

19  
2ej.



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

FACULTAD DE INGENIERÍA



DESARROLLO DE UN SISTEMA DE INFORMACION  
AUTOMATIZADO DE INVENTARIOS APLICABLE A LA EMPRESA  
COMERCIAL EMYCO, S.A. DE C.V.

## **TESIS PROFESIONAL**

PARA OBTENER EL TITULO DE  
**INGENIERO EN COMPUTACION**

PRESENTAN:

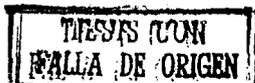
**AGUSTIN CARRILLO DIAZ**

**ALBERTO MORALES BARAJAS**

**MARTIN MARTINEZ GARCIA**

MEXICO, D. F.

1992





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

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

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

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

# INDICE TEMATICO

## Introducción

Objetivo	4
Justificación	4
Plan de trabajo	4

## Capítulo 1 Antecedentes teóricos

1.1 Bases de datos	9
1.1.1 Introducción y definiciones	9
1.1.2 Sistemas de manejo de bases de datos	10
1.1.3 Análisis de los principales DBMS	11
1.1.4 El enfoque relacional	14
1.2 Ingeniería de Software	16
1.2.1 Definición del sistema	17
1.2.2 Desarrollo	21
1.3 Redes de computadoras	23
1.3.1 Sistemas centralizados	23
1.3.2 Sistemas distribuidos	23
1.3.3 Redes de computadoras	25
1.3.4 Estructura de redes	27
1.3.5 Arquitecturas de redes	29
1.3.6 Telecomunicaciones	33

## Capítulo 2 Planteamiento del problema

2.1 Descripción general del sistema actual	37
2.1.1 Elementos del sistema actual	37
2.1.2 Entradas y salidas del sistema actual	39
2.2 Planteamiento general de soluciones	40
2.2.1 Alternativas para comunicación de datos	41
2.2.2 Alternativas para redes de área extendida	44
2.2.3 Rendimiento de los medios de comunicación	56
2.2.4 Manejadores de bases de datos actuales	57

## **Capítulo 3 Análisis de Hardware y Software**

3.1 Análisis de elementos del sistema	67
3.1.1 Alcances del sistema	67
3.1.2 Análisis del servicio de telecomunicación	67
3.1.3 Equipo central	72
3.1.4 Equipo de sucursales	74
3.1.5 Análisis del Software de desarrollo	75
3.1.6 Características del Clipper 5.01	77
3.1.7 Análisis del Software de comunicación	81

## **Capítulo 4 Desarrollo del SRDI**

4.1 Especificación de requerimientos de Software	89
4.1.1 Introducción	89
4.1.2 Descripción general	91
4.1.3 Requerimientos específicos	91
4.2 Diseño	116
4.2.1 Introducción	116
4.2.2 Consideraciones	116
4.2.3 Diagrama de estructura	117
4.2.4 Almacenamiento de datos	119
4.2.5 Descripción del módulo de actualización de existencias	122
4.2.6 Interfaz con el usuario	124
4.2.7 Recuperación de fallas	124

## **Capítulo 5 Resultados y conclusiones**

### **Apéndice A Metodologías de análisis**

A.1 Metodología de Pressman	A3
A.2 Metodología de Yourdon	A4

### **Apéndice B Protocolo X.25**

### **Apéndice C Diagramas de flujo de datos del SRDI**

### **Glosario**

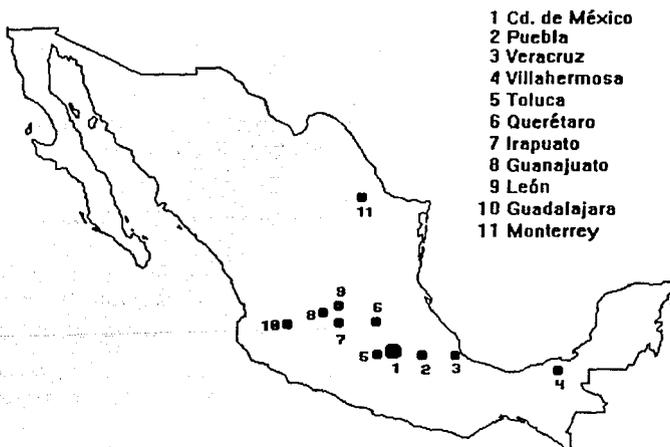
### **Bibliografía**

# INTRODUCCION

En las comunicaciones, como en la electrónica en general, los métodos y procedimientos de manejo de información han venido mejorando, motivo por el cual el hombre ha buscado nuevas formas de manejar la información de manera oportuna y precisa, de tal manera que la empresa que no cuenta con información confiable y veraz se va retrasando respecto de sus competidores, lo cual impide su crecimiento y con el tiempo propicia su posible extinción. Es por eso que las empresas de hoy en día cuentan con equipos de trabajo muy complejos, los cuales les permiten obtener información oportuna y confiable, buscando con ello ser el número uno en el mercado en que compiten.

El procesamiento de la información en las empresas es un factor determinante para el desarrollo de las mismas, tal es el caso de "**Comercial Emyco, S. A. de C.V.**", la cual es una empresa que se dedica a la venta y compra de calzado y accesorios de vestir.

Esta organización cuenta con 23 tiendas, distribuidas en toda la República Mexicana. Existe una oficina central, localizada en el Distrito Federal, que controla la adquisición y distribución de productos, registros de ventas y la generación de información estadística, tablas comparativas y estimaciones de venta.



*Distribución de las Sucursales de Emyco.*

"Comercial Emyco, S.A. de C.V." es una empresa creada en 1926, por lo cual su larga trayectoria en el mercado zapatero le ha permitido ser una de las mejores dentro de su ramo, además de ser una empresa 100% mexicana.

## OBJETIVO

Con base en lo expuesto anteriormente, el objetivo de la presente tesis es el desarrollar un sistema de inventarios, y la selección de un sistema de comunicación en red que optimice la transferencia de datos entre las diferentes sucursales de la empresa "Comercial Emyco, S.A. de C.V."

## JUSTIFICACION

Actualmente, la transferencia de información entre las sucursales y la oficina central es vía terrestre, cada tres días en forma de paquetería, lo cual provoca retraso en el procesamiento de información y la toma de decisiones sobre las políticas de la empresa.

Lo expuesto anteriormente motiva al uso de un medio de comunicación más efectivo entre las sucursales y la oficina central.

"Comercial Emyco, S.A. de C.V.", cuenta con un *Sistema de Inventarios de Control Automático (SIDCA)*, desarrollado en 1985 en lenguaje *Cobol* y que corre en un equipo *TOWER XP 1632* con cuatro terminales, bajo ambiente *UNIX*.

Dadas las necesidades de la empresa de obtener información oportuna y confiable, este sistema resulta ya obsoleto, tanto en aspectos técnicos como funcionales, por lo que se propone el desarrollo de un nuevo sistema que se ajuste a las necesidades actuales.

## PLAN DE TRABAJO

Para lograr un desarrollo estructurado del trabajo de tesis, se aplican conocimientos de bases de datos, ingeniería de *Software* y redes de computadoras.

Se utilizan además, algunas herramientas de desarrollo y apoyo, como son: manejadores de bases de datos, procesadores de palabra, programas para definición de gráficos, etc.

Así, en el capítulo 1 se da una breve introducción a los conceptos teóricos que se utilizarán en las partes de la selección del sistema de comunicación y el desarrollo del *Software*. De esta manera, partiendo de la teoría, se hace un planteamiento general del problema y sus posibles soluciones.

El capítulo 2 comprende el planteamiento de necesidades y posibles soluciones, así como la viabilidad para el desarrollo del sistema.

En el capítulo 3 se realizan las fases de análisis económico y técnico del sistema. Se definen también las funciones de cada parte del sistema.

En el capítulo 4 se desarrolla el Sistema de Inventarios. En la primera parte de este capítulo se definen las características del sistema, haciendo un análisis del mismo, para luego planificar los requerimientos necesarios. La segunda parte comprende el desarrollo del *Software* con base en el análisis teórico realizado, incluyendo las fases de diseño del *Software*, codificación y pruebas de funcionamiento. En la tercera parte se hacen consideraciones acerca del mantenimiento del sistema, incluyendo corrección, adaptación y modificaciones.

En los capítulos 2 y 3 se aplica la metodología de análisis de sistemas de Pressman (apéndice A).

Se concluye la tesis especificando los alcances y limitaciones del sistema.

Se anexan al trabajo los apéndices que contienen documentación técnica del *Software* desarrollado o utilizado y un glosario de términos.

# **CAPITULO 1**

## **ANTECEDENTES TEORICOS**

# 1.1 BASES DE DATOS

## 1.1.1 INTRODUCCION Y DEFINICIONES

El procesamiento de grandes volúmenes de información es un problema actual al que se enfrenta todo el mundo, como son: gobierno, escuelas, librerías, etc. Nuestra sociedad requiere contar con información veraz y precisa que le permita tomar decisiones correctas.

La invención de la computadora marcó una revolución en los métodos tradicionales del procesamiento de la información. Fue a principios de los años sesentas cuando algunas empresas prestigiadas decidieron computarizar sus sistemas de información. Los primeros sistemas de administración de datos por computadora se desarrollaron como aplicaciones independientes enfocadas a obtener resultados específicos, situación que provocó la duplicación de datos. A estos sistemas se les conoce como *Sistemas de Manejo de Archivos (FMS)*.

A finales de los años sesentas surgen los *Sistemas de Bases de Datos (DBS)* que permiten superar los problemas asociados con los sistemas de información tradicional. Archivos individuales se integran en una sola base de datos para ser compartidos por todos los usuarios, evitándose la duplicación de datos.

Los *DBS* involucran algunos conceptos básicos como son los siguientes:

- Campo:** Es la unidad mínima de almacenamiento que contiene una base de datos (columna).
- Registro:** Es el conjunto de campos relacionados entre sí. Cada registro define una entidad de la base de datos (renglón).
- Archivo:** Es una colección de registros del mismo tipo. En un archivo, cada renglón equivale a un registro y cada columna a un campo.

Existen dos tipos de registros:

- Registro Lógico:** Es la manera como el usuario maneja la información almacenada en la base de datos.
- Registro Físico:** Es la unidad de transferencia de información entre el dispositivo de almacenamiento secundario y la memoria primaria. Un registro físico puede equivaler a varios registros lógicos.

## 1.1.2 SISTEMAS DE MANEJO DE BASES DE DATOS

Un *DBS* es un sistema computarizado de información cuyo manejo se efectúa por medio de paquetes de *Software*, llamados *Sistemas de Manejo de Bases de Datos (DBMS)*. Los tres componentes principales de un *DBS* son el *Hardware*, el *Software (DBMS)* y los datos por manejar.

### Base de datos

Una base de datos es una colección de archivos interrelacionados creados con un *DBMS*. El contenido de una base de datos se obtiene combinando datos de todas las diferentes fuentes en una organización, de tal manera que los datos estén disponibles para todos los usuarios, y los datos redundantes puedan eliminarse, o al menos minimizarse.

### Sistemas de manejo de bases de datos

El *DBMS* interpreta y procesa las peticiones del usuario para recobrar información de la base. En la mayoría de los casos las peticiones de consulta tendrán que atravesar varios procesos antes de que se pueda acceder físicamente la base de datos. El *DBMS* le permite al usuario el acceso y modificación de datos en cualquier parte de la base sin necesidad de conocer la organización física dentro del dispositivo de almacenamiento.

Se puede decir que el *DBMS* es un intermediario entre el usuario y la base de datos.

### Hardware para sistemas de bases de datos

El componente de *Hardware* de un *DBS* consiste en el *Dispositivo de Almacenamiento de Acceso Directo (DASD)*, la computadora central, el procesador de entrada/salida y la unidad de control.

Los dos elementos principales para el almacenamiento de datos de una computadora son la memoria primaria o principal (circuitos integrados) y los dispositivos de almacenamiento secundaria tales como cintas magnéticas, discos duros, discos flexibles, tambores, etc. La *Unidad Central de Procesamiento (CPU)* permite entre otras actividades el acceso a la memoria primaria.

La velocidad de acceso a la memoria de un dispositivo de almacenamiento secundario es mucho más lenta que a la memoria primaria, debido a que se involucran procesos electromecánicos; en cambio, la memoria primaria sólo realiza procesos electrónicos.

Los dispositivos de almacenamiento secundario se caracterizan por ser baratos y permitir almacenar datos de manera permanente.

### Administradores de bases de datos

En la mayoría de las ocasiones los usuarios finales de las empresas no tienen control directo sobre la manera en que se procesa la información de la base de datos. Para ello se asigna personal o equipo al que se le llama *Administrador de la Base de Datos (DBA)*. El personal que conforma el *DBA* generalmente tiene experiencia en el uso de *DBMS*, diseño de bases de datos, sistemas operativos y programación.

En las dos últimas décadas, el *Software* para el manejo de datos evolucionó gradualmente desde el proceso secuencial hasta los sistemas actuales. Es mucho más sencillo desarrollar un sistema de manejo de datos mediante lenguajes de alto nivel que a través del lenguaje ensamblador. A principios de los años sesentas el almacenamiento de datos se limitaba a dispositivos secuenciales. Para mediados de los sesentas aparecen los dispositivos de almacenamiento directo, como son los discos magnéticos, permitiendo crear estructuras de almacenamiento más sofisticadas. En la actualidad, los fabricantes de computadoras incluyen como parte del sistema operativo, métodos básicos de acceso secuencial y aleatorio.

En la figura 1.1 se muestra la evolución del *Software* y el *Hardware* en las últimas décadas:

	Principios de los sesentas	Mediados de los sesentas	Años setentas	Años ochentas
<i>Hardware</i> : elementos de almacenamiento secundario	cinta magnética cinta de papel	Disco magnético Tambor magnético	Disco magnético Tambor magnético	Máquina base de datos PC. CD-ROM.
<i>Software</i>	COBOL FORTRAN	Método básico de acceso del Sistema Operativo	FMS DBMS	Mainframe/interface PC, bases de datos distribuidas
Tipo de procesamiento	Secuencial	Aleatorio	Base de datos	Distribuidas

Figura 1.1 Evolución del *Hardware* y el *Software*.

### 1.1.3 ANALISIS DE LOS PRINCIPALES DBMS

Un *DBMS* es un paquete sofisticado de programación, el cual se compone de rutinas relacionadas encaminadas a realizar aplicaciones especializadas sobre la base de datos. Por lo regular el *DBMS* opera bajo las características del sistema operativo, es por ello que algunas aplicaciones se encuentran diseñadas para un solo sistema operativo. Por ejemplo, el sistema *IMS* corre únicamente en

equipos *IBM*, pero por otro lado la aplicación llamada *TOTAL* está diseñada para operar en varios sistemas operativos.

La gran mayoría de *DBMS* existentes en el mercado pueden ser clasificados dentro de una de las tres siguientes categorías: red, jerárquico y relacional. Esta clasificación se basa en la estructura lógica utilizada en el diseño del *DBMS*. La razón principal por la cual los datos son organizados con base en alguna estructura lógica, como puede ser por tablas, árboles, listas, redes etc., es debido a que el acceso a los datos puede ser de forma aleatoria. Por otro lado, la estructura física de los archivos es totalmente transparente para el usuario.

Algunos *DBMS* basan su funcionamiento en modelos híbridos. Por ejemplo, un *DBMS* jerárquico o de red puede ser equipado con recursos para indexar y proporcionar posibilidades semirrelacionales.

Para que el *DBMS* pueda comunicarse con el usuario existen diferentes lenguajes de bases de datos. A continuación se describen algunos de ellos:

- 1) Lenguaje de manejo de datos: El *DBMS* proporciona un lenguaje de manejo de datos para que emita proposiciones de entrada/salida para poder acceder la base de datos. Este lenguaje puede ser una extensión del lenguaje de la computadora central. Las proposiciones pueden ser tecleadas de manera interactiva desde una terminal o pueden ser emitidas desde un programa en lenguaje de alto nivel.
- 2) Lenguaje de consulta: Este lenguaje permite a los usuarios recuperar información de la base sin necesidad de escribir programas. Por lo regular se basa en algoritmos con los cuales el usuario puede teclear proposiciones de consulta conversacionales, son prácticos para la realización de transacciones aisladas.
- 3) Lenguaje de programación: La mayoría de los administradores de bases de datos de macrocomputadoras cuentan con interfaces con lenguajes de alto nivel; sin embargo, la mayoría de los *DBMS* de microcomputadoras no cuentan con dicha interfaz.

### **El enfoque Jerárquico**

El *DBMS* de enfoque jerárquico utiliza tres estructuras de árbol para la representación de arreglos lógicos. En el enfoque jerárquico se parte de una sola raíz, cada nodo excepto la raíz se encuentra conectado con un solo nodo en el nivel superior. Las características de una estructura de base de datos jerárquica son las siguientes:

- 1) Las entidades de un archivo jerárquico están conformadas por estructuras de árbol.
- 2) Los diferentes tipos de registro conceptual están enlazados por medio de relaciones uno-a-muchos.
- 3) Cada nodo padre puede tener distinto número de hijos.
- 4) Un registro hijo no puede existir si no existe un registro padre.
- 5) Para eliminar un registro padre será necesario borrar todos los registros hijo que le pertenezcan.

### **El enfoque de Red**

La mayoría de los sistemas de manejo de redes de bases de datos se apegan a las recomendaciones *CODASYL*. Una estructura de datos tipo red abarca más que las estructuras de árbol porque un nodo hijo puede tener más de un padre. Se puede decir que la estructura de árbol se considera un caso especial de estructura de red. Las características de una estructura de base de datos de red son las siguientes:

- 1) El registro padre de un conjunto es llamado el propietario, mientras que el registro hijo es llamado miembro. Un conjunto se forma por un registro propietario y uno o varios registros miembro.
- 2) Un registro miembro puede asociarse con más de un propietario.
- 3) Un registro puede ser miembro en un tipo de conjunto y al mismo tiempo propietario en otro conjunto.

### **El enfoque Relacional**

En el *DBMS* relacional una estructura lógica se representa por medio de tablas bidimensionales llamadas relaciones. Una entidad se representa por un renglón de la tabla.

La ventaja principal del enfoque relacional está en la simplicidad de su representación en la estructura lógica de la base de datos y la flexibilidad para establecer relaciones de datos por medio de campos de conexión. Todas las entidades de una base relacional se representan como tablas separadas sin encontrarse en alguna jerarquía fija.

El enfoque relacional incrementa la independencia de los datos usando campos de conexión. En lugar de usar señaladores para establecer relaciones uno-a-muchos, entre un registro padre de un archivo y un registro hijo de otro, el

sistema relacional establece la asociación insertando el valor clave-principal de registro padre en un archivo en el campo de conexión correspondiente de todos sus registros hijos.

Una característica única del sistema relacional es su independencia de trayectorias de entrada/salida. Ya que una base de datos relacional consta de una colección de tablas separadas, cualquier tabla puede ser accesada sin necesidad de acceder otras relaciones.

La terminología relacional es la siguiente:

- 1) Relación: A una tabla también se le llama relación.
- 2) Atributo: Cada columna de una relación representa un atributo o campo. Una entidad o registro está definida por varios atributos.
- 3) Dominio: Es el conjunto de todos los valores posibles de entre los cuales el atributo puede tomar su valor.
- 4) Ada: Nombre que también se le da a los renglones.
- 5) Archivo neto: Un archivo se considera como neto si es que ninguno de sus campos contiene ocurrencias múltiples de valores de datos.

### 1.1.4 EL ENFOQUE RELACIONAL

El sistema relacional es el rival más importante del sistema *CODASYL* para el manejo de bases de datos, aun cuando estos sistemas se lanzaron al mercado hasta principios de los ochentas, más de una década después de la aparición de los sistemas de red y jerárquico.

Una de las características del enfoque relacional es la simplicidad de su representación lógica; para los usuarios es más fácil entender las tablas que los complejos árboles o estructuras de red.

En un modelo relacional una tabla es una entidad separada que no ocupa un nivel jerárquico fijo, o sea que en el modelo de datos no se describen relaciones padre-hijo o propietario-miembro. Consecuentemente se puede acceder directamente cualquier registro en un archivo relacional sin necesidad de navegar por una red de varios niveles desde la raíz.

**E.F. Codd** planteó el enfoque relacional a fines de los sesentas, razón que promovió la implantación de tres importantes prototipos experimentales: *System R*, *Ingres* y *Query-By-Example (QBE)*.

Aunque desde el punto de vista del usuario el enfoque relacional parece más sencillo, su implantación puede resultar complicada. Las características relacionales se alcanzan vía un *Software* sofisticado que requiere de tiempo extra del *CPU* para su ejecución.

Como consecuencia, uno de los retos más importantes era alcanzar un mejor nivel de desempeño, en especial para las grandes bases de datos. Finalmente, los sistemas relacionales de manejo de bases de datos se comercializaron hasta principios de los ochentas, después de una década de investigación y desarrollo. A continuación se muestra una tabla con algunos de los sistemas relacionales actuales y los sistemas operativos que los soportan:

### **DBMS RELACIONALES**

*Query-By-Example* de IBM

*SQLIDS* de IBM

*DB2* de IBM

*Ingres* de *Relational Technology*

*Oracle* de *Relational Software*

*dBASE IV* de *Borland*

### **SISTEMAS OPERATIVOS**

*VM* de IBM

*DOS/VSE, VM/CMS* de IBM

*MVS* de IBM

*VMS, UNIX* de DEC

*RSX, VMS, UNIX* de DEC

*MS-DOS (PC's)*

### **Arquitectura de un Sistema Relacional**

La arquitectura relacional consta de los siguientes componentes:

- 1) **Modelo relacional de datos:** En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones almacenadas. Cada registro de tipo conceptual en un modelo relacional de datos, se implanta como un archivo almacenado distinto.
- 2) **Submodelo de datos:** Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de uno o más escenarios para describir los datos requeridos para una aplicación dada. Un escenario puede incluir datos de una o más tablas de datos. Cada programa de aplicación está provisto de un *Buffer* donde el *DBMS* deposita los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de que el *DBMS* las escriba en la bases de datos.
- 3) **Esquema de almacenamiento:** En el nivel interno, cada tabla base se implanta con un archivo almacenado. Para las recuperaciones sobre las claves primaria o secundaria se pueden establecer uno o más índices para acceder un archivo almacenado.

- 4) **Sublenguaje de datos:** El sublenguaje de datos es un lenguaje de manejo de datos para el sistema relacional. Originalmente **Codd** propuso dos sublenguajes de datos, que son el álgebra relacional y el cálculo relacional.

## **1.2 INGENIERIA DE SOFTWARE**

### **Sistemas**

En su sentido más amplio, un sistema es un conjunto de componentes que interactúan para alcanzar algún objetivo. Los sistemas son de hecho todo lo que rodea al ser humano. Por ejemplo: se sienten sensaciones físicas originadas por un complejo sistema nervioso, un conjunto de partes que incluyen al cerebro, espina dorsal, nervios y células sensitivas especiales debajo de la piel que trabajan conjuntamente para hacer sentir dolor, calor, comezón, etc.

El ser humano se comunica a través del lenguaje que es un sistema altamente desarrollado de palabras y símbolos que tienen significado. Vive también de acuerdo a un sistema económico en el cual los bienes y los servicios se cambian por otros de valor comparable y por medio de los cuales, por lo menos en teoría, los participantes de estos intercambios se benefician.

Cuando se comienza a ver lo abundante que son los sistemas, no sorprende darse cuenta que cada sistema del negocio depende de una o más entidades abstractas, llamadas sistemas de información. Por medio de estos, los datos pasan de una persona a otra o de un departamento a otro y puede realizarse cualquier cosa, desde comunicaciones entre oficinas y comunicaciones telefónicas hasta un sistema de computadoras que generen informaciones periódicas a diferentes usuarios. Los sistemas de información sirven de hecho a todos los sistemas de un negocio. Ellos son el lazo que mantienen unidos a diferentes componentes en forma total, que pueden trabajar de manera efectiva hacia el mismo objetivo.

### **Ciclo de vida del desarrollo de sistemas**

El desarrollo de sistemas es un proceso que consiste en dos etapas principales: análisis y diseño de sistemas. Comienza cuando la gerencia o en algunas ocasiones el personal de desarrollo de sistemas se da cuenta de que cierto sistema del negocio necesita mejorarse.

El ciclo de vida del desarrollo de sistemas es el conjunto de actividades que los analistas, desarrolladores y usuarios necesitan llevar a cabo para desarrollar y poner en marcha un sistema de información. Se debe tener presente que en la

mayoría de las situaciones del negocio las actividades están íntimamente relacionadas y son inseparables.

En muchas ocasiones el orden de las etapas será difícil de determinar, las diferentes partes de un proyecto pueden encontrarse al mismo tiempo en diversas fases. Algunos componentes pueden encontrarse en la etapa del análisis mientras otros se hayan en etapas avanzadas del diseño.

#### **Etapas del ciclo de vida del desarrollo de sistemas:**

- 1) Investigación preliminar
- 2) Determinación de requerimientos
- 3) Desarrollo del sistema prototipo
- 4) Diseño del sistema
- 5) Desarrollo de *Software*
- 6) Prueba de los sistemas
- 7) Puesta en marcha
- 8) Mantenimiento

### **1.2.1 DEFINICION DEL SISTEMA**

Para que un sistema de información por computadora sea útil, debe funcionar adecuadamente, ser fácil de manejar y adecuarse a la empresa para la cual se ha diseñado. Si le ayuda al personal a hacer su trabajo en una forma más eficiente se le utilizará, de lo contrario con seguridad lo evitarán.

Para desarrollar y diseñar sistemas de información se requieren diversas habilidades:

- 1) La capacidad para examinar una petición para recibir apoyo de la computadora y determinar si debe ser considerada como solución para el problema.
- 2) El conocimiento para recopilar e interpretar los datos que ayudan a diagnosticar algún problema de la empresa, su relación con los sistemas de información y de cómputo.

- 3) El entendimiento para determinar después de examinar una situación en la empresa, donde sí es aconsejable utilizar la computadora y donde son más efectivos los procedimientos y sistemas manuales.
- 4) La capacidad para diseñar y desarrollar especificaciones de un sistema de información conforme se determinaron a través de un examen del sistema actual.
- 5) El conocimiento para seleccionar los mejores métodos de entradas de datos, almacenamiento de archivos, acceso, proceso de salida para una situación dada.
- 6) Tener un conocimiento interno del desarrollo del *Software*.
- 7) La capacidad para comunicarse efectivamente con otras personas.
- 8) Experiencia en cada uno de los puntos anteriores a través de la aplicación de problemas de desarrollo y casos de estudio.

### **Análisis del sistema**

En términos generales, en las empresas, el análisis y diseño de sistemas se refiere al proceso de examinar una situación de la empresa con la intención de mejorarla mediante nuevos procedimientos y métodos.

El diseño de sistemas es el proceso de planeación de un nuevo sistema dentro de la empresa para reemplazar o completar el existente, pero antes de que esto pueda llevarse a cabo primero se debe entender por completo el sistema anterior y determinar cómo se puede utilizar la computadora en forma óptima, si es posible, para hacer esta operación en forma más efectiva.

Por lo tanto el análisis de sistemas es el proceso que sirve para recopilar e interpretar los hechos, diagnosticar problemas y utilizar estos hechos a fin de mejorar el sistema, este es el trabajo del analista de sistemas.

En otras palabras se debe comprender cómo trabaja el sistema actual y más específicamente cuál es el flujo de información por el que atraviesa el sistema. También es importante saber por qué desea cambiar sus operaciones actuales, sólo después de recabar estos datos se puede empezar a definir cómo se puede beneficiar con un sistema de información basado en computadora y que sea útil a todos los usuarios del sistema.

El objetivo de un sistema es la razón de su existencia. Para cumplir sus propósitos, los sistemas interactúan con sus medios ambientes. Los sistemas que interactúan con su medio ambiente recibiendo entradas y produciendo

salidas se les llama sistemas abiertos. En contraste, los sistemas que no interactúan con sus alrededores se conocen como sistemas cerrados. Todos los sistemas en marcha son sistemas abiertos, por lo tanto los sistemas cerrados solo existen en forma conceptual.

Aunque en la actualidad la computadora es un objeto común en los negocios, gobierno, medicina, industria y aún en la política, la gente prefiere mantenerse al margen y acudir a quienes entienden el poder de la computadora.

Los analistas de sistemas ejercen una influencia considerable en la organización para la que trabajan, con base a sus recomendaciones se instalan nuevos sistemas y se descartan los viejos.

Mediante la información generada como resultado de los sistemas que desarrolló el analista, los gerentes pueden decidir el curso de la acción para un producto nuevo o ya existente, también es responsable de los informes empleados en la asignación de una estrategia que determine la imagen de toda la compañía.

No todos los sistemas requieren procesamiento de datos, del mismo modo no todos los requerimientos de información del usuario acerca del trabajo del sistema se convertirán en proyectos de desarrollo de sistemas.

En los negocios las computadoras y los sistemas de información ocupan un lugar muy especial, ya que hacen posible la funcionalidad de las oficinas de reservaciones aéreas, de departamentos de registros hospitalarios, funciones de nóminas, contabilidad, operaciones bancarias electrónicas, sistemas de intercomunicación telefónica, entre otras muchas e incontables aplicaciones tanto de empresas grandes como pequeñas.

La tecnología ha avanzado a una velocidad impresionante, pero el aspecto más importante en cualquier sistema es el conocimiento humano y el uso de las ideas para aprovechar la computadora de una manera que realice las tareas requeridas. Este proceso es esencialmente en lo que consiste el desarrollo de sistemas.

Los analistas de sistemas no sólo resuelven problemas actuales, frecuentemente se les llama para manejar la expansión planeada de una empresa, en este caso el estudio de sistemas se orienta hacia el futuro, dado que no existe ningún sistema actual, el analista considera tan cuidadosamente como sea posible cuales serán las necesidades de la empresa y en qué área deberá considerar los cambios para que coincidan con estas necesidades. En este caso y en la gran mayoría de las circunstancias los analistas pueden recomendar formas alternas para mejorar la situación.

Al trabajar con los gerentes y empleados de la empresa, el analista de sistemas recomienda qué opción debe adoptarse para una solución, la elección debe basarse en aspectos como la adaptabilidad de la solución a la estructura de la empresa, así como el apoyo que deberá tener por parte de los empleados. Si los empleados que usarán el sistema no se sienten a gusto con éste, fallará en su propósito por mejorar la compañía.

Algunas veces el tiempo que lleva el desarrollar una opción comparada con otras será el aspecto más difícil. Los costos y beneficios financieros también son importantes de determinar, la gerencia es la que decidirá qué opción va a aceptar. El analista de sistemas puede recomendar, pero la gerencia que es la que va a pagar y utilizar los resultados es la que realmente decide.

Una vez que se toma la decisión, se desarrolla un plan para poner en marcha la recomendación. El plan incluye todas las características del diseño de sistemas, como son necesidades nuevas de captación de datos, especificaciones de archivos, procedimientos de operación y necesidades de equipo y personal. De hecho los diseños establecerán el trabajo que desempeñará el personal y el que realizarán las computadoras.

"El análisis del sistema especifica qué es lo que el sistema debe hacer y cómo alcanzar el objetivo".

Seguramente se ha advertido que en cada uno de los pasos anteriores se involucra al personal. Los gerentes y los empleados saben lo que funciona y lo que no es útil para la empresa, así mismo conocen qué es lo que ocasiona problemas y los lugares donde se necesita o no el cambio y específicamente los lugares donde afectará la innovación y donde no afectará. Aún cuando la tecnología abunda en muchas empresas y negocios, el personal es todavía el que logra que esta organización funcione, por lo tanto la comunicación y trato con los empleados son parte muy importante en el trabajo del analista de sistemas.

Se sabe que el analista de sistemas es quien estudia los sistemas de la empresa para conocer los métodos actuales y buscar la efectividad.

Cuantas veces se ha estado en situaciones donde se pregunta si existe alguna mejor manera de hacer algo.

Se puede iniciar una petición de desarrollo de sistemas por otras muchas razones, pero la clave es que alguien, ya sea un gerente o un empleado o un especialista de sistemas, inicie un requerimiento para recibir ayuda de un sistema de información. Cuando ese requerimiento se realiza, la primera actividad de sistemas (la investigación preliminar) se inicia.

El análisis de sistemas involucra:

- 1) Clarificación del requerimiento
- 2) Estudio de factibilidad
- 3) Aprobación del requerimiento

El resultado será aprobar el requerimiento para su atención posterior o rechazarlo como no factible para un desarrollo futuro.

Un punto clave del análisis de sistemas se consigue al adquirir un conocimiento detallado de todas las facetas importantes dentro del área de negocios que se investiga, por esta razón, a menudo esta actividad se conoce como investigación detallada.

## 1.2.2 DESARROLLO

### Diseño del Software

El diseño de un sistema de información produce los elementos que establecen cómo el sistema cumplirá los requerimientos identificados durante el análisis de sistemas. A menudo los especialistas de sistemas se refieren a esta etapa como el diseño lógico, en contraste con el desarrollo del *Software* de programas que se conoce como diseño físico.

Los analistas de sistemas empiezan por identificar los informes y otras salidas que el sistema producirá, a continuación los datos específicos de cada uno de estos se señalan incluyendo su localización exacta sobre el papel, la pantalla u otras formas o medios de despliegue.

Usualmente los diseñadores dibujan la forma o la visualización como la esperan cuando el sistema esté terminado. El diseño del sistema también describe los datos calculados o almacenados que se introducirán, los grupos de datos individuales, los procedimientos de cálculo, la selección de las estructuras de los archivos y los dispositivos de almacenamiento, etc. Los procedimientos que ellos describen muestran cómo se van a procesar los datos y a producir la salida. Los documentos que contienen las especificaciones de diseño utilizan muchas formas para representar los diseños, diagramas, tablas y símbolos especiales.

### Codificación

De la información del diseño detallado se pasa al grupo de programación para que pueda comenzar el desarrollo del *Software*, los diseñadores son

responsables de proporcionar a los programadores las especificaciones completas y escritas con claridad que establezcan lo que debe hacer el *Software*. Conforme comienza la programación, los diseñadores están pendientes para contestar preguntas, esclarecer ideas confusas y manejar los problemas que confronten los programadores cuando utilicen las especificaciones de diseño.

Los desarrolladores del *Software* pueden por ejemplo instalar o modificar *Software* comercial que se haya comprado o pueden escribir programas nuevos diseñados a la medida. La decisión de qué se va a hacer, depende del costo de cada una de las opciones, el tiempo disponible para escribir el *Software* y la disponibilidad de programadores. Los programadores son los responsables de documentar los programas, e incluir los comentarios que expliquen cómo se utilizó cierto procedimiento y cómo se codificó en cierta forma.

La documentación es esencial para probar el programa y darle mantenimiento una vez que la aplicación se haya puesto en marcha.

### **Pruebas de Software**

Durante la prueba, los sistemas se utilizan en forma experimental para asegurar que el *Software* no falle, es decir, que correrá de acuerdo a las especificaciones y a la manera con la que los usuarios esperan que lo haga.

Se examinan datos especiales de prueba en la entrada del procesamiento y los resultados para localizar algunos problemas inesperados. Cuando el personal de sistemas verifica y pone en uso el nuevo equipo, entrena al personal usuario, instala la nueva aplicación y construye los archivos de datos que se necesitan, entonces se dice que el sistema está puesto en marcha.

De acuerdo con el tamaño de la empresa que empleará la aplicación y el riesgo asociado con su uso, los desarrolladores del sistema pueden escoger una prueba piloto para la operación del sistema, sólo en una área de la compañía. A veces correrán paralelamente, tanto el nuevo sistema como el viejo, para comparar resultados de ambos. En otras ocasiones los desarrolladores de sistemas pararán el sistema anterior por completo un día y al siguiente empezarán a utilizar el nuevo.

### **Mantenimiento**

Una vez instalado el sistema, se utilizará la aplicación durante muchos años, sin embargo, tanto la empresa como los usuarios cambiarán y el medio ambiente será diferente conforme transcurre el tiempo, por lo tanto, la aplicación indudablemente necesitará mantenimiento, es decir, se harán cambios y modificaciones al *Software* y a los archivos y procedimientos para cumplir con

los requerimientos nuevos de los usuarios. Los sistemas de la empresa y el medio ambiente de los negocios están en continuo cambio, los sistemas de información deben de mantenerse de la misma forma en este sentido.

## **1.3 REDES DE COMPUTADORAS**

### **1.3.1 SISTEMAS CENTRALIZADOS**

Tradicionalmente, y hasta principios de los setentas, la función de procesamiento de datos estaba organizada en una forma centralizada. En un *Sistema Centralizado de Datos* el soporte de procesamiento de datos es provisto por una o un grupo de computadoras generalmente grandes, localizadas en una instalación central de procesamiento de datos. Muchas de las tareas realizadas por tal instalación son iniciadas en el centro y el resultado es producido en el mismo centro.

Un *Sistema Centralizado de Datos* es centralizado en muchos sentidos:

- 1) Computadoras centralizadas. Una o más computadoras están localizadas en una instalación central.
- 2) Datos centralizados. Todos los datos son almacenados en archivos y bases de datos en la instalación central.
- 3) Control centralizado. Un administrador de procesamiento de datos o sistemas de información tiene la responsabilidad del procesamiento de datos centralizado.
- 4) Personal de soporte centralizado. Una instalación de procesamiento de datos centralizado debe incluir un personal de soporte técnico que opere y mantenga el equipo de procesamiento de datos.

Un ejemplo de procesamiento de datos centralizado es un sistema de reservaciones en una línea aérea. Típicamente tales sistemas están organizados de tal forma que los datos y el procesamiento de datos son centralizados. La instalación central mantiene un registro de vuelos y asientos disponibles y puede actualizar la base de datos para hacer y cancelar reservaciones. Los pagos pueden también ser realizados por el sistema. El agente de la línea aérea posee una terminal sin capacidad de procesamiento local.

### **1.3.2 SISTEMAS DISTRIBUIDOS**

El término de *Sistema Distribuido* se ha utilizado para denominar a diferentes clases de sistemas informáticos, en los que el procesamiento de la información se encuentra repartido en el espacio y ha aparecido como una alternativa a los

sistemas en los que la potencia de tratamiento de la información se encuentra concentrada en un único elemento.

Se puede hacer una clasificación de los distintos tipos de soluciones que pueden encontrarse bajo la denominación de sistema distribuido, a partir de las distancias físicas entre los elementos de tratamiento de la información:

- 1) Redes de computadoras
- 2) Redes locales de computadoras
- 3) Sistemas multicomputadores
- 4) Sistemas multiprocesadores

### **Redes de computadoras**

Esta denominación comprende a máquinas interconectadas a grandes distancias (más de un kilómetro).

Las *Redes de Computadoras* surgieron como una solución a la necesidad de conectar computadoras situadas en lugares remotos con el objetivo fundamental de compartir recursos, es decir, permitir a cualquier usuario de cualquier computadora acceder y utilizar recursos de *Hardware* o *Software*, del conjunto de las máquinas que constituyen la red. Se les denomina también *Redes de Area Extendida (WAN)*.

### **Redes locales de computadoras**

Al conjunto de computadoras interconectadas a distancias entre 10 metros y un kilómetro se le denomina *Red de Area Local (LAN)*. La experiencia alcanzada en el campo de las *WAN* tuvo influencia decisiva en el desarrollo de las redes locales y, los primeros trabajos que se realizaron fueron simplificaciones y optimizaciones de aquellas, aprovechando las ventajas aportadas por la disminución de distancias entre las máquinas.

### **Sistemas multicomputadores**

Desde la década de los sesentas, los fabricantes de equipo de cómputo empezaron a desarrollar y aplicar la idea de descentralización de las funciones de la computadora. Así aparecieron unidades especializadas localizadas a pequeñas distancias del sistema central para el control de periféricos y comunicaciones.

## **Sistemas multiprocesadores**

Estos sistemas tienen como propósito el aumentar la potencia de procesamiento de la información mediante la utilización de elementos especializados funcionando en paralelo y comprendidos en un circuito integrado, una tarjeta de circuito impreso o un equipo electrónico.

### **1.3.3 REDES DE COMPUTADORAS**

La fusión de las computadoras y las comunicaciones ha tenido una profunda influencia en la forma en que los sistemas de cómputo están organizados. El viejo modelo de tener una sola computadora para satisfacer todas las necesidades de cálculo de una organización está siendo reemplazado con gran rapidez por otro que considera un número grande de computadoras separadas, pero interconectadas, que efectúan el mismo trabajo. A este modelo se le denomina *Red de Computadoras* y puede definirse como "la colección interconectada de computadoras autónomas para el intercambio de información", entendiéndose por computadora autónoma aquella con capacidad de procesamiento propio y que no es controlada por otra.

Con el reciente progreso en la tecnología de redes y comunicaciones, ha sido posible ligar computadoras en un número de maneras útiles. Un ejemplo es la red *ARPA* instalada en 1968 por la Agencia de Desarrollo de Proyectos Avanzados del Departamento de Defensa de los Estados Unidos de Norteamérica, la cual para 1979 conectó sobre 100 computadoras en 60 lugares de Norteamérica y Europa.

La tecnología de red es importante por varias razones:

- 1) Las redes de computadoras soportan la comunicación entre los usuarios de computadoras individuales. Ellas hacen posible el movimiento rápido y confiable de datos de un punto a otro y el soporte de aplicaciones de comunicación tales como el correo electrónico.
- 2) Las redes permiten compartir recursos costosos entre una comunidad de usuarios. Por ejemplo, una impresora puede ser compartida por varios usuarios de procesadores de palabras.
- 3) Las redes soportan la distribución de aplicaciones muy grandes para ser realizadas por una sola computadora. Tales aplicaciones son divididas y cada parte es asignada a otra computadora de la red.

Puede usarse una amplia variedad de redes de computadoras para el soporte de sistemas distribuidos, pero deben considerarse cinco criterios principales para la selección de la tecnología adecuada:

- 1) Ancho de banda alto: La velocidad de transmisión de datos entre las computadoras conectadas debe ser alta, es necesario para permitir la distribución de sistemas de almacenamiento. Debe ser capaz de intercambiar grandes cantidades de datos muy rápidamente con las otras partes del sistema.
- 2) Baja Latencia: Debería tomar muy poco tiempo para que un solo *bit* viaje de una computadora a otra. La baja latencia resulta en un rápido intercambio de mensajes cortos y por ello conduce a una comunicación eficiente.
- 3) Rango amplio: Los sistemas de computadoras que se extienden en continentes son ya bastante comunes en la actualidad y requieren tecnologías de red que puedan transmitir datos a miles de kilómetros.
- 4) Alta confiabilidad: Idealmente sólo una fracción del mensaje debería experimentar errores de transmisión; el ancho de banda y la latencia son desafortunadamente afectados por la necesidad constante de retransmitir mensajes.
- 5) Bajo costo: La economía de la computación distribuida depende de la disponibilidad de tecnología de red barata.

La principal razón por la cual existen tantas tecnologías de red es que ninguna de ellas se ajusta a los cinco criterios. Cada tecnología es deficiente en algún aspecto con respecto a las demás.

La base inicial más común para la selección de una red es el tamaño del área que la red puede cubrir. Las tecnologías de red caen en las dos categorías ya mencionadas en la clasificación de *Sistemas Distribuidos*, *WAN* y *LAN*. Hay una distinción importante entre estas dos tecnologías, que es determinante de las otras propiedades importantes de las redes de computadoras. Las tecnologías de área extendida son lentas en velocidad y altas en latencia y proporción de error; las pocas excepciones tienden a ser costosas. Las *LAN* son, en contraste, rápidas y libres de error. La mayoría de los intentos serios de procesamiento distribuido han usado por esta razón tecnología de área local; las *WAN* han sido usadas principalmente para transmitir datos y no han probado ser muy útiles para la distribución de tareas computacionales.

Hay algunos conceptos que se aplican a todas las redes de computadoras y se cubren a continuación, estos incluyen conmutación, topologías de red y el uso de circuitos virtuales.

### 1.3.4 ESTRUCTURAS DE REDES

En toda red existe una colección de máquinas destinadas para correr programas de aplicación. Cada una de estas máquinas recibe el nombre de huésped o estación. Los huéspedes están conectados mediante una subred de comunicación que tiene la función de enviar mensajes entre ellos.

Una subred en la mayor parte de las WAN consiste de dos componentes diferentes: las líneas de transmisión física o canales y los elementos de conmutación.

Los elementos de conmutación, también conocidos como nodos, son computadoras especializadas que se utilizan para conectar dos o más líneas de transmisión, y a su vez, se encuentran conectadas a huéspedes. Sin embargo, en muchas LAN, los nodos son circuitos simples que interfieren lo menos posible con el tránsito de los datos. Cuando los elementos llegan por una línea de entrada, el nodo deberá seleccionar una línea para reenviarlo. La figura 1.2 muestra cómo cada uno de los huéspedes se encuentra conectado a un nodo. Por éste transitan los datos que envía o recibe el huésped.

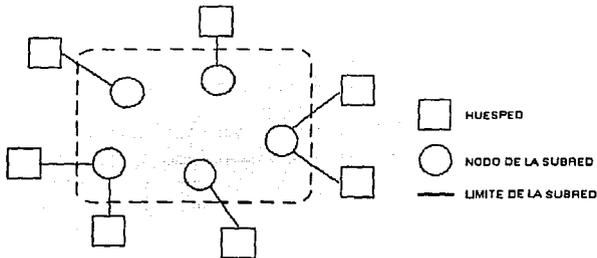


Figura 1.2 Relación entre huéspedes y subred de comunicación.

Existen diferentes formas para la configuración de las líneas de comunicación de una subred de comunicaciones o topología de redes. Sin embargo, existen sólo dos tipos de diseños para una subred de comunicación:

- 1) **Canales punto a punto:** Existe una línea de conexión entre pares de nodos. Si dos nodos desean comunicarse y no comparten una línea común, deben hacerlo a través de otro nodo. Cuando un mensaje se

envía de un nodo a otro a través de uno o más intermediarios, el mensaje es recibido y almacenado por cada uno de ellos hasta que haya una línea de salida disponible para reenviarlo.

En el contexto de una red, a un mensaje se le denomina paquete. La *Organización Internacional de Estándares (ISO)* define a un paquete como "un conjunto de datos y otros elementos de control organizados en un formato determinado y que se transmiten con un todo de acuerdo con un determinado procedimiento de transmisión".

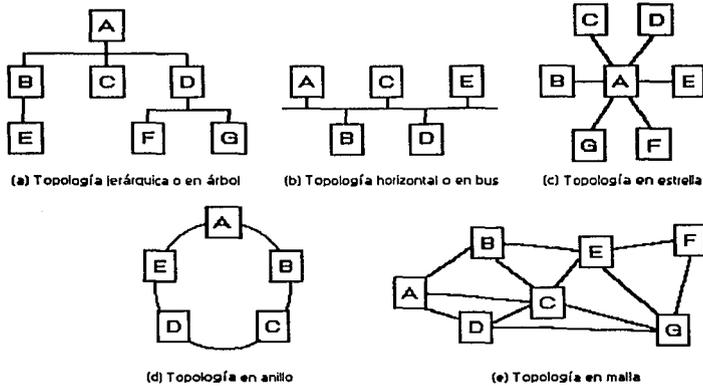


Figura 1.3 Topologías de subred punto a punto.

- 2) **Canales de difusión:** Tienen una sola línea de comunicación que es compartida por todas las máquinas que constituyen la red. Los paquetes que envía una máquina son recibidos por todas las demás. El campo de dirección, localizado entre los elementos de control dentro del paquete, especifica a cual va dirigido. En el momento en que se recibe un paquete se verifica este campo de dirección y, si está destinado a otra máquina, es ignorado.

Los sistemas de difusión admiten la posibilidad de dirigir un paquete a todos los huéspedes mediante un código especial incluido en el campo de dirección.

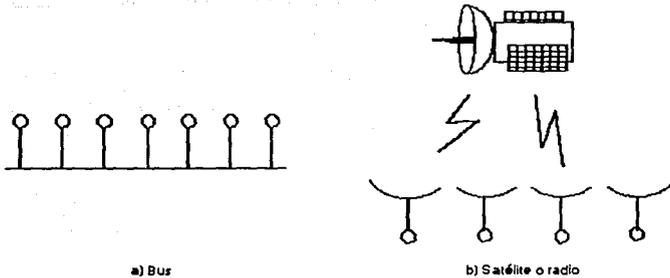


Figura 1.4 Topologías de subred de difusión.

### 1.3.5 ARQUITECTURAS DE REDES

Por lo general, en el análisis de los sistemas de comunicación en las redes de computadoras, se utiliza la metodología consistente en una estructura de niveles o capas, con el objeto de disminuir su complejidad. Cada uno de estos niveles se construye sobre su predecesor y tiene el propósito de ofrecer servicios a los niveles superiores, para los cuales es desconocida su estructura.

El modelo de comunicación más ampliamente usado es el modelo de referencia para *Interconexión de Sistemas Abiertos (OSI)* de la ISO. Este involucra siete niveles, con el nivel de aplicación a lo alto y el nivel físico en la parte inferior (figura 1.5).

Los niveles  $n$  correspondientes conversan entre sí utilizando reglas y convenciones que se conocen como *protocolo* del nivel  $n$ , indicado en la figura 1.5 por las líneas  $P_n$ . No existe una transferencia directa de datos entre niveles correspondientes, sino que pasan la información al nivel inmediatamente inferior, y así sucesivamente, hasta llegar al nivel más bajo de la estructura. Debajo de este primer nivel está el medio físico a través del cual se realiza la comunicación real.

La arquitectura de red involucra el desarrollo de especificaciones que permitan la construcción de *Hardware* o *Software* correspondiente a cada nivel, que siga en forma correcta el *protocolo* apropiado.

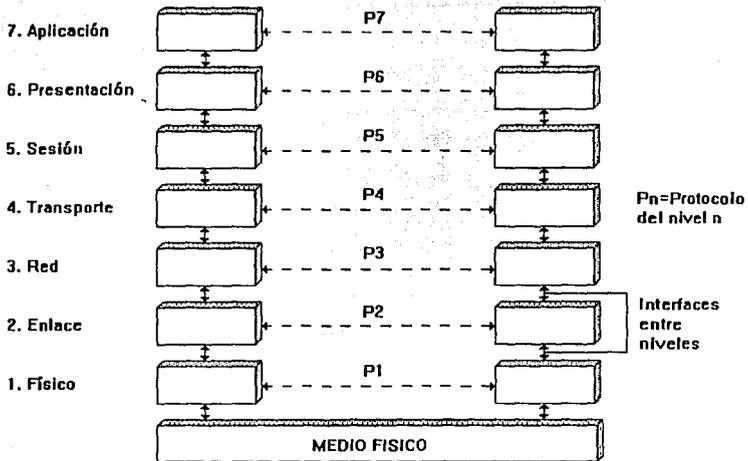


Figura 1.5 Capas, protocolo e interfaces según el modelo de referencia OSI.

### Modelo de referencia de la ISO

El modelo de referencia *OSI* se refiere a la interconexión de sistemas abiertos, es decir, a aquellos sistemas dispuestos a establecer comunicación con otros distintos. El modelo *OSI*, como se mencionó antes, se compone de siete niveles, que fueron definidos bajo ciertos principios:

- 1) Cada nivel será creado en situaciones donde se necesite un nivel diferente de abstracción.
- 2) Cada nivel deberá tener una función bien definida.
- 3) Los límites de los niveles deberán seleccionarse tomando en cuenta la minimización del flujo de información a través de las interfaces.
- 4) El número de niveles deberá ser el suficiente para que funciones diferentes no tengan que ponerse en el mismo nivel. Por otra parte, este número no deberá ser lo suficientemente pequeño para que su arquitectura no sea difícil de manejar.

En el modelo *OSI* se emplea el término *ente* o *entidad* para definir a los elementos activos de un sistema, sean *Software* (proceso) o *Hardware*, que se encuentran en cada uno de los niveles.

## **Nivel 7. Aplicación**

En este nivel se llevan a cabo las funciones específicas de comunicación entre los diferentes procesos de aplicación que constituyen el sistema de red.

Se debe considerar que los procesos de aplicación que utilizan el mecanismo de interconexión se encuentran distribuidos y deben comunicarse para llevar a cabo objetivos comunes. Desde el punto de vista del usuario, un proceso se comunica con otros procesos, y esta operación se lleva a cabo a través del sistema operativo.

## **Nivel 6. Presentación**

El objetivo de los elementos situados a este nivel, es proporcionar un conjunto de servicios a los *entes* que constituyen el nivel superior. Dichos servicios están fundamentalmente orientados a la interpretación de la estructura de las informaciones intercambiadas por los procesos de aplicación. Ejemplos de las funciones que conciernen a este nivel son:

- 1) Selección del tipo de terminal
- 2) Manejo de los formatos de presentación de los datos
- 3) Ordenes de manejo y formateado de los archivos
- 4) Conversiones de códigos de los datos
- 5) Formateado de los datos y órdenes de control
- 6) Control de la forma de transferir informaciones

## **Nivel 5. Sesión**

El objetivo de los elementos situados en este nivel es proporcionar un soporte a la comunicación entre los *entes* del nivel de presentación.

Cada vez que se desea establecer comunicación entre sistemas de elementos distintos, se establece una sesión entre los correspondientes *entes* del nivel de presentación. La sesión regula el diálogo entre ellos y deja de existir cuando este finaliza. Cada *ente* del nivel sesión se identifica mediante una dirección, asociada a un elemento capaz de almacenar la información que se intercambia.

## **Nivel 4. Transporte**

El objetivo de los elementos que componen a este nivel consiste en proporcionar un servicio de transporte de la información a través del sistema.

Este servicio deberá ser transparente para los usuarios (elementos del nivel sesión), liberándolos de ese modo de todo lo referente a la forma de llevar a cabo dicho transporte. El nivel de transporte proporciona principalmente tres servicios:

- 1) Servicios orientados al establecimiento de una conexión
- 2) Servicios orientados hacia la realización de transacciones
- 3) Servicios orientados hacia la difusión de información a múltiples destinatarios

Una de las razones que justifican este nivel es la optimización de los recursos de comunicaciones con objeto de minimizar el costo de dichos intercambios de información.

### **Nivel 3. Red**

El objetivo del nivel de red es proporcionar los elementos necesarios para intercambiar información entre los *entes* del nivel de transporte, a través de una red de transmisión de datos. Las funciones asignadas a los *entes* de este nivel cobran pleno sentido cuando en la comunicación se utiliza una red de transmisión de datos. Un punto de suma importancia en su diseño es la determinación sobre cómo encaminar la información del origen al destino.

### **Nivel 2. Enlace**

El objetivo de este nivel es proporcionar los elementos necesarios para establecer, mantener, y terminar interconexiones de enlace de datos entre los *entes* del nivel de red.

Un enlace de datos se establece siempre entre dos puntos físicos de conexión del sistema. En todos los casos se considera que un enlace es siempre direccional. Existen en la práctica diferentes tipos de *protocolos* de enlace utilizados en el intercambio de información entre sistemas informáticos.

### **Nivel 1. Físico**

En este nivel se definen y materializan las características mecánicas, eléctricas, funcionales y de procedimiento para establecer, mantener y terminar la interconexión física entre nodos y huéspedes.

Para que se lleve a cabo el intercambio de información entre dos niveles debe existir un acuerdo sobre un conjunto de reglas acerca de la interfaz. En una interfaz típica la entidad de la capa (n+1) pasa una unidad de datos de la

interfaz a la entidad del nivel  $n$ , a través de un punto de acceso al servicio. La unidad de datos de la interfaz consiste de una unidad de datos del servicio y de alguna información de control. La unidad de datos del servicio es la información que se pasa, a través de la red, a la entidad del mismo nivel de la otra máquina y posteriormente al nivel  $(n+1)$ . La información de control es necesaria porque ayuda a que los niveles inferiores realicen su trabajo, pero no forma parte de los datos.

Para hacer la transferencia de una unidad de datos del servicio, puede ser necesaria su fragmentación por parte del nivel de la entidad  $n$  en varias partes, de tal forma que a cada una de ellas se le asigne una cabecera y se envíe por una unidad de datos del *protocolo* distinta. Las entidades correspondientes al mismo nivel utilizan las cabeceras de la unidad de datos del *protocolo* para llevar a cabo su *protocolo* de igual a igual. Por medio de ellos se identifica cuales son las unidades de datos del *protocolo* que contienen datos y cuales las que llevan información de control.

Los niveles pueden ofrecer dos tipos diferentes de servicios a los niveles superiores: uno orientado a conexión y otro sin conexión.

El servicio orientado a conexión se modeló basándose en el sistema telefónico. Para utilizar una red con servicio de este tipo, el usuario establece primero una conexión, la utiliza y después termina la conexión. A diferencia de esto, en el servicio sin conexión, cada mensaje lleva una dirección del destino y cada uno de ellos se encamina, en forma independiente, a través del sistema.

### 1.3.6 TELECOMUNICACIONES

En los últimos años se han creado nuevas compañías de telecomunicaciones que ofrecen servicios de comunicación de datos especializados, que por lo general compiten directamente con las compañías telefónicas. Algunas de estas compañías ofrecen servicios de larga distancia con un rendimiento muy alto (mediante el empleo de satélites), en tanto que otras ofrecen tiempo compartido, redes u otros servicios usando facilidades de comunicación que ellas mismas rentan a otros proveedores de servicios portadores comunes.

A Continuación se describen algunos tipos de telecomunicación.

- 1) Red télex: Es una red telegráfica conmutada que permite comunicarse a nivel nacional o al extranjero con la ayuda de un teleimpresor.
- 2) Circuitos alquilados o dedicados: En este tipo de servicio las líneas telefónicas se constituyen entre los puntos de manera permanente, sin limitación de tiempo ni utilización.

- 3) **Red de uso privado:** Incrementa la rentabilidad de una red dedicada, con configuración punto a punto compartiendo una sola línea entre terminales.
- 4) **Red telefónica conmutada:** Actualmente se da a los usuarios de la red telefónica conmutada la posibilidad de interconectar equipos informáticos mediante una selección previa por marcaje con un aparato telefónico.
- 5) **Red VSAT:** Funciona a través del sistema de satélites Morelos, mediante una antena central coordinadora que se enlaza con las antenas remotas de los usuarios, permitiendo así la comunicación entre una y otra.

## **CAPITULO 2**

# **PLANTEAMIENTO DEL PROBLEMA**

## 2.1 DESCRIPCION GENERAL DEL SISTEMA ACTUAL

El sistema que se encuentra en operación actualmente en "**Comercial Emyco, S.A. de C.V.**", presenta muchas deficiencias, debido basicamente a que su desarrollo se realizó desde hace varios años con la tecnología de *Hardware* y *Software* que se ajustaba a las necesidades de esa época.

Algunas de las deficiencias son las siguientes:

- 1) El proceso de actualización de datos de ventas y movimientos de las tiendas es muy lento, además durante dicho proceso no se permite el acceso al sistema por cualquier otra terminal.
- 2) Las veintitres sucursales ubicadas en el interior de la república envían la información a la oficina central localizada en la Ciudad de México, cada tres días por vía terrestre en forma de paquetería. De las seis sucursales del Distrito Federal sólo cuatro de ellas cuentan con punto de venta. Las sucursales con punto de venta envían la información en un disco flexible diariamente. La información recibida por paquetería tiene que ser capturada en la oficina central.
- 3) El sistema presenta inconsistencias en la validación de la información recibida de las sucursales.
- 4) Algunos reportes se tardan hasta dos días en ser generados.
- 5) El disco duro es demasiado pequeño para soportar el volumen de información.
- 6) El sistema actual no cuenta con un proceso automático de reconstrucción de índices. En caso de fallas en la energía eléctrica, este proceso debe realizarse manualmente utilizando herramientas externas al sistema.

Con base en las desventajas mencionadas anteriormente y las necesidades actuales de crecimiento de la empresa, es necesario desarrollar un nuevo sistema que cubra estas deficiencias y al mismo tiempo permita la facilidad de integrar nuevos módulos de procesamiento de información.

A continuación se presenta un panorama del estado actual del sistema.

### 2.1.1 ELEMENTOS DEL SISTEMA ACTUAL

En la tabla que se muestra en la figura 2.1 se describen los diferentes elementos de *Hardware* que componen al equipo de cómputo actual:

Componente de Hardware	Características
CPU	NCR TOWER 1632
Microprocesador	MC68010 a 10 MHz
Terminales	4 unidades marca TELEVIDEO
RAM	1 MB
Discos Duros	2 unidades de 46 MB cada uno
Discos Flexibles	1 unidad de diskettes de 5.25" HD
Unidades de Cinta	1 de 46 MB

*Figura 2.1 Componentes de Hardware.*

Por otro lado, los componentes de *Software* del sistema *TOWER 1632* son los que se muestran en la figura 2.2:

Componente de Software	Características
Sistema Operativo	UNIX V versión 2.01
Sistema de Inventarios (SIDCA)	Desarrollado en lenguaje COBOL
Sistema de Nómina (NOM 12)	Desarrollado en lenguaje COBOL
Sistema de Contabilidad (EMYCON)	Desarrollado en lenguaje COBOL

*Figura 2.2 Componentes de Software.*

Específicamente, el sistema de inventarios *SIDCA* realiza las funciones de:

- 1) Mantenimiento de archivos de artículos, tablas y listados múltiples.
- 2) Permite fijar los precios del inventario por los métodos de costo estándar y costo promedio.
- 3) Proporciona métodos flexibles para fijación de precios y capacidad para cambio automático de precios.
- 4) Permite la entrada de transacciones al inventario y su actualización con listados y cifras de control para su verificación.
- 5) Permite la consulta de existencias de los artículos, incluso a nivel de colores y tallas.
- 6) Permite el manejo de diversos tipos de artículos con o sin color y talla.
- 7) Emite reporte de existencias y ventas, tanto individual como comparativo entre tiendas.
- 8) Permite el registro de los movimientos al inventario, controlándolos por tienda y por lotes con cifras de control.

- 9) Imprime catálogo de artículos para oficina central o para tienda.
- 10) Posee una interfaz con los sistemas de punto de venta y de facturación.
- 11) Controles para realizar los cierres periódicos y anuales.

La empresa también posee 7 equipos *IBM PS/2* en la oficina central y otros 4 equipos similares en tiendas que manejan punto de venta.

El *Punto de Venta* es un sistema de *Software* que corre bajo ambiente *MS-DOS*, el cual se utiliza para realizar diversas actividades internas de una tienda, como son: ventas, devoluciones, embarques, transferencias, etc.

## 2.1.2 ENTRADAS Y SALIDAS DEL SISTEMA ACTUAL

Para obtener un mejor entendimiento del sistema que se planea realizar, a continuación se listan las diferentes entradas y salidas de datos del sistema actual:

### Entradas:

- 1) Reporte de venta diaria de tapas
- 2) Embarques
- 3) Devoluciones a proveedor
- 4) Traspasos o transferencias
- 5) Ajustes
- 6) Memorandums de precios
- 7) Auditorías
- 8) Información de clientes de la tienda
- 9) Etiquetación de mercancía de la tienda
- 10) Altas, bajas y cambios de artículos
- 11) Depuración de artículos a determinada fecha
- 12) Altas, bajas y cambios de proveedores
- 13) Altas, bajas y cambios de clientes

## **Salidas:**

- 1) Reporte de existencias y ventas mensuales
- 2) Tabla comparativa de tiendas en existencias y ventas
- 3) Reporte de existencias a nivel de talla-color
- 4) Reporte de venta y existencias mensuales de 1 a 24 meses atrás, acumulados por grupo o línea
- 5) Reporte de lotes de captura (movimientos o sólo totales)
- 6) Lista de precios
- 7) Diferencias de marcaje y actualización
- 8) Reporte de baja de un color
- 9) Tabla de tiendas, colores, tallas, sublíneas, modelos, grupos, líneas, características y proveedores
- 10) Reporte de archivo de control
- 11) Reporte comparativo de existencias y ventas
- 12) Cierre de mes
- 13) Cierre anual
- 14) Reporte de la historia de un artículo
- 15) Reporte de clientes

## **2.2 PLANTEAMIENTO GENERAL DE SOLUCIONES**

Una alternativa general de solución a las deficiencias del sistema actual es la agilización de los procesos, dándole a cada tienda la facilidad de contar con un sistema local para el manejo de sus inventarios, así como poder transferir los movimientos del día a la oficina central, donde se lleve a cabo el análisis final de la información. Esto implica la necesidad de recurrir a un medio de transmisión de datos entre ciudades, debido a la distribución geográfica de las sucursales.

El planteamiento general de soluciones involucra los dos aspectos importantes en el desarrollo de sistemas, que son: el *Software* y el *Hardware*.

En lo que se refiere al *Software* a desarrollar, este depende del *Hardware* elegido, en aspectos como: el sistema operativo a usar, memoria disponible, capacidad de almacenamiento y velocidad.

## 2.2.1 ALTERNATIVAS PARA COMUNICACION DE DATOS

En el diseño de un sistema apropiado de teleinformática, varios elementos entran en juego, siendo los cuatro siguientes los más importantes:

- 1) Velocidad
- 2) Medio de transporte
- 3) Terminal remota
- 4) Sistema central y aplicación

Además de estos elementos se debe de considerar el costo de instalación y su posible crecimiento. Un punto muy importante es que el sistema proyectado debe apegarse a los reglamentos vigentes de la *Secretaría de Comunicaciones y Transportes (SCT)*, así como la obtención de los permisos correspondientes.

Para las necesidades de comunicación de datos a gran distancia, es posible utilizar los servicios de una red *WAN*. Esta red puede ser propiedad de la organización que la usa (red privada) o provista por una tercera parte (red pública); en este último caso la red es compartida por un número de organizaciones. A las *WAN* se les conoce también como redes de comunicación conmutadas. Estas consisten de una colección de nodos interconectados, en los cuales la información es transmitida de una estación fuente a una estación destino, siendo encaminada a través de la red de nodos. La figura 2.3 ilustra este concepto en forma simplificada. Los nodos son conectados por vías de transmisión. Las señales que entran a la red desde una estación son encaminadas al destino siendo conmutadas de nodo a nodo. Por ejemplo, la información de la estación *A* destinada a la estación *F*, es enviada al nodo 4 y posteriormente a los nodos 5 y 6 o los nodos 7 y 6 hacia el destino.

Existen dos tecnologías muy diferentes que son usadas en las redes conmutadas de área extendida: la conmutación de circuitos y la conmutación de paquetes. Estas dos tecnologías difieren en la forma en que los nodos conmutan la información de una liga a otra en el camino de la fuente al destino.

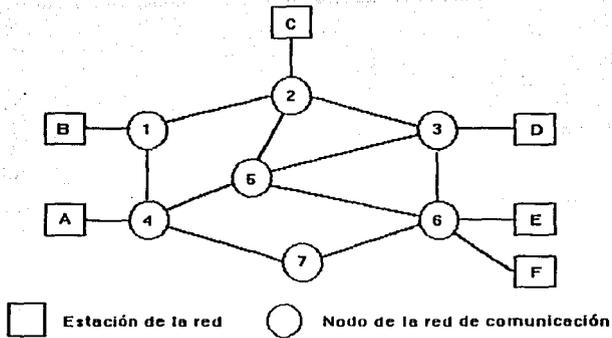


Figura 2.3 Red de conmutación genérica.

### Conmutación de circuitos

La ISO define a la conmutación de circuitos como el procedimiento que enlaza a voluntad dos o más equipos terminales de datos y que permite la utilización exclusiva de un circuito de datos durante la comunicación.

La conmutación de circuitos es la tecnología actualmente dominante para comunicación de voz y datos. La comunicación vía conmutación de circuitos implica que haya una ruta de comunicación dedicada entre dos estaciones. Esta ruta es una secuencia conectada de ligas entre los nodos de la red. En cada liga física, un canal es dedicado a la conexión, el ejemplo más común es la red telefónica.

La ruta de conexión es establecida antes de que la transmisión de datos comience. De este modo la capacidad del canal debe ser reservada entre cada par de nodos entre la ruta, y cada nodo debe tener disponible capacidad interna de conmutación para manejar la conexión requerida. Los conmutadores deben de tener la inteligencia para hacer estas asignaciones y para trazar un camino a través de la red.

### Redes de conmutación de paquetes

La ISO define la conmutación de paquetes como un procedimiento de transferencia de datos mediante paquetes provistos de direcciones, en el que la vía de comunicación se ocupa unicamente durante el tiempo de transmisión de un paquete, quedando a continuación la vía disponible para la transmisión de otros paquetes.

En la conmutación de paquetes los datos son transmitidos en paquetes cortos. Un límite típico en la longitud del paquete es 1000 bytes. Si una fuente tiene un paquete para enviar más largo, el mensaje es fragmentado en una serie de paquetes. Cada paquete contiene una porción de datos de usuario más alguna información de control. La información de control, en un mínimo, incluye la información que la red requiere para ser capaz de encaminar el paquete a través de la red y dirigirla al destino propuesto. En cada nodo de la ruta, el paquete es recibido, almacenado brevemente y pasado al siguiente nodo.

Los paquetes son inicialmente enviados al nodo al cual se conecta la estación enviante. Cuando cada paquete llega a este nodo, el nodo almacena el paquete brevemente, determina la siguiente rama de la ruta, y forma el paquete en una cola para enviar por esa liga. Cuando la liga está disponible, el paquete es transmitido al siguiente nodo. Todos los paquetes trabajan eventualmente su camino a través de la red y son entregados en el destino propuesto.

El enfoque de la conmutación de paquetes tiene un número de ventajas sobre la conmutación de circuitos:

- 1) La eficiencia en línea es mayor, ya que una simple liga nodo a nodo puede ser compartida dinámicamente por muchos paquetes sobre el tiempo. Los paquetes son puestos en línea de espera y transmitidos tan rápidamente como sea posible sobre la liga. En contraste con la conmutación de circuitos, el tiempo en una liga nodo a nodo es preasignado usando multiplexaje síncrono por división en el tiempo. En mucho de ese tiempo tal liga puede estar ociosa, debido a que una porción de su tiempo es dedicado a una conexión que está ociosa.
- 2) Una red de conmutación de paquetes puede llevar a cabo conversión de proporción de datos. Dos estaciones de diferentes proporciones de datos pueden intercambiar paquetes, ya que cada una se conecta a su nodo a su propia razón de datos.
- 3) Cuando el tráfico se hace pesado en una red de conmutación de circuitos, algunas llamadas son bloqueadas; esto es, la red se rehusa a aceptar requerimientos de conexión adicionales hasta que la carga en la red disminuya. En una red de conmutación de paquetes, los paquetes son aún aceptados, pero el retraso en la entrega se incrementa.
- 4) Pueden usarse prioridades. Así, si un nodo tiene un número de paquetes en cola para transmisión, puede transmitir los paquetes de mayor prioridad primero. Estos paquetes experimentarán por ello menor retardo que los paquetes de menor prioridad.

## 2.2.2 ALTERNATIVAS PARA REDES DE AREA EXTENDIDA

La conmutación de paquetes suma diversas alternativas nuevas para la red de área extendida, en adición a aquellas que pueden ser provistas usando tecnología de conmutación de circuitos. Así como hay redes de conmutación de circuitos públicas y privadas, hay también redes de conmutación de paquetes públicas y privadas. Una red de conmutación de paquetes pública trabaja como una red telefónica pública. En este caso la red provee un servicio de transmisión de paquetes a una variedad de suscritos. Típicamente, el proveedor de la red posee un conjunto de nodos de conmutación de paquetes y los liga con líneas rentadas provistas por terceros. A tal red se le denomina *Red de Valor Agregado (VAN)*, reflejando el hecho de que la red suma valor a las facilidades de transmisión. En la mayoría de los países hay una sola red pública propiedad o controlada por el gobierno y referida como una *Red Pública de Datos (RPD)*.

La otra alternativa de conmutación de paquetes es una red dedicada a las necesidades de una organización. La organización puede poseer los nodos de conmutación de paquetes o rentar una red entera de conmutación de paquetes de un proveedor de red. En cualquier caso, las ligas entre los nodos son de nuevo líneas de telecomunicaciones rentadas.

En las figuras 2.4 y 2.5 se resumen las características principales de la conmutación de circuitos y la conmutación de paquetes.

### CONMUTACION DE CIRCUITOS

Ventajas	Desventajas
<p>Compatible con voz. Economías de gran escala pueden ser realizadas usando la misma red para voz y datos.</p> <p>Generalización de procedimientos de llamada para voz y datos. No se requiere entrenamiento de usuario o protocolos para manejar el tráfico de datos.</p>	<p>Sujeto a bloqueos. Esto hace difícil clasificar la red propiamente. El problema es menos severo con el uso de técnicas de encaminamiento dinámicas no jerárquicas.</p> <p>Requiere compatibilidad del suscrito. Los dispositivos en cada final del circuito deben ser compatibles en términos de protocolo y razón de datos, ya que el circuito es una conexión transparente. Además, para cada terminal conectada a un <i>host</i>, se necesita una línea física en el <i>host</i>.</p> <p>Gran carga de procesamiento y señales. Para aplicaciones tipo transacción, las llamadas de datos son de corta duración y necesitan ser activadas rápidamente. Esto incrementa proporcionalmente la sobrecarga de la red.</p>

Figura 2.4 Características de la conmutación de circuitos.

## CONMUTACION DE PAQUETES

Ventajas	Desventajas
<p>Proporciona conversión a velocidad. Dos dispositivos conectados con diferentes razones de datos pueden intercambiar datos; la red coloca en un <i>buffer</i> los datos y los dirige a la razón de datos apropiada.</p> <p>Desaparece el bloqueo. Conforme se incrementa la carga de la red, el retardo se incrementa, pero usualmente son permitidos nuevos intercambios.</p> <p>Utilización eficiente. Se usan conmutadores en lugar de capacidad dedicada a una llamada particular.</p> <p>Multiplexaje lógico. Un sistema <i>host</i> puede tener conversaciones simultáneas con un número de terminales sobre la misma línea.</p>	<p>Encaminamiento y control complejos. Para lograr eficiencia y elasticidad, una red de conmutación de paquetes debe emplear un conjunto de algoritmos de encaminamiento y control complejo.</p> <p>Retardo. El retardo está en función de la carga. Puede ser grande y es variable.</p>

*Figura 2.5 Características de la conmutación paquetes .*

Existe una variedad de servicios que facilitan la transmisión de datos, de entre los cuales, dadas las características del problema a resolver, se mencionan las siguientes:

### **Enlace privado**

En diferentes ciudades se tiene la necesidad de capturar un gran volumen de información, que debe tenerse casi de manera inmediata en el sistema central de la Ciudad de México. La solución a esta necesidad sería instalar una terminal en cada ciudad, con todos los requerimientos de teleproceso y todos sus equipos adecuados.

La terminal se conecta a un *modem*, el cual a su vez se conecta a una línea privada dedicada las 24 horas del día al uso exclusivo de la compañía. Esta línea llega a una central telefónica, desde donde se envía la información a la Ciudad de México a través de microondas o cualquier otro medio de enlace a distancia. De la central telefónica de la Ciudad de México, la información es enviada al domicilio adecuado. La línea se conecta al *modem* correspondiente y de ahí a la computadora central, por medio de los respectivos controladores de comunicaciones.

### **Red conmutada**

La *Red Conmutada* es una red tipo malla que se compone por una gran cantidad de pares de cables delgados (hilos) que llegan de una central telefónica a la casa de una persona que cuenta con un auricular en el cual el

sonido que viaja por el cable en forma de señales eléctricas es transformado a señales acústicas.

En un principio el teléfono era conducido por varias personas llamadas telefonistas, que se encargaban desde una central telefónica de recibir la llamada de una zona X y comunicarla a una zona Y, esto por medio de un cable que se conecta a un canal origen que hace la llamada a un canal destino que la recibe.

A medida que fue creciendo la cantidad de líneas telefónicas, las telefonistas no se daban abasto, por lo cual se crearon los circuitos de conmutación (*switches*), los cuales utilizando equipo electrónico (*microprocesadores*) han logrado automatizar el proceso de una llamada telefónica.

Los principales problemas que presenta la *Red Conmutada* son el ruido en las líneas, congestión de tráfico, la falla de los equipos telefónicos y su demora en arreglarlos.

Desde el momento en que se hace la llamada, ésta se conecta a una central telefónica que se encarga de recibir las llamadas de los usuarios y encaminarlas a su destino. Cada central maneja 40,000 líneas o 4 series.

Ejemplo:

2 22	13	15
1	2	3

1: Número de serie : 2 22

2 y 3: 0000 a 9999 o sea 10,000 líneas o abonados por serie

Si se trata de alguna de las series que maneja la central, ésta se conecta directamente al abonado destino sin salir de la central.

Si la serie que marca no se encuentra en la misma central, ésta se dirige hacia la central que maneje la serie correspondiente, si ésta no puede contestar se le envía *altandem*. La función del *tandem* es encontrar la ruta más apropiada para conectar la llamada (distribuir el tráfico).

Cuando la llamada es foránea, la central telefónica envía la señal a la *Central Automática de Larga Distancia (CALD)*, la cual se comunica directamente a la otra *CALD* de la ciudad correspondiente, y ésta se encarga de transmitirla a la central telefónica a la que pertenece la serie de la llamada. La *CALD* almacena

en cinta el número telefónico que hace la llamada, el número que recibe, la fecha, la hora y la duración; para posteriormente incluirla en el recibo telefónico.

Los *tandem* manejan alrededor de 25 centrales y están interconectados entre sí.

Dentro de la nueva administración de *Telmex*, se están sustituyendo las líneas telefónicas por fibra óptica, la cual mejora la calidad y al mismo tiempo incrementa el número de líneas disponibles y se aumenta la velocidad de transmisión de datos.

*Telmex* facilita a sus usuarios la posibilidad de realizar la interconexión entre equipos informáticos a través de la *Red Conmutada*, que es la base de la infraestructura existente para la telefonía.

En consecuencia la *Red Conmutada* permite a un usuario, mediante una selección previa por marcaje con un aparato telefónico, la comunicación con cualquier otro usuario del mundo.

Actualmente se dispone en los *modems* de los correspondiente dispositivos de respuesta y desconexión automática para cuando la terminal actúa de receptor.

El modo de operación puede ser tan sencillo como establecer la conexión manualmente a través del operador de la terminal o computadora, pasando posteriormente mediante el correspondiente conmutador de "datos", que puede encontrarse en el propio *modem* o en un equipo auxiliar llamado unidad de control voz/datos, a estar en disposición de transmisión o recepción, siempre que en el otro extremo el operador haya realizado la operación de conmutar a datos, si la conexión se hace de forma manual, o si ha recibido una señal acústica del *modem* distante si la respuesta es automática.

En cuanto a la posibilidad de la llamada automática o "*autocall*" (recomendación V-25) está prevista en breve su incorporación a los equipos conversores de señales.

Por ser una red pública, en principio, cualquier usuario puede ponerse en comunicación con otro, debiendo ser el receptor en especial si la respuesta es automática, el que realice los controles pertinentes (por ejemplo definición de los correspondientes "*password*") para que sólo accedan los usuarios autorizados.

La disponibilidad de acceso a una *Red Conmutada* es casi inmediata, pues es solo cuestión de tomar un teléfono y efectuar la llamada correspondiente, ya sea local o de larga distancia.

Las líneas de transmisión de *Red Conmutada*, se sabe por experiencia, son de muy mala calidad, comparadas con las redes de enlace privado.

Para tener acceso a la *Red Conmutada* se deben de seguir los siguientes pasos:

- 1) Se marca el teléfono de la red.
- 2) La red contesta por medio de un *modem* de respuesta automática con un sonido agudo.
- 3) Se transfiere la llamada a la terminal correspondiente.
- 4) Aparece identificación de la red en la terminal.
- 5) Se identifica la terminal para pedir acceso a la red y el direccionamiento del punto remoto, que es el sistema central.
- 6) Posteriormente, el sistema central pide la identificación de la terminal para concederle acceso al sistema de la compañía.

En la figura 2.6 se muestra la relación de costo por minuto aproximado (sin IVA) entre la Ciudad de México y algunas ciudades de provincia (horario diurno).

Ciudad destino	Importe (por minuto)
Villahermosa	\$ 1,815.00
Guanejuato	\$ 1,422.00
Guadaleajara	\$ 1,715.00
Irapuato	\$ 1,389.00
León	\$ 1,554.00
Puebla	\$ 1,009.00
Queretero	\$ 1,280.00
Monterrey	\$ 1,706.00
Veracruz	\$ 1,611.00

Figura 2.6 Precio por minuto.

Las comunicaciones con computadoras personales poseen muchas características comunes a las de los *Main Frames*. No obstante, existen también algunas diferencias que conviene tener presentes a la hora de utilizar la *PC* para la conexión en una red o para las comunicaciones de datos en general. En primer lugar, la mayoría de las computadoras personales son asíncronas.

No existe ninguna razón técnica por la cual un equipo personal no pueda ser síncrono, pero lo cierto es que la mayoría de los fabricantes de *PC's* no han incluido en sus tarjetas de comunicaciones (puertos) las líneas correspondientes a las señales de temporización necesarias para las transmisiones síncronas. No

obstante, cada día salen al mercado más productos para *PC* en entornos síncronos, como el *SDLC*, *BSC*, e incluso *X25*.

Las *PC*'s asíncronas pueden comunicarse también con sistemas síncronos, si se intercala entre la *PC* y el *modem* una unidad adaptadora de asíncrono a síncrono. Este adaptador también puede colocarse en forma de tarjeta. El adaptador proporciona las señales necesarias para llevar a cabo la sincronización entre la *PC* y el *modem* (líneas 24,15 y 17 de *RS-232-C* y líneas 113, 114 y 115 de las interfaces *V.24*).

Algunas *PC*'s emplean *Modulación de Frecuencia (FM)* para transmitir la señal por el canal telefónico. El método más habitual es la *Modulación por Desplazamiento de Frecuencia (FSK)*. La idea es simple, se utilizan dos tonos de frecuencias diferentes para representar el 1 y el 0. Casi todas las *PC*'s emplean también *modems* en *full duplex*.

## **Telepac**

*Telepac* es una red especializada en la transmisión de datos, ya que emplea la técnica de conmutación de paquetes a través de varios protocolos, permitiendo a los usuarios contar con un servicio confiable, de alta calidad y bajo costo.

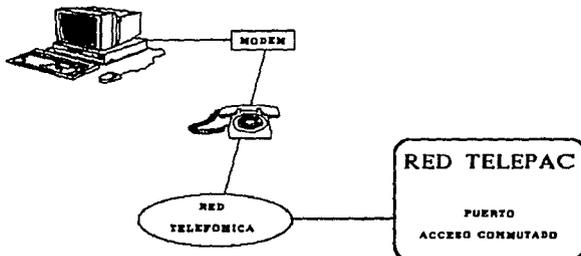
### *Telepac y diversos sistemas de cómputo*

- 1) Usuario con configuración *Host*/Terminal (síncrono)
- 2) Usuario con configuración *Host* (síncrono)
- 3) Usuario con configuración de Terminal (asíncrono)
- 4) Usuario con acceso conmutado con equipo *LapTop* o *PC* (asíncrono)

De los diversos sistemas de cómputo mostrados en los incisos anteriores, se utilizará la opción número 4, ya que es la que se apega al equipo de *Hardware* que se utilizará. Las características de esta configuración son:

- 1) Acceso a la red telefónica conmutada
- 2) Tipo de transmisión asíncrono de 300 a 2400 *bps*
- 3) Clave de acceso con un *ID* de 7 caracteres alfabéticos y un *Password* de 6 caracteres numéricos.
- 4) Terminal inteligente (equipo *PC*).

En la figura 2.7 se muestra el esquema de conexión a la red *Telepac* a través de la línea telefónica conmutada.



*Figura 2.7 Conexión a la red Telepac vía línea telefónica conmutada.*

Telepac opera con los protocolos: X.25, X.3, X.28, X.29 y X.75.

La infraestructura de comunicación está conformada siguiendo un diseño de tipo malla, que la divide en dos subredes: la red de acceso y la red de transporte.

Actualmente Telepac cuenta con una cobertura de 55 ciudades del interior de la República (Figura 2.8).

Acapulco	Cuicatán	Mexicali	Saillilo
Aguascalientes	Durango	México, D.F.	San Luis Potosí
Campeche	Eneenada	Monclova	Tampico
Cancún	Guadalajara	Monterrey	Tepic
Cd. Delicias	Guanajuato	Morelia	Tijuana
Cd. Juárez	Hermoaillo	Nogales	Tlalnepantla
Cd. Obregón	Jalapa	Nuevo Laredo	Tlaxcala
Cd. Victoria	La Paz	Oaxaca	Toluca
Celaya	León	Pachuca	Torreón
Chetumal	Los Mochis	Piedras Negras	Tuxtla Gutiérrez
Chihuahua	Manzanillo	Puebla	Veracruz
Coatzacoacoas	Mazatlán	Puerto Vallarta	Villa Hermosa
Colima	Mérida	Querétaro	Zacatecas
Cuernavaca	Matamoros	Reynosa	

*Figura 2.8 Ciudades de México que cuentan con servicio de Telepac*

### *Tarifas (Proporcionadas por Telecomm)*

Las tarifas de la red pública de transmisión de datos *Telepac* se aplican conforme a las siguientes cuotas:

A. Servicio Permanente	Cuota
I. Por suscripción	\$ 150,000.00
II. Por cada cambio de identificador de red asignado	\$ 50,000.00
III. Por conexión o reconexión	
a) Por puerto, por ocasión	
1. Por enlace dedicado	\$ 280,000.00
2. Por red conmutada	\$ 70,000.00
IV. Por cada identificador de red asignado, mensualmente	\$ 44,000.00
V. Por uso de puerto, mensualmente	
a) Modalidad asíncrona	
1. Por enlace dedicado	\$ 288,000.00
2. Por red conmutada	\$ 28,000.00
b) Modalidad síncrona	
1. Por enlace dedicado	\$ 576,000.00
B. Servicio nacional por enlace dedicado o por red conmutada (mensualmente)	
I. Por cantidad de información (por kilosegmento)	\$ 1,720.00
II. Por tiempo de conexión (por minuto)	\$ 160.00

De ser seleccionado este servicio, solo se pagarían las tarifas referentes a acceso vía red telefónica conmutada.

### **Red Digital Integrada (RDI)**

Actualmente *Teléfonos de México (Telmex)* ofrece el servicio de telefonía digital integrada, gracias a complejos sistemas de fibra óptica y equipos de microondas digitales instalados en las ciudades de México, Monterrey, Guadalajara, Cd. Juárez, Nuevo Laredo, Chihuahua y Reynosa; con una capacidad de servicio de 40,000 líneas digitales de alta velocidad para conmutadores y 1,300 líneas de acceso múltiple. La *RDI* está basada en tres redes públicas de transporte de información:

- 1) **Red Digital Terrestre:** Es una red adicional a la red telefónica pública, constituida por nodos de transmisión digital interconectados entre sí. Esta red no utiliza ninguno de los recursos actualmente en uso de la

red conmutada analógica; además, cuenta con nodos de conmutación para la interconexión de grandes usuarios, lo que permite ofrecer a la comunidad de negocios del país servicios avanzados en telecomunicaciones.

- 2) **Red Satelital Multiusuario:** Complementa los servicios de la red digital terrestre. Para ello, se instalaron dos estaciones maestras, una en la Ciudad de México y otra en Ciudad Juárez. Con esto se enlaza la comunicación de voz y datos a través del satélite Morelos y cinco estaciones de banda ancha en los principales polos de desarrollo turístico e industrial del país. Cuenta con una capacidad para conectar hasta mil estaciones remotas, en donde sean requeridas, brindando a los usuarios la posibilidad de rentar circuitos vía satélite en condiciones competitivas de precio y calidad, evitando los altos costos de inversión requeridos al instalar una red propia.
- 3) **Red de Paquetes de datos:** Apoyados en la infraestructura digital antes mencionada, *Telmex* tiene en operación la red de paquetes de datos. A través de esta red se ofrecen servicios como correo electrónico, servicio de videotexto, intercambio de información entre sistemas de cómputo para soporte de operaciones a tiempo que requieran los usuarios. Algunos de los principales usuarios de esta red son: Banamex, Bancomer, Banco Internacional, Probursa, Asemex, Inverlat, Celanese y Visa. Este servicio se ofrece actualmente en solamente algunas de las ciudades más importantes del país.

También está en proceso de instalación una red multiusuario de satélite para incorporar a un mayor número de poblaciones del territorio nacional a los servicios de la novedosa *RDI*, puesta en operación en febrero de 1991.

Según *Telmex*, la *RDI* ofrece tres ventajas:

**Disponibilidad:**

- 1) Se tiene la infraestructura para brindar el servicio inmediatamente.
- 2) Respaldo de la instalación y supervisión por compañías de prestigio mundial.
- 3) Tiempos mínimos de respuesta en el servicio.

**Confiabilidad:**

- 1) El medio de transmisión es fiable, ya que utiliza fibra óptica, un medio de transmisión inmune al ruido de interferencia.

- 2) Precisión para completar llamadas con el uso de centrales de conmutación digital.
- 3) Respaldo asegurado mediante la instalación de radios digitales y fibras ópticas de soporte.

#### **La tecnología empleada:**

La *RDI*, también llamada "red superpuesta" (porque esta superpuesta a la red telefónica convencional, a la que complementa), trabaja mediante fibra óptica, material que permite una señal digital de 2.04 *Mbps*.

La velocidad de transmisión es de 140 *Kbps*, permitiendo así el envío de datos, voz, imágenes, etc. El usuario puede conectarse mediante un conmutador privado (deberá ser digital y contratar el servicio de *RDI*). El enlace se hace vía la *Unidad de Servicio Remota (USR)*.

Para convertir la señal se utilizan *multiplexores Mx serie 7000* y equipo de potencia de 54 watts, ambos suministrados por *Ericsson*.

Todos los datos se mandan a la estación central de la *RDI Telmex*, ubicada en Carrasco, al sur de la Ciudad de México. En este centro de control, con ayuda de equipo *Unys X* versión 5.0, y tecnología *Accunet* de la *AT&T*, se manejan, controlan, almacenan y actualizan los sistemas de conexión digital de los subsistemas que pasan por la red *DACS (Digital Access Crossconexion System)*, la cual monitorea permanentemente todos los sistemas.

El usuario puede contar con los canales que desee, aunque generalmente hay dos: *PC 0940*, con canal asociado, "que maneja 40 más un canales", y *PC 0860*, sin canal asociado, "que maneja 30 más dos canales".

#### **Servicios Principales:**

La *RDI* ofrece a sus usuarios diversos servicios:

Acceso digital a un conmutador electrónico digital. Mediante este servicio, los conmutadores del usuario pueden ser conectados a la *RDI* vía troncales de 64 *Kbps* y enlaces de 2.04 *Mbps*, permitiendo la transmisión de información tanto de voz como de datos.

Acceso directo. Este servicio permite que las extensiones del conmutador del usuario, conectadas a la *RDI*, puedan ser accesadas desde el exterior con un número directo, sin necesidad de la intervención de la operadora.

Centrex básico. En las oficinas del usuario se instala un módulo de la central de conmutación de la *RDI*, dotando así al cliente de los servicios y funciones básicas equivalentes de un conmutador privado.

Videoconferencia. Es la capacidad de transmitir señales de video interactivo y en diversas localidades a través de enlaces de la *RDI*, estableciendo una comunicación efectiva y dinámica que permite optimizar tiempo y costos de las empresas.

Enlace digital de alta velocidad. Se utiliza un canal de 2.04 *Mbps* punto a punto para la transmisión de señales de información, como voz, datos e imágenes. El enlace digital de alta velocidad mejora las comunicaciones al facilitar la administración de su capacidad, ya que puede modularse de acuerdo con la necesidades de cada usuario. Puede manejar desde 30 comunicaciones de voz comprimida, así como 30 de datos de 64 *Kbps* hasta 180 de 9.6 *Kbps*.

Red privada metropolitana. Es la capacidad de interconectar a través de la *RDI* todas las ubicaciones de un cliente en una misma ciudad, con las facilidades y servicios de una red privada con enlaces de alta calidad y velocidad que permiten la administración adecuada de recursos de la empresa mediante la tecnología digital, pudiendo seleccionar diversas modalidades de transmisión, como son voz, datos e imágenes, como si fuera en un solo edificio, optimizando la operación diaria.

Cruce fronterizo. Ofrece a las empresas e instituciones de las ciudades fronterizas la posibilidad de conectarse con clientes o sucursales de ciudades de Estados Unidos a través de enlaces de la *RDI* de hasta 2.04 *Mbps*.

Red global. Permite el enlace de redes digitales avanzadas en el país con localidades de diversas naciones, para el establecimiento de comunicaciones efectivas y competitivas.

Red privada de voz y datos. Integra todas las funciones que una empresa lleva a cabo en diferentes localidades mediante los servicios de la *RDI*, los cuales ofrecen una conectividad total entre domicilios y ciudades.

Red de paquetes de datos. La *RDI* permite la transferencia electrónica de datos, el acceso a bases de datos (videotexto) y el uso del correo electrónico entre empresas e instituciones.

Red satelital. Esta red permite el envío de información a aquellas empresas que se encuentran en ciudades que no cuentan con infraestructura terrestre digital.

Así mismo, la tecnología *RDI* facilita su mantenimiento y continuidad al incorporar sistemas de administración de red computarizados. Estos sistemas están en los

centros de control y se encargan del diagnóstico y monitoreo sistemáticos del servicio. En caso de ser necesario, reenrutran la señal de inmediato a medios de transmisión de respaldo, asegurando tiempos mínimos de respuesta al usuario.

### **Terminal de apertura muy pequeña (VSAT)**

El Servicio de red pública de acceso múltiple por satélite *VSAT (Very Small Aperture Terminal)* es un servicio que se presta bajo la modalidad de tiempo compartido, mediante la técnica de conmutación de paquetes a usuarios con necesidades de comunicación de datos, a velocidades desde 9.6 hasta 64 Kbps.

Este sistemas se diseñó como un sistema de red pública de transmisión de datos bajo la modalidad de tiempo compartido. Consiste basicamente en la estación terrestre maestra de *Telecomm* y un conmutador de paquetes de datos, que junto con las varias estaciones remotas de los usuarios ubicadas en cualquier lugar del país configuran una red tipo estrella.

La red *VSAT* funciona a través del sistema de satélites Morelos, mediante una antena central coordinadora que enlaza las antenas remotas de los usuarios, permitiendo así la comunicación entre una y otra.

La red requiere de:

- 1) Una antena remota modelo *Skylinx 25* de *Scientific Atlanta*
- 2) Un radio *modem* de protocolo *X.25*
- 3) Cualquier computadora capaz de soportar los volúmenes de datos que se demanden

Dentro del rango de mayor capacidad, el sistema está en condiciones de operar a velocidades de 64 Kbps y con cuatro puertos, cada uno con velocidades de hasta 19.2 Kbps respectivamente.

En caso de utilizar un *protocolo* diferente al *X.25*, se cobra un cargo adicional a la tarifa normal.

En la figura 2.9 se muestra graficamente la configuración *VSAT*:

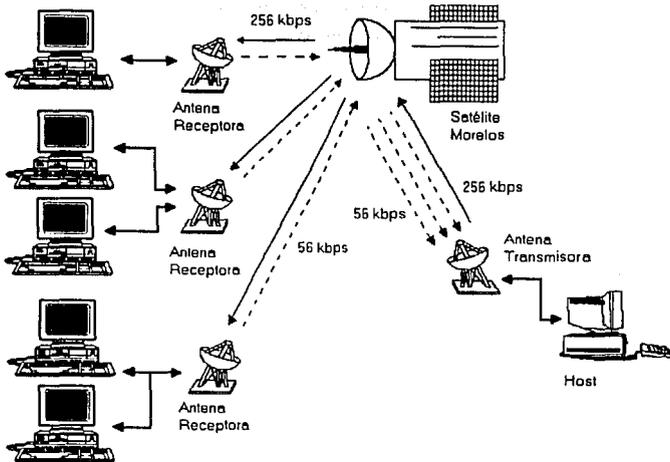


Figura 2.9 Configuración VSAT

## 2.2.3 RENDIMIENTO DE LOS MEDIOS DE COMUNICACION

En la figura 2.10 se ilustra el rendimiento entre las líneas telefónicas conmutadas, líneas privadas y redes de paquetes públicas y privadas.

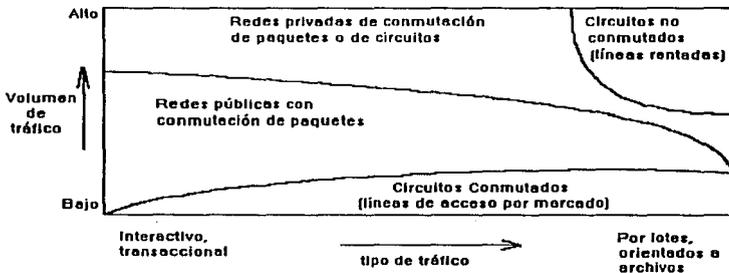


Figura 2.10 Rendimiento de los medios de comunicación de datos

Las líneas privadas alquiladas constituyen la opción más recomendable para las organizaciones que generen un tráfico intenso y constante durante las 24 hrs.

del día, o para aquellas que no puedan tolerar los retardos de las líneas normales. Una línea alquilada puede usarse en cualquier momento, ya que siempre está conectada. Por otro lado, los canales alquilados suelen ser la opción idónea para aquellas empresas que establezcan conexiones multipunto a través de sus canales privados, ya que la estructura multipunto permite compartir mejor el canal.

Para las organizaciones con un volumen de tráfico bajo o medio suele resultar interesante suscribirse a una red pública de paquetes. Las empresas cuyo tráfico sea más bien ocasional pueden tener suficiente con la red telefónica pública. También para las organizaciones esparcidas a lo largo de una amplia región geográfica suele resultar rentable la red de paquetes pública *Telepac*, ya que la mayoría de estas redes gravan el volumen, y no la distancia entre los equipos terminales de datos.

En muchas empresas se han establecido redes privadas de paquetes o sistemas privados de conmutación de circuitos. Estos sistemas presentan ciertas ventajas, para volúmenes de tráfico medios o intensos, suelen resultar más rentables que las líneas conmutadas. Además, si el tráfico se produce a ráfagas intensas, una red de paquetes privada suele ofrecer mejor servicio y es más económica que una línea privada.

## **2.2.4 MANEJADORES DE BASES DE DATOS ACTUALES**

La selección del *DBMS* es una tarea de gran importancia para cualquier empresa, ya que una mala elección podría propiciar un mal procesamiento de datos, o incluso, la pérdida de los mismos. Antes de adquirir el *DBMS*, es importante realizar un estudio de mercado para conocer las características de los existentes actualmente.

A continuación se narra la historia de una empresa que tuvo serios problemas debido a la mala selección del *DBMS*:

Hace algunos años, una compañía instaló una *LAN*, y adquirió algunas herramientas de *Software* disponibles en esa época. Los *DBMS* eran muy pocos y extremadamente caros, así que se seleccionó uno que no trabajaba en red.

El *DBMS* que se seleccionó parecía perfecto, ya que el ambiente de trabajo era muy amigable, los usuarios podían diseñar sus propias formas de entrada y salida de datos, contaba con la opción de impresión en láser, ahorrando la necesidad de transferir la información a un procesador de palabra. Durante los siguientes meses se capturaron miles de registros de información.

Debido a que el número de archivos y directorios se incrementó de manera alarmante, se decidió encontrar la manera de agrupar los archivos pequeños en archivos más grandes. Se contactó al distribuidor para pedir asesoría y se enteraron de la existencia de nuevos y más poderosos manejadores, con la capacidad de operar en red.

La empresa decidió adquirir un nuevo manejador, pero se toparon con una gran cantidad de problemas, como fue el hecho de convertir los archivos del manejador anterior al nuevo, la creación de nuevos formatos de captura y reporte, etc. Como consecuencia, los costos se duplicaron y casi toda la información se tuvo que volver a capturar.

En la actualidad existen muchos *DBMS* en el mercado, lo cual propicia que la selección del manejador apropiado sea más complicada. Sus características y alcances son muy variados, así como los requerimientos de *Hardware*. El manejador apropiado depende de las necesidades de cada empresa.

En la figura 2.11 se muestran algunos datos correspondientes a los principales *DBMS* disponibles actualmente en el mercado.

Algunas de las diferentes variantes son la que a continuación se muestran:

### **Windows vs. No-Windows**

Si la empresa ha estandarizado sus aplicaciones bajo el ambiente *Windows*, probablemente será recomendable adquirir un manejador que sea compatible con *Windows*, pese a que actualmente no hay muchas opciones disponibles. La ventaja de estos manejadores es que permiten trabajar en ambiente gráfico, pero el *Hardware* necesario debe cumplir con características más sofisticadas, como es un procesador 80386SX mínimo, de 4 a 6 MB de RAM y un monitor VGA de preferencia de color.

Por otro lado, los manejadores que no operan bajo ambiente *Windows*, son más variados y los costos de equipo que requieren son mucho menores.

### **Plataforma múltiple**

Actualmente, en el mercado existen diferentes equipos de cómputo con características propias, sistemas operativos diferentes, etc. Algunos *DBMS* permiten operar bajo diferentes plataformas de *Hardware* (*Oracle*, *Progress*, *dBASE IV*). Desafortunadamente, algunas aplicaciones diseñadas para *MS-DOS*, por lo regular presentan problemas al correr bajo *UNIX* o *VMS* y viceversa. Es recomendable adquirir un manejador que opere bajo múltiples plataformas, sólo cuando sea necesario.

Compañía	Producto	S.O.	Multi-usuario	Lang. progr.	Interf. usuario	Soporte Server	Precio
1 on 1 Computer Solutions, Inc.	1 on 1 = 3 Integrated v. 2.0	DOS	No	Xbase	SI		\$96 a \$499
Ace Software Corp.	Ace File v. 1.0	Windows	SI		SI		\$295
Actus, Inc.	4th. Dimension 2.2	DOS	SI	4D	SI	DAL, BY	\$795
Blyth Software	Omnis Even v. 1.0	DOS, Windows	SI	Omnis SQL	SI	DAL, OR, MS	\$1250 a \$2500
Borland International, Inc.	dBASE IV v. 1.1	DOS, Windows, UNIX, VAX	SI	dBASE	SI	MS, BY	\$795 a \$1295
Borland International, Inc.	Paradox v. 3.5	DOS	SI	PAL	SI	EE, OR	\$795
ButtonWare, Inc.	PC-File v. 6.0	DOS	SI		SI		\$150
Condor DBMS Services, Inc.	Condor3 v. 2.30	DOS	No	Condor	SI		\$395
Cyclic Software, Inc.	MultiBase v. 4.3	DOS	SI	Xbase	No		\$499 a \$995
Data Access Corp.	DataFlex v. 3.0	DOS	SI	DataFlex	SI		\$795 a \$1450
DataEase International, Inc.	DataEase v. 4.5	DOS, OS	SI	DGL	SI	EE, OR, MS, BY	\$795
Dorsak Software	Force v. 2.1E	DOS	SI	Xbase	No		\$199
Evolution Trading, Inc.	Anthora v.1.0	Windows	SI	Anthora	SI		\$295
FFE Software, Inc.	FirstSQL v. 2.0	DOS	SI	FirstSQL, SQL	SI		\$495 a \$995
FFE Software, Inc.	FirstSQL C	DOS	SI	C	No		\$295 a \$699
Fox Software, Inc.	FoxPro v. 2.0	DOS	SI	SQL, Xbase	SI		\$795 a \$995
Fox Software, Inc.	FoxBASE+ v. 2.10	DOS	SI	Xbase	SI		\$395 a \$700
Gupta Technologies, Inc.	SQL Windows 3.0	Windows	SI	SQL	SI	EE, G, N, OR, MS	\$1595 a \$2495
IBM	OS/2 Extended Edition Database Manager	OS/2	SI		SI	EE	\$400
Informix Software, Inc.	Informix-SQL	DOS, OS/2, UNIX	SI	Informix, SQL	SI	INF	\$1200
Matsyco Corp.	ObjectView v. 1.05	DOS, OS/2, Windows	SI	BASIC, C	SI	OR, MS, BY	\$899
mdbe, Inc	MDBS v. 4.3	DOS, OS/2, UNIX, VAX	SI		SI		\$995
Microform	R:BASE v. 3.1	DOS, OS/2	SI	SQL, R:BASE	SI	OR	\$795 a \$995
Nantucket Corp.	Clipper v. 5.01	DOS	SI	Clipper, Xbase	No		\$795
Novell, Inc.	Btrieve v. 5.10	DOS, Novell, OS/2, Windows	SI		No		\$595
Odessa Corp.	Double Helix v 3.5	MS SQL	SI		SI		\$995
Oracle Corp.	ORACLE v. 6.0	DOS, MS SQL, OS/2, UNIX, VAX	SI	Oracle, SQL	SI	OR	\$799 y más

Figura 2.11 Tabla comparativa de características de DBMS.

## Archivos hijos vs. Relacionales

Si las necesidades de almacenamiento y procesamiento de datos son sencillas, es recomendable adquirir un manejador de archivo fijos, ya que por lo regular son amigables y fáciles de usar (*PC-FILE, Q&A*), el *Hardware* que requieren es modesto (modelos *XT*), pero también se encuentran muy limitados en sus aplicaciones.

Es importante analizar el futuro de la empresa, ya que si posteriormente se requiere un manejador más poderoso, será necesario realizar la transferencia de información. Para facilitar dicha transferencia, algunos manejadores de archivos fijos, por lo regular, cuentan con un hermano mayor, como son *Paradox Special Edition* y *Personal R:BASE*.

Los manejadores relacionales son más flexibles, cuentan con lenguajes de programación, permiten la creación de reportes sofisticados, etc. Debe aclararse que los llamados manejadores relacionales para computadoras personales no cumplen por lo regular con todas las características, más bien son manejadores de archivos ligados.

### **Multi-usuario vs. Usuario-sencillo**

Por *Multi-usuario* se entiende que más de un usuario puede acceder al *DBMS* para utilizar un mismo archivo al mismo tiempo, dentro de una red.

Actualmente muchos manejadores cuentan con ambas opciones, además de que existen muy pocas diferencias de operación desde el punto de vista del usuario, la mayor parte de las variantes de operación le son transparentes. En donde si se presentan diferencias muy marcadas es cuando se tienen que crear programas de aplicación. Los manejadores multi-usuario son funcionales sólo cuando se cuenta con una red.

### **Capacidad de Programación**

Los *DBMS* se pueden clasificar en tres tipos:

- 1) Solo lenguaje de programación (*Clipper, Quicksilver*)
- 2) Lenguaje de programación y modo interactivo (*dBASE, Paradox*)
- 3) No programables (*Alpha Four, Magic*)

Por lo regular, los manejadores que sólo cuentan con lenguaje de programación, poseen una mayor cantidad de instrucciones y permiten crear programas ejecutables desde sistema operativo. En el caso de *Clipper*, permite programación orientada a objetos y enlace con otros lenguajes de programación como *C* y *Ensamblador*.

Los manejadores de propósito general cuentan con lenguajes de programación más limitados, pero a cambio le permiten al usuario realizar operaciones sin necesidad de programar. Por lo regular estos manejadores presentan varios niveles de complejidad y trabajan bajo varias plataformas.

Los manejadores no programables se utilizan por lo regular para realizar aplicaciones sencillas a través de un limitado juego de procedimientos.

### **Soporte Técnico**

El soporte técnico consiste en la disponibilidad del distribuidor para atender las necesidades del cliente. Algunas empresas cobran cargos adicionales por prestar este servicio y en algunas ocasiones el tiempo de respuesta es muy grande.

### **Seguridad de datos**

La seguridad de la información almacenada es de especial importancia. Existen algunos *DBMS* que permiten regresar los datos a su estado original (*rollback*) cuando algún proceso en especial presenta problemas. Un ejemplo de este tipo de manejador es *FoxPro/LAN*.

### **Soporte para SQL(Structured Query Language)**

Por lo regular los manejadores que utilizan *SQL* corren en *minicomputadoras* o *main frames*. Algunos manejadores para *PC* que permiten uso de *SQL* son *Oracle*, *dBASE IV* y *Fox-Pro 2.0*, pero lo hacen de manera muy limitada. Antes de decidir utilizar *SQL* es importante investigar cuales son los beneficios que ofrece.

### **Consideraciones especiales**

Además de todo lo mencionado en los párrafos anteriores, es importante comentar algunas otras herramientas que pueden ser importantes, como son: la disponibilidad de utilizar imágenes, generar sonidos, interfaz con digitalizadores y dispositivos de medición, velocidad de procesamiento, etc.

### **Advanced DB Master 4.1 (ESP Computer Systems)**

Es un *DBMS* multi-usuario, su ambiente de trabajo se basa en el manejo de menús. Permite la creación de menús definidos por el usuario, varios niveles de *password*, protección para campos y funciones de programas, longitud variable de campos, diccionario de datos, etc. Se encuentra disponible en formato usuario sencillo y multi-usuario, requiere una versión 2.1 o superior de *MS-DOS*. Es compatible con diferentes tipos de redes, como son: *Novell*, *3Com*, *PC-Network*, *Star-Lan* y *Lantastic*. La versión para usuario sencillo cuesta \$695 y la versión para multi-usuario tiene un precio de lista de \$1,195 U.S.D.

### **Alpha Four 1.1 (Alpha Software)**

Es un *DBMS* diseñado para no-programadores. Su interfaz con el usuario a través de menús permite crear aplicaciones de base de datos en equipos *IBM PC* y compatibles. Los usuarios pueden definir campos y crear formas en archivos múltiples, puede trabajar con hasta 10 bases de datos simultáneamente. Su precio de lista es de \$595 *U.S.D.*

### **Clipper 5.01 (Nantucket Corporation)**

Es un *DBMS* para equipos *IBM PC* y compatibles. Permite la creación de arreglos multidimensionales, incluye el *RTLink* que permite depurar el tamaño de los archivos ejecutables, permite la lectura de código de otros programas, se pueden escribir varios comandos en una sola línea de código, incluye características limitadas de programación orientada a objetos. Su precio de lista es de \$795 *U.S.D.*

### **DataEase 4.2 (DataEase Inc.)**

Es un *DBMS* que opera en modo usuario sencillo o multi-usuario. Cuenta con el lenguaje de programación *DQL* de cuarta generación, utiliza técnicas de programación orientada a objetos, provee de acceso a redes *LAN*. Corre bajo sistema operativo *UNIX*. Tiene un precio de lista de \$895 para usuario sencillo, y de \$1,195 *U.S.D.* para versión multi-usuario.

### **DataFlex 3.0 (Data Access)**

Es un *DBMS* que utiliza lenguaje de programación de cuarta generación, acepta el uso de *mouse*, cuenta con barras de desplazamiento y botones en pantalla. Se puede correr bajo *MS-DOS*, *OS/2*, *UNIX* y *VMS*. Su precio de lista varía entre \$695 y \$9,000 *U.S.D.*

### **dBASE IV 1.1 (Borland Inc.)**

*dBASE IV* está diseñado para incrementar la productividad tanto de usuarios finales como de desarrolladores. El centro de control con que cuenta permite la utilización de datos sin necesidad de programar, además permite la creación de pantallas de captura, formatos de reporte y etiquetas postales, posee un generador de aplicaciones. La versión *IV* puede ser utilizada con el lenguaje estándar o con *SQL*. Además cuenta con un manejador de memoria dinámica y *password* de protección. Soporta redes *Novell*, *Net-BIOS*, *3Com* e *IBM PC LAN*. La edición estándar tiene un precio de lista de \$795 y la edición para desarrolladores cuesta \$1,295 *U.S.D.*

### **dbFAST Windows 1.50 (Genesoft)**

*dbFast/PLUS* y *dbFast/Windows* cuentan con compiladores para crear aplicaciones. *dbFast/PLUS* permite transportar código entre *DOS*, *MS-Windows* y *Mac*. Ofrece ambiente de trabajo gráfico, menús, ventanas, botones, cajas de diálogo, total uso del *mouse*. *dbFast/Windows* permite la integración de imágenes gráficas. *dbFast/Plus* corre en equipos *XT*, *AT* o compatibles, requiere disco duro con 4 MB, 640 KB RAM, *MS-DOS* ver. 3.0 en adelante. *dbFast/Windows* requiere además de *MS Windows 2.1* o superior.

### **FoxPro 2.0 (Fox Software)**

Es un *DBMS* relacional que corre bajo sistema operativo *MS-DOS*. Se caracteriza por correr a gran velocidad, cuenta con lenguaje *SQL*. Cuenta con lenguaje de programación de cuarta generación y programación orientada a objetos, permite acceso a rutinas en lenguaje *C* y *ensamblador*. Los índices que utiliza son rápidos y pequeños, da oportunidad de manejar memoria virtual. La versión para usuario sencillo tiene un precio de \$795 y la versión para red cuesta \$1295 U.S.D.

### **Oracle Tools and Database 6.0 (Oracle)**

Oracle es un *DBMS* que permite desarrollar aplicaciones y ser conectadas en diferentes ambientes operativos. Incluye lenguaje de programación de cuarta generación y *SQL*. En computadoras personales requiere 8 MB de RAM, ocupa más de 20 MB de disco duro, no genera programas ejecutables y tiene un costo de \$1,500 U.S.D.

### **Informix (Informix Software, Inc.)**

El *DBMS Informix* es una colección de programas de *Software* que permiten al usuario crear un sistema completo de base de datos, a menudo sin alguna programación de la acostumbrada. Cuando esta programación es requerida, generalmente puede ser hecha con lenguajes de escritura de reportes, que la hacen rápida y sencilla. Aún, si estas herramientas no son bastante flexibles, se dispone de interfaces entre los lenguajes de programación convencional y las rutinas de almacenamiento y recuperación de *Informix*. Las principales utilerías de *Informix* para captura y recuperación de datos son *Perform*, *Acego* y el *Informer*. El precio de lista es de \$1,200.00 U.S.D.

## **CAPITULO 3**

# **ANALISIS DE HARDWARE Y SOFTWARE**

## 3.1 ANALISIS DE ELEMENTOS DEL SISTEMA

Haciendo un análisis de la información recopilada en el capítulo 2 y con base en las características y alcances del sistema que se pretende desarrollar, en el presente capítulo se seleccionará el medio a través del cual se realizará la comunicación y transferencia de información entre las diferentes sucursales y la oficina central. A continuación se elegirá el equipo de cómputo que realice el procesamiento de datos, y por último se seleccionará el *Software (DBMS)* que facilite el desarrollo del sistema de inventarios, que sea compatible con el equipo de cómputo y que se ajuste al medio de comunicación.

Además de los aspectos técnicos se considerarán los económicos, de manera que la selección se ajuste a las posibilidades reales de la empresa.

### 3.1.1 ALCANCES DEL SISTEMA

Se planea interconectar las sucursales de "**Comercial Emyco S.A de C.V.**" distribuidas en el país a la central que se encuentra en el D.F., sólo para transferencias de archivos. No habrá interconexión entre sucursales, sino de sucursal a central únicamente, es decir, se empleará una topología de red en estrella, la cual se implantará usando alguno de los servicios de telecomunicaciones disponibles.

El control de inventarios se realizará localmente en cada una de las sucursales, liberando de carga de trabajo a la central. La central sólo se encargará de recibir datos, procesarlos y enviar información mínima a las sucursales (cambios de precios, cambios de modelo-color-talla, comunicados, ordenes de resurtido, entre otros).

Un punto importante a considerar es la capacidad de procesamiento y de almacenamiento en la computadora central. Esta computadora podría ser el servidor de archivos (*server*) de una LAN que proporcionara información a las gerencias de la empresa (Gerencia General, Gerencia Comercial, Gerencia de Administración y Gerencia de Recursos Humanos).

Debe aclararse que la finalidad de este trabajo es lograr el control de inventarios y la transmisión de información entre las sucursales y la central, por lo que el análisis e implantación de una red local queda fuera de sus alcances.

### 3.1.2 ANALISIS DEL SERVICIO DE TELECOMUNICACION

Para decidir el tipo de telecomunicación a emplear en el diseño de la red de computadoras para "**Comercial Emyco S.A de C.V.**" se considerarán varios aspectos:

- 1) Geografía
- 2) Confiabilidad
- 3) Tiempo de utilización del servicio y volumen de información
- 4) Velocidad
- 5) Conexiones externas y expectativas a futuro
- 6) Costos de instalación y tarifas

Se analizarán estos aspectos para cada uno de los servicios de telecomunicaciones disponibles en México (ya descritos en el capítulo 2):

- 1) *Red Conmutada*
- 2) *Telepac*
- 3) *RDI*
- 4) *VSAT*

Este análisis servirá para determinar cuál es el más apropiado a las necesidades y características de la empresa.

### **Geografía**

Las sucursales con que cuenta la empresa se sitúan en las ciudades listadas a continuación:

- |                |                 |                  |
|----------------|-----------------|------------------|
| 1. Guadalajara | 5. México, D.F. | 9. Toluca        |
| 2. Guanajuato  | 6. Monterrey    | 10. Veracruz     |
| 3. Irapuato    | 7. Puebla       | 11. Villahermosa |
| 4. León        | 8. Querétaro    |                  |

La *Red Conmutada* cubre todas las ciudades, ya que a final de cuentas sólo es necesario contar con una línea telefónica, un *modem* y un teléfono.

*Telepac* no cubre la ciudad de Irapuato. Esta deficiencia se puede remediar haciendo la transmisión de datos de la ciudad que no cuenta con el servicio

hacia un nodo cercano, que sí cuenta con dicho servicio, a través de un enlace telefónico o por microondas.

*RDI* solo cubre la Ciudad de México, Monterrey y Guadalajara, pero se cuenta con un sistema satelital que le permite cubrir las demás zonas donde se encuentran las sucursales.

*VSAT* puede ser instalada en cualquier punto de la república mexicana, ya que la recepción de la señal se realiza mediante una antena.

### **Confiabilidad**

Basicamente, la información que se transmite son movimientos de ventas, devoluciones, embarques, transferencias y memorandums. Se requiere que esta transmisión garantice la integridad y seguridad de los datos.

La *Red Conmutada* sólo permite transmisiones confiables a muy bajas velocidades, además, conforme aumenta la distancia, la probabilidad de que la información que se transmite sufra interferencias es mayor.

*Telepac* puede ser accesada a través de líneas privadas ó líneas conmutadas. La confiabilidad de la transmisión está en función del medio de comunicación que se esté utilizando. Las líneas privadas son más recomendables.

*RDI* realiza la comunicación de datos a través de fibra óptica, que es un material inmune al ruido e interferencia.

*VSAT* realiza la comunicación de datos vía satélite, brindando una confiabilidad muy alta.

### **Tiempo de utilización del servicio y volumen de información**

Se planea que la transferencia de información sea durante tiempos cortos y de manera esporádica a lo largo de un día. Los volúmenes que serán transmitidos son pequeños (aproximadamente 1 MB por día), esto significa que no es necesario contar con un servicio dedicado, ya que sería aprovechado en un porcentaje muy bajo.

La *Red Conmutada* es muy práctica en este aspecto, ya que el cobro se realiza en base al tiempo de utilización del servicio. La tarifa está en relación directa con la distancia de transmisión. Es recomendable solamente para volúmenes pequeños de información.

*Telepac* realiza la transmisión de datos mediante la técnica de conmutación de paquetes. El costo del servicio está en función del volumen de transmisión.

**RDI** se caracteriza por utilizar fibra óptica como medio de comunicación. La cantidad de información transmitida es independiente del precio del servicio, ya que se cobra una renta fija. Este medio de comunicación es muy recomendable para grandes volúmenes de información, dada su excelente calidad.

**VSAT** realiza la transmisión de información mediante la técnica de conmutación de paquetes vía satélite. El costo está en función del volumen de información transmitido y de la distancia.

### **Velocidad**

Las necesidades de la empresa no requieren de un tiempo de respuesta inmediato, ya que no se realizarán operaciones en línea y los volúmenes de transmisión son muy pequeños. Es por esto que este aspecto no es muy relevante para decidir cuál servicio se va a utilizar. La figura 3.1 muestra la comparación de velocidad de transmisión entre los diferentes servicios:

Servicio	Velocidad de Transmisión
Red Conmutada	hasta 9.6 Kbps (varía según el <i>modem</i> )
Telepac	hasta 2.4 Kbps
RDI	hasta 2.04 Mbps
VSAT	hasta 64 Kbps

*Figura 3.1 Comparación de velocidades de transmisión.*

### **Conexiones externas y expectativas a futuro**

Debe considerarse la posibilidad de crecimiento de la red debido al surgimiento de nuevas sucursales. El servicio deberá ser capaz de cubrir el lugar de ubicación de éstas. La política actual de la empresa es ubicar sus tiendas en centros comerciales que cuentan con los servicios de comunicación básicos (teléfono).

Se espera que la *Red Conmutada* en el futuro (1998), utilice fibra óptica como medio de comunicación, lo cual incrementará la calidad y velocidad de transmisión. Se espera que la cantidad de usuarios aumente.

*Telepac* tendrá que sufrir cambios en su infraestructura debido a la integración de la *Red Conmutada* vía fibra óptica. El futuro de *Telepac* es realmente incierto ya que el número de usuarios muy probablemente se reducirá. Principalmente en el área metropolitana se presentan problemas de acceso a la red debido a sobrecarga de usuarios.

*RDI* tiene buenas expectativas en el futuro, ya que con el incremento de las conexiones vía fibra óptica, se espera que el servicio cubra todo el país y su costo no sea tan elevado.

Se espera que *VSAT* sea el medio de comunicación más utilizado, debido a su alta tecnología y al rápido incremento en la cantidad de satélites disponibles mundialmente.

### Costos de instalación y tarifas

La figura 3.2 muestra un resumen de los costos aproximados de instalación y tarifas por uso de una línea para los distintos servicios de comunicación de datos:

Servicio	Por instalación	Tarifa Mensual	Por uso
Red Conmutada	\$ 2,000,000	\$ 20,000	variable
Telepac	\$ 220,000	\$ 72,000	\$ 1,720 (ksegmento)
RDI (por línea)	\$ 2,542,000	\$ 55,800	variable
VSAT	\$ 6,200,000	\$ 988,900	variable

Figura 3.2 Tabla comparativa de precios por servicio de comunicación.

Es importante mencionar que para el uso de *VSAT* es necesario contar con una estación terrena tipo *VSAT SKYLINX.25* para cada una de las sucursales. Para *RDI* es necesario adquirir paquetes entre 200 y 250 líneas, lo que ocasionaría un gran desperdicio de líneas. Para *Telepac* es necesario contar con un *modem* y un teléfono en cada sucursal.

En la figura 3.3 se muestra un resumen de evaluación de las características de los servicios de comunicación de datos que se han considerado:

Servicio / Característica	Red Conmutada	Telepac	RDI	VSAT
Geografía	SI	SI	No	SI
Confiabilidad	SI*	SI	SI	SI
Tiempo de utilización y volumen de información	SI	SI	No	No
Velocidad	SI	SI	SI	SI
Conexiones externas y expectativas a futuro	SI	SI	SI	SI
Costos de instalación y mantenimiento	SI	SI	No	No

Figura 3.3 Evaluación de las características de los servicios de comunicación.

\* Solo es confiable a bajas velocidades (2400 bps)

La solución óptima en cuanto a velocidad de transmisión es el uso de la *RDI* ó *VSAT*, pero como se mencionó anteriormente, este aspecto no es el de mayor prioridad, además los costos de instalación y tarifas son muy elevados en relación a la *Red Conmutada* y *Telepac*. Estos costos sólo serían justificados si se necesitara un procesamiento en línea para grandes volúmenes de información, como sucede en bancos, casas de bolsa, telereservaciones, etc.

Como puede observarse en la tabla de la figura 3.3, los servicios que se ajustan más a los requerimientos del sistema son la *Red Pública de Transmisión de Datos (Telepac)* y la *Red Conmutada*.

Se plantea la utilización de la *Red Conmutada* ya que haciendo una comparación de precios entre esta y *Telepac*, resulta ser más económica. Por ejemplo si se desea realizar una transmisión de 30 kilosegmentos entre la Ciudad de México y Villahermosa, se requiere una comunicación de aproximadamente 7 minutos. Considerando la renta mensual más el costo de comunicación, por *Telepac* el costo sería de aproximadamente \$55,000 y por *Red Conmutada* sería de \$13,260. La velocidad máxima de transmisión para *Telepac* mediante línea telefónica es de 1,200 kbps, en cambio la *Red Conmutada* permite una transmisión confiable a 2,400 Kbps.

En el aspecto de confiabilidad, la *Red Conmutada* no es la más favorable, pero este problema puede remediarse a través del uso de algún paquete de comunicación de datos comercial (*Blast, Crosstalk, Mirror*), ya que estos cuentan con mecanismos de comprobación de datos y de recuperación en caso de interrupciones en la comunicación.

### 3.1.3 EQUIPO CENTRAL

El equipo central se encarga de recibir diariamente la información referente a los movimientos realizados en todas las sucursales. Actualmente el equipo central ha presentado problemas en cuanto a velocidad de procesamiento y a capacidad de almacenamiento del disco duro. Se espera que a futuro la cantidad de sucursales se vea incrementada, lo que provocará un aumento en la carga de trabajo.

Existen tres clases de equipos de cómputo en el mercado:

- 1) *Equipo Main Frame*. Es el equipo más grande, permite un gran número de usuarios conectados a una sola computadora a través de terminales. Sus costos son muy elevados y por lo regular son utilizadas en empresas o instituciones educativas muy grandes, las cuales necesitan compartir su información. Estos equipos requieren

de un espacio grande y un medio ambiente especial, como es control de temperatura y humedad entre otros.

- 2) Equipo *Minicomputadora*. Es un equipo que posee las mismas características que un *Main Frame*, solo que la velocidad, la capacidad de almacenamiento y la cantidad de terminales es menor. No requiere de un ambiente especial de operación y su espacio es mínimo. Es una buena opción para empresas medianas.
- 3) Equipo *Microcomputadora*. Con el avance de la tecnología se han desarrollado equipos pequeños cada vez más poderosos que en sus inicios eran enfocados al desarrollo de actividades a nivel personal. Hoy en día estos equipos cuentan con suficiente capacidad para compararse en algunas de sus funciones con equipos mayores, como es la velocidad de procesamiento, la capacidad de almacenamiento, la posibilidad de comunicación en red con otras computadoras. Una característica importante de estos equipos es su bajo costo y su fácil manejo. Es una buena opción para implantación de *Sistemas Distribuidos*.

Con base en las necesidades actuales y a futuro de la empresa, el equipo central con que se cuenta actualmente resulta ser obsoleto, por lo que se planea adquirir un nuevo equipo de cómputo que cumpla con los requerimientos de desarrollo y presupuesto de la empresa.

El sistema a desarrollar es un *Sistema Distribuido*, ya que los procesamientos de información se pueden realizar de manera independiente entre las sucursales y la oficina central. El equipo central se encargará de recibir y procesar la información de las sucursales para posteriormente generar reportes que permitan un mejor control de los inventarios de las tiendas. Debido a que se planea incrementar la cantidad de tiendas y en base a que el equipo actual es ya cubierto por una *Microcomputadora*, se puede decir que la mejor opción, tanto en precio como en diseño es una *PC*. Analizando las expectativas a futuro, el equipo *PC* más apropiado para ser utilizado como equipo central es una *PC/486* dadas sus características. Una *PC/486* permite opciones de expansión mayores a cualquier otra *PC*, además permite multi-tareas y manejo de memoria virtual. Si en el futuro se pretendiera crear una *LAN* en la oficina central, el equipo *PC/486* puede operar como el *server*.

Es importante que el equipo de cómputo cuente con un coprocesador matemático para agilizar los procesos de manejo de bases de datos. Pensando a futuro, si el equipo es utilizado como *server*, será necesario un mínimo de 4 *MB* de *RAM*. El disco duro de más de 300 *MB* permite un amplio crecimiento en el almacenamiento de información de las tiendas. Las unidades de disco flexible

basicamente se utilizan para transporte de información y la cinta magnética será utilizada como un respaldo de información.

A continuación se listan las características del equipo *PC/486* a utilizar:

- 1) Microprocesador: *80486*
- 2) Velocidad del Microprocesador: *25 Mhz* en adelante
- 3) Coprocesador Matemático: *80487*
- 4) Memoria *RAM*: *4 MB* mínimo, con capacidad de expansión
- 5) Disco Duro: *330 MB*
- 6) Discos Flexibles: Una unidad de *5.25"* y una de *3.5"* de alta densidad
- 7) Video: Despliegue *VGA*
- 8) Puertos: 2 puertos seriales y un puerto paralelo
- 9) Unidad de Cinta Magnética: *330 MB*

En la figura 3.4 se listan algunas modelos de computadoras existentes en el mercado mexicano que se ajustan a los requerimientos antes mencionados:

Marca	Modelo	Veloc. (Mhz)	RAM (MB)	Precio (U.S.D.)
IBM	PS/2 mod. 90 XP	33	8 exp a 64	\$ 13,170.00
Hewlett Packard	VECTRA 486/33T	33	4 exp a 64	\$ 13,355.00
Acer	Acer 1173E	33	8 exp a 26	\$ 6,800.00
AST	Premium 486/33/TE	33	8 exp a 16	\$ 12,680.00
SAMSUNG	SD 930	33	4 exp a 32	\$ 6,105.00
COMPAQ	COMPAQ DESKPRO 486	33	8 exp a 64	\$ 11,000.00

*Figura 3.4 Equipos 486 disponibles en México.*

Con base en la tabla anterior, y conociendo la calidad y soporte de los equipos, se considera que la mejor opción es la computadora *Acer 1173E* ya que combina bajo costo y buena calidad pese a que *IBM, HP* y *Compaq* son de mejor calidad.

### 3.1.4 EQUIPO DE SUCURSALES

Las sucursales realizarán por separado el procesamiento de los movimientos de cada día, por lo que no es necesario utilizar equipos de cómputo tan sofisticados para dichas tareas. Los movimientos principales son la captura de

ventas, devoluciones al proveedor y al cliente, transferencias de información y embarques.

Una computadora modelo *PC XT* puede realizar dichos procesos, pero no es conveniente ya que se encuentran descontinuadas, por lo tanto se seleccionarán equipos *PC AT/286*, los cuales brindan una mayor velocidad y mayor capacidad de almacenamiento a bajo costo.

A continuación se muestra una configuración básica de equipos *PC AT/286*:

- 1) Microprocesador: 80286
- 2) Velocidad del Microprocesador: 12 Mhz en adelante
- 3) Memoria RAM: 1 MB mínimo, con capacidad de expansión
- 4) Disco Duro: 40 MB
- 5) Discos Flexibles: Una unidad de 5.25" de alta densidad
- 6) Video: Monocromático de alta resolución
- 7) Puertos: 2 puertos seriales y un puerto paralelo

En la figura 3.5 se listan algunas modelos de computadoras existentes en el mercado mexicano que se ajustan a los requerimientos antes mencionados:

Marca	Modelo	Veloc. (Mhz)	RAM (MB)	Precio (U.S.D.)
Hewlett Packard	VECTRA 286/12	12	1 exp a 4	\$ 2,970.00
Acer	915V-II	16	1 exp a 5	\$ 1,379.00
AST	Bravo/286-16	16	1 exp a 16	\$ 1,664.00
SAMSUNG	SD 620	12	1 exp a 2	\$ 1,320.00
AUVA	AUVA 230	16	1 exp a 4	\$ 1,150.00

*Figura 3.5 Equipos 286 disponibles en México.*

Analizando la tabla anterior se observa que el equipo *Acer 915V-II* es económico, además de ser confiable.

### 3.1.5 ANALISIS DEL SOFTWARE DE DESARROLLO

Para seleccionar el producto de *Software* de desarrollo, tres aspectos son importantes: La flexibilidad del lenguaje de programación, la velocidad de ejecución y el costo.

Algunos productos como *dBASE IV*, *FoxPro* y *Oracle* no crean programas ejecutables desde la línea de comandos del sistema operativo. Estos ofrecen un compilador que crea archivos objeto, que aumentan la velocidad de ejecución de los programas al eliminar el proceso de interpretación de instrucciones, pero que siguen siendo dependientes del producto para su ejecución, ya sea a través de la interfaz con el usuario o de un módulo *run-time*. Un punto importante a considerar es que para cada aplicación distribuida debe adquirirse una copia del producto.

En el lado opuesto se encuentran los compiladores para aplicaciones de bases de datos. Utilizando estos productos (*Force*, *Clipper*) pueden crearse programas ejecutables desde la línea de comandos del sistema operativo, sin la necesidad de *Software* adicional. Por lo general contienen un lenguaje de programación potente que puede adaptarse con mayor facilidad a las necesidades del programador.

Para el sistema de inventarios que se desarrollará, debido a que se planea un proceso distribuido, cada sucursal tendrá una copia de la aplicación. Si se desarrollara con un *DBMS* con interfaz con el usuario, debería tenerse además de la aplicación una copia del *DBMS* en cada sucursal. Teniendo en cuenta lo anterior, lo más apropiado es utilizar como herramienta de desarrollo un *DBMS* que genere programas ejecutables, ya que una aplicación creada de esta forma no requiere tener al *DBMS* en la instalación donde se use la aplicación, ni es necesario pagar alguna cuota extra por licencia de uso.

Como punto de comparación puede mencionarse que cada copia de *dBASE IV* cuesta aproximadamente lo mismo que el compilador *Clipper 5.01* de *Nantucket Corp.*

Dentro de los *DBMS*, el que ofrece mayor flexibilidad en la programación es *Clipper 5.01*, ya que cuenta con características muy interesantes como son: la compilación condicional de código, inclusión de archivos fuente dentro de un archivo en el tiempo de compilación, uso avanzado de atributos de variables, y en general, características de los lenguajes de programación de alto nivel como *C* y *Pascal*.

De acuerdo a las necesidades del sistema a desarrollar y considerando las características que *Clipper* ofrece, resulta ser la mejor opción para la implementación del mismo.

Existe una gran cantidad de bibliotecas de funciones para *Clipper*, destinadas a salvar sus deficiencias o para aumentar su poder. Así, existen bibliotecas para comunicaciones, uso de gráficos y *Programación Orientada a Objetos (OOP)*, entre otras.

### 3.1.6 CARACTERÍSTICAS DE CLIPPER 5.01

*Asthon-Tate* creó más de un programa de bases de datos cuando produjo *dBASE II*, *dBASE III* y *dBASE III PLUS*. Ahora, cuando se menciona *dBASE*, nos referimos al lenguaje llamado *dBASE*, no sólo al producto de *Software*.

*Nantucket Corporation* ha lanzado al mercado desde 1985 un compilador (*Clipper Winter85*) para el lenguaje *dBASE*. Actualmente *Nantucket* ha tomado un rumbo propio para la creación de su propio estándar para aplicaciones de bases de datos.

*Clipper* es un medio para desarrolladores que usa como su conjunto de comandos estándar una extensión del lenguaje *dBASE III Plus*. No hay un *prompt* o centro de control que provea una interfaz interactiva con el usuario, como puede encontrarse en *dBASE III Plus*, *dBASE IV* o *FoxPro*.

En cambio, *Clipper* se concentra en la tarea de crear aplicaciones de manejo de datos que pueden ser distribuidas como programas ejecutables (.EXE). Los usuarios no necesitan aprender nada acerca de *dBASE*. Todo lo que necesitan saber es cómo usar su programa, lo que puede mejorar grandemente su productividad.

Los comandos y funciones de *Clipper* son un superconjunto del lenguaje *dBASE III Plus*. Este lenguaje mismo es parte de la arquitectura abierta de *Clipper*. *Clipper* fue el primer lenguaje *dBASE* en ofrecer funciones definidas por el usuario y ahora, en la versión 5.0 (5.01) es el primero en ofrecer comandos definidos por el usuario.

Además, el sistema extendido de *Clipper* (*Clipper Extended System*) ha sido mejorado para permitir acceso más fácil a rutinas y funciones creadas en lenguaje *C* y *ensamblador*. Actualmente existen en el mercado una gran variedad de paquetes de biblioteca, creados usando el sistema extendido, que dan a *Clipper* una potencia adicional en manejo de archivos, impresoras, *mouse*, pantalla, comunicaciones, etc.

En adición a estas características, *Nantucket* ha incluido un preprocesador para el compilador accesible al usuario, un enlazador que hace uso óptimo de la memoria y un *debugger* sofisticado.

#### Especificaciones Técnicas

El ambiente de *Clipper* es similar en muchas formas al del *dBASE III Plus*, pero muchas de las características del *dBASE III Plus* han sido expandidas en *Clipper*.

A continuación se muestra una lista de las especificaciones técnicas de *Clipper*.

1) Archivos de datos:

Máximo mil millones de registros o el espacio disponible en disco

Compatibilidad total con *dBASE III Plus*

En adición, *Clipper* puede usar *drivers* de bases de datos alternos para usar datos en otros formatos

2) Registros:

Longitud de campo fija

Tamaño limitado sólo por *RAM*

El número máximo de campos es 1024

3) Tipos de campos:

Carácter: máximo 64 KB

Numérico: máximo 19 bytes

Fecha: 8 bytes

Lógico: 1 byte

Memo: máximo 64 KB

## Indices

Se tienen disponibles dos tipos de indexado. Se pueden usar los archivos de índices estándar de *dBASE III Plus* (.NDX), si los datos usados por la aplicación no exceden el tamaño y límites del *dBASE III Plus* y se ha enlazado el *driver* de *dBASE III Plus* a la aplicación. El segundo tipo, cuando no se considera la compatibilidad, el archivo de índices de árbol B+ de *Clipper* (.NTX), ofrece ejecución más rápida. Los archivos .NTX de *Clipper* pueden tener expresiones llave de hasta 250 bytes. A diferencia de los archivos de índices .NDX, los .NTX pueden contener campos de fecha en las expresiones de llave, sin conversión de tipo previa.

## Variables de memoria

*Clipper* puede manejar hasta 1024 variables de memoria públicas y privadas y los arreglos soportan los siguientes tipos de datos:

1) Carácter: máximo 64 KB

- 2) Numérico: máximo 19 bytes
- 3) Fecha: 8 bytes
- 4) Lógico: 1 byte
- 5) NIL: Es un tipo de datos que indica la ausencia de un valor. NIL puede ser usado sólo por variables y elementos de arreglo
- 6) Bloques de código sin tamaño: Los bloques de código son una manera ingeniosa de hacer una variable dentro de una pieza pequeña de código ejecutable
- 7) Memo: Las variables memo no son específicamente soportadas, pero la función *memoedit()* hace a las variables memo y carácter casi intercambiables en uso, excepto que los campos *memo* son almacenados en un archivo *.DBT* en segmentos de 512 bytes y las variables carácter son de longitud fija

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

## Arreglos

*Clipper* puede crear y usar arreglos multidimensionales. Los arreglos cuentan como una variable de memoria. Los arreglos multidimensionales son creados anidando arreglos en arreglos.

Cada arreglo puede consistir de 4,096 elementos. Los elementos son independientes en tipo, y cada elemento sigue las mismas reglas que las variables de memoria. Los elementos *memo* no son soportados, pero los elementos *carácter* pueden ser de hasta 64 KB.

## Archivos

Bajo versiones *DOS 3.3* o posteriores, *Clipper* puede abrir hasta 250 archivos de todos los tipos. A continuación se listan los tipos y extensiones por defecto:

- |    |                       |             |
|----|-----------------------|-------------|
| 1) | Bases de datos        | .DBF        |
| 2) | Memos                 | .DBT        |
| 3) | Variables almacenadas | .MEM        |
| 4) | Etiquetas de correo   | .LBL        |
| 5) | Reportes              | .FRM        |
| 6) | Formatos              | .FMT o .PRG |

- |    |                |      |
|----|----------------|------|
| 7) | Programas      | .PRG |
| 8) | Alternos       | .TXT |
| 9) | Procedimientos | .PRG |

### **Áreas de trabajo**

*Clipper* soporta hasta 250 áreas de trabajo, con un archivo de datos abierto en cada una. Soporta hasta 15 archivos de índices abiertos en cada área de trabajo. A cada área de trabajo se le puede dar un *alias* distinto al nombre de archivo de base de datos abierto.

### **Manejo de memoria**

Una forma en la cual *Clipper* logra eficiencia en el manejo de memoria es basándose en un sistema de asignación dinámica de memoria a los programas. En las versiones anteriores, las aplicaciones mayores que la cantidad de *RAM* disponible tenían que construirse usando *overlays* estáticos.

*Clipper* provee un versión especial del enlazador *RTLlink*, de *PocketSoft*, que le da un nuevo esquema de manejo de memoria usando métodos de *overlays* dinámicos. Una aplicación creada con *RTLlink* es automáticamente compuesta de un módulo de carga y un *overlay* sin considerar el tamaño de la aplicación. El módulo de carga es hecho del código de inicialización y lo esencial para tener un programa corriendo. El *overlay*, que está contenido en el archivo *.EXE*, contiene todas las funciones y procedimientos que hacen la aplicación.

Se reserva una porción de *RAM* para traslapar el código *overlay* en memoria. Tanto código de *overlay* como sea posible es colocado en esta área cuando el programa mismo se sitúa en memoria. Si la aplicación completa se ajusta, la aplicación completa es cargada en *RAM*. Cuando las aplicaciones grandes necesitan un módulo (procedimiento o función) que no está ya en memoria, se usa un algoritmo de *el menos recientemente usado* para determinar que módulo en memoria es quitado para que el traslape pueda tener lugar.

El sistema de *overlay dinámico* permite a una aplicación de 500 KB correr en una máquina que tiene sólo 256 KB de *RAM*.

### **Manejo de memoria virtual**

*Clipper* usa memoria convencional y memoria expandida para proveer un esquema de *Manejo de Memoria Virtual (VMM)*, para almacenar variables, arreglos y *buffers* de bases de datos. Cuando una aplicación corre fuera de espacio en memoria convencional para uno de estos elementos, es usada la

memoria expandida. Cuando la memoria expandida esta agotada, se usa un traslape de archivo a disco.

Lo importante de este sistema es que trabaja automáticamente sin la intervención del programador.

### **Objetos de Clipper**

El equipo de desarrollo de *Nantucket* ha usado algunos conceptos orientados a objetos como un punto de partida para *NFT (Nantucket Future Technology)*. *Clipper 5.01* lleva *NFT* al mercado.

*Clipper 5.01* tiene una implantación de principios de *Programación Orientada a Objetos (OOP)*. Cada tipo diferente de objeto es llamado una *clase*. En ambientes reales de *OOP* se pueden crear nuevas clases de objetos que realizan funciones específicas. *Clipper 5.01* está restringido a sólo cuatro clases de objetos: clase *Error*, clase *Get*, clase *TBrowse* y clase *TBColumn*. Cada objeto individual se dice que es una instancia de su clase.

Actualmente existen en el mercado dos productos que extienden a *Clipper* crear y manipular clases y los objetos (o instancias de una clase) creados de ellas. *Class(y)* e *Integrated Development Corp.* y *SuperClass* de *Island Publishing* fueron diseñados para *Clipper 5.01* y se dice serán compatibles con las siguientes versiones de *Clipper*.

## **3.1.7 ANALISIS DEL SOFTWARE DE COMUNICACION**

Un obstáculo para el acceso a la información usando las líneas telefónicas es el ruido de la línea, que puede enmascarar los tonos que acarrear la información entre los *modems*, causando errores. Como los *modems* con corrección de errores son relativamente nuevos, las rutinas de transferencia de archivos en *Software* se desarrollaron primero para proveer control de errores que los *modems*.

La mayoría de los *protocolos* de transferencia de archivos envían la información en paquetes (*packets*) que incluyen alguna forma de detección de errores, a menudo una suma de verificación o una *Prueba de Redundancia Cíclica (CRC)*. Tanto la suma de verificación como la *CRC* son valores que se derivan de la información usando algoritmos matemáticos. El *protocolo* envía el valor junto con los *bits* del paquete. El programa receptor inspecciona la información que se recibe y compara los valores de verificación que recibe con los valores que calcula. Si los valores de verificación no coinciden, el receptor pide que se retransmita el paquete.

La mayoría de los *protocolos* más anticuados requieren que se reciba una confirmación de que los valores de verificación concuerden antes de enviar otro paquete. Algunos de los *protocolos* más nuevos permiten la transmisión en múltiples paquetes antes de recibir confirmación del primer paquete. Este tipo de *protocolo* con *ventanas* es particularmente útil en los circuitos de comunicación con demoras prolongadas tales como los enlaces por satélite. Si un *protocolo* de *ventana deslizante* detecta un error, se retransmite sólo el paquete que tiene error. Los *protocolos* de *corrida libre* transmiten un archivo entero antes de que esperen una confirmación.

### Tipos de protocolos

Un *protocolo* de transferencia de archivos es simplemente un conjunto de reglas que sigue un programa mientras transfiere datos a otra computadora.

El tamaño de un paquete afecta el rendimiento total de la transferencia de archivos. Los paquetes pequeños requieren procesamiento adicional pero se transmiten más rápidamente. Los paquetes mayores logran un mayor rendimiento a menos que experimenten un número apreciable de retransmisiones.

El *protocolo* de transferencia de archivos más simple es la copia de un archivo ASCII hacia el puerto serial. La computadora básicamente "mecanografía" un archivo hacia el puerto serial sin que se detecte ningún error.

Uno de los *protocolos* más ampliamente utilizado es el *XModem*, también conocido como *protocolo* de Christiansen. Cada bloque de *XModem* tiene un tamaño fijo de 128 bytes con un número de bloque y una suma de verificación añadida. El *XModem*, un *protocolo* simple y uno de los primeros de dominio público, ha inspirado un número de variaciones. Estas difieren principalmente en el *protocolo* de corrección de error. La popularidad del *XModem* se deriva de su amplia distribución aunque es uno de los *protocolos* disponibles más lentos.

*XModem 1K* es una versión del *XModem* que usa bloques de 1024 bytes. Muchos programas se refieren al *XModem 1K* como *YModem*. Pero, como se creó originalmente, el *YModem* también añade una habilidad de archivo en lote, de forma que se pueden efectuar transferencias de archivos múltiples en la misma operación. El *YModem-G*, una versión de *corrida libre* del *YModem* puede usarse con *modems* que tienen control de errores, detecta los errores pero no los corrige. Si se encuentra un error usando el *YModem-G* y un *modem* con corrección de errores, es probable que no sea un error de transmisión, sino un error entre el *modem* y la PC.

*Kermit*, creado en la Universidad de Columbia, permite la transferencia de archivos binarios de 7 u 8 bits o ASCII. También es el estándar aceptado para las comunicaciones de *Main Frames*. A pesar de sus muchas variaciones uno de los puntos fuertes de *Kermit* es su habilidad para negociar un nivel común de opciones entre las diferentes versiones. También puede adaptar su tamaño de bloque a la información, pero esto frecuentemente añade más sobrecarga que *XModem*. Dependiendo de la implementación, *Kermit* puede brindar menos rendimiento que *XModem*.

*ZModem* presenta una sobrecarga baja, con velocidad y confiabilidad grandes. El *ZModem* ajusta dinámicamente su tamaño de paquete dependiendo de las condiciones de la línea y una verificación de error tipo CRC de 32 bits muy confiable. Con su opción de recuperación de archivo, si el *ZModem* aborta una transferencia, puede reinicializarla desde el punto en que se abortó en lugar de volver a enviar el archivo entero.

El protocolo *Blast* emplea un diseño avanzado *full duplex* que opera eficientemente en líneas telefónicas ruidosas y en circuitos sujetos a retardo de propagación, tales como ligas de satélite y circuitos virtuales de red. El tamaño del bloque puede ser variado para optimizar la sobrecarga de datos sobre diversas condiciones en la línea. Si la línea es ruidosa los datos serán altamente susceptibles a la corrupción. Cada vez que un bloque o parte de un bloque es corrompido, el bloque debe ser transmitido otra vez. Por ello los pequeños bloques son generalmente preferidos en medios ruidosos, porque los datos perdidos pueden ser retransmitidos cuando ocurre el error. Los bloques pequeños son más probables de llegar libres de error.

*Blast* usa la técnica estándar CRC para la detección de errores en bloques de datos y la técnica de *ventana deslizante*.

### **Lenguajes de guión**

Los lenguajes de guión de los programas de comunicación son la clave para automatizar las tareas repetitivas de comunicación. La mayoría de los programas tienen una opción de aprendizaje que graba el flujo de la información que se recibe y las respuestas que se dan. Luego la opción de aprendizaje genera un archivo que repite las respuestas en los momentos apropiados.

La mayoría de los lenguajes de aprendizaje se inspiran en lenguajes de computación de alto nivel como *Pascal* o *C*. Algunos como el del *Crosstalk Mk.4*, ofrece compiladores que le permiten optimizar el rendimiento de los guiones. Los guiones pueden no solamente conectar al usuario a un servicio, sino que pueden llamar al servicio a una hora determinada, conectarse, cargar archivos y mensajes y colgar, todo esto sin intervención de un operador.

## Programas de comunicación disponibles

Existen muchos programas de propósito general, pero no todos apoyan las últimas técnicas que permite realizar transferencias de información rápidas y exactas.

En la tabla de la figura 3.6 se muestran las principales características de los programas de comunicación.

Marca	Producto	Precio (USD)	Requerimientos
Digital Communications Associates, Inc.	Crosstalk Communicator ver. 2.0	\$ 99.00	320 K de RAM, MS-DOS 2.0 o posterior
Digital Communications Associates, Inc.	Crosstalk for Windows ver. 1.1	\$ 195.00	150 K de RAM, MS-Windows 2.x o posterior
Digital Communications Associates, Inc.	Crosstalk Mk.4 ver. 2.0	\$ 245.00	320 K de RAM, MS-DOS 2.1 o posterior
Future Soft Engineering Inc.	DynComm Asynchronous Edition ver 3.00.08	\$ 295.00	512 K de RAM, MS-Windows 2.x o posterior
SoftKlone	Mirror III ver. 2.0	\$ 149.00	470 K de RAM, MS-DOS 2.0 o posterior
Dataform Technologies Inc.	ProComm Plus ver. 2.0	\$ 119.00	192 K de RAM, MS-DOS 2.0 o posterior
Hayes Microcomputer Products Inc.	Smartcom Exec ver 2.0	\$ 129.00	512 K de RAM, MS-DOS 2.0 o posterior
Communication Research Group	Blast PC Plus	\$ 495.00	320 K de RAM, MS-DOS 2.0 o posterior
Communication Research Group	Blast PC	\$ 445.00	320 K de RAM, MS-DOS 2.0 o posterior

Figura 3.6 Principales programas de comunicación.

### Crosstalk Communicator

Ofrece un producto de comunicación de bajo costo y fácil de usar, especialmente para comunicaciones de punto a punto. Incluye un *buffer* de búsqueda y una documentación bien escrita tanto para los niveles básicos como para los avanzados. Se puede ajustar la velocidad hasta 115,200 *bps*.

### Crosstalk for Windows

Es un programa de comunicaciones para *Windows* muy intuitivo. La extremadamente amistosa interfaz para el usuario lo hace muy práctico para los usuarios de *Windows*. Su lenguaje de guiones es similar al de *Crosstalk Mk.4*, con funciones añadidas para trabajar con el intercambio dinámico de datos, el cuál deja crear enlaces dinámicos entre los múltiples programas activos en una sesión de *Windows*.

### **Crosstalk Mk.4**

Prueba que un programa de comunicaciones sofisticado y poderoso puede ser todavía fácil de utilizar y amistoso para el usuario. Combinando el lenguaje compilable de guiones y el extenso número de emulaciones de terminal se tendrá uno de los mejores productos de comunicación en *Software* sofisticados que existen actualmente en el mercado.

### **DynaComm Asynchronous Edition**

Es un producto poderoso pero caro. Incluye muchas opciones valiosas tales como el acceso a los *modems* de red y la emulación de terminal 3270 pero el dominarlo lleva tiempo. El lenguaje versátil de guiones puede crear una variedad de aplicaciones.

### **Mirror III**

Ofrece un producto de comunicaciones amistoso con el poder para conversar con eficiencia *BBS's*, *Main Frames* y servicios en línea. Sus poderosas funciones en menús proveen flexibilidad para todas las operaciones y puede añadir los estándares de control de errores y compresión de información *MNP* al *modem*.

### **ProComm Plus**

Es un producto poderoso e intuitivo que presenta menús desplegables. El mismo incluye apoyo para más de 100 *modems*, un modo de diálogo, y un directorio de llamada que contiene hasta 200 nombres, números y ajustes de sesión.

### **Smartcom Exec**

Ofrece una variada emulación de terminales y *protocolos* de transferencia, un lenguaje completo de guiones y un sistema de menú de simple operación. También tiene alto rendimiento y el respaldo de una compañía con una extensa experiencia en comunicaciones.

### **Blast**

Corre en diversas marcas y modelos de equipos: *Micros*, *Minis* y *Main Frames* bajo más de 30 sistemas operativos diferentes. Algunas de las características del *Blast* son las siguientes:

- 1) Transfiere archivos 100% sin errores aún en líneas telefónicas ruidosas.
- 2) Funciona en X.25.

- 3) Opera en radio enlace. En frecuencia *UHF*, *VHF* y onda corta.
- 4) Soporta retardos de propagación que se presentan en comunicación por satélite y en redes X.25 y amplias.
- 5) Emula una gran variedad de terminales (*VT100*, *VT220*, *VT320*, *Teletideo*, *Data General*, *HP*, etc).
- 6) Lenguaje de programación propio para integrarlo en aplicaciones o bien para operación desatendida.
- 7) Retransmite a partir del punto de interrupción. Si por alguna razón se cae el enlace, cuando se restablece, *Blast* sólo transmite saldos de los archivos sin necesidad de empezar la transferencia de nuevo.
- 8) *Full Duplex* real. En una sola línea de dos hilos envía y recibe simultáneamente lo cual lo hace muy rápido.
- 9) Para *PC*'s transfiere en *Background* (las transferencias de archivos las efectúa sin esclavizar la *PC*, es decir que mientras está corriendo cualquier aplicación, al mismo tiempo se realiza la transferencia de archivos) y control remoto (con esta característica es posible tomar control de un *PC* remota desde una computadora local vía *modem*).

La única diferencia entre *Blast PC* y *Blast PC Plus* consiste en que *Blast PC* no cuenta con la opción de control remoto.

*Crosstalk Mk.4* con su muestrario de 21 emulaciones de terminal, opciones flexibles de comunicación en red, un extenso lenguaje de guiones con más de 300 verbos, un preprocesador fácil de configurar, 40 cadenas de configuración de *modem* prefijadas, y una excelente documentación, prueba ser un poderoso programa de comunicaciones, pero su costo es un poco elevado comparado con otros paquetes similares.

Otro producto superior a los demás es *Smartcom Exec*, que es el más rápido de todos los programas en la ejecución de *ZModem* basado en caracteres. Aunque las emulaciones de terminal y los *protocolos* de transferencia de archivo de sus menús no son tan completos como los de otros programas, su propio lenguaje de guiones es muy extenso, con 191 verbos.

Debido a que el sistema a desarrollar no requiere de emulaciones de terminal, ni procesos de comunicación complejos, *Smartcom Exec* resulta ser la mejor opción ya que es un producto económico, rápido y de operación simple.

# **CAPITULO 4**

## **DESARROLLO DEL SRDI**

En el presente capítulo se llevarán a cabo la especificación de requerimientos y diseño del sistema para la recopilación de datos de los inventarios de las sucursales, con el fin de generar información de ventas y existencias globales, así como comparativos entre tiendas.

El capítulo incluye la *Especificación de Requerimientos de Software (ERS)* y, en forma resumida, la fase de diseño. La *ERS* se ha realizado después de hacer un estudio del funcionamiento del sistema actual y de determinar y analizar los requerimientos que debe cubrir el nuevo sistema, y sirve como base para las fases de diseño y desarrollo.

Para el desarrollo de este capítulo se emplean técnicas de análisis y desarrollo de Yourdon, representante de la corriente metodológica más difundida y que incluyen: diagramas de flujo de datos, modelo entidad-relación, diccionario de datos y diagrama de estructura.

## **4.1 ESPECIFICACION DE REQUERIMIENTOS DE SOFTWARE**

### **4.1.1 INTRODUCCION**

#### **4.1.1.1 Objetivo**

El propósito de esta *Especificación de Requerimientos de Software* es presentar de manera sencilla los requerimientos necesarios para el desarrollo de un *Sistema de Recopilación de Datos de Inventarios (SRDI)* de las sucursales de **Comercial Emyco, S.A. de C.V.** Incluye una descripción técnica y lógica de lo que realizarán las funciones del *Software*.

#### **4.1.1.2 Alcance de SRDI**

El *SRDI* recopilará diariamente los movimientos de mercancía registrados en cada una de las sucursales y los concentrará, para después obtener información comparativa entre ellas de existencias y ventas. El sistema llevará también un registro de los movimientos de artículos de cada una de las sucursales con el fin de poder formular pedidos de artículos que serán enviados a cada una de ellas. El *Software* también registrará los catálogos de precios de artículos y los enviará a las sucursales junto con información de apoyo para que ellas puedan realizar su registro de inventarios de manera conveniente.

#### **4.1.1.3 Definiciones, Acrónimos y Abreviaciones**

Grupo. Es la clasificación del artículo de acuerdo al uso y persona al que está destinado, por ejemplo: caballero vestir, dama vestir, caballero sport, etc.

**Línea.** Define a la compañía que provee el artículo.

**Sublínea.** Es la clasificación de artículos con características comunes, pertenecientes a un grupo-línea.

**Modelo.** Es cada una de las variaciones en artículos de una sublínea.

**Corrida.** Es un grupo de tallas en las que puede existir un artículo

**Movimiento.** Cada una de las entradas o salidas que tiene un artículo en el almacén. Incluye:

- a) Embarques. Los artículos que llegan al almacén directamente del proveedor.
- b) Entradas por transferencias. Los artículos que entran al almacén enviados por otra de las sucursales de la organización.
- c) Devoluciones a proveedor. Los artículos que son enviados de vuelta al proveedor por razones de defecto o de mala calidad.
- d) Salidas por transferencias. Los artículos que son enviados del almacén de una tienda al de otra tienda de la organización.
- e) Ventas. Salidas del almacén por entrega de artículos a algún cliente.
- f) Devoluciones de cliente. Entradas al almacén por devolución de mercancía por parte del cliente.

#### **4.1.1.4 Consideraciones**

El resto de la *ERS* contiene un mayor detalle del proyecto. Las principales secciones contienen lo siguiente:

La sección 4.1.2 contiene la descripción general de cada uno de los requerimientos, junto con todas las suposiciones que se han hecho sobre el funcionamiento del sistema y su interrelación con los sistemas de inventarios de las sucursales.

La sección 4.1.3 contiene la descripción técnica de cada uno de los requerimientos funcionales.

En el apéndice C se incluyen el Diagrama de Contexto del *SRDI*, los Diagramas de Flujo de Datos con las especificaciones de cada nivel y el Diccionario de Datos que describe la composición de los datos relacionados con el sistema.

## **4.1.2 DESCRIPCION GENERAL**

### **4.1.2.1 Introducción**

En esta sección se describen todos los factores que afectan al desarrollo del sistema y, de forma general, los requerimientos que se deben cumplir, que sirvan de base para definir en la sección siguiente los requerimientos de forma más específica.

### **4.1.2.2 Perspectiva del SRDI**

Como ya se mencionó, el *Software* a desarrollar debe tener una interfaz con los sistemas de inventarios de las sucursales. Esta interfaz se hará mediante el uso de un software de comunicaciones comercial para la transferencia de archivos *ASCII*. Este paquete se encargará de realizar la transmisión de datos y de la colocación de estos en un directorio especial donde puedan ser tomados por el *SRDI* en caso de recepción o en caso de envío.

Al momento de activarse el sistema deberá reconocer los archivos enviados por las sucursales y con ellos actualizar un archivo maestro de movimientos, antes de actualizar el archivo maestro de existencias y desechar estos de envío.

### **4.1.2.4 El usuario**

El operador del *SRDI* deberá tener práctica en el manejo de *PC*, con conocimientos del *MS-DOS*, específicamente en el arranque de un programa de aplicación, y de alguna impresora. Deberá tener conocimientos del paquete de comunicaciones para poder realizar los enlaces con las sucursales para la transferencia de datos.

### **4.1.2.5 Restricciones y requerimientos generales**

El sistema deberá ser capaz de incorporar a su base de datos, archivos en formato de texto y generar archivos de igual formato que puedan ser enviados a través de la *Red Conmutada*.

## **4.1.3 REQUERIMIENTOS ESPECIFICOS**

### **4.1.3.1 Introducción**

Esta sección contiene los requerimientos específicos para el desarrollo del *SRDI*. Cada requerimiento funcional tendrá una descripción general, sus entradas, procesamiento y salidas.

#### **4.1.3.2 Requerimientos funcionales**

Para la determinación de las funciones del *SRDI* se recurre al Diagrama de Contexto y los Diagramas de Flujo de Datos del apéndice C, de donde se distinguen las siguientes funciones :

##### **4.1.3.2.1 Función de recepción de datos (Proceso 1.0 DFD)**

###### **4.1.3.2.1.1 Función de Identificación de Tienda (Proceso 1.1 DFD)**

###### **Introducción**

Los movimientos se reciben en un archivo con el nombre en formato PMMDDTTT.TXT, donde:

- P indica que se trata de un paquete de movimientos
- MM es el mes de envío
- DD es el día de envío
- TTT es el número de la tienda que envía el paquete de datos

Al recibir los movimientos de una tienda, debe actualizarse el archivo de tiendas con la fecha de recepción, para validar que se haga el cierre periódico de inventario, sólo cuando todas las tiendas hayan enviado sus movimientos de todo el mes.

###### **Entradas**

- Paquete\_movims

###### **Procesamiento**

Si se detecta en el directorio del sistema un archivo con el formato MTTTDDMM.TXT, se identifica a la tienda que lo envió por los caracteres segundo al cuarto.

Se copia el contenido del archivo detectado a un archivo de recepción llamado TMOV, que antes debe ser vaciado, para que únicamente contenga los datos nuevos.

Se actualiza el día del último envío en el archivo de tiendas con los datos del día y el mes detectados (dígitos cinco al ocho).

## Salidas

- Detalle\_movims

### 4.1.3.2.1.2 Función de Actualización de Existencias (Proceso 1.2 DFD)

#### Introducción

Se actualiza el archivo de existencias para la tienda de entrada de acuerdo a las cantidades y tipos de movimientos indicados en los detalles de entrada.

El proceso también actualiza el costo promedio en el catálogo de artículos. Este costo promedio se maneja a nivel de organización, por lo que es afectado por los movimientos de todas las sucursales. Se considera que las existencias están ordenadas por artículo a nivel modelo, al igual que los movimientos.

#### Entradas

- Detalle\_movims

- Existencia\_día

- Costo\_promedio

#### Proceso

Mientras haya movimientos

existencia\_inicial=0

cantidad\_articulos=0

nuevo\_costo\_promedio=0

existencia\_nueva=0

Buscar el artículo a nivel color-talla del movimiento en el almacén de existencias respectivo a la tienda de entrada.

Si no existe el artículo-color-talla

    Agregar un nuevo registro al archivo con la identificación del artículo y con saldo igual a cero

existencia\_inicial= la existencia actual del artículo

$\text{cantidad\_articulos} = (\text{la cantidad de artículos registrados en el movimiento, multiplicándola de acuerdo a la tabla de la figura 4.1})$

$\text{existencia\_nueva} = \text{existencia\_inicial} + \text{cantidad\_articulos}$

Si el movimiento no es un ajuste

$\text{nuevo\_costo\_promedio} = (\text{precio de costo del movimiento} * \text{cantidad\_articulos} + \text{precio de costo del catálogo de artículos} * \text{existencia\_inicial}) / \text{existencia\_nueva}$

en caso contrario

$\text{nuevo\_costo\_promedio} = \text{precio de costo del catálogo de artículos}$

Reemplazar el costo promedio del artículo con  $\text{nuevo\_costo\_promedio}$

Reemplazar la existencia del artículo con  $\text{existencia\_nueva}$

Tipo de Movimiento	Acción a realizar
Embarque	multiplicar por (+)
Transferencia (entrada)	multiplicar por (+)
Devolución de cliente	multiplicar por (+)
Transferencia (salida)	multiplicar por (-)
Devolución a proveedor	multiplicar por (-)
Venta a cliente	multiplicar por (-)
Ajuste de inventario	multiplicar por (+)

*Figura 4.1 Afectación de los diferentes tipos de movimientos a las existencias*

## Salidas

- Existencia\_nueva
- Nuevo\_costo\_promedio

### 4.1.3.2.1.3 Función de Concentrado de Movimientos (Proceso 1.3 DFD)

#### Introducción

Se actualiza archivo de concentrado de movimientos con los detalles de entrada.

#### Entradas

- Detalle\_movims

## **Proceso**

Se graba cada detalle de movimientos en el concentrado de movimientos

## **Salidas**

- Detalle\_movims

### **4.1.3.2.2 Función de Reporte de Movimientos (Proceso 2.0 DFD)**

El reporte del Concentrado de Movimientos genera un resumen de los movimientos realizados por artículo-tienda, indicando para cada uno de ellos la existencia al inicio de mes y la existencia final y precio de costo, calculados a partir de sus movimientos.

#### **4.1.3.2.2.1 Función de Selección de Movimientos (Proceso 2.1 DFD)**

## **Introducción**

Este proceso realiza el filtrado de datos de movimientos de acuerdo a un rango de tiendas proporcionado por el usuario y a la fecha proceso, tomada del reloj del sistema.

## **Entradas**

- Fecha\_proceso
- Rango\_de\_tiendas
- Detalle\_movims

## **Proceso**

Se pide al usuario el rango de tiendas que serán incluidas en el reporte de existencias y ventas

Si el número de la primera tienda es mayor al de la segunda

Desplegar error: "Error en rango de tiendas"

en caso contrario

Se toma la fecha del sistema, que será la de proceso

Del concentrado de movimientos, se filtra la información menor o igual a la fecha proceso y que esté dentro del rango de tiendas especificado.

## **Salidas**

- Movims\_filtrados (mes-tienda)

### **4.1.3.2.2.2 Función de Totalizado por Tienda-Artículo (Proceso 2.2 DFD)**

#### **Introducción**

Esta función tiene como finalidad totalizar los movimientos por artículo a nivel de modelo. Se sugiere utilizar una estructura de datos temporal para concentrar este totalizado que tenga un campo para cada tipo de movimiento y uno para totalizar la existencia de artículos en el siguiente proceso (Recuperación de existencias DFD 2.3).

#### **Entradas**

- Movims\_filtrados (Mes-Tienda)

#### **Proceso**

Ordenar los movimientos por Tienda-Artículo-Tipo de movimiento

Mientras haya los movimientos

Agregar registro en el archivo de Totalización, grabar el artículo y el número de tienda

Mientras la tienda y el artículo sean iguales:

$Total\_articulos=0$

Mientras el Tipo de movimiento sea igual:

$Total\_articulos=Total\_articulos+la\ cantidad\ en\ el\ movimiento$

Grabar en el registro de Totalización, en el tipo de movimiento correspondiente, el total de artículos resultante.

## **Salidas**

- Movims\_tienda\_articulo

#### **4.1.3.2.3 Función de recuperación de existencias (Proceso 2.3)**

##### **Introducción**

Ya que el Totalizado de movimientos se hizo a nivel de modelo, el totalizado de existencias debe hacerse también a este nivel. En el proceso de actualización de existencias (DFD 1.2), este se hace a nivel de Color-Talla, por lo que se requiere un algoritmo de totalización. Para el algoritmo de cálculo de existencia a nivel modelo, se considera que las existencias están ordenadas por Artículo-Color-Talla. Las existencias que se recuperan son las de inicio de mes, ya que en el reporte se calcularán las existencias hasta la fecha de impresión.

##### **Entradas**

- Movims\_tienda-artículo

##### **Proceso**

Mientras haya movimientos

    Total\_artículos=0

    Buscar el Artículo en el almacén de existencias de la tienda correspondiente

    Mientras el Artículo-Tienda sea el mismo

        Total\_artículos=Total\_artículos+existencia en el almacén al inicio del mes

    Grabar en el archivo de totalizado el total de artículos como la existencia a imprimir en el reporte de existencias y ventas.

##### **Salidas**

- Movims\_y\_existencias

#### **4.1.3.2.4 Reporte de Existencias y Ventas (Función 2.4 DFD)**

##### **Introducción**

El reporte de Existencias y Ventas se crea a partir del totalizado de movimientos y existencias creado en la función 4.1.3.2.2.3. En él se incluirá el precio de costo promedio, tomado del catálogo de artículos (Función 4.1.3.2.1.2). El reporte debe ser ordenado por tienda y artículo.

##### **Entradas**

- Movims\_y\_existencias

- Sublínea
- Descripción\_artículo
- Costo\_promedio

### **Proceso**

Para cada registro del archivo de existencias y ventas, se imprimen los datos en forma horizontal y en el orden siguiente:

- a) Sublínea
- b) Código de artículo
- c) Descripción del artículo
- d) Costo promedio
- e) Existencia inicial
- f) Embarques
- g) Devoluciones de cliente
- h) Entregas (entradas por Traspaso)
- i) Ajustes
- j) Traspasos
- k) Devoluciones a proveedor
- l) Ventas
- m) Existencia final

### **Salidas**

- Existencias\_y\_ventas

#### **4.1.3.2.3 Función de Impresión de Comparativo de Existencias y Ventas (Proceso 3.0 DFD)**

Este proceso tiene el objeto de generar un reporte de existencias y ventas, con totales por tienda-artículo a nivel color, a diferencia del proceso 2.0 DFD, en el

cual se hacen totales por artículo-tienda. Para cada artículo se imprimen los totales de existencias y ventas para cada una de las tiendas.

#### **4.1.3.2.3.1 Función de Selección de Movimientos (Proceso 3.1 DFD)**

##### **Introducción**

Tiene el propósito de filtrar los movimientos por el rango de tiendas a incluir y con fecha menor o igual a la proporcionada.

##### **Entradas**

- Fecha
- Rango\_de\_Tiendas
- Detalle\_Movims

##### **Proceso**

Se pide al usuario el rango de tiendas que serán incluidas en el reporté de existencias y ventas

Si el número de la primera tienda es mayor al de la segunda

Desplegar error: "Error en rango de tiendas"

en caso contrario

Se toma la fecha del sistema, que será la fecha de proceso

Del concentrado de movimientos, se filtra la información correspondiente menor o igual a la fecha de proceso y que esté dentro del rango de tiendas especificado.

##### **Salidas**

- Movims\_filtrados

#### **4.1.3.2.3.2 Función de Totalizado por Artículo-Color (Proceso 3.2 DFD)**

##### **Introducción**

Esta función tiene como finalidad totalizar los movimientos por artículo a nivel de color. Se sugiere utilizar una estructura de datos temporal para concentrar este totalizado, que tenga un campo para cada tipo de movimiento y uno para

totalizar la existencia de artículos en el siguiente proceso (Recuperación de existencias DFD 3.3).

### **Entradas**

- Movims\_filtrados (Mes-Tienda)

### **Proceso**

Ordenar los movimientos por Artículo-Tienda-Tipo de movimiento

Mientras haya los movimientos

Grabar en el archivo de Totalización el artículo

Mientras el artículo (a nivel color) y la tienda sean iguales:

Total\_articulos=0

Mientras el Tipo de movimiento sea igual:

Total\_articulos=Total\_articulos+la cantidad en el movimiento

Grabar en el archivo de Totalización el total de artículos resultante en el tipo de movimiento correspondiente.

### **Salidas**

- Movims\_artículo-color

## **4.1.3.2.3.3 Función de Recuperación de Existencias (Proceso 3.3 DFD)**

### **Introducción**

Ya que el Totalizado de movimientos se hizo a nivel de color, el totalizado de existencias debe hacerse también a este nivel. En el proceso de actualización de existencias (DFD 1.2), este se hace a nivel de Color-Talla, por lo que se requiere un algoritmo de totalización. Para el algoritmo de cálculo de existencia a nivel color, se considera que las existencias están ordenadas por Artículo-Color

### **Entradas**

- Movims\_artículo-color

- Existencia\_inicial

## **Proceso**

Mientras haya movimientos

Total\_artículos=0

Buscar el Artículo en el almacén de existencias de la tienda correspondiente

Mientras el Artículo-Color-Tienda sea el mismo

Total\_artículos=Total\_artículos+existencia en el almacén

Grabar en el archivo de totalizado el total de artículos como la existencia a imprimir en el reporte de existencias y ventas.

## **Salidas**

- Movims\_y\_existencias

### **4.1.3.2.3.4 Función de Impresión de Reporte Comparativo (Proceso 3.4 DFD)**

#### **Introducción**

El reporte de Existencias y Ventas se crea a partir del totalizado de movimientos y existencias creado en la función 4.1.3.2.2.3. En él se incluirá el precio de costo promedio, tomado del catálogo de artículos (Función 4.1.3.2.1.2). El reporte debe ser ordenado por tienda y artículo.

#### **Entradas**

- Movims\_y\_existencias

- Sublínea

- Descripción\_artículo

- Costo\_promedio

#### **Proceso**

Para cada artículo-color de movims\_y\_existencias

Imprimir los datos del artículo en forma horizontal y en el orden siguiente:

- a) Sublínea
- b) Código de artículo

- c) Descripción del artículo
- d) Costo promedio

Para cada color del artículo

Imprimir color del artículo

Para cada tienda

Imprimir

- a) Existencia
- b) Ventas

Acumular para el artículo-color las cantidades por movimiento y existencias

Imprimir los acumulados y existencias de movimientos para el artículo-color

- a) Existencia inicial (existencia final - totales de (b), (c), (d), (e), (f) y (g))
- b) Devoluciones de cliente
- c) Entregas (entradas por traspaso)
- d) Traspasos (salidas)
- e) Ajustes
- f) Devoluciones a proveedor
- g) Ventas
- h) Existencia final

#### Salidas

- Comparativo\_existencias\_y\_ventas

#### 4.1.3.2.4 Función de Cierre de Inventario (Proceso 4.0 DFD)

Este proceso tiene la función de actualizar las existencias de inicio de mes con las existencias finales, para irse acumulando en el mes que comienza. A continuación se realiza un respaldo de los movimientos efectuados durante el

mes, antes de ser eliminados todos sus registros para la acumulación del nuevo mes.

El acceso a este proceso debe estar restringido a personal autorizado para realizarlo, ya que su activación actualiza los saldos de los artículos (de inicio del mes y del día de labor) y elimina de la base de datos los movimientos del mes, lo que provocaría resultados incorrectos en los reportes por tienda y comparativo de existencias y ventas .

#### **4.1.3.2.4.1 Función de Verificación de Envíos de Sucursal (Proceso 4.1 DFD)**

##### **Introducción**

El proceso (1.1) del *DFD* de la recepción de datos de la sucursal actualiza los datos de cada tienda con una fecha de último envío de datos de la sucursal. El proceso de esta sección utiliza las fechas para determinar si se puede realizar el Cierre de Inventario.

##### **Entradas**

- Fecha\_sistema
- Fechas de último envío de movimientos

##### **Proceso**

El proceso de cierre de inventario es activado por el usuario y se debe realizar sólo si:

- a) No se ha realizado el Cierre de Inventario para el mes que se pretende
- b) La fecha de último envío de las tiendas es menor o igual a la fecha del reloj del sistema

Si no se cumple cualquiera de las condiciones de arriba, el sistema deberá enviar un mensaje explicando la razón por la cuál no se permite hacer el Cierre. En caso contrario se realizan los procesos (4.2) y (4.3) del *DFD*.

##### **Salidas**

- Mes\_de\_Cierre

#### **4.1.3.2.4.2 Función de Actualización de Existencias Iniciales (Proceso 4.2 DFD)**

##### **Introducción**

Este proceso es el que realiza específicamente el Cierre del Inventario. Hace la actualización de las existencias iniciales del mes que comienza y al primer día de labor de este mes. Estas existencias son utilizadas para realizar los reportes por tienda y el comparativo de existencias y ventas acumuladas por mes.

##### **Entradas**

- Mes\_de\_cierre
- Existencia\_inicial

##### **Proceso**

Mientras haya artículos en el almacén de existencias

Reemplazar la existencia\_mes por la existencia\_día (existencia\_inicial)

Reemplazar la existencia\_día por cero (existencia\_nueva)

##### **Salidas**

- Existencia\_nueva

#### **4.1.3.2.4.3 Función de Respaldo de Información (Proceso 4.3 DFD)**

##### **Introducción**

Después de que se han actualizado las existencias de los artículos, se realiza el respaldo de movimientos del mes que se ha cerrado, para posteriormente ser eliminados del almacén y que este comience el siguiente mes vacío. Esto evita la utilización innecesaria de espacio de almacenamiento de datos.

##### **Entradas**

- Mes\_de\_cierre
- Detalle\_movims

## **Proceso**

Con el mes de cierre, se filtran los movimientos recibidos a lo largo de éste y son enviados a un disco flexible. El proceso debe tener la capacidad de enviar los respaldos a discos de 5 1/4" o 3 1/2".

El formato para nombrar los archivos de respaldo deberá ser de la siguiente forma:

RMMAANN.[Extensión de archivo de datos]

donde:

R Significa Archivo de Respaldo

MMAA Es el mes y el año de respaldo

NN Es el número consecutivo de discos de respaldo para un mismo mes

Después de haber completado el respaldo, la misma información filtrada deberá ser eliminada del almacén de movimientos.

## **Salidas**

- Detalle\_movims (a disco)

### **4.1.3.2.5 Función de Ajustes de Inventario (Proceso 5.0 DFD)**

#### **4.1.3.2.5.1 Función de Captura del Ajuste (Proceso 5.1 DFD)**

## **Introducción**

Este proceso captura los datos para ajuste de inventario de una tienda particular, validándolos antes de aceptarlos.

## **Entradas**

- Ajuste

## **Proceso**

Para el registro de un ajuste se deberán pedir al usuario los datos de entrada y deberán ser validados de la siguiente forma:

- a) El número de tienda deberá existir en el almacén de tiendas

- b) El código del artículo deberá existir en el almacén de artículos, en el color y la talla especificados
- c) La cantidad de artículos de ajuste deberá ser diferente de cero

Se pueden registrar varios ajustes para una misma tienda, que cumplan con las condiciones anteriores y serán agrupados para poder ser utilizados por el proceso (5.2) del *DFD*.

### **Salidas**

- Grupo\_ajustes

#### **4.1.3.2.5.2 Función de Creación de Paquete de Ajustes (Proceso 5.2 DFD)**

### **Introducción**

El proceso genera un archivo que será enviado a una tienda, donde será leído por el sistema de inventarios para realizar los movimientos de ajuste.

### **Entradas**

- Grupo\_ajustes

### **Proceso**

Para la creación de un paquete de ajustes, se toma el grupo de ajustes de entrada y se genera un archivo de texto con un ajuste por renglón.

El nombre del archivo de ajustes a generar será de la forma ANNNNTTT.TXT, donde:

- A Indica que se trata de un ajuste
- NNNN indica el número de paquete de ajustes (consecutivo)
- TTT indica el número de la tienda destino

Capturar el número de tienda

Verificar que la tienda exista en el archivo de tiendas

Repetir hasta que el usuario utilice un comando de paro

- Capturar el código del artículo y verificar que exista en el archivo de artículos

- Capturar el código de color y verificar que exista en el archivo de artículo-color
- Capturar el código de talla y verificar que se encuentre dentro de la corrida, contenida en el archivo de artículos
- Capturar la cantidad de ajuste y verificar que sea mayor o menor que cero, pero no igual
- Si no existió error en la captura de datos, grabar datos en el archivo de salida

Dentro del archivo, los datos deberán colocarse en la siguiente disposición:

código\_artículo, código\_color, código\_talla, cantidad

### **Salidas**

- Paquete\_de\_ajustes

### **4.1.3.2.6 Función de Registro de Tiendas (Proceso 6.0 DFD)**

#### **Introducción**

En este proceso se registran los datos de las tiendas pertenecientes a la empresa. Este registro se utilizará para las validaciones de tiendas de los distintos procesos que las requieran.

#### **Entradas**

- Número\_tienda
- Nombre\_tienda
- Dirección\_tienda
- Teléfono\_tienda
- Representante\_tienda

#### **Proceso**

Se deberán tener las siguientes opciones

- a) Alta

Repetir hasta que el usuario utilice un comando de paro o se graben los datos

Capturar los datos de la tienda

Validar que se proporcionen todos los datos de la tienda. Si esto no se cumple, se generará un mensaje de error

Validar que no se duplique el número de la tienda. Si se intenta duplicar, se generará un mensaje de error

Si no se produjo mensaje de error

Agregar los datos de la tienda al archivo de tiendas

#### b) Baja

Pedir al usuario el número de tienda

Si la tienda existe

Borrar la tienda del archivo de tiendas

#### c) Modificación

Pedir al usuario el número de tienda

Si la tienda existe en el archivo de tiendas

Recupera los datos de la tienda

Repetir hasta que el usuario utilice un comando de paro o se graben los datos

Permitir modificar los datos de la tienda, exceptuando el número que la identifica

Validar que los datos capturados no estén vacíos. Si alguno lo está, enviar mensaje de error

Si no se produjo mensaje de error

Actualizar en el archivo los datos de la tienda

#### d) Reporte

Repetir hasta que el usuario utilice un comando de paro o se impriman los datos

Pedir al usuario el rango de tiendas

Si la tienda inicial es mayor a la final

desplegar mensaje de error

en caso contrario

filtrar los datos de las tiendas que cumplan con el rango

imprimir los datos de las tiendas

### **Salidas**

- Datos\_tienda

### **4.1.3.2.7 Función de Registro de Proveedores (Proceso 7.0 DFD)**

#### **Introducción**

En este proceso se registran los datos de los proveedores de la empresa. Este registro se utilizará para las validaciones en el registro de artículos y para el registro de movimientos de artículos en las sucursales.

#### **Entradas**

- Número\_proveedor
- Nombre\_proveedor
- Dirección\_proveedor
- Teléfono\_proveedor
- Representante\_proveedor

#### **Proceso**

Se deberán tener las siguientes opciones

a) Alta

Repetir hasta que el usuario utilice un comando de paro o se graben los datos

Capturar los datos del proveedor

Validar que se proporcionen todos los datos del proveedor. Si esto no se cumple, se generará un mensaje de error

Validar que no se duplique el número de proveedor. Si se intenta duplicar, se generará un mensaje de error

Si no se produjo mensaje de error

Agregar los datos del proveedor al archivo de proveedores

b) Baja

Pedir al usuario el número de proveedor

Si el proveedor existe

Borrar el proveedor del archivo de proveedores

c) Modificación

Pedir al usuario el número de proveedor

Si el proveedor existe en el archivo de proveedores

Recuperar los datos del proveedor

Repetir hasta que el usuario utilice un comando de paro o se graben los datos

Permitir modificar los datos del proveedor, exceptuando el número que lo identifica

Validar que los datos capturados no estén vacíos. Si alguno lo está, enviar mensaje de error

Si no se produjo mensaje de error

Actualizar en el archivo de proveedores los datos del proveedor

d) Reporte

Repetir hasta que el usuario utilice un comando de paro o se impriman los datos

Pedir al usuario el rango de proveedores

Si el proveedor inicial es mayor al final

desplegar mensaje de error

en caso contrario

filtrar los datos de los proveedores que cumplan con el rango

imprimir los datos de los proveedores

### **Salidas**

- Datos\_proveedor

#### **4.1.3.2.8 Función de Registro de Artículos (Proceso 8.0 DFD)**

En este proceso se registran los datos de los artículos que se manejarán en los inventarios de las sucursales. Estos datos son la parte más importante de todo el sistema porque alrededor de ellos giran todos los procedimientos del control de inventarios.

##### **4.1.3.2.8.1 Función de Captura de Descripción de Artículo (Proceso 8.1 DFD)**

### **Introducción**

Un aspecto importante para este proceso es el de tener una estructura de datos en la que se puedan registrar todas las tiendas en donde se encuentre disponible el artículo, así como todos los colores en los cuales existe.

### **Entradas**

- Código\_artículo
- Grupo
- Línea
- Sublínea
- Modelo
- Descripción
- Precio\_venta
- Código\_corrida

- Código\_color

### Proceso

Se deberán tener las siguientes opciones

#### a) Alta

Repetir hasta que el usuario utilice un comando de paro o se graben los datos

Capturar los datos de la tienda

Serán motivo de error de captura los siguientes casos:

- 1) No se proporcionan todos los datos del artículo
- 2) Se intenta duplicar el código del artículo
- 3) No existe la línea (número de proveedor)
- 4) Se intenta duplicar el grupo-línea-sublínea-modelo
- 5) El precio de venta es menor que cero

Si se cumple una de las condiciones anteriores

enviar el mensaje de error correspondiente

en caso contrario

Agregar los datos del artículo al archivo de artículos

#### b) Baja

Pedir al usuario el código del artículo

Si el artículo existe

Presentar datos del artículo

Presentar al usuario mensaje de confirmación

Si el usuario confirma el borrado

Borrar el artículo del archivo de artículos

#### c) Modificación

Pedir al usuario el código del artículo

Si el artículo existe en el archivo de artículos

Recuperar los datos del artículo

Repetir hasta que el usuario utilice un comando de paro o se graben los datos

Permitir modificar los datos del artículo, exceptuando el código

Si se cumple alguna de las condiciones de error listados en el inciso (a)

Desplegar el mensaje de error correspondiente

en caso contrario

Actualizar en el archivo los datos del artículo

#### d) Reporte

Repetir hasta que el usuario utilice un comando de paro o se impriman los datos

Pedir al usuario el rango de artículos (por código de artículo)

Si el código inicial es mayor al final

Desplegar mensaje de error

en caso contrario

Filtrar los datos de los artículos que cumplan con el rango

Imprimir los datos de los artículos, incluyendo los colores en que existe y las tiendas en que está disponible

#### **Salidas**

- Datos\_tienda

#### **4.1.3.2.8.2 Función de Registro en Tiendas (Proceso 8.2 DFD)**

##### **Introducción**

Este proceso registra en qué tiendas existirá el artículo que se capturó en el proceso 8.1 del *DFD*.

##### **Entradas**

- Código\_artículo
- Número\_tienda

##### **Proceso**

Pedir al usuario el código del artículo

Si el artículo existe

Repetir hasta que el usuario utilice un comando de paro

Capturar el número de la tienda

Si la tienda existe

Buscar el artículo y tienda en el archivo art-tienda

Si no existen

Agregar al archivo art-tienda los códigos de artículo y tienda

en caso contrario

Desplegar mensaje de error

en caso contrario

Desplegar mensaje de error

##### **Salidas**

- Código\_artículo
- Número\_tienda

#### **4.1.3.2.8.3 Función de Registro de Colores (Proceso 8.3 DFD)**

##### **Introducción**

En este proceso se registran los colores en que existe el artículo capturado en el proceso 8.1 del *DFD*.

##### **Entradas**

- Código\_artículo
- Código\_color

##### **Proceso**

Pedir al usuario el código del artículo

Si el artículo existe

Repetir hasta que el usuario utilice un comando de paro

Capturar el código de color

Si el código de color existe

Buscar el artículo y color en el archivo art-tienda

Si no existe

Agregar al archivo art-color los códigos de artículo y color

en caso contrario

Desplegar mensaje de error

en caso contrario

Desplegar mensaje de error

##### **Salidas**

- Código\_artículo
- Código\_color

### **4.1.3.3 Atributos**

#### **4.1.3.3.1 Confiabilidad**

*SRDI* será ampliamente probado para evitar que ocurran errores en los cálculos. El código será escrito en una forma modular para hacer las modificaciones tan fáciles como sea posible.

#### **4.1.3.3.2 Mantenibilidad**

Los módulos del sistema serán los más independiente posible para que los cambios en alguno de ellos no produzca error en cualquier otra parte del sistema.

#### **4.1.3.3.3 Seguridad**

El acceso a los procesos más importantes del sistema estará protegido contra personal no autorizado para ejecutarlo, mediante el uso de claves de acceso a los mismos.

#### **4.1.3.3.4 Integridad**

*SRDI* tendrá la capacidad para la fácil reconstrucción de los archivos de índices dañados por fallas en la energía eléctrica.

## **4.2 DISEÑO**

### **4.2.1 INTRODUCCION**

El objetivo de este apartado es el de mostrar de forma general la estructura del sistema físico que satisface los requerimientos planteados en la *ERS*, desarrollada en la primera parte de este capítulo.

Las especificaciones de programas de cualquier sistema pueden llevarse cientos o miles de hojas. En este trabajo sólo se incluyen la descripción de archivos y las especificaciones de un módulo (Actualización de existencias).

### **4.2.2 CONSIDERACIONES**

Por razones prácticas, la codificación del módulo se realizará directamente en el lenguaje *Clipper*. Esto puede ser justificado con el conocimiento de las características del lenguaje y su adaptabilidad al tipo de aplicación a desarrollar. Una justificación de selección en base a comparación de características con

otras herramientas ya ha sido dada en el capítulo 3, en la sección de Análisis de Software de Desarrollo.

### 4.2.3 DIAGRAMA DE ESTRUCTURA

Las figuras 4.2 a 4.6 muestran el diagrama de estructura de SRDI de forma general y los módulos resultantes.

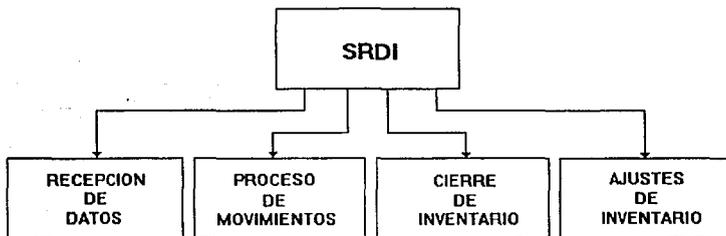


Figura 4.2 Primer nivel del diagrama de estructura.

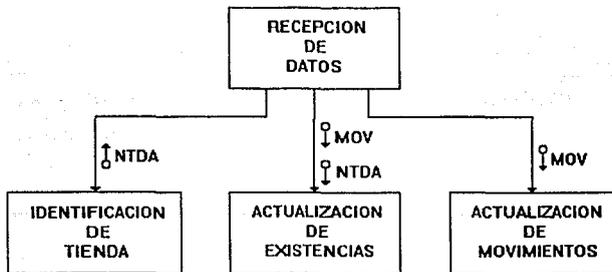


Figura 4.3 Módulos para la recepción de datos.

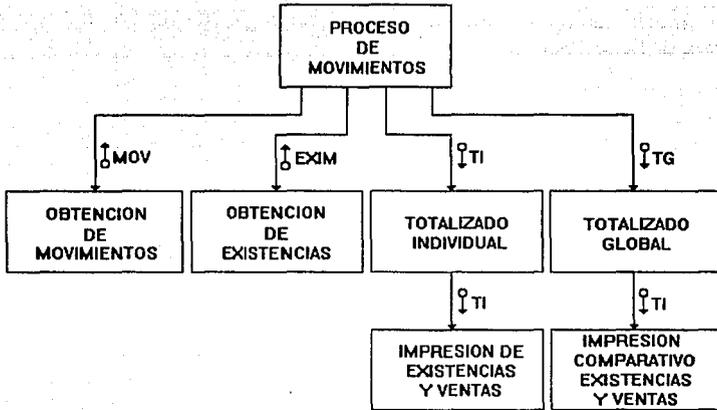


Figura 4.4 Módulos para proceso de movimientos.

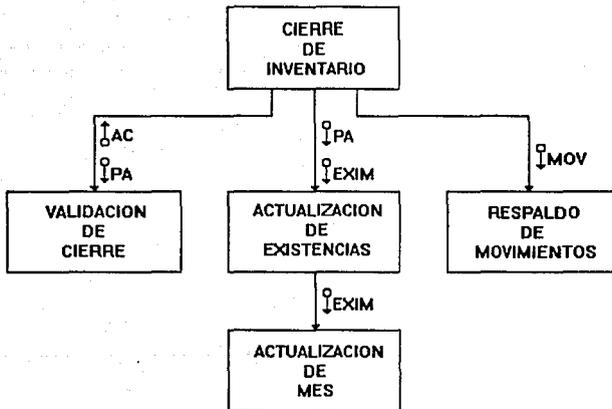


Figura 4.5 Módulos para el cierre de inventarios.

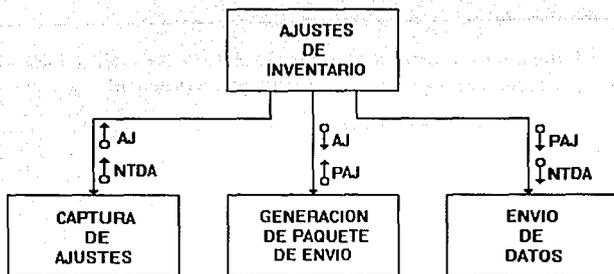


Figura 4.6 Módulos para ajustes de inventario.

## 4.2.4 ALMACENAMIENTO DE DATOS

Se han definido ocho archivos para el almacenamiento permanente de datos del sistema. En base a los atributos del artículo, se definieron para él cuatro tablas. A continuación se presentan las estructuras y los índices para acceso y ordenamiento de cada uno de los archivos. Los tipos de campo se abrevian: C (carácter), N (numérico) y F (fecha).

### 4.2.4.1 Almacén CART

Descripción: Datos generales del artículo.

Campo	Tipo	Longitud	Decimales
ccart	C	7	0
grupo	C	2	0
línea	C	4	0
sublínea	C	5	0
modelo	C	6	0
descr_art	C	40	0
corrida	C	2	0
pventa	N	8	0
pcosto	N	7	0

Índices:

- a) CART1. ccart
- b) CART2. grupo+línea+sublínea+modelo

#### 4.2.4.2 Almacén ACOL

Descripción: Relaciona los artículos con los colores. Los colores en que existe un artículo. Se utiliza en la impresión del catálogo de artículos.

Campo	Tipo	Longitud	Decimales
ccart	C	7	0
cod_color	C	2	0

Indices:

- a) ACOL1. ccart+cod\_color

#### 4.2.4.3 Almacén ATDA

Descripción: Relaciona el artículo con las tiendas. Las tiendas que expenden el artículo. Se utiliza en la impresión del catálogo de artículos.

Campo	Tipo	Longitud	Decimales
ccart	C	7	0
numtda	C	3	0

Indices:

- a) ATDA. ccart+numtda

#### 4.2.4.4 Almacén SBL

Descripción: Sublíneas de artículos. La sublínea esta relacionada con el proveedor, que es la línea. Se utiliza en la captura de artículos.

Campo	Tipo	Longitud	Decimales
línea	C	4	0
sublínea	C	5	0
descr_sbl	C	20	0

Indices:

- a) SBL. línea+sublínea

#### 4.2.4.5 Almacén MOVD

Descripción: Archivo maestro de movimientos. En él se almacenan los movimientos enviados por las tiendas y de él se obtienen los reportes de existencias y ventas.

Campo	Tipo	Longitud	Decimales
numtda	C	3	0
ccart	C	7	0
tipomov	C	1	0
fecha	F	8	0
cod_color	C	2	0
cod_talla	C	2	0
cantidad	C	4	0
pcosto	N	7	0

Indices:

- a) MOVD1. fecha+numtda+ccart+tipomov
- b) MOVD2. fecha+ccart+numtda+cod\_color+tipomov

#### 4.2.4.6 Almacén EXI

Descripción: Almacenamiento de existencias iniciales diarias y de mes. Es actualizado con los movimientos enviados por las tiendas. Se utiliza en los reportes de existencias y ventas.

Campo	Tipo	Longitud	Decimales
numtda	C	3	0
ccart	C	7	0
cod_color	C	2	0
cod_talla	C	2	0
saldod	C	4	0
saldom	C	4	0

Indices:

- a) EXI1. numtda+ccart+cod\_color+cod\_talla
- b) EXI2. ccart+cod\_color+numtda

#### 4.2.4.7 Almacén TDA

Descripción: Datos generales de las tiendas.

Campo	Tipo	Longitud	Decimales
numtda	C	3	0
nomtda	C	30	0
domtda	C	30	0
coltda	C	30	0
pobtda	C	30	0
ciudtda	C	2	0

cptda	C	5	0
telta	N	20	0
repta	C	30	0
fecenvio	C	2	0

Indices:

a) TDA1. numtda

#### 4.2.4.8 Almacén PRO

Descripción: Datos generales de proveedores.

Campo	Tipo	Longitud	Decimales
numpro	C	3	0
nompro	C	30	0
dompro	C	30	0
colpro	C	30	0
poppro	C	30	0
ciudpro	C	2	0
cpro	C	5	0
telpro	N	20	0
repro	C	30	0

Indices:

a) PRO1. numpro

### 4.2.5 DESCRIPCION DEL MODULO DE ACTUALIZACION DE EXISTENCIAS.

#### 4.2.5.1 Introducción

El módulo de actualización de existencias, desde ahora denominado ACT\_EXI actualiza las existencias de los artículos registrados en un paquete de movimientos enviado por alguna tienda.

#### 4.2.5.2 Detalle del proceso

El proceso toma los movimientos contenidos en un archivo de recepción de movimientos (TMOV) y actualiza el archivo de existencias (EXI) de acuerdo a los tipos de movimientos. También son actualizados los costos promedio de los artículos en el catálogo de artículos (CART).

En la figura 4.7 se muestra el código del módulo ACT\_EXI, de acuerdo a las especificaciones declaradas en la ERS (Función 4.1.3.2.1.2). Las variables utilizadas son comentadas dentro del código.

```

*-----
* Modulo : ACT_EXI
* objetivo : Actualización de existencias
* F.Crea. : 28/Jun/92
* Ult.Rev : 28/Jun/92
*-----

```

```

// Los archivos utilizados por el módulo son abiertos en
// el módulos de recepción de datos de la siguiente forma:

```

```

// select 1
// use tmov
// select 2
// use exi index exi1
// select 3
// use cart index cart1

```

```

// Declaracion de variables
local xsaldod //existencias iniciales
local xnumarts //cantidad de articulos en el movimiento
local xnvosaldo //nuevo saldo
local xppcosto //promedio del precio de costo

```

```

// Definición de tipos de movimiento

```

```

#Define EMBARQUE '01'
#Define TRAN_ENT '02'
#Define DEV_CLIÉ '03'
#Define TRAN_SAL '04'
#Define DEV_PROV '05'
#Define VENTA '06'
#Define AJUSTE '07'

```

```

// Se relacionan los detalles del movimiento con el
// catalogo de artículos para tener el precio de costo

```

```

select tmov
set relation to ccart into cart
go top
do while .not.eof()
// Busca existencia del artículo. Si no hay existencia
// registradas, se graba un registro.
// Guarda la existencia actual en xsaldod
xsaldod =0
xnvosaldo =0
xnumarts =0
xppcosto =0
select exi
seek tmov->numtda+tmov->ccart+tmov->cod_color+tmov->cod_talla
if eof()
append blank
replace numtda with tmov->numtda ;;
ccart with tmov->ccart ;;
cod_color with tmov->cod_color ;;

```

```

        cod_talla with tmov->cod_talla      ;;
        saldod with 0                       ;;
        saldorm with 0
    endif
    xsaldod=saldod
// Clasifica el movimiento como entradas (sumar) o salidas (restar)
select tmov
do case
case tipomov=EMBARQUE.or.tipomov=TRAN_ENT.or.;
tipomov=DEV_CLIE.or.tipomov=AJUSTE
xnumarts=cantidad
case tipomov=TRAN_SAL.or.tipomov=DEV_PROV.or.tipomov=VENTA
xnumarts=-cantidad
endcase
xnvosaldo=xsaldod+xnumarts
// Se calcula el promedio del precio de costo a nivel modelo,
// nivel determinado por el código del artículo
// Un ajuste no afecta el costo promedio, por lo tanto se toma el
// costo promedio del catálogo de artículos, y no del movimiento
if tipomov<>AJUSTE
xppcosto=(xnumarts*tmov->pcosto+xsaldod*cart->pcosto) / xnvosaldo
else
xppcosto=cart->pcosto
endif
// Actualiza precio de costo
replace cart->pcosto with xppcosto
// Actualiza existencias
replace exi->saldod with xnvosaldo
select tmov
skip
enddo
select tmov
set relation to
return

```

*Figura 4.7 Listado del programa de actualización de existencias*

## 4.2.6 INTERFAZ CON EL USUARIO

La interfaz del sistema con el usuario será mostrando menús de barras para el acceso a las opciones disponibles, Dispondrá de pantallas de ayuda para la captura de claves de artículos, colores, tallas, proveedores, y tiendas.

La activación será a través del comando `c:[directorio]\srdi`

## 4.2.7 RECUPERACION DE FALLAS

El sistema contará con un módulo de reconstrucción de índices dañados por alguna causa imprevista, como fallas en el suministro de energía eléctrica o del equipo de cómputo.

## **CAPITULO 5**

# **RESULTADOS Y CONCLUSIONES**

Tomando en cuenta el avance de la tecnología en nuestros tiempos y los cambios que se están dando en nuestro país en materia de comercio (Tratado de Libre Comercio), se espera tener un mercado más competitivo y organizado, donde la información a tiempo y confiable representa un parámetro muy importante en la toma de decisiones, ya que es un reflejo de lo que se está presentado en la venta y manejo del producto y sirve a los directivos de empresas para detectar sus aciertos o puntos de corrección para lograr ser competitivos a nivel nacional o internacional.

El sistema de inventarios ofrece un procesamiento distribuido que le permite a cada sucursal, tener un control directo y automático de sus inventarios, así como mantener comunicación con la central vía *modem* para transferencia de movimientos que serán tomados para mantener actualizado y correcto su inventario, tanto en central como en sucursal.

Los sistemas de sucursal y central se han diseñado de tal manera que cubran los requisitos que la empresa necesita para funcionar correctamente en sus operaciones, como lo es el manejo a nivel detalle de todos los artículos. Esto permite saber en qué color y talla se vendió un artículo y su afectación en el inventario, así como datos de acumulados de ventas y existencias, para que con esta información se pueda obtener un estimado para el siguiente año tanto de ventas como de la existencia que se necesita tener.

Las sucursales podrán saber qué mercancías son las que tienen rezagadas en sus almacenes y cómo poder ofertarlas para obtener un beneficio aceptable y poder reciclar su inventario.

Es importante mencionar que el sistema no realiza la transmisión, sino solamente genera un archivo *ASCCI* con todos los movimientos (Embarques, Devoluciones, Traspasos, Ventas y Ajustes de la tienda) del día, que posteriormente, es tomado por un paquete de comunicaciones el cuál se encarga de transmitirlo a la central.

Para los equipos que se van a utilizar se han presentado las posibles alternativas de solución y la selección se ha hecho en base a presupuestos obtenidos de proveedores, considerando una serie de características de desempeño similares.

Apoyándonos en la base de que los equipos de Microcomputadoras cumplen todas las expectativas que el cliente necesita, se tomó la decisión de utilizar un equipo *PC* para el desarrollo de nuestra aplicación. Se utilizó el lenguaje de programación *Clipper*, que ofrece una plataforma de desarrollo flexible y versátil, como se mostró en el capítulo 3.

Para el análisis y diseño del sistema, se utilizó la metodología *Yourdon*, que comparada con otras metodologías se consideró la más práctica, por el desglose simple de cada una de las fases de desarrollo y por su enfoque altamente gráfico, que permite describir de una manera sencilla los elementos de un sistema.

La experiencia que el sistema desarrollado nos ha aportado es provechosa, ya que ha permitido aplicar gran parte de los conocimientos adquiridos en la Universidad. No obstante, conforme el tiempo avanza, los métodos y técnicas utilizados pueden variar de manera sensible, por lo que el Ingeniero debe mantenerse actualizado, buscando siempre seguir adelante en su tarea de crear y mejorar los procesos productivos, para bien de la sociedad.

**APENDICE A**

**METODOLOGIAS DE ANALISIS**

En este apéndice se presentan las metodologías utilizadas para la realización del sistema.

La metodología de *Pressman* es aplicable al desarrollo global de un sistema, en tanto que la metodología de *Yourdon* está dirigida al análisis y desarrollo de sistemas de *Software*.

## A.1 METODOLOGIA DE PRESSMAN

*Roger S. Pressman* presenta una metodología para el análisis de sistemas en su libro "*Ingeniería de Software*", un enfoque práctico [PRE88].

Algunas veces se cae en la confusión porque el término análisis de sistemas se usa a menudo en un contexto que alude sólo a las actividades de análisis de los requerimientos del *Software*.

El análisis de sistemas se realiza con los siguientes objetivos presentes:

- 1) Identificar las necesidades del cliente
- 2) Evaluar la viabilidad del sistema
- 3) Realizar un análisis técnico y económico
- 4) Asignar funciones al *Software*, al *Hardware*, a la gente, a la base de datos y a otros elementos del sistema
- 5) Establecer restricciones de costo y de tiempo
- 6) Crear una definición del sistema que sea la base para todo el trabajo de ingeniería subsecuente

### Metodología de Análisis

- 1) Identificación de las necesidades. El analista asiste al cliente definiendo los objetivos del sistema: la información que se va a producir, a suministrar, las funciones y el rendimiento requerido. Se debe distinguir entre lo que el cliente necesita y lo que el cliente quiere.
- 2) Estudio de viabilidad. El estudio de viabilidad se centra en 4 áreas de interés primario.  
  
-Viabilidad económica. Una evaluación del costo de desarrollo frente al beneficio final producido por el sistema desarrollado.

-Viabilidad técnica. Un estudio de funcionalidad, rendimiento y restricciones que pueden afectar la posibilidad de realizar un sistema aceptable.

-Viabilidad legal. Una determinación de cualquier infracción, violación o ilegalidad que pudiera resultar del desarrollo del sistema.

-Alternativas. Una evaluación de aproximaciones alternativas al desarrollo del sistema.

Un estudio de viabilidad no está garantizado para los sistemas en los que la justificación económica es obvia, el riesgo técnico es bajo, se esperan pocos problemas legales, y no existe una alternativa razonable. Sin embargo o si cualquiera de las condiciones falla, un estudio debe ser realizado.

- 3) Análisis económico. El análisis de costos y beneficios marca los costos del desarrollo del proyecto y los contrasta con los beneficios tangibles e intangibles de un sistema.
- 4) Análisis técnico. Se evalúan los elementos técnicos del concepto del sistema, mientras que al mismo tiempo se recoge información adicional sobre el rendimiento, la fiabilidad, la facilidad de mantenimiento y posibilidad de producción.

## A.2 METODOLOGIA DE YOURDON

Generalmente se entiende por metodología refiriéndonos a cualquier ámbito o trabajo, a un sistema ordenado de proceder para la obtención de un fin [FUE91].

Si nos centráramos en el entorno informático, es decir en la producción o desarrollo de sistemas informáticos, es evidente que el uso de una metodología en este proceso aporta unas ventajas que hacen aconsejable su uso.

El problema en el momento actual es elegir una de las disponibles en el mercado con el suficiente conocimiento de causa.

Una primera observación nos lleva a una separación entre metodologías *Públicas*, o sea aquellas cuya utilización no lleva al usuario al pago de ninguna cantidad a las entidades que la crearon, y *Privadas* aquellas desarrolladas por entidades del mismo tipo y que, por tanto, basan sus beneficios en el cobro de licencias de uso como si de cualquier otro producto *Software* se tratara.

Dentro de las metodologías públicas se pueden distinguir tres corrientes actualmente:

- 1) La francesa, que dió como fruto la metodología *Merise*, potenciada por la administración francesa a partir del año 1977.
- 2) La inglesa, también impulsada por la administración en Gran Bretaña y que dió lugar, a partir de 1981, *Ssadm*.
- 3) La americana, basada en la teorías de *Edward Yourdon* y que tiene algunas variantes aportadas por otros autores como *DeMarco*, *Gane* y *James Martin*.

El resto de las metodologías existentes, tanto públicas como privadas, deben considerarse como adaptaciones, más o menos mejoradas, de las citadas anteriormente.

También hay que considerar que los objetivos que persiguen todas ellas son parecidos, por tanto, es evidente que para hacer un desarrollo estructurado y ordenado de una aplicación, los caminos seguidos, o sea, las fases y su cronología, no pueden ser muy diferentes.

Pero el futuro del uso de las metodologías pasa por su soporte en herramientas informatizadas, de las englobadas actualmente bajo el nombre de *CASE*. Es por esto que la tendencia general es que, cada vez menos, cada cual haga la guerra por su cuenta. Afortunadamente, van pasando los tiempos en que cada casa fabricante seguía su línea y se imponen líneas comunes a la hora de buscar estándares en casi todos los terrenos.

A nivel europeo, todos trabajan en la elaboración del *Eurométodo*, proyecto impulsado por la Comunidad Europea en un intento de unificar las tendencias actuales sobre metodología.

Yourdon es el representante de la corriente metodológica más importante de Estados Unidos, aunque hay numerosos autores que aportan variantes, matices y formalismos de representación al método de Yourdon.

A lo largo de sus obras Yourdon describe técnicas para la realización de análisis estructurado de sistemas basado principalmente en los siguientes conceptos:

- 1) Diagramas de flujo de datos para la representación de procesos.
- 2) Diagramas de transición de estados para la representación estructurada de las funciones a realizar en los procesos.

- 3) **Modelo Entidad / Relación** para la representación conceptual de datos.
- 4) **Diccionario de datos** como base o soporte de información del sistema.
- 5) **Diagramas o mapas de estructura** para la representación modular de los procesos y las variables intercambiadas entre ellos.
- 6) **Especificaciones de programas** basadas en lenguaje estructurado y tablas de decisión.

### **Diagramas de flujo de datos**

Se utilizan para la representación gráfica de procesos y datos. En los *DFD's*, se usan los siguientes elementos:

#### ***Procesos:***

Se representan por medio de círculos poniendo en el interior el nombre del proceso.

Representan las operaciones, manuales o mecanizadas, que se realizan con los datos.

Deben tener, al menos, un flujo de entrada de datos y uno de salida.

Para indicar la secuencia de los procesos, en el interior del círculo, además del nombre del proceso, se pondrá un número de orden de ejecución del mismo.

En una primera fase deben ser muy generales con la idea de dar una visión global del sistema y de los objetivos a cubrir. Posteriormente, cada círculo o proceso dará lugar en un *DFD* de mayor nivel de detalle a una representación gráfica del mismo. Al *DFD* más general se le llama también *Diagrama De Contexto*. Dará una imagen de las relaciones del sistema con otros subsistemas y determinará los límites del mismo.

#### ***Flujo de datos:***

Representa el movimiento de información o de objetos entre las personas o departamentos contemplados.

Gráficamente se dibujan como líneas que unen al emisor con el receptor de la información u objeto, indicando el sentido del movimiento por medio de una punta de flecha.

Sirven como base a los lineamientos administrativos de la empresa.

**Entidades:**

Son las personas o servicios que reciben o emiten algún flujo de información.

Se representan como rectángulos en cuyo interior figura el nombre de la entidad.

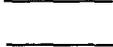
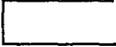
**Almacenamiento:**

Se representa con dos líneas paralelas en cuyo interior se escribe el nombre del archivo.

Corresponden a los conjuntos básicos de información de la empresa y podrán ser manuales o informatizados y serán origen o destino de un flujo de datos. A su vez proporcionarán entradas de información a los procesos o serán salida de los mismos.

En cuanto a los símbolos empleados para la representación de los DFD's, existen dos variantes principales: la de *Yourdon/DeMarco* y la de *Gane/Sarson*.

A continuación se muestran las diferencias:

CONCEPTO	YOURDON/DEMARCO	GANE/SARSON
PROCESO		
ENTIDAD		
FLUJO DE DATOS		
ALMACENAMIENTO		

*Figura A.1 Símbolos de los DFD's según Yourdon/DeMarco y Gane/Sarson*

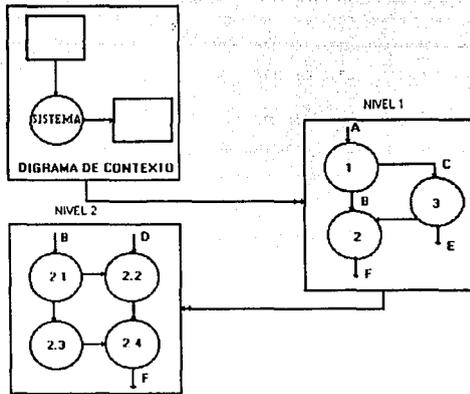


Figura A.2 Niveles de descomposición de los DFD's

### Diagramas de transición de estados

Mediante estos diagramas se representan las diferentes funciones a realizar indicando su secuencia y las condiciones que manejan su ejecución.

Cada proceso se ve representado por una serie de acciones enmarcadas en rectángulos. Se pasa de una acción a la siguiente a través de una conexión en la que se representa el par condición-acción, es decir, la condición que se debe cumplir para seguir la secuencia por ese lado del diagrama y la acción a realizar en ese caso.

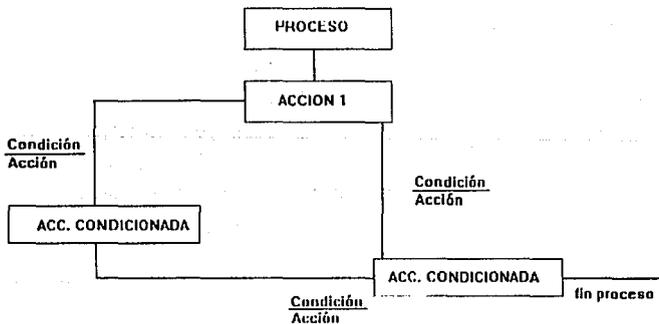


Figura A.3 Diagramas de transición de estados

## Modelo entidad/relación

Para realizar el modelo conceptual de datos, Yourdon propone un esquema de entidades y relaciones del sistema, pasando después a normalizar esta estructura. Esto es una tendencia general en todas las metodologías que es asumida también en ésta.

## Diccionario de datos

Sirve para describir los datos manejados por el sistema, sus características y posibles valores.

Para la descripción de los diferentes datos se utilizan los símbolos siguientes :

- = está compuesto de
- + para unir campos elementales
- () campos opcionales
- [ ] selección de una entre varias opciones
- { } iteraciones
- \*\* comentarios
- @ identificación o clave de un almacenamiento
- | separación de opciones alternativas

Mediante esta notación podría representarse la sintaxis de los datos del sistema. Enseguida se muestra un ejemplo:

- Dirección = Calle + Población + Provincia
- Calle = { Carácter }
- Población = { Carácter }
- Provincia = { Número } O Bien
- Provincia = 1 {Número} 52 \* De 1 A 52 \*
- Caracteres = [ A-Z | 0-9 | , | . ]
- Numero = [ 0-9 ]

## Diagramas de estructura

Mediante ellos se representa gráficamente la estructura de un proceso descomponiéndolo en módulos que intercambian información y se encadenan hasta completar las funciones a realizar.

Cada unidad de un diagrama de estructura se representaría de la siguiente manera, según la notación *Yourdon/Constantine*:

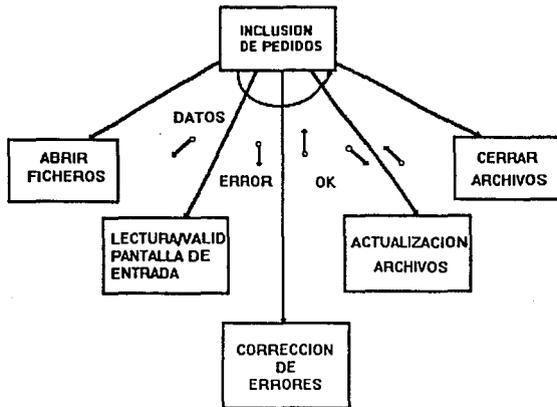


Figura. A.4 Unidad de un diagrama de estructura

Se pueden definir, partiendo de este tipo de unidad, estructuras complejas que representen la secuencia de ejecución de funciones. En este tipo de diagramas se añaden los parámetros o variables que intercambian los diferentes módulos.

## METODOLOGIA

Yourdon define las siguientes etapas y niveles en el ciclo de vida de los sistemas informáticos:

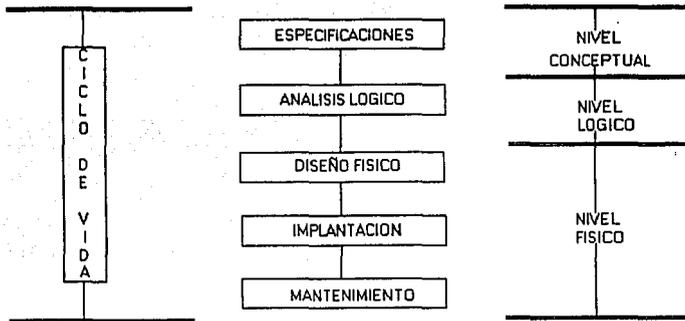


Figura A.5 Ciclo de vida según Yourdon

Por tanto, al aplicar las técnicas descritas anteriormente a lo largo de estas etapas, se distinguen las siguientes actividades a realizar:

### Estudio de viabilidad

En este punto se debe identificar el proyecto a realizar, los usuarios responsables y se debe hacer un estudio de la situación actual, representando la misma a través de *DFD's* de primer nivel o diagramas de contexto en los que, de forma simple, se indiquen los procesos simples más relevantes.

En este estudio de la situación actual se identificarán las deficiencias del mismo como puntos a resolver por el nuevo sistema. Esta actividad debe ocupar entre un 5 y un 10 % del total del proyecto.

### Análisis del sistema

En esta fase se debe representar mediante las técnicas ya vistas de diagramas de flujo, modelo entidad/relación, diagramas de transición de estado, etc, del sistema a desarrollar.

En esta parte se deben tener en cuenta los requerimientos de los usuarios relativos a cambios o funcionalidad del sistema y esta labor es facilitada por herramientas *CASE* de diagramación y por técnicas de prototipo del sistema.

### Diseño

En esta tercera fase, se pasa del nivel conceptual descrito anteriormente a un nivel de representación lógica de los datos mediante un diseño dependiente del modelo de base de datos elegida (*CODASYL*, relacional, etc.) y una

estructuración de los procesos utilizando diagramas de estructura de los mismos y generando las especificaciones de programa correspondientes.

#### **Implantación o producción**

Comprende la generación de código y el ensamblaje e integración de todos los módulos.

#### **Pruebas del sistema**

Realizar con la totalidad del sistema hasta llegar a la aceptación del mismo por parte del usuario. En esta fase se harán pruebas de integración de funcionamiento conjunto de programas y cadenas.

#### **Control de calidad**

El objetivo de esta actividad es garantizar los controles de calidad del *Software* que puedan estar definidos para la empresa. Esta fase complementa la anterior de forma que el producto final sea de un buen nivel de calidad y cumpla los estándares fijados.

#### **Documentación**

En este apartado se generará toda la documentación necesaria para la instalación del sistema: manuales de usuario, de operación, etc. La documentación interna, o sea, las especificaciones de programas, habrán sido creadas ya anteriormente y utilizadas por los programadores. Con ello el nuevo sistema queda completamente documentado, interna y externamente.

#### **Conversión de los datos del sistema anterior**

La ejecución de esta fase depende evidentemente, del estado anterior a la mecanización del entorno afectado por el proyecto. Si existía ya un sistema informatizado se deben realizar los programas de conversión de datos al nuevo sistema y si anteriormente los archivos eran manuales puede requerirse una grabación y carga previa a la puesta en marcha del sistema.

#### **Instalación**

Comprende la puesta en marcha del sistema y en esta fase son de aplicación las consideraciones, aspectos tales como formación y entrenamiento del usuario, entrega de manuales, procesos paralelos, etc.

**APENDICE B**

**PROTOCOLLO X.25**

A lo largo de la década de los setentas se hizo patente que la solución para establecer a mediano plazo un servicio público internacional de conmutación y transporte de datos era la tecnología de conmutación de paquetes.

Un aspecto técnico importante de las redes de conmutación de paquetes es la interfaz entre los dispositivos conectados y la red.

Una red de conmutación de circuitos provee una vía transparente para los dispositivos conectados que hace parecer que las dos estaciones en comunicación tienen un enlace directo. Sin embargo, en el caso de la red de conmutación de paquetes, las estaciones conectadas deben organizar sus datos en paquetes para la transmisión. Esto requiere un cierto nivel de cooperación entre la red y las estaciones conectadas. Esta cooperación es materializada a través de un estándar de interfaz. El estándar casi universalmente usado para este propósito es el X.25.

En 1974, el *Consejo Consultivo Internacional de Telefonía y Telegrafía (CCITT)* emitió el primer borrador de X.25. Este documento sería revisado en 1976, 1978 y 1980, y de nuevo en 1984 para dar lugar al texto definitivo publicado en 1985. El documento inicial incluía una serie de propuestas sugeridas por *Datapac*, *Telenet* y *Tymnet*, tres nuevas redes de conmutación de paquetes. Desde aquel 1975, X.25 ha ido ampliándose e incorporando numerosas opciones, servicios y funciones.

En la actualidad, X.25 es la norma de interfaz orientada al usuario de mayor difusión de las redes de paquetes de gran cobertura.

En X.25 se definen los procedimientos que realizan el intercambio de datos entre los *Dispositivos de Usuario (ETD)* y un nodo de red encargado de manejar los paquetes (*ETCD*). Su título formal es la interfaz entre equipos terminales de datos y equipos de terminación del circuito de datos para terminales que trabajan en modo paquete sobre redes de datos públicas.

Las redes utilizan el *protocolo X.25* para establecer los procedimientos mediante los cuales dos *ETD* que trabajan en modo paquete se comunican a través de la red.

El *protocolo* del nivel físico define la interfaz física con la red, que será *X.21* a velocidades síncronas entre 2.4 y 48 *Kbps*.

Actualmente la línea de acceso entre la terminal y el centro de conmutación es siempre un circuito fijo; sin embargo, no se excluye la posibilidad de que en un futuro próximo el acceso se obtenga a través de la red telefónica con conmutación o de una red de conmutación de circuitos.

El nivel de enlace transforma el sistema de transmisión de señales que constituye el nivel físico en un sistema que permite la comunicación fiable entre las dos entidades que dialogan mediante el *protocolo* correspondiente.

El nivel de red o simplemente nivel 3, es el propiamente específico del acceso de terminales a redes públicas con conmutación de paquetes ya que especifica la señalización entre el *ETD* y la red, de forma que sea posible establecer y liberar llamadas virtuales, así como controlar las transferencias de datos una vez establecidas las mismas.

La red recibe un paquete de solicitud de llamada a través de un canal lógico no. X indicando el *ETD* destinatario de la llamada.

Cada *ETD* conectado a una red pública de datos tiene asignado un número único en la misma forma que cada teléfono tiene asignado internacionalmente un número único que lo identifica. A partir de este número de *ETD*, la red encamina el paquete hacia el *ETD* destino. Una vez alcanzando dicho *ETD*, la red genera un paquete de llamada entrante a través de un canal lógico libre de la interfaz del *ETD*.

Si el *ETD* destino desea entablar la comunicación virtual, generará un paquete de llamada aceptada por el canal lógico no. Y; el extremo que originó la llamada será informado por la red del establecimiento de la llamada mediante un paquete de llamada conectada por el canal lógico no. X. A partir de este instante la llamada está establecida y ninguno de los paquetes que se intercambien necesita llevar indicación del *ETD* destino sino simplemente el número de canal lógico asignado a la comunicación virtual en cada interfaz local (X o Y respectivamente).

Durante la fase de transferencia de información, es posible la transmisión dúplex de informaciones de usuario por medio de paquetes de datos. Los paquetes de datos son numerados en secuencia por la estación emisora de los mismos; la estación receptora puede controlar el ritmo de entrega de paquetes de datos mediante la generación de las validaciones oportunas que se envían en los paquetes o sobre los propios paquetes de datos.

Inicialmente la estación receptora concede un crédito a la estación transmisora, que indica el máximo número de paquetes de datos que pueden ser emitidos sin recibir validación alguna; al enviar validaciones la estación receptora de datos va actualizando los créditos de la estación transmisora.

El mecanismo por el cual la red, como estación receptora, genera validaciones, se conoce por algoritmo de rotación de ventana.

## **APENDICE C**

# **DIAGRAMAS DE FLUJO DE DATOS DEL SRDI**

Este apéndice contiene los modelos gráficos utilizados en el análisis estructurado del sistema.

En primer lugar se muestra la simbología contenida los diagramas de flujo de datos y en el modelo *Entidad-Relacionamiento*.

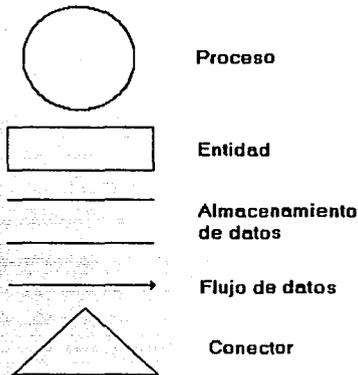


Figura C.1 Simbología.

La figura C.2 es una representación global del SRDI.

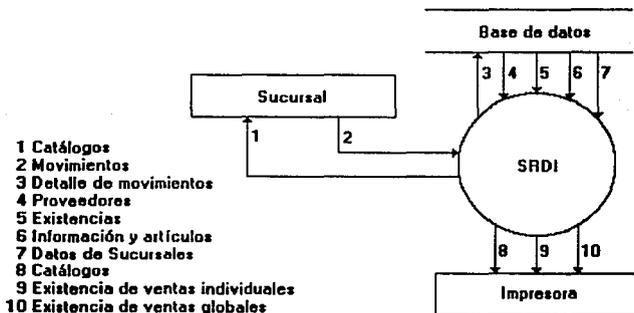


Figura C.2 Diagrama de Contexto.

Los DFD's para SRDI son detallados hasta el segundo nivel. La figura C.3 muestra el diagrama de primer nivel, y las figuras C.4 a C.9 detallan el segundo nivel de los procesos 1.0, 2.0, 3.0, 4.0, 5.0 y 8.0.

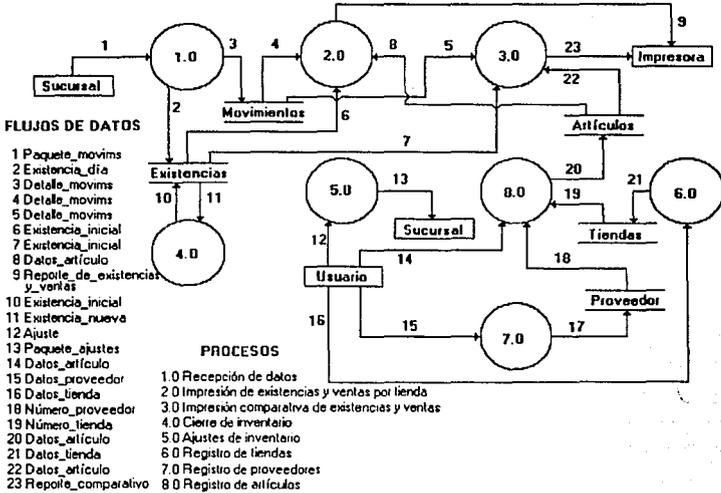


Figura C.3 Diagrama de primer nivel.

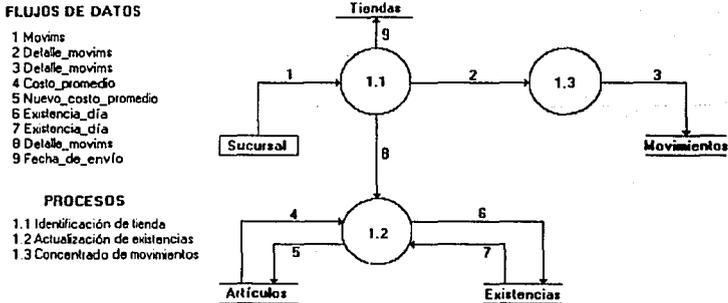


Figura C.4 Diagrama de recepción de datos.

**FLUJOS DE DATOS**

- 1 Rango\_de\_tiempos
- 2 Movims\_filtrados
- 3 Fecha
- 4 Detalle\_movims
- 5 Datos\_articulo
- 6 Movims\_y\_existencia
- 7 Existencias\_y\_ventas
- 8 Existencia\_inicial
- 9 Movims\_tienda\_articulo

**PROCESOS**

- 2.1 Selección de movimientos
- 2.2 Totalizado de movimientos por artículo
- 2.3 Recuperación de existencias
- 2.4 Reporte de existencias y ventas

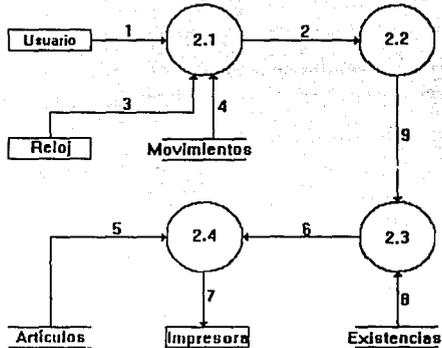
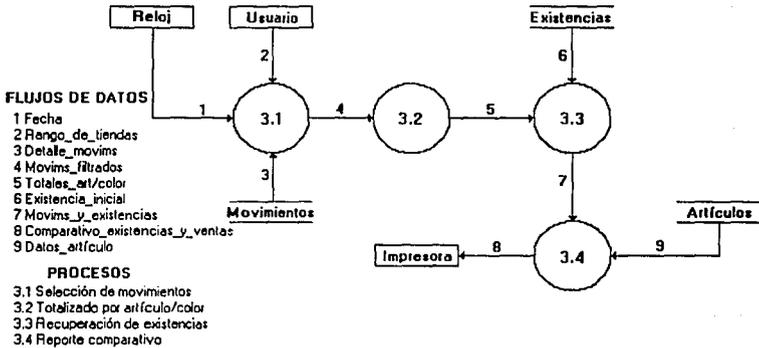


Figura C.5 Diagrama de impresión de existencias y ventas



**FLUJOS DE DATOS**

- 1 Fecha
- 2 Rango\_de\_tiempos
- 3 Detalle\_movims
- 4 Movims\_filtrados
- 5 Totales\_art/color
- 6 Existencia\_inicial
- 7 Movims\_y\_existencias
- 8 Comparativo\_existencias\_y\_ventas
- 9 Datos\_articulo

**PROCESOS**

- 3.1 Selección de movimientos
- 3.2 Totalizado por artículo/color
- 3.3 Recuperación de existencias
- 3.4 Reporte comparativo

Figura C.6 Diagrama comparativo de existencias y ventas

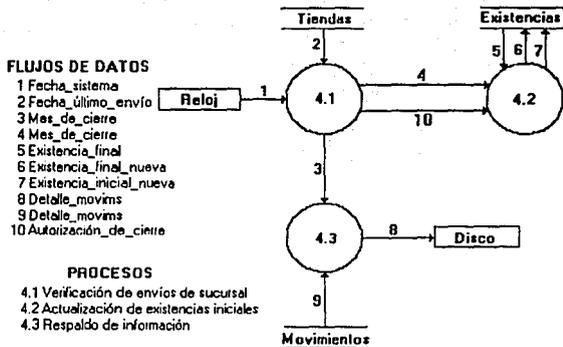


Figura C.7 Diagrama de cierre de inventario.

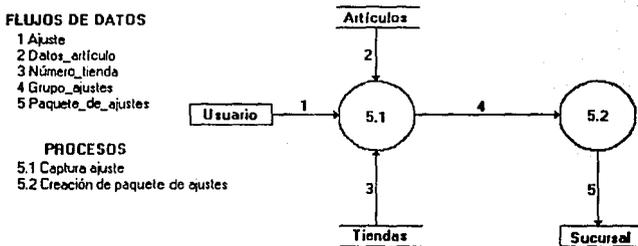


Figura C.8 Diagrama de ajuste de inventarios.

### FLUJOS DE DATOS

- 1 Datos\_artículo
- 2 Número\_proveedor
- 3 Datos\_artículo
- 4 Código\_artículo
- 5 Código\_color
- 6 Número\_tienda
- 7 Código\_color
- 8 Código\_artículo
- 9 Número\_tienda
- 10 Código\_artículo

### PROCESOS

- 8.1 Captura de descripción de artículo
- 8.2 Registro en tiendas
- 8.3 Ajuste de colores

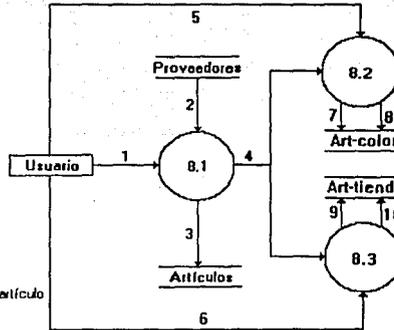


Figura C.9 Diagrama de captura de artículos.

## DICCIONARIO DE DATOS

El diccionario de datos es una descripción detallada de los flujos detectados en los DFD's mostrados anteriormente.

Los símbolos del diccionario de datos son usados en la siguiente forma :

= Esta compuesto de

+ Y

{ } Iteración

\* \* Comentarios

| Separa alternativas en la construcción [ ]

AJUSTE = CODIGO\_ARTICULO + NUMERO\_TIENDA + CODIGO\_COLOR + CODIGO\_TALLA + CANTIDAD

ART\_COLOR = CODIGO\_ARTICULO + CODIGO\_COLOR

ART\_TIENDA = CODIGO\_ARTICULO + NUMERO TIENDA

DATOS\_ARTICULO = CODIGO\_ARTICULO + GRUPO + LINEA + SUBLINEA + MODELO + CODIGO\_CORRIDA + DESCRIPCION + COSTO\_PROMEDIO + PRECIO\_VENTA

DATOS\_PROVEEDOR = NUMERO\_PROVEEDOR + NOMBRE\_PROVEEDOR + DIRECCION\_PROVEEDOR + TELEFONO\_PROVEEDOR + REPRESENTANTE\_PROVEEDOR

DATOS\_TIENDA = NUMERO\_TIENDA + NOMBRE\_TIENDA + DIRECCION\_TIENDA + TELEFONO\_TIENDA + REPRESENTANTE\_TIENDA + ULTIMO\_ENVIO

DETALLE\_MOVIMS = NUMERO\_TIENDA + CODIGO\_ARTICULO + TIPO\_MOVIMIENTO + CODIGO\_COLOR + CODIGO\_TALLA + CANTIDAD

EXISTENCIA = NUMERO\_TIENDA + CODIGO\_ARTICULO + CODIGO\_COLOR + CODIGO\_TALLA + EXISTENCIA\_MES + EXISTENCIA\_DIA

MOVIMS\_Y\_EXISTENCIAS = NUMERO\_TIENDA + CODIGO\_ARTICULO + TIPO\_MOVIMIENTO + CANTIDAD\_EMBARQUES + CANTIDAD\_TRASP\_ENT + CANTIDAD\_TRASP\_SAL + CANTIDAD\_DEV\_PROV + CANTIDAD\_DEV\_CLI + CANTIDAD\_VENTAS + CANTIDAD\_AJUSTES + EXISTENCIA\_INICIAL

ALFANUMERICO = [ DIGITO | CARACTER ]

AÑO = { DIGITO } 2

AUTORIZA\_CIERRE = \* SEÑAL DE AUTORIZACION DE CIERRE DE INVENTARIO \*

CANTIDAD = { DIGITO } 3

CARACTER = [ A | a | B | b | C | c | ... Z | z ]

CODIGO\_ARTICULO = { DIGITO } 7

CODIGO\_TALLA = DIGITO + DIGITO

COMPARATIVO\_EXISTENCIAS\_Y\_VENTAS = \* REPORTE COMPARATIVO \*

COSTO\_PROMEDIO = { ALFANUMERICO } 7

DESCRIPCION = { ALFANUMERICO } 40

DIA = { DIGITO } 2

DIGITO = [ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ]

DIRECCION\_PROVEEDOR = { ALFANUMERICO } 60

DIRECCION\_TIENDA = { ALFANUMERICO } 60

EXISTENCIA = { DIGITO } 5

EXISTENCIA\_DIA = EXISTENCIA

EXISTENCIA\_FINAL = EXISTENCIA

EXISTENCIA\_FINAL\_NUEVA = EXISTENCIA

EXISTENCIA\_INICIAL = EXISTENCIA

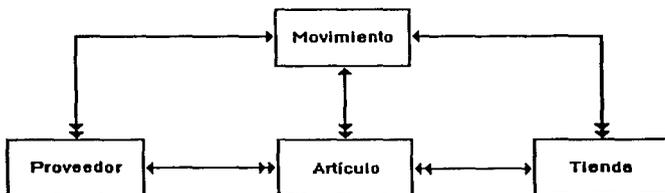
EXISTENCIA\_INICIAL\_NUEVA = EXISTENCIA

EXISTENCIA\_MES = EXISTENCIA

EXISTENCIAS_Y_VENTAS =	MOVIMS_Y_EXISTENCIAS
FECHA =	DIA + MES + AÑO
FECHA_DE_ENVIO =	FECHA
FECHA_SISTEMA =	FECHA
FECHA_ULTIMO_ENVIO =	FECHA
GRUPO =	{ DIGITO } 2
LINEA =	{ DIGITO } 3
MES=	{ DIGITO } 2
MODELO =	{ DIGITO } 6
MOVIMS_FILTRADOS =	* UN GRUPO DE DETALLE_MOVIMS *
NOMBRE_PROVEEDOR =	{ ALFANUMERICO } 30
NOMBRE_TIENDA =	{ ALFANUMERICO } 30
NUEVO_COSTO_PROMEDIO =	{ DIGITO } 7
NUMERO_PROVEEDOR =	{ DIGITO } 4
NUMERO_TIENDA =	{ DIGITO } 3
PAQUETE_DE_AJUSTES =	{ AJUSTE } * PARA LA MISMA TIENDA *
PAQUETE_DE_MOVIMS =	{ DETALLE_MOVIMS } * DE UNA TIENDA *
PRECIO_VENTA =	{ DIGITO } 7
RANGO_DE_TIENDAS =	* TIENDA_INICIAL + TIENDA_FINAL *
REPRESENTANTE_PROVEEDOR=	{ ALFANUMERICO } 30
REPRESENTANTE_TIENDA =	{ ALFANUMERICO } 30
SUBLINEA =	{ DIGITO } 2
TELEFONO_PROVEEDOR =	{ ALFANUMERICO } 20
TELEFONO_TIENDA =	{ ALFANUMERICO } 20
TIENDA_FINAL =	NUMERO_TIENDA
TIENDA_INICIAL =	NUMERO_TIENDA
TIPO_MOVIMIENTO =	* EMBARQUE   TRASPASO_ENTRADA   TRASPASO_SALIDA   DEVOLUCION_PROVEEDOR   DEVOLUCION_CLIENTE   AJUSTE   VENTA *
TOTALES_ART_COLOR =	* MOVIMIENTOS TOTALIZADOS POR ARTICULO-COLOR *
TOTALES_TIENDA_ART=	* MOVIMIENTOS TOTALIZADOS POR TIENDA-ARTICULO *

## MODELO ENTIDAD-RELACIONAMIENTO

La figura C.10 muestra las entidades para *SRDI* y sus interrelaciones.



*Figura C.10 Modelo Entidad-Relacionamiento.*

# GLOSARIO

**Almacenamiento de Datos** Cualquier lugar de un sistema donde se almacenan los datos entre transacciones o entre ejecuciones del sistema (incluye archivos manuales y legibles por máquina, base de datos y tablas).

**Arbol** Estructura de datos en la que cada nodo, excepto la raíz, tienen únicamente un padre

**Archivo** Colección de registros con la misma composición de campo.

**ASCII** American Standard Code for Information Interchange: Código estándar americano para el intercambio de información. Existen dos tipos de códigos ASCII: ASCII-7 y ASCII-8.

**Base de datos** Conjunto de archivos interrelacionados creados y manipulados por un DBMS.

**Bit** Unidad mínima de almacenamiento de información. Puede tomar valores de 0 o 1

**Bps** Bits por segundo.

**Buffer** Espacio temporal de memoria reservado por el sistema operativo. Su contenido puede ser transferido a memoria primaria o secundaria.

**Byte** Unidad de almacenamiento de memoria equivalente a 8 bits, usado para representar un carácter.

**CALD** Central Automática de Larga Distancia.

**Campo** Es la unidad mínima de almacenamiento que contiene una base de datos.

**CCITT** Consejo Consultivo Internacional de Telefonía y Telegrafía

**CODASYL** Conference on Data System Language. Organización que establece los estándares para el lenguaje cobol y los sistemas administradores de bases de datos CODASYL.

**CPU** Central Processing Unit.

**CRC** Prueba de redundancia cíclica.

**DACS** Digital Access Crossconexion System.

**DASD** Direct Access Storage Device. Dispositivo de almacenamiento de acceso inmediato, ejemplos: disco magnético, memoria burbuja.

**DBA** Database Administrator. Uno o más individuos responsables del control y manejo del sistema de la base de datos.

**DBMS** Database Management System. Sistema de software utilizado para el soporte de bases de datos.

**DBS** Database System.

**Debugger** Es una utilidad que ejecuta un programa línea a línea y ofrece comandos específicamente diseñados para ayudar en la búsqueda de errores.

**Diagrama de Estructura** Un modelo lógico de una jerarquía modular, que muestra invocación, comunicación inter-modular (datos y control), y la ubicación de lazos y decisiones importantes.

**Diagrama de Flujo de Datos (DFD)** Una representación de los flujos de datos a través de un sistema de cualquier tipo, que muestra las entidades externas que son fuente o destino de los datos, los procesos que transforman los datos, y los lugares donde los datos son almacenados.

**Diccionario de Datos** Un almacenamiento de datos que describe la naturaleza de cada parte de los datos empleados en el sistema, que a menudo incluye las descripciones de procesos, entradas del glosario, y otros ítem.

**Diseño** El proceso (iterativo) de tomar un modelo lógico de un sistema con un conjunto de objetivos fuertemente establecidos para este sistema, y producir la especificación de un sistema físico que satisfaga dichos objetivos.

**Diseño Estructurado** Un conjunto de guías para producir una jerarquía de módulos lógicos que representan un sistema altamente modificable.

**Driver** Rutina que controla un periférico.

**Entidad** 1. Entidad externa: fuente o destino de datos en un diagrama de flujo de datos. 2. Algo sobre lo cual se almacena información en un almacenamiento de datos, por ejemplo, clientes, empleados.

**Especificación de Requerimientos (ERS)** Un documento que describe claramente y con precisión cada uno de los requerimientos esenciales del software y las interfaces externas.

**Estructura de Datos** Uno o más elementos de datos con una forma de relación particular, que generalmente se utilizan para describir alguna entidad.

**ETD** Equipo terminal de datos.

**Fibra óptica** Filamento delgado de vidrio o algún otro material transparente, a través del cual puede ser transmitida información.

**FM** Modulación de frecuencia.

**FMS** File Management System. administrador de archivos.

**FSK** Desplazamiento de frecuencia.

**Hardware** Los circuitos y dispositivos periféricos de una computadora.

**Host** Equipo central en una red de computadoras.

**Índice** Un almacenamiento de datos que como parte del proceso de recuperación toma información sobre los valores de algún atributo(s) y retorna con información que permite que el/registro(s) con esos atributos sea(n) recuperado(s) rápidamente.

**Interfaz** Es una definición del formato o estructura en el cual la información es transferida entre dos partes de un sistema.

**ISO** International Standards Organization. Se ocupa de la elaboración de recomendaciones internacionales a partir de propuestas de los países miembros y otros organismos profesionales.

**Kbps** Miles de bits por segundo.

**LAN** Local Area Network Red de computadoras conectadas a una distancia menor a 1 Km.

**Main frame** Término reservado para las computadoras más grandes.

**Mbps** Millones de bits por segundo.

**Memoria primaria** Espacio de almacenamiento accesible al CPU. Contiene programas en ejecución y los datos necesarios para éstos.

**Memoria secundaria** Espacios de almacenamiento de acceso indirecto al CPU. Algunos ejemplos de memorias secundarias son los discos duros y los discos flexibles.

**Microcomputadora** Una computadora pequeña y de bajo costo que incorpora un microprocesador

**Minicomputadora** Una computadora de tamaño mediano, que por lo regular soporta varios usuarios.

**Modem** Modulador-demodulador. Es un dispositivo que convierte las señales digitales en señales acústicas y viceversa.

**Módulo** 1. Módulo lógico: Una función o un conjunto de funciones referidas por un nombre. 2. Módulo físico. Una secuencia de sentencias contiguas de programa, limitada por un elemento vecino, y referida por un nombre.

**Mouse** Dispositivo de entrada manual, el cual opera arrastrándolo en una superficie plana.

**Multiplexor** Dispositivo que permite que varios ETD o puertos compartan una misma línea de comunicaciones, por lo general un canal telefónico.

**NFT** Nantucket Future Technology.

**Nodo** Entidad que ocupa una posición en una estructura de datos.

**OOP** Oriented Object Programming.

**OSI** Open System Interconnection. Es un modelo de comunicación entre dispositivos de cooperación.

**Overlay** Módulo con código ejecutable que puede ser colocado en memoria cuando sea requerido por un módulo de carga.

**Password** Clave de acceso a un sistema.

**Programación Estructurada (codificación)** La construcción de programas empleando un pequeño número de construcciones lógicas, cada una con una entrada, una salida, en una jerarquía enlazada.

**Protocolo** Es un conjunto de reglas bajo las cuales dos procesos pueden ser comunicados. Este término también se utiliza para varios procesos en diferentes computadoras en una red.

**Pseudocódigo** Una herramienta para la especificación de un programa lógico en un formato legible, en forma similar al lenguaje corriente, pero sin observar las reglas sintácticas de ningún lenguaje de programación en particular.

**QBE** Query-By-Example. Es una consulta por medio de ejemplos; utilizado por DBMS relacionales.

**RAM** Random Access Memory. Memoria digital de acceso directo, frecuentemente utilizada como la memoria primaria de una computadora.

**RDI** Red Digital Integrada

**Registro** Entidad representada por un cierto número fijo de campos.

**Relación** Un archivo representado en formato normalizado como una tabla bidimensional de elementos de datos.

**ROM** Read Only Memory.

**RPD** Red pública de datos.

**Runtime** Sistema que demanda respuesta rápida del CPU y que posee una alta prioridad de trabajo como para interrumpir otros procesos.

**SCT** Secretaría de Comunicaciones y Transportes

**Server** Computadora administradora de una red.

**SIDCA** Sistema de Inventarios de Control Automático

**Sistema Operativo** Sistema de Software que administra los recursos de una computadora.

**Software** Programas y datos de una computadora

**SQL** Structure Query Language. Lenguaje de consulta estructurado utilizado por algunos DBMS relacionales.

**Tandem** Equipo de cómputo utilizado para administrar las llamadas telefónicas.

**Telecomm** Telecomunicaciones de México

**Telepac** Red especializada en la transmisión de datos que emplea la técnica de conmutación de paquetes.

**Telmex** Teléfonos de México

**Terminal** Es un monitor y sus dispositivos de entrada asociados, utilizado para interactuar con una computadora de propósito general.

**Transmisión Asíncrona** Transmisión en la cual cada carácter de información es sincronizado independientemente.

**Transmisión Full Duplex** Transmisión de datos en ambas direcciones al mismo tiempo.

**Transmisión Half Duplex.** Transmisión de datos en ambas direcciones, una dirección a la vez.

**USR** Unidad de Servicio Remota

**VAN** Red de valor agregado.

**VMM** Virtual Memory Management

**VSAT** Very Small Aperture Terminal.

**WAN** Wide Area Network. Redes de computadoras conectadas a distancias mayores a 1 Km.

## BIBLIOGRAFIA

---

- [ALA86] Alabau, A., *Teleinformática y Redes de Ordenadores*, Marcombo, 1986
- [BLA91] Black, U., *Redes de Computadoras, Protocolos, Normas e Interaces*, Macrobit, ra\*ma, 1990.
- [BYR91] Byrd, M., "Software de Comunicación Asíncrona: Mejorando el acceso a la información", *PC Magazine*, Año 2, No. 8, Agosto 1991 PP. 30-42.
- [FUE91] Fuensalida, A.L., *Metodologías de Desarrollo: Producción automática de software con herramientas CASE*, Macrobit, ra\*ma, 1991.
- [GAN91] Gane, C. y T. Sarson, *Análisis Estructurado de Sistemas*, El Ateneo, 1991.
- [GER84] Gerez, V., M. Mier, R.Nieva y G. Rodríguez, *Desarrollo y Administración de Programas de Computadora (Software)*, CECSA, 1984.
- [NEW87] Newman, W.M., *Designing Integrated Systems for the Office Environment*, McGraw Hill, 1987.
- [PER92] Perschke, S. y M. Liczbanski, "Pick the Righth Database Management System from the Start", *Data Based Advisor*, Vol. 10, No. 2, Febrero 1992, PP. 98-118.
- [PRE88] Pressman, R.S., *Ingeniería del Software: Un enfoque práctico*, McGraw Hill, 1988.
- [SEN92] Sen, J.A., *Sistemas de Información*, McGraw Hill, 1992.
- [STA90] Stallings, W., *Business Data Communications*, Macmillan Publishing Company, 1990.
- [TAN91] Tanenbaum, A.S., *Redes de Ordenadores, Prentice Hall Hispanoamericana*, 1991.
- [THA90] Thayer, R.H., "Software Requirements Analysis and Specifications: a two-day seminar", *Technology Training Corporation*, 1990.
- [TIL91] Tiley, W.E., *Using Clipper*, Que Corporation, 1991.
- [TSA91] Tsai, A.Y.H., *Sistemas de Base de Datos: Administración y Uso*, Prentice Hall Hispanoamericana, 1991.