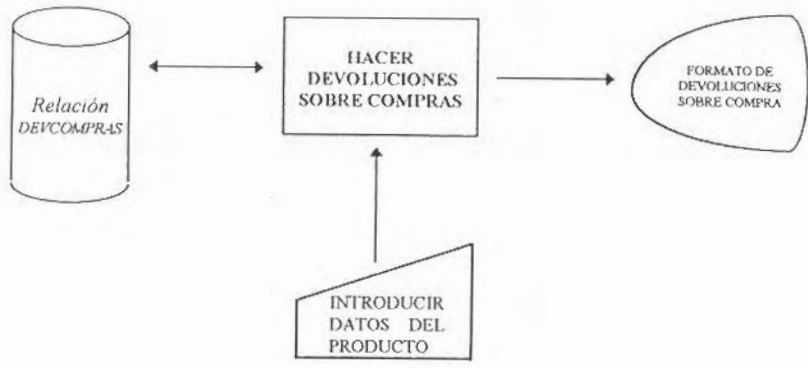
## 3.4. DIAGRAMAS DE FLUJO.

A continuación se muestran los diagramas de flujo para el SCIF de manera general y para cada uno de los procesos antes mencionados. Esta representación es a través de un *Diagrama de flujo de sistema*, el cual nos muestra el flujo de la información entre los diferentes soportes físicos que integran el sistema .

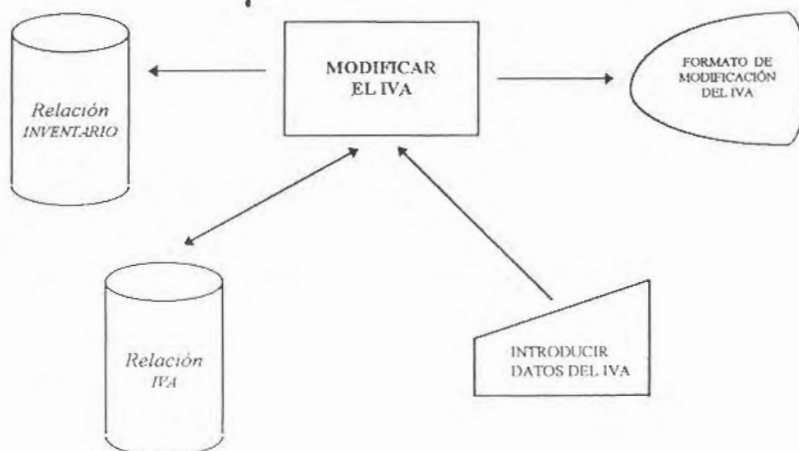
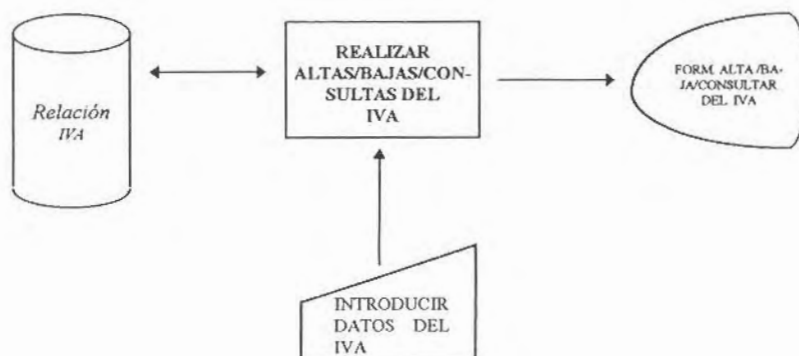
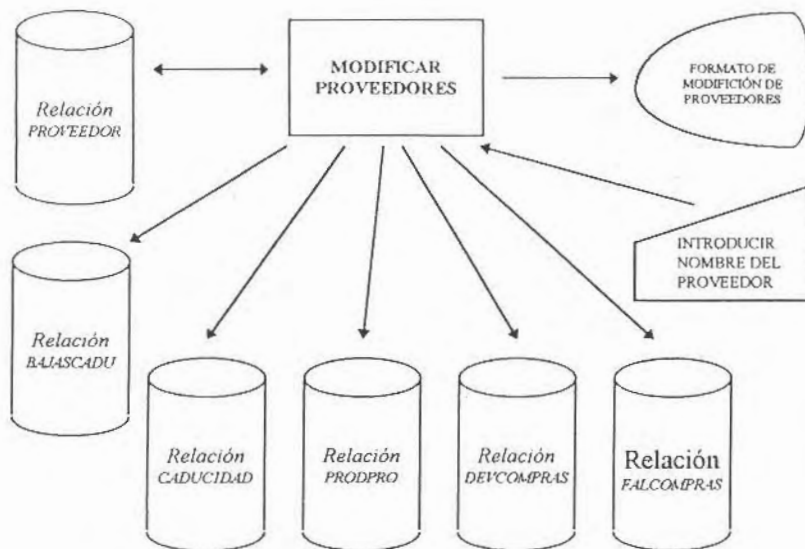












*Cuadro comparativo entre el SCIF y la manera de como se hace el trabajo en la farmacia*

SIN SISTEMA	CON EL SCIF
Cuando llega un cliente a la farmacia y pregunta por un algún producto, la encargada se va a buscar en los anaqueles a revisar si lo tiene.	Hay que teclear el nombre del producto. Se informará si existe o no.
Una vez encontrado el producto y con la ayuda de una calculadora se procede a calcular el precio. Aplicándole al precio marcado en el producto su descuento correspondiente.	Al teclear el nombre del producto y si este se encuentra, aparecerán sus datos correspondientes, entre ellos: su precio ya con su descuento y el número de piezas en existencia.
Cuando se realiza una venta hay que obtener el total del precio de los productos y aplicar el descuento. Valiéndose para esto de una calculadora.	El sistema cuenta con <i>punto de venta</i> , sólo hay que pasar el código de barras de los productos por el lector óptico y en pantalla irán apareciendo los productos que van a formar parte de la venta y el total a pagar. Informará también la diferencia a entregar al cliente.
Una vez realizada la venta, en una lista, se anotan los productos vendidos así como su precio. Diario deben anotarse los productos que se venden.	Los datos de las ventas son almacenadas en la base de datos. Con lo que podemos disponer en todo momento de la información referente a las ventas, solo hay que teclear las fecha de la cual se quiere saber cuales fueron las ventas de ese día, o en su defecto se pueden introducir lapsos de fechas.
Cada mes hay que hacer las notas que corresponden a las ventas de ese período. Para esto se recurre a las listas de ventas de ese mes.	Sólo hay que teclear las fechas de las cuales se desea saber las ventas realizadas en ese período.
A principio de año se revisan todos los productos para saber su fecha de caducidad. Si caducan en el transcurso del año que comienza, son anotados en una libreta separados por meses. Los productos que llegan en el transcurso de ese año no son anotados sus caducidades.	Cada que se adquieren productos es necesario poner su fecha de caducidad. Con lo que se tiene la información de caducidad de cada uno de los productos.
Cada cierto período de tiempo hay que tener listos los productos que están próximos a caducar, y que se encuentran entre los meses que solicitó con anterioridad el agente. Para ello hay que revisar la lista hecha a principios de año.	Se deben introducir las fechas de las que se quiere saber que productos caducan en un determinado período, mostrándose la lista de estos. Dicha lista puede ser impresa.

SIN SISTEMA	CON EL SCIF
A finales de cada año y por cuestiones contables es necesario saber el total del inventario. Para ellos se recurre a un recuento físico de todos y cada uno de los productos que se tienen en existencia.	El sistema tiene incluida una opción de ver todo el inventario, así como el total de los mismos. También permite imprimir su reporte.
Cuando se tiene una devolución o un faltante sobre compras, hay que anotar en un formato, los productos en cuestión. Después cuando llega la nota de crédito se debe revisar el formato y hacer los cálculos correspondientes para obtener el valor de la bonificación.	Estos datos hay que introducirlos en la computadora y con una sola consulta se puede saber el total de la bonificación.
	Se puede obtener el total en ventas efectuadas por períodos que elija el usuario
	Se puede consultar los productos más vendidos en ciertos períodos de tiempo.

*Cuadro comparativo entre un Sistema de Información Gerencial y el SCIF.*

SIG	SCIF
La información suministrada es oportuna, precisa y verídica.	La información que maneja es la que requiere el dueño una farmacia y esta todo el tiempo disponible, y puede obtener la información que se desee en cualquier momento. Comprueba que los datos que se introducen sean verdaderos en el sentido de que sean válidos.
Provee información interna, la cual se origina dentro de las empresas y son necesarias para llevar a cabo las operaciones diarias.	Provee información respecto al manejo de los inventarios y las operaciones que tienen que ver con los mismos, como son: conocer cual es la situación de un determinado producto como su precio, piezas en existencia; relación de las ventas efectuadas.
Suministran información a usuarios externos.	Imprime reportes que son necesarios para cuestiones contables y que se entregan al contador, entre ellos están el total y descripción detallada de los productos que forman el inventario, y los ingresos obtenidos en ciertos lapsos de tiempos por concepto de las ventas.

SIG	SCIF
Permite realizar un gran número de operaciones, que tienen que ver dentro de la empresa.	Existen varios procesos que se pueden llevar a cabo y con los cuales se puede manipular la información que se tiene. Por ejemplo, realizar cambios de precios, dar de alta o baja los productos u obtener un listado de los mismos, bajo ciertos criterios ya establecidos.
Deben poder ofrecer a los directivos de las empresas, la ayuda necesaria en cuestión de información, para la toma de decisiones.	Se pueden obtener informes de los productos más vendibles, así como saber como estuvieron las ventas en determinadas épocas del año, y con ello tener un seguimiento de cómo va el negocio, con lo cual el dueño pueda tomar decisiones en base a dicha información.

### 3.5. CARACTERÍSTICAS DEL SCIF

En esta sección se hablará de las características más importantes del SCIF en cuanto a las restricciones de integridad y seguridad se refieren. Como vimos en el apartado 1.1.7., estas restricciones son un medio para asegurar que los cambios que se vayan a realizar dentro de la base de datos por usuarios autorizados, no resulten en pérdida de información, que nos puedan originar una inconsistencia en los datos.

Analizaremos en primer lugar las *restricciones de dominio*. Este tipo de restricciones nos van a indicar el conjunto de valores de un mismo tipo, de los cuales podrán tomar su valor los diferentes atributos dentro de las relaciones. El SCIF cuenta con un *manejador de errores*, que detecta cualquier valor que no corresponda con los datos que se le están pidiendo. La Figura 3.27 presenta un ejemplo de este caso.

Esta es la ventana para dar de alta a los productos. Como se puede observar, al introducir un valor para el código del producto, se presenta un mensaje avisándonos de un error en los datos que acabamos de introducir. Como vimos en el análisis, el código de un producto (que en este caso será su código de barras) está formado por un conjunto de valores enteros y aquí aparecen letras, el manejador de errores se da cuenta y envía el mensaje correspondiente. Mientras no se corrija este error, el SCIF no permitirá al usuario realizar ninguna otra acción.

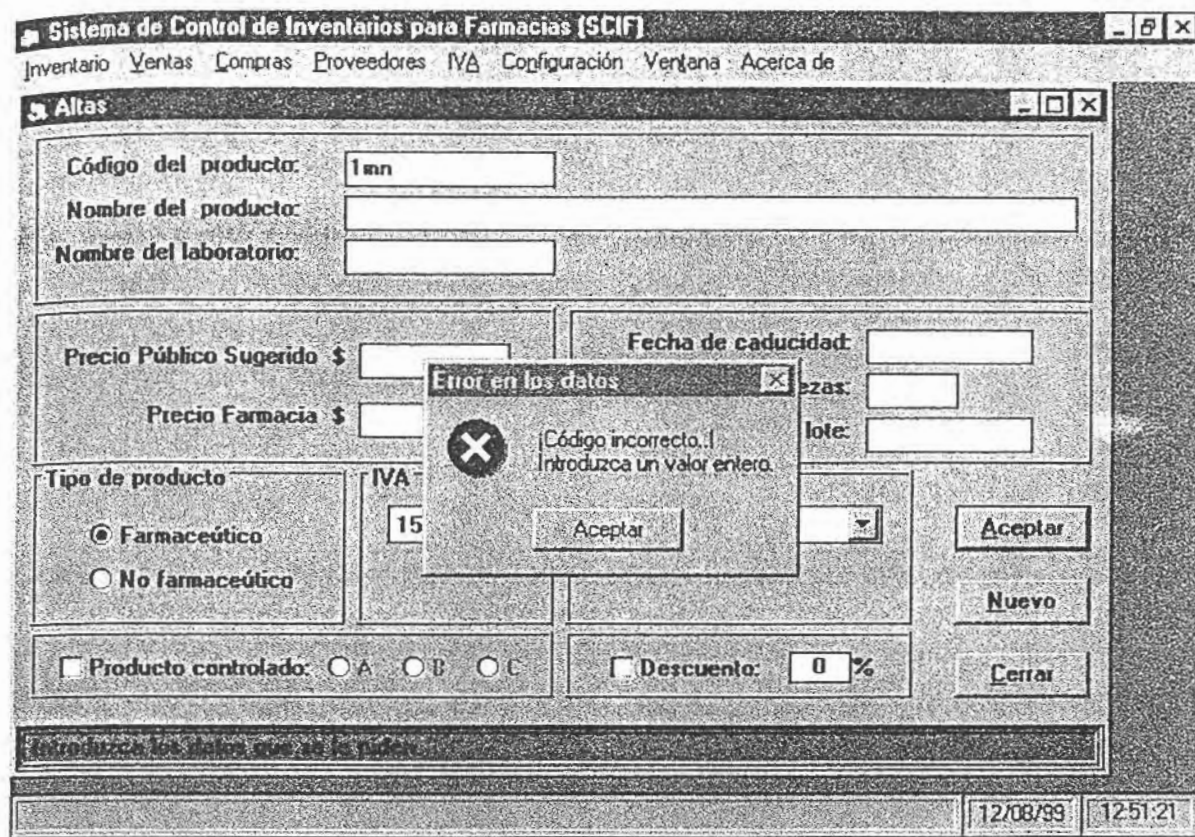


Figura 3.27 Respuesta del SCIF ante datos erróneos.

También el SCIF cuenta con un *manejador de datos estándar*, que permitirá que los datos se guarden bajo una misma presentación, cuando estos sean cadenas. Por ejemplo, para los nombres la primera letra será mayúscula y las demás minúsculas, y sólo habrá un espacio entre palabra y palabra. Lo que se busca con lo anterior es que al presentar datos en pantalla estos sean de modo uniforme. Este manejador también tiene otro fin, el de quitar espacios a los datos que no lo requieran y que por algún error se hayan introducido, podemos mencionar aquí las fechas de caducidad, precios, número de piezas, etc. Las Figuras 3.28 y 3.29 presentan estos casos.

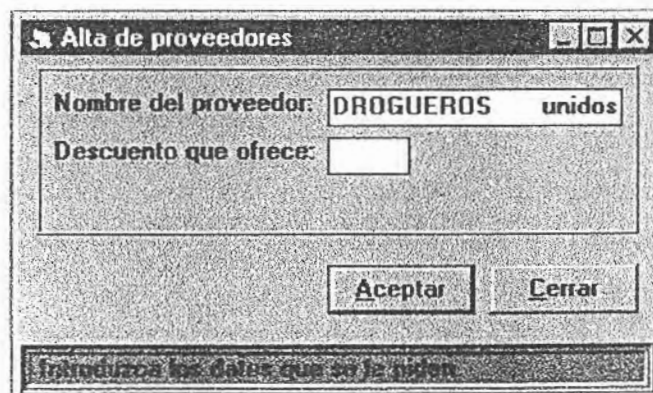


Figura 3.28 Entrada de datos para el nombre del proveedor.

Figura 3.29 Cambio realizado por el SCIF.

Con lo que respecta a *valores nulos*, el SCIF sólo permite este tipo de valores en los atributos que guardan información acerca de la fecha de caducidad, ya que hay productos que no tienen fecha de caducidad. Para los demás datos es necesario introducir valores válidos (Figura 3.30).

Figura 3.30 Respuesta del SCIF ante la falta de datos.

Si se han introducido todos los valores y son correctos, se podrá llevar a cabo la acción seleccionada, pero si falta algún valor será necesario introducirlo o de lo contrario no podrá efectuar ninguna operación. El SCIF avisará de la falta de algún dato, en caso de que no estén completos.

En cuanto a la *integridad referencial*, se toma como base a los diagramas de flujo y en ellos se ve, que relaciones se ven afectadas al realizar un cambio en una determinada relación.

Pasando a aspectos de seguridad, el SCIF cuenta con clave de acceso para poder ingresar al sistema (Figura 3.31). Sólo se permite 2 intentos de equivocación al teclear la clave, ya que en caso de una tercera el sistema se saldrá.

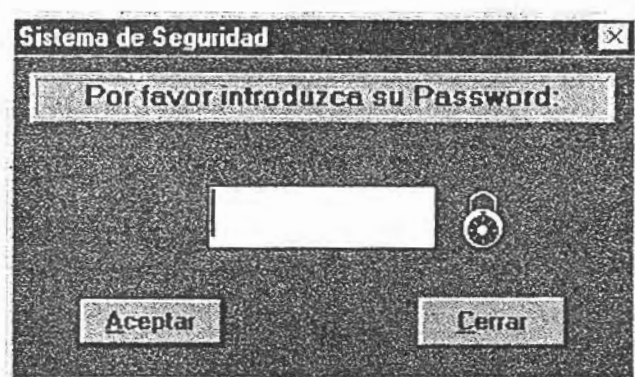


Figura 3.31 Ventana inicial para poder acceder al SCIF.

Al teclear el usuario su clave esta aparece en forma de asteriscos, esto es con la finalidad de que no la pueda ver otra persona (Figura 3.32).



Figura 3.32 Al teclear su clave el usuario esta no será vista en pantalla.

Si su clave es correcta el candado que aparece en la figura se abrirá y entrará sin ningún problema a l sistema. En caso contrario se le enviará un mensaje de error (Figura 3.33).

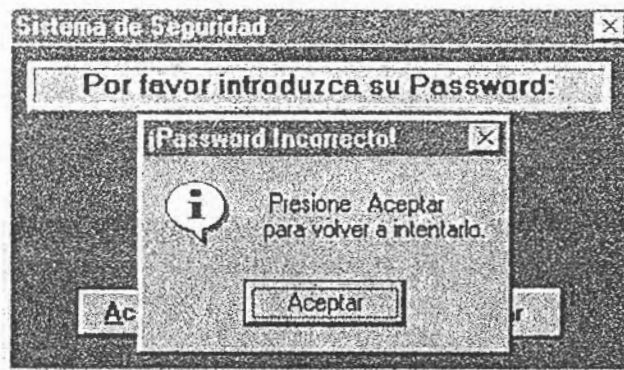


Figura 3.33 Mensaje dado por el SCIF cuando la clave es incorrecta.

Dentro del menú del sistema existe una opción llamada *Configuración* en la cual puede el usuario cambiar su clave si así lo desea.

Como parte de la seguridad del sistema, se pueden realizar respaldos de la información en discos flexibles

Otro aspecto importante a cuidar dentro del SCIF, fue el de la atomicidad, es decir, que al llevarse a cabo una operación dentro del sistema, está se llegue a realizar completamente. Lo anterior se refiere a que si se necesitan afectar varias relaciones dentro de cualquier operación, todas se afecten, de lo contrario si sólo se llega a realizar cambios sobre una parte tendremos como resultado una inconsistencia en nuestra base de datos. Existen factores<sup>9</sup> que pueden ocasionar que no se llegue a completar una operación. Dentro de la programación se utilizó una instrucción, que maneja Visual BASIC, que es la siguiente:

1. *BeginTrans*
2. *CommitTrans*
3. *Rollback*

Con la primera instrucción marcamos donde empieza una operación, y todos los cambios que se realicen a partir de ahí, se harán en un buffer de memoria. Al terminar la operación tenemos dos opciones que se hagan los cambios físicamente en la base de datos o no se lleve a cabo ningún cambio. Si elegimos la primera opción, se activará la instrucción *CommitTrans* e inmediatamente, en una sola operación de escritura se llevarán a cabo los cambios en la base de datos, pero si elegimos la instrucción *Rollback* los cambios no se realizarán nunca. Por lo tanto si llega a haber un fallo en el sistema que no permita llegar al final de la instrucción *CommitTrans*, entonces los cambios no se realizarán y la base no estará en un estado inconsistente.

Para finalizar, el SCIF puede trabajar con un *Lector de Código de Barras*. Lo que lee este lector de los productos viene siendo el código de barras que aparece en cada producto y es precisamente este código el que se solicita en las diferentes opciones del sistema, con lo cual se vuelve más ágil la búsqueda de los datos de un determinado producto.

<sup>9</sup> Para más detalles ver la sección 1.1.8.



## CONCLUSIONES

Las aportaciones que el SCIF ofrece se pueden resumir en los siguientes puntos:

El sistema va enfocado a pequeñas y medianas farmacias, lo cual no quiere decir que las grandes farmacias no lo puedan utilizar. Sin embargo, tiene una limitante en cuanto a que no trabaja bajo un ambiente de red por el tipo de farmacia al que el sistema está enfocado.

La interfaz con el usuario es diferente a la de otros sistemas que existen en el mercado, mientras los otros se manejan en un ambiente para DOS, este se maneja en un ambiente para Windows, lo que hace que su presentación sea mejor y le da confianza al usuario.

El sistema es "amigable con el usuario", ya que le permite trabajar con toda confianza, pues cuenta con un manejador de errores que impide la entrada de datos que no sean los adecuados, además de ventanas de diálogo de confirmación las cuales aparecen en caso de que vaya a realizar alguna actualización dentro del sistema. Esto es muy importante sobre todo cuando se van a borrar o modificar datos pues el usuario decidirá si se lleva a cabo o no la operación. Por otro lado, se da una pequeña descripción de cada uno de los botones que aparecen en el sistema con solo seleccionarlo o poniendo el puntero del Mouse sobre el botón, con esto el usuario sabe que acción se realizará si lo presiona. Lo anterior le da seguridad al usuario y hace que le pierda el miedo a trabajar con el sistema.

El sistema está adecuado a las necesidades propias de una farmacia, es decir, realmente lo que el usuario de este ramo necesita está presentado en este sistema. La mayoría de los sistemas que ya existen son demasiado complejos, ya que son diseñados y pensados para ser manejados por grandes farmacias, además no proporcionan la información que realmente se requiere. Esto resultó después de haber analizado 2 sistemas de este tipo, creados por empresas del ramo farmacéutico y de los comentarios de dueños de los negocios que se visitaron.

La realización de sistemas con las características del SCIF, resulta de interés para ser adquiridos por pequeños empresarios, ya que en la actualidad, un sistema de este tipo es muy costoso y sólo algunas farmacias lo pueden adquirir.

Con el sistema propuesto todo ese trabajo tedioso se hará ahora de manera rápida y confiable a través de la computadora.

### *Trabajos a futuro*

Gracias a las características de Visual Basic, se pueden agregar más módulos para realizar otras operaciones si así se requieren, esto permite que el SCIF pueda seguir creciendo. Entre otros proyectos, está el de poder integrar un módulo para consultas utilizando estructuras del lenguaje SQL, pensando en usuarios más especializados. Esto sería de gran ayuda porque no se estaría limitado a las consultas que sólo permite el sistema.

Así también, el de poder trabajar con el SCIF a nivel multiusuario, lo cual soporta VB. Como en todo el sistema se utilizaron objetos para acceder a la base de datos, esta implementación no sería muy difícil de llevar a cabo.

## BIBLIOGRAFIA Y REFERENCIAS

- ACKOFF, Russel L.: "Fundamentos de investigación de operaciones", Noriega editores, México 1994.  
[30] pág. 136  
[31] págs. 138-140
- ANDERSON, Taylor William: "La computación en el comercio", McGraw-Hill, México 1992.
- BECK, Leland: "Software de sistemas: Introducción a la programación de sistemas", Addison-Wesley Iberoamericana, México 1990.
- CASTELLANOS, García J.: "Introducción a la Computación", Interamericana, México 1991.
- DATE, C. J.: "Introducción a los sistemas de bases de datos", Volumen. I, 5a. edición, Addison- Wesley Iberoamericana, E.U.A 1993  
[12] pág. 245
- EDITORIAL Multicolor: "Pasatiempo farmacéutico", Volumen 7, Número 12, Diciembre 1997.
- EDITORIAL Multicolor: "Pasatiempo farmacéutico", Volumen 8, Número 3, Marzo 1998.
- EDWARD, Jones: "Aplique dBase IV", McGraw-Hill, 1a. edición, México 1990.
- GONZÁLEZ, Alfons.: "Programación de bases de datos con Visual Basic", Computec Ra-ma, México 1997.
- HAMPTON, David R.: "Administración", McGraw-Hill, 3a. edición, México 1993.
- HEYMAN, Mark S.: "La esencia de Visual Basic 4", Prentice-Hall, México 1995.
- HURSCH, Carolyn J.: "SQL, el lenguaje de consulta estructurado SQL", Ra-ma España 1990.
- JOYANES, Aguilar Luis: "Metodología de la programación", McGraw-Hill, España.

- KORTH, Henry F.: "Fundamentos de bases de datos", 2a. edición, McGraw-Hill, España 1993.
- LAWRENCE, S. Orilia: "Computación aplicada a los negocios", McGraw-Hill, México 1990.
- LEVINE, Gutiérrez Guillermo: "Introducción a la computación y a la programación estructurada", McGraw-Hill, México 1990.  
[ 1 ] pág. 392  
[18] pág. 406
- MILLER, H. George: "La computación y sus aplicaciones", Addison-Wesley Iberoamericana, México 1993.
- OCÉANO, Grupo Editorial: "Enciclopedia audiovisual-educativa, Computación", Océano Multimedia, Volumen 1, España 1996.
- OCÉANO, Grupo Editorial: "Enciclopedia audiovisual-educativa, Computación", Océano Multimedia, Volumen 2, España 1996.
- PRESSMAN, S. Roger: "Ingeniería del software", McGraw-Hill, 3a. edición, España 1993.  
[22] págs. 26-27  
[23] págs. 29-30  
[24] págs. 30-32
- RODRÍGUEZ, A. Miguel Ángel: "Bases de datos", McGraw-Hill, España 1992.
- RAUSTENSTRAUCH, W.: "El presupuesto en el control de las empresas industriales", Fondo de Cultura Económica, 2a. edición, México 1984.  
[33] págs. 150-155
- SIMPSON, Alan: "Cómo usar dBASE III plus", incluye avance de dBASE IV, Macrobit. México 1990.
- TORRES, T. David: "Contabilidad I", Diana, México 1994.  
[32] pág. 426

## APÉNDICE A

## Código de Barras

El Código de Barras es un conjunto de rayas verticales u horizontales de distinto espesor que contiene información del producto sobre el que figuran. Se usa este código en productos de alimentación, revistas, medicamentos, etc. Y es el Lector de Código de Barras el que reconoce los datos contenidos en la combinación de rayas verticales u horizontales.

La entrada de datos a la computadora siempre ha sido un trabajo pesado y aburrido. Aunque se trate de una persona con mucha experiencia, puede resultar excesivo el tiempo empleado en introducir datos a la computadora.

Para evitar estos tiempos perdidos y el costo por ellos generado, han surgido unos periféricos llamados lectores ópticos de caracteres. Uno de estos periféricos puede analizar páginas de texto, digitalizarlas y convertirlas en código de caracteres que pueden ser manipulados por cualquier programa de la computadora. Un lector óptico de caracteres puede ser de 10 a 20 veces más rápido que una persona experimentada y además la información introducida por el lector tendrá mayor fiabilidad.

Una aplicación del lector óptico de caracteres es el que se usa sobre el Código de Barras de los artículos preempacados que se venden en las tiendas comerciales. La persona encargada de verificar mueve los artículos por una abertura, a través de la cual un haz láser de helio y neón se dirige hacia la etiqueta con el Código de Barras. Las zonas oscuras del código absorben la luz, y las zonas claras las reflejan. Un detector sensible a la luz lee el patrón del Código de Barras en el patrón de reflexión. Esta información pasa a una computadora central, la cual identifica el producto y envía el precio programado para el artículo a la caja registradora electrónica.

En el año de 1986 nació la Asociación Mexicana del Código del Producto, A.C. (Amecop), con el propósito de promover el uso de código de barras. Al principio esta tarea resultó difícil, pues los industriales argumentaban que de nada serviría hacer la inversión de imprimir en sus productos códigos de barras si las tiendas carecían de lectores ópticos para leerlos. Sin embargo se dieron cuenta de que esta inversión podría ser factible, ya que al haber una gran cantidad de productos en el mercado con código de barras las tiendas se verían en la necesidad de implantar cajas registradoras con lectores ópticos, ya que esto sería de gran ayuda para ellos. Y finalmente así sucedió. En el año de 1990 se adoptó el cobro de cuentas a partir del código de barras, y en la actualidad infinidad de productos cuentan con este sistema para su rápida identificación.

En conjunto, tanto el Código de Barras como el Lector de Código de Barras proveen un rápido y exacto medio para identificar a los productos, así como la entrada de datos a los sistemas de información computarizados.

## APÉNDICE B

## Requerimientos para trabajar con el SCIF:

Procesador:	486 o superior.
Sistema operativo:	3.1 o superior.
Memoria requerida:	8 MB en RAM.
Espacio en disco:	20 MB.

## Instalación del SCIF

Para instalar el SCIF al disco duro, sólo hay que copiar la carpeta llamada SCIF que viene en el disco de instalación con todos los archivos que están dentro de ella. Una vez copiada la carpeta al disco, ábrala y localice el archivo llamado **Scif.exe** haga doble click sobre éste e inmediatamente se empezará a cargar el sistema. Se recomienda crear un *acceso directo* al SCIF.

Al cargarse el programa aparecerá la siguiente ventana:



Pedirá introducir el Password, al teclearlo correctamente se abrirá el candado que aparece en la ventana y entrará al menú principal del sistema, en donde ya se podrá empezar a trabajar con el SCIF.

La ventana principal será la que aparece a continuación.

