

16  
2eg



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE INGENIERIA

CONTROL DE INVENTARIOS PARA LOS  
ALMACENES DE UNA EMPRESA TELEFONICA.

**T E S I S**  
QUE PARA OBTENER EL TITULO DE  
**INGENIERO EN COMPUTACION**  
P R E S E N T A N  
**CHIMAL MARTINEZ HECTOR**  
**GUERRERO ARENAS PATRICIA**

DIRECTOR: M. en I. JUAN CARLOS ROA BEIZA



MEXICO, D. F.

1999

TESIS CON  
FALLA DE ORIGEN

280081



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

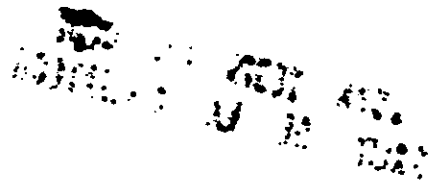
**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

PAS/NACION

DISCONTINUA.



## AGRADECIMIENTOS

**A Dios:** Por estar siempre conmigo, cuidarme y guiarme.

**A mis padres Mario y Lucía:** Porque me han enseñado y corregido, han estado presentes cuando los he necesitado. Porque siempre me dieron lo más dulce, más amado y más deseado: **Libertad** (que significa: "responsabilidad"). Que suerte tengo de tener unos padres como ustedes.

**A mi Tío Saúl:** Que siempre ha sido también mi padre y un gran amigo, que no espera nada a cambio de su amistad y cariño. Que se acerca a mi vida tanto en las buenas como en las malas.

**A mis tíos Moisés y Rosita:** Por que a un niño le encantan los osos grandes, las mochilas nuevas, los pasteles...

**A mi Tita †, mi abuelita Catalina y mis abuelitos, Pafnuncio † y Vicente † :** Porque sé que en donde estan, nos cuidan y bendicen.

**A mis hermanos Claudia I; Luz Ma; Rubén, Jorge, Araceli G. y Vicente:** Porque hemos compartido alegrías y tristezas, les dejo esto: Ser hombre es comprender que la vida no es algo que se nos da ya hecho, sino que es la oportunidad de hacer algo bien y de trascendencia.

**A Miguel Angel:** Por darme tu amor, felicidad y confianza. Por estar a mi lado siempre y ser mi inspiración para conseguir nuevas metas. Te quiero por ignorar mis debilidades y por permanecer firmemente atado a las posibilidades de lo bueno que hay en mi.

**A todos aquellos:** Amigos y familiares que es difícil de enlistar, ustedes saben a quienes me refiero. A todos ustedes gracias por el apoyo y la confianza.

**Patricia Guerrero Arenas**

**A mis Padres Sara y Juan,** por haberme dado la vida y transmitirme una formación de buenos principios para poderme enfrentarme a la vida con valor y determinación y lograr en mi un ser de bien.

**A mi Esposa Diana,** por darme la dicha de ser padre y por ser la compañera, la amiga, la confidente, con quien compartir mis tristezas, alegrías, fracasos, éxitos, gracias por todo tu apoyo incondicional.

**A todos y cada uno de mis Hermanos y sus esposas,** por creer en mi, y siempre estar conmigo en las buenas y en las malas, gracias por todo el apoyo que de mucho me ha servido para realizar este trabajo.

**A todos y cada uno de los compañeros de escuela,** por permitirme compartir sus conocimientos, alegrías en tiempos de estudiantes, gracias.

**A mis Abuelitos Ma. Teresa, \*Manuela, \*J. Carmelo y \*Lorenzo,** por sus consejos que en vida me dieron y me dan, y que siempre llevare conmigo y que forman una parte importante en mi vida.

**A mis Hijos Sigry Itzel, Hectorín y Gerardito,** por ser los seres que Dios me ha permitido tener y poder transmitir en ellos los principios y experiencias que la vida me ha dado.

**A todos y cada uno de mis Maestros,** por haberme transmitido sus conocimientos, que fueron y son la base en mi formación profesional, gracias a todos y cada uno de ellos.

***Héctor Chimal Martínez***

**A la Universidad Nacional Autónoma de México:** Por habernos recibido en sus aulas y darnos con ello bases para superarnos día con día y ser mejores individuos.

**A todos nuestros maestros:** Que nos dieron sus enseñanzas y experiencias para que pudiéramos desempeñar con eficiencia y calidad nuestras labores profesionales.

**A Teléfonos de México:** Que por medio de Inttelmex, nos brindo todo el apoyo para llevar a cabo la conclusión de nuestros estudios profesionales.

**Al Ing. Alejandro Mayoral, Alejandro Méndez, Germán Torres y todos mis compañeros de trabajo** Por su apoyo y amistad incondicional.

**A Gabriel Aceves Hernández y Margarito Guerrero Madrigal:** Por su apoyo y amistad incondicional.

**A nuestra querida Facultad de Ingeniería:** Ya que en sus aulas tuvimos la fortuna de adquirir los conocimientos, y aunque fue severa, siempre nos inculcó amor y respeto al estudio y compañerismo universitario.

**Al M.I. Juan Carlos Roa Beiza:** Por habernos guiado, corregido y dedicado tiempo, para que con su experiencia y conocimiento, se llegara a buen termino este trabajo de Tesis.

**A Fco. Alan Sánchez y Luis Romero:** Por apoyarnos, brindarnos su amistad, tiempo y conocimientos para concluir este trabajo.

**A todos nuestros Amigos y Familiares:** Que durante las distintas etapas de nuestra vida, nos han apoyado y acompañado en los buenos y malos momentos.

**Al ing. Raúl González Martínez:** Por su apoyo y comprensión.

**Patricia Guerrero Arenas.**

**Héctor Chimal Martínez.**

## ÍNDICE TEMÁTICO

### CAPÍTULO 1. MARCO TEÓRICO

1.1. Conceptos básicos de inventarios.....	1
1.1.1 Introducción.....	1
1.1.2 Categorías de inventarios.....	8
1.1.3 Regla de administración de los inventarios.....	9
1.1.4 Costo de abastecimiento y de almacenamiento.....	17
1.1.5 Política de administración de inventarios.....	23
1.1.6 Planificación global.....	24
1.1.7 Etapas de la planificación y del control de inventarios.....	26
1.1.8 Planificación de los inventarios.....	28
1.1.9 Ventajas y desventajas de los inventarios.....	28
1.2. Teoría de programación estructurada.....	30
1.3. Metodología de diseño.....	42
1.3.1. Yourdon.....	43
1.3.2. Chen.....	56
1.3.3. Merise.....	68
1.3.4. Elección de la metodología óptima.....	74
1.4. Sistemas Operativos.....	75
1.4.1. Unix.....	77
1.4.2. Windows 9x/Windows NT.....	94
1.5. Comunicaciones.....	108
1.5.1. Redes de Computadoras.....	108
1.5.2. Modelos de interconexión.....	114
1.5.3. Modelo OSI.....	118
1.5.4. Arquitectura TCP/IP.....	120
1.5.5. Pasado y presente.....	122
1.5.5.1. Servicio en modo conexión y en modo sin conexión.....	122
1.5.5.2. Direccionamiento.....	122



1.5.5.3.	Ruteo.....	130
1.5.5.4.	Servicio de Nombres .....	132
1.5.5.5.	Suite del protocolo TCP/IP .....	139
1.5.5.5.1.	Aplicaciones y utilidades.....	139
1.5.5.5.2.	Protocolo de control de transferencia .....	144
1.5.5.5.3.	Protocolo de transferencia de archivos simple .....	149
1.5.5.5.4.	Protocolos TCP/IP .....	150
1.5.5.5.5.	Protocolos de acceso de Red.....	156
1.6	Modelos de Bases de datos.....	160
1.6.1	Jerárquico.....	161
1.6.2	De red.....	163
1.6.3	Relacional.....	164
1.7	Teoría de Bases de Datos Relacionales.....	171
1.7.1	Elementos del Modelo de Datos.....	172
1.7.2	La estructura de la Base de Datos .....	177
1.7.3	Normalización.....	194
1.8.	Características, Ventajas y desventajas de Visual Basic 5.0 .....	203
1.9.	Características, Ventajas y desventajas de Access .....	219

## **CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN**

2.1.	Situación actual.....	239
2.2.	El usuario y sus requerimientos .....	250
2.3.	Recopilación y análisis de la información.....	256
2.4.	Planteamiento del problema.....	271
2.5.	Descomposición funcional.....	281
2.6.	Opciones de solución .....	285
2.7.	Elección de la solución óptima .....	300

**CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA**

3.1. Aplicación de la metodología elegida (Back End para cada módulo.....)	309
3.1.1. Diagrama de contexto.....	309
3.1.2. Diagrama de flujo de datos.....	310
3.1.3. Diccionario de datos.....	324
3.1.4. Diagrama de entidad Relación.....	345
3.1.5. Normalización.....	347
3.2. Generación de código para el procesamiento de la información.....	354
3.3. Diseño y construcción del Front -End.....	374
3.4. Pruebas e integración del Sistema.....	385
3.5. Generación de reportes para la toma de decisiones.....	408
3.6. Factibilidad Técnica y operativa.....	421
<b>MANUAL DE USUARIO.....</b>	<b>429</b>
<b>MANUAL TÉCNICO.....</b>	<b>461</b>
<b>CONCLUSIONES.....</b>	<b>485</b>
<b>BIBLIOGRAFÍA.....</b>	<b>488</b>
<b>APÉNDICE A.....</b>	<b>A-1</b>
Conceptos básicos de Contabilidad Financiera.....	A-1
Origen y Definición.....	A-1
Importancia y utilidad.....	A-2
Principios de contabilidad generalmente aceptados.....	A-4
Registro de operaciones.....	A-8
Ecuación básica de la contabilidad.....	A-9
Definición y clasificación del activo.....	A-9
Definición y clasificación del pasivo.....	A-12
Definición de capital contable.....	A-14
Estados financieros.....	A-16
Objetivo y utilidad.....	A-16

Balance General .....	A-18
<b>APÉNDICE B</b> .....	B-1
Diccionario de Datos .....	B-1
<b>APÉNDICE C</b> .....	C-1
Normalización .....	C-1
<b>APÉNDICE D</b> .....	D-1
Generación de Código .....	D-1
<b>APÉNDICE E</b> .....	E-1
Glosario .....	E-1

# **CAPITULO 1. MARCO TEORICO**

## **1.1. Conceptos básicos de inventarios**

### **1.1.1. Introducción**

La escasez, sobreabundancia, y falta de abastecimiento de materias primas, así como otros problemas, que han experimentado las empresas, son aspectos que han contribuido a la toma de conciencia sobre la importancia de una administración económica de los inventarios.

Los materiales son las materias primas, los componentes, los subensambles y los artículos que se usan para producir un bien o servicio.

La mayoría de los materiales son transformados en productos terminados, pero las provisiones de ellos son consumidas en las operaciones diarias. Los materiales se convierten en costos directos, mientras que las provisiones son clasificadas como costos indirectos.

La administración de materiales es la planeación, la organización y el control del flujo de materiales, desde su compra inicial, pasando por las operaciones internas, hasta la distribución de los productos finales. La figura 1.1.1.1. identifica los principales aspectos de la administración de materiales como:

- 1) Compras.
- 2) Transporte (interno y externo).
- 3) Control a través de la administración de producción de inventarios (incluyendo recepción, almacenamiento, embarque, manejo de materiales y conteo de inventarios).
- 4) Almacenamiento y distribución.

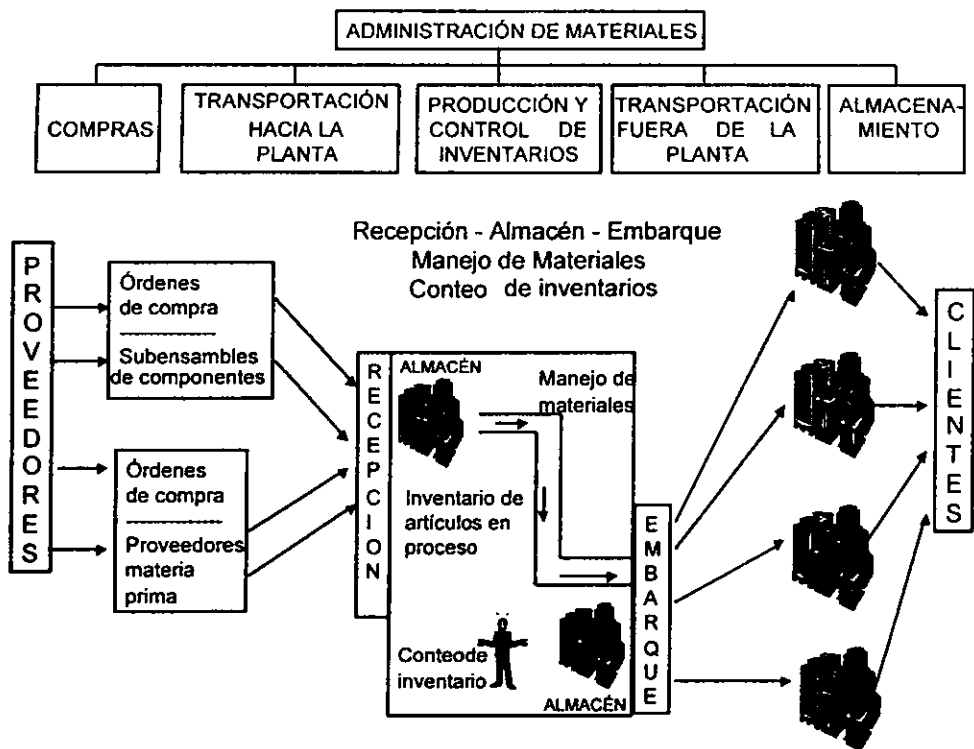


Figura 1.1.1.1. Principales elementos del sistema de administración de materiales.

### Propósito de los inventarios

Los inventarios son recursos ociosos que poseen un valor económico. Las empresas generalmente clasifican sus inventarios como:

- Materias primas
- Producto en proceso
- Producto terminado

Todos los inventarios representan una inversión designada para facilitar las actividades de producción y servir a los consumidores. Sin embargo, mantener inventarios consume capital de trabajo, el cual puede no estar proporcionando un ingreso o un

rendimiento en una inversión y puede ser requerido urgentemente en cualquier momento.

Consecuentemente el problema de la administración de inventarios (figura 1.1.1.2. y 1.1.1.3.) es el mantenimiento de niveles de inventarios adecuados, pero no excesivos.

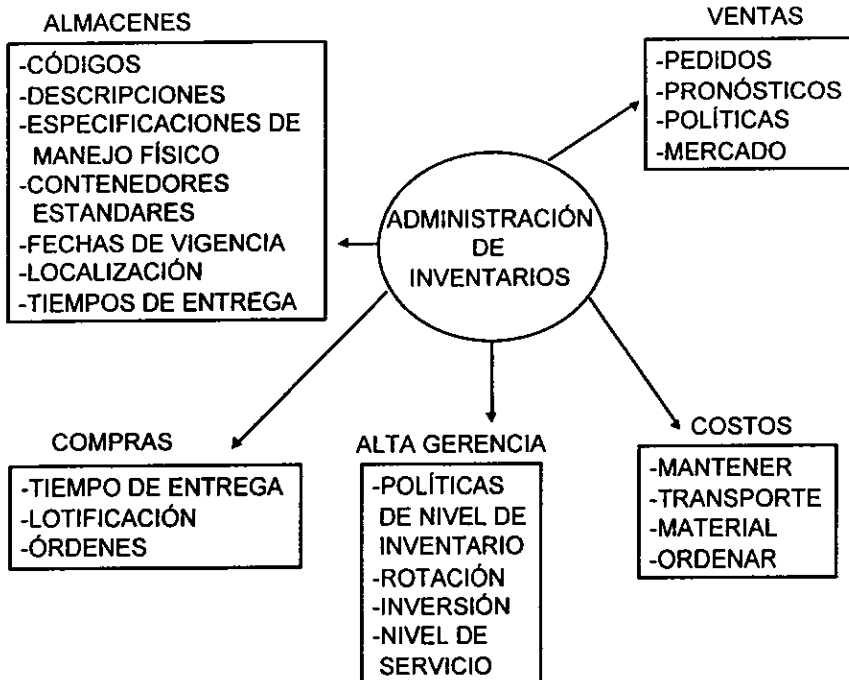
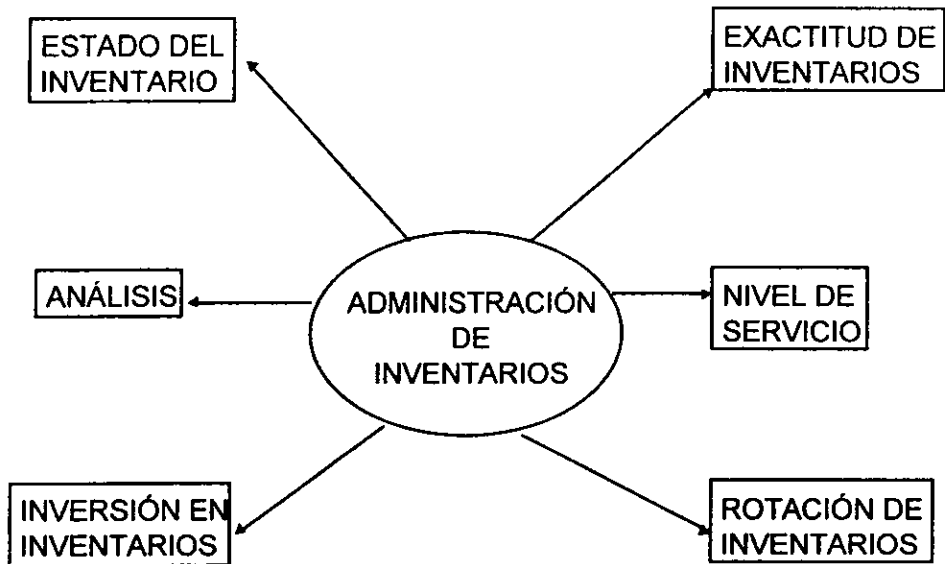


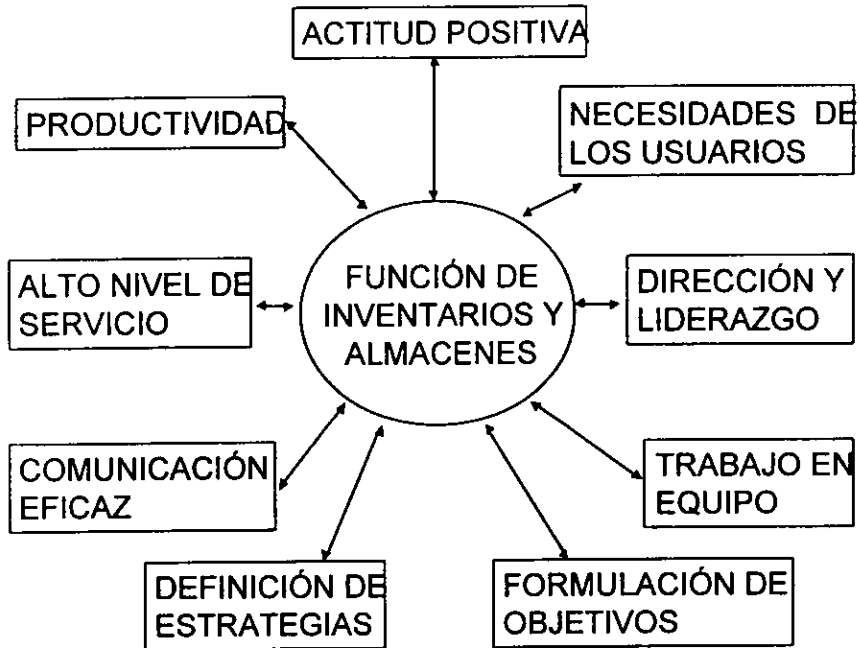
Figura 1.1.1.2. Administración de inventarios(entradas).



**Figura 1.1.1.3. Administración de inventarios (salidas).**

**Principales razones para llevar inventarios (figura 1.1.1.4.)**

- 1.- Proporcionar servicio a clientes con demandas variables (inmediatas o estacionales).
- 2.- Protegerse contra los errores de los proveedores, la escasez y los faltantes.
- 3.- Mejorar el nivel de las actividades de producción, estabilizando el empleo y optimizar las tareas de trabajo.



**Figura 1.1.1.4. Función de inventarios y almacenes.**

- 4.- Desfasar las etapas sucesivas en operaciones para que las fallas no detengan el sistema.
- 5.- Facilitar la producción de diferentes productos en las mismas instalaciones.
- 6.- Proporcionar un medio de obtener y manejar materiales en lotes de tamaño económico y obteniendo descuentos por cantidad.
- 7.- Proporcionar un medio para eliminar los riesgos de incertidumbre sobre los precios futuros y las entregas, tales como huelgas, incrementos de precio e inflación.



## **Demanda dependiente e independiente**

Un inventario de demanda dependiente esta compuesto por las materias primas, los componentes y los subensambles que son usados en la producción de artículos que sirven para la fabricación de otros artículos o para la fabricación de productos finales. Por ejemplo la demanda de teclados de computadora depende del artículo original, las computadoras. El inventario de producción es muy dependiente y predecible. Los requerimientos de todos los componentes encarados con otros componentes son fijados por el diseño, y las cantidades de producción son dictadas por la programación maestra de la empresa.

Los inventarios de demanda independiente constan de los productos terminados, las partes de servicio, y otros artículos cuya demanda aumenta mas directamente dependiendo del ambiente incierto del mercado. Por esto, la distribución de inventarios generalmente tiene una demanda altamente incierta e independiente. Las demandas dependientes normalmente pueden calcularse, mientras que las demandas independientes, usualmente requieren alguna clase de pronóstico.

El objetivo de esta tesis es desarrollar un sistema de control de inventarios para una planta telefónica; los métodos usados en la misma para controlar dichos recursos no han sido confiables, dado que se debe de tener material suficiente para abastecer a las entidades que lo requieran, de acuerdo a un cálculo de necesidades de las mismas y así mismo contar con material suficiente para cubrir emergencias; ya que dentro de la planta telefónica, día con día se tienen contratiempos y daños en la red, lo cual ocasiona pérdidas económicas muy grandes tanto para la telefónica como para los usuarios.

El buen diseño de un sistema computarizado de inventarios puede resolver este y varios problemas mas.

El control interno de los inventarios es muy importante porque las mercancías que lo componen son el alma de una empresa y sobre todo en el caso de una empresa de servicios telefónicos, por las razones anteriormente expuestas. Los elementos de un buen control de inventarios son los siguientes:

· Contar los inventarios físicamente por lo menos una vez cada año, cualquiera que sea el sistema que se use.

Mantener procedimientos eficientes de compras, recepción y embarques.

Almacenar los inventarios para protegerlos de robos, daños y deterioro.

Limitar el acceso a los inventarios a personal que no esté autorizado a manipular los registros contables.

Mantener registros de inventarios perpetuos para mercancías con alto costo unitario.

Comprar inventarios en cantidades económicas.

Mantener en existencia el inventario suficiente para evitar situaciones de falta de productos, que conducen a la pérdida de productividad en una empresa.

No mantener almacenado un inventario demasiado grande, evitando de esta forma el gasto de inmovilizar dinero en artículos que no se necesitan.

El conteo anual de los inventarios es necesario porque es la única forma de estar seguros de la cantidad de mercancías en existencia.

En este capítulo sentaremos las bases de los elementos de la administración de inventarios, los cuales serán muy útiles durante el desarrollo e implementación del sistema.

### **1.1.2. Categorías de inventarios**

Los inventarios varían dependiendo de las actividades. En una empresa industrial se encuentran inventarios de materias primas, de productos en curso, de productos terminados y de mantenimiento. En una empresa comercial existen inventarios de productos terminados y de muebles de oficina. En general los inventarios pueden dividirse en 4 categorías.

a) **Inventario de fabricación.** Es el formado por las materias primas brutas, las piezas y los productos semiterminados que entran en la composición de los productos terminados.

b) **Inventario de productos en curso.** Se trata de las componentes que se encuentran en las diferentes etapas de la fabricación. Dichos productos pueden almacenarse en los locales de fabricación si el procedimiento de producción implica etapas sucesivas, como ocurre por ejemplo en una etapa de ensamble.

c) **Inventario de productos terminados.** Estos productos que son el resultado final del sistema de producción, se guardan en almacenes apropiadamente acondicionados hasta el momento de su expedición.

d) **Inventario MRO (Mantenimiento, Reparación, Operaciones).** Estos productos no forman parte integral de un producto terminado, pero intervienen directamente en el proceso de fabricación. El aceite, el jabón, la grasa, las piezas de repuesto para las máquinas y los muebles de oficina son algunos ejemplos. Se le conoce también como inventario de abastecimiento.

### 1.1.3. Regla de administración de los inventarios

Una representación gráfica de la evolución de los inventarios permitirá comprender más fácilmente los elementos necesarios para elaborar una regla de administración. Por lo general esta regla debe definir los siguientes elementos: nivel del inventario, inventario activo, inventario de seguridad, punto de reorden, tasa de agotamiento, tasa de reabastecimiento o de entrega (fig. 1.1.3.1.).

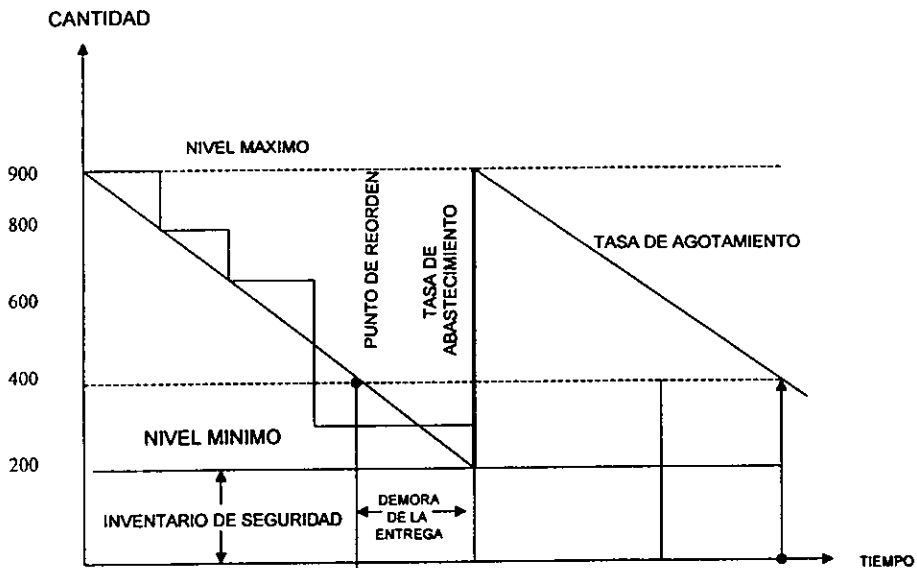


Figura 1.1.3.1. Evolución de los inventarios.

#### Niveles de inventario

Los niveles de inventario representan los límites predeterminados de las cantidades por almacenar. Estas cantidades varían entre un nivel máximo y un nivel mínimo. La determinación de estos 2 niveles depende del consumo anual de la tasa de agotamiento, del costo unitario del producto, de las demoras de entrega, etc.

### **Inventario activo**

Este es el inventario que varía constantemente al ritmo de las entradas y salidas del almacén, y puede corresponder a la cantidad económica o al consumo actual.

### **Nivel de servicio**

Este se refiere a la intensidad con la cual la empresa desea satisfacer la demanda. El nivel de servicio puede concebirse en dos formas:

a) La relación entre el número de unidades ofrecidas y el número demandado. De la cual se obtiene la ecuación 1.1 con la cual obtendremos el porcentaje con el cual la empresa desea satisfacer la demanda.

$$Uo/Nd \times 100 = \% D \quad (1.1)$$

Donde  $Uo$  = Unidades ofrecidas

$Nd$  = Número demandado de unidades

$\%D$  = Porcentaje de la demanda

Por ejemplo si 10 clientes demandan 1400 unidades entre si y solamente se ofrecen 900, obtener el porcentaje con el cual la empresa desea satisfacer la demanda.

De la ecuación 1.1

$$900/1400 \times 100 = 64.3 \% \text{ (porcentaje de la demanda)}$$

b) La relación entre el número de clientes que han comprado el producto y los que lo han demandado.

$$\text{NCC/CD X100} = \%CS \quad (1.2)$$

Donde NCC = Numero de clientes que han comprado el producto

CD = Clientes que han demandado el producto

%CS = % Clientes satisfechos

Supongamos, por ejemplo, que el número de clientes por atender es de 10 unidades cada uno y el décimo demanda 500. Si las nueve primeras demandas son satisfechas y la décima no se atiende, el nivel de servicio será igual a:

De la ecuación 1.2

$$9/10 \times 100 = 90\% \text{ (porcentaje de la clientela satisfecho)}$$

Con la primera definición se conoce el porcentaje de la demanda que ha sido satisfecho, mientras que la segunda definición indica el porcentaje satisfecho de la clientela, independientemente de las unidades demandadas.

### **Punto de reorden**

Este es el nivel de inventario, a partir del cual se decide ordenar el producto.

Este punto, que se establece para asegurar la disponibilidad de los productos en los periodos de reabastecimiento, designa una cantidad que está en función de la tasa de la demanda durante el período de reabastecimiento y de la demora de la entrega.

El siguiente ejemplo (fig. 1.1.3.2.) ilustra una manera de determinar en forma aproximada el punto de reorden. Supongamos que, para un artículo determinado, los consumos mensuales y las demoras de entrega son las siguientes:

<b>Mes</b>	<b>Consumo mensual (unidades)</b>	<b>Demora de la entrega (días)</b>
Enero	90	3
Febrero	110	5
Marzo	120	6
Abril	80	4
Mayo	100	3
Junio	100	3
<b>Total</b>	<b>600</b>	<b>24</b>

**Figura 1.1.3.2. Punto de reorden**

El consumo promedio mensual se calcula de acuerdo a la siguiente ecuación:

$$PM = TCM/NM \quad (1.3)$$

Donde PM = Promedio mensual

TCM = La suma del total de los consumos mensuales

NM = Número de meses de dicho consumo

Por lo tanto, de acuerdo a la ecuación 1.3

$$PM = TCM/NM = 600/6 = 100 \text{ unidades}$$

Para calcular el consumo diario de unidades por día Considerando meses de 30 días, se obtiene la siguiente ecuación.

$$CD = PM/DPM \quad (1.4)$$

Donde CD = Consumo diario  
PM = Promedio mensual  
DPM = Días por mes considerados

Por lo tanto, de acuerdo a la ecuación 1.4. el consumo diario sería de:

$$CD = PM/DPM = 100/30 = 3.3 \text{ Unidades/día}$$

La demora promedio de entrega se obtiene de la ecuación 1.5.

$$DPE = TDE/NM \quad (1.5)$$

Dónde DPE = Demora de la entrega  
TDE = Total de la suma de días de demora  
NM = Número de meses

Por lo tanto, de acuerdo a la ecuación 1.5. la demora promedio de entrega sería

$$DPE = TDE/NM = 24/6 = 4 \text{ días}$$

El punto de reorden se obtiene de acuerdo a la ecuación 1.6.

$$PRO = CD/DPE \quad (1.6)$$



Dónde

PRO = Punto de reorden

CD = Consumo diario

DPE = Demora de la entrega

Por lo tanto, de acuerdo a la ecuación 1.6

$$PRO = (CD)(DPE) = (3.3)(4) = 13.2 \text{ unidades}$$

Por tanto debe levantarse la orden en el momento en que el inventario llega a aproximadamente a 14 unidades. En este ejemplo no se ha mencionado la existencia de un inventario de seguridad que permita evitar la escasez durante los periodos o demoras de entregas que excedan de cuatro días.

Costo del inventario

Los principales costos asociados con acopiar y mantener inventarios son los siguientes:

- a) Costos de ordenar (CO) e iniciar la producción para colocar pedidos, expedición, inspección y cambio o establecimiento de instalaciones para producir internamente.
- b) Costos de mantener (CM) en inversión de capital, manejo, almacenamiento, seguros, impuestos, obsolescencia, deterioro y procesamiento de datos.
- c) Costos de compra (CC) incluyendo el precio pagado o la mano de obra, los materiales y los cargos indirectos necesarios para producir el artículo.

El costo total (CT) de un inventario es la suma del costo de ordenar, mas el costo de mantener y el costo de compra. Si D es igual a la demanda en unidades en una base anual, CO es igual al costo de preparar o hacer un pedido, CC es igual al costo de matener una unidad en inventario durante un período de tiempo dado, P es igual al

costo de compra, Q es igual al tamaño del lote, y  $Q/2$  equivale al inventario promedio, por lo tanto la expresión matemática queda expresada en la ecuación 1.7:

$$CT = CO + CM + CC \quad (1.7)$$

Donde :

$$CO = (Co\$/orden)(Orden/Unidades)(Unidades/año) \quad (1.8)$$

$$CM = (Cc\$/unidad-año)(Unidades)/2 \quad (1.9)$$

$$CC = (P\$/unidad)(Unidades/año) \quad (1.10)$$

### Conteo de inventarios

Los registros de inventarios deben ser altamente seguros (98-99 %) para facilitar los sistemas de producción automatizada. Dos métodos de auditar registros de inventarios son:

a) Por conteo físico periódico (una vez al año)

b) Y conteo cíclico

El conteo cíclico, es un conteo físico continuo del inventario, por lo que todos los artículos son contados con una frecuencia específica, y los registros de inventarios son periódicamente confrontados con los datos reales. Un ciclo es el tiempo requerido para contar todas las partidas en inventario al menos una vez.

Por ejemplo la compañía X tiene 6400 artículos en inventario, 400 son clase A, 1000 son artículos B y 5000 son clase C. La compañía opera 250 días por año y desea contar A, B y C con una frecuencia relativa de 5, 2, y 1 vez al año. ¿Cuántos artículos debe la compañía contar por día, en promedio?

Artículo	Artículo	Frecuencia de conteo	Cuentas totales
Tipo	Número		
A	400	5	2000
B	1000	2	2000
C	5000	1	<u>5000</u>
		Cuenta total	9000

**Figura 1.1.3.3. Ejemplo, compañía X.**

Donde

NDAC = Número de artículos contados día

CT = Cuentas totales

ND = Número de días

Por lo tanto

$$NDAC = CT/ND \quad (1.11)$$

$$NDAC = 9000/250 = 36 \text{ artículos contados/día}$$

### **Inventario de Seguridad**

Este tiene como finalidad impedir toda interrupción en el aprovisionamiento, causada por demoras en la entrega o por un aumento imprevisto de la demanda durante el periodo de reabastecimiento.

La importancia del inventario de seguridad esta ligada al nivel de servicio, la fluctuación de la demanda y la variación de las demoras de entrega. Si una empresa desea aumentar su nivel de servicio, deberá acrecentar su inventario de seguridad a fin de poder responder al alza imprevista de la demanda y adaptar el punto de reorden en consecuencia.

#### 1.1.4. Costo de abastecimiento y de almacenamiento

El abastecimiento tiene relación con la recepción y el almacenamiento de artículos terminados y la entrega de ellos a los clientes. Las principales decisiones se relacionan con:

- a) La localización y el tamaño de los almacenes.
- b) El pedido y el manejo de materiales.
- c) Llevar registros.

Una etapa importante en el estudio de una política de administración de inventarios es el análisis de los diferentes costos asociados con la compra, el almacenamiento y la utilización de los productos. Estos Costos pueden clasificarse en tres categorías:

Costo de aprovisionamiento

Costo de almacenamiento

Costo de escasez

Los elementos necesarios para el cálculo de estos costos pueden obtenerse en el departamento de contabilidad, el cual también debe clasificarlos en costos fijos y costos variables.

Costo de aprovisionamiento

Este se refiere a la adquisición o renovación del inventario. Este costo también denominado costo de adquisición, comprende el costo de la orden y el precio pagado por la mercancía. El costo de la orden incluye los gastos inherentes a la emisión de una solicitud de pedido, el transporte, la recepción y la inspección.

Ciertos elementos del costo de la orden son fijos e independientes del número de pedidos emitidos o de la cantidad de artículos por pedido. Por ejemplo los gastos de administración del personal y del material correspondientes al establecimiento de los pedidos son invariables, cualquiera que sea el número de pedidos. Otros gastos varían en función del trabajo suplementario y contribuyen solo en cierta medida al costo del pedido. Por tanto, en todo los gastos pueden distinguirse una parte fija y una parte variable. En el caso del costo del pedido, los gastos que intervienen son los siguientes:

a) Costo de mano de obra.

Oficina de compras (Director, compradores y otros).

Almacén (Personal dedicado a la recepción de la mercancía).

b) Gastos inmobiliarios (Superficie y mantenimiento de la oficina y del almacén).

c) Deudas pasivas (intereses sobre préstamos).

d) Costo del suministro.

e) Comunicaciones.

f) Transporte y distribución.

g) Recepción e inspección.

La distribución entre gastos fijos y variables depende de la organización interna de la empresa.

A continuación (fig. 1.1.4.1.) se da un ejemplo de cálculo del costo de la orden. Los porcentajes que se indican son solo promedios.

Elementos de gasto	Costo total (\$)	% Variable	Costo variable(\$)
Mano de obra	200 000	60%	120 000
Inmobiliario	13 500	75%	10 125
Deudas pasivas	6 500	90%	5 850
Suministros	6 500	50%	3 250
Comunicaciones	6 500	50%	3 250
Transporte	33 500	25%	8 375
Inspección	53 500	25%	13 375
<b>Total</b>	<b>320 000</b>	<b>100%</b>	<b>164, 225</b>

Fig. 1.1.4.1. Cálculo del costo de la orden.

Número total de pedidos emitidos por año: 16 000

El promedio del costo total por pedido se obtiene de acuerdo a la ecuación 1.12

$$PCTP = CT/PEA \quad (1.12)$$

Dónde

PCTP = Promedio del costo total por pedido

CT = Costo total

PEA = Pedidos emitidos por año

Por lo tanto, de acuerdo a la ecuación 1.12 el costo total por pedido sería:

$$PCTP = CT/PEA = 320,000/16,000 = 20.00$$

El costo variable por pedido se obtiene de acuerdo a la ecuación 1.13.

$$CVP = CV/PEA \quad (1.13)$$

Dónde

CVP = Costo variable por pedido

CV = Total de la suma del costo variable

PEA = Pedidos emitidos por año

Por lo tanto, de acuerdo a la ecuación 1.13, el costo variable por pedido sería:

$$CVP = CV/PEA = 164,225/16,000 = 10.26$$

#### Costo de almacenamiento

Por lo común, el costo anual de almacenamiento representa mas del 25% del valor promedio de los productos almacenados (este porcentaje se sitúa entre 14 y 36%). En principio, este costo se compone de los siguientes elementos:

- a) Valor promedio del inventario (para un año determinado).
- b) Intereses sobre la inversión. Es importante tener en cuenta los gastos correspondientes a los intereses y el rendimiento que sería posible obtener si el capital se invirtiera en alguna otra cosa. La tasa de interés puede evaluarse en función de la tasa bancaria en curso (entre 10 y 15% del valor promedio del inventario).
- c) Gastos de seguros. Numerosas compañías se aseguran, contra incendios, robos, o cualquier otra forma de daños. Este costo se sitúa entre 1 y 3% de su valor promedio del inventario.

d) Impuestos prediales. Estos representan de un 2 a un 4% del valor inmobiliario (terreno, almacén).

e) Mano de obra. Los salarios pagados a los empleados por el control y la manipulación de los inventarios constituyen un cargo fijo.

f) Costo de ocupación. El almacén se deprecia a una tasa de entre 1 y 5% por año.

g) Costo de obsolescencia. Ciertos productos terminados o el material que se utiliza en la fabricación se vuelven obsoletos con la introducción de nuevos productos. Este costo puede representar entre un 4 y un 10% del valor promedio del inventario.

Elementos de gasto	Costos (\$)	Porcentaje de valor promedio del inventario
Intereses	150 000	7.5
Seguros	600	0.03
Impuestos	75 000	3.75
Prediales	60 000	3.00
Mano de Obra	100 000	5.00
Ocupación	40 000	2.00
Obsolescencia	400	0.02
Deterioro	10 000	0.5
Reciclaje		
<b>Total</b>	<b>436 000</b>	<b>21.8</b>
Inventario Promedio	2 000 000	

Figura 1.1.4.2. Variaciones de los costos.



h) Costo de deterioro. El deterioro puede deberse al almacenamiento, la manipulación u otras causas. Debe representar un máximo de 1% del valor promedio del inventario.

De una forma general, estos costos varían con el incremento o decremento del inventario. Es por ello que dichos costos se expresan como un porcentaje del valor promedio del inventario. En la figura 1.1.4.2. se da un ejemplo de su distribución.

#### **Costo de escasez o falta de inventario**

Este costo corresponde al monto de las ventas perdidas como consecuencia de la falta de inventario, del costo de detención de la producción, de los gastos suplementarios o del costo de los trabajos administrativos suplementarios. El costo de la escasez se considera uno de los más difíciles de evaluar.

### **1.1.5. Política de administración de inventarios**

Esta política consiste en el conjunto de reglas y procedimientos que aseguran la continuidad de la producción de una empresa permitiendo una seguridad razonable en cuanto a la escasez de materia prima e impidiendo el exceso de inventarios con el objeto de mejorar la tasa de rendimiento.

Esta política puede variar de una empresa a otra, de un período a otro, y de un producto o grupo de productos a otro. Su éxito reside en el respeto de los siguientes imperativos:

Establecer relaciones exactas entre las necesidades probables y los abastecimientos de diferentes productos.

Definir categorías para el inventario y clasificar cada mercancía en la categoría adecuada.

Mantener los costos de abastecimiento y almacenamiento al mas bajo nivel posible.

Mantener un nivel adecuado de inventarios.

Satisfacer rápidamente la demanda

Recurrir a la informática

La política de administración de los inventarios debe también estar adaptada al sistema de producción. Las necesidades de materiales y de servicios no son las mismas para la producción en serie y para la producción intermitente o por unidad.

### 1.1.6. Planificación global

La planificación es una etapa esencial que precede a los trabajos y engloba un objetivo determinado. Estos planes futuros de acción se inscriben dentro de un contexto dinámico que lleva al administrador a ajustar sus planes al ritmo de los cambios.

La planificación es una etapa esencial que precede a los trabajos y engloba todas las previsiones inherentes a la elaboración de planes de acción eficaces.

La planificación se hace a largo plazo (construcción de una nueva fábrica, expansión de la gama de productos), a plazo mediano (elaboración de los planes de producción y de venta) y a corto plazo (calendario de producción), así como a diferentes niveles del sistema de producción. Cada tipo de planificación responde a cierta necesidad de información y de control del administrador. La planificación global define, para un periodo determinado, las orientaciones de la empresa en materia de producción de bienes y servicios.

La finalidad de la planificación global es evaluar el conjunto de recursos materiales, humanos y financieros necesarios para las operaciones de producción de un periodo dado. El principal objetivo es satisfacer, al mas bajo costo posible, las previsiones de demanda de este periodo.

La realización de este objetivo está sujeta a restricciones internas y externas.

La política de la empresa en materia de mano de obra, horas suplementarias, almacenamiento y nivel de servicio a la clientela constituyen las restricciones internas. Las restricciones externas provienen de las condiciones tecnológicas las cuales limitan la capacidad técnica de producción y de la situación sociológica, económica y de la competencia, las cuales condicionan las variaciones de la demanda.

El resultado de la planificación Global , es el programa general de producción y se elabora a partir de las previsiones de la demanda para el conjunto de productos ofrecidos por la empresa, y define las cantidades por producir, los niveles de inventario y la composición de la mano de obra para cada período.

En la siguiente figura 1.1.6.1. se da una lista de los elementos de información necesarios para elaborar un programa general de producción (PGP).

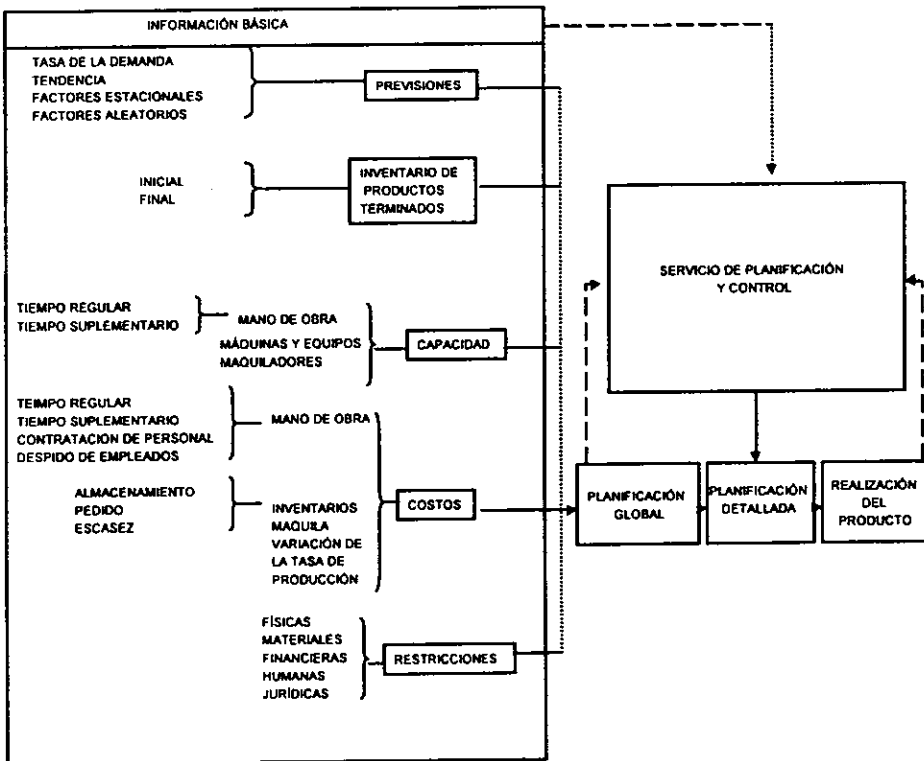


Figura 1.1.6.1. Elementos de información necesarios, para elaborar un programa general de producción.

### **1.1.7. Etapas de la planificación y del control de inventarios**

En las etapas de la planificación y del control de inventarios se utilizan diferentes formulas para encauzar la información al departamento de programación, a fin de repartir los trabajos, seguirlos y controlarlos. En la empresa pequeña y mediana, los trabajos del departamento de programación, son generalmente ejecutados por el responsable de grupo.

Los calendarios de compras se elaboran posteriormente en función del horario global y del nivel actual de los inventarios.

Al final de cada período de trabajo, el departamento de programación debe transmitir los informes referentes a la producción al departamento de planificación, a fin de que este tome las decisiones que sean necesarias. En la figura 1.1.7.1. se sitúan las principales etapas de la planificación de la producción en serie y del control de inventarios.

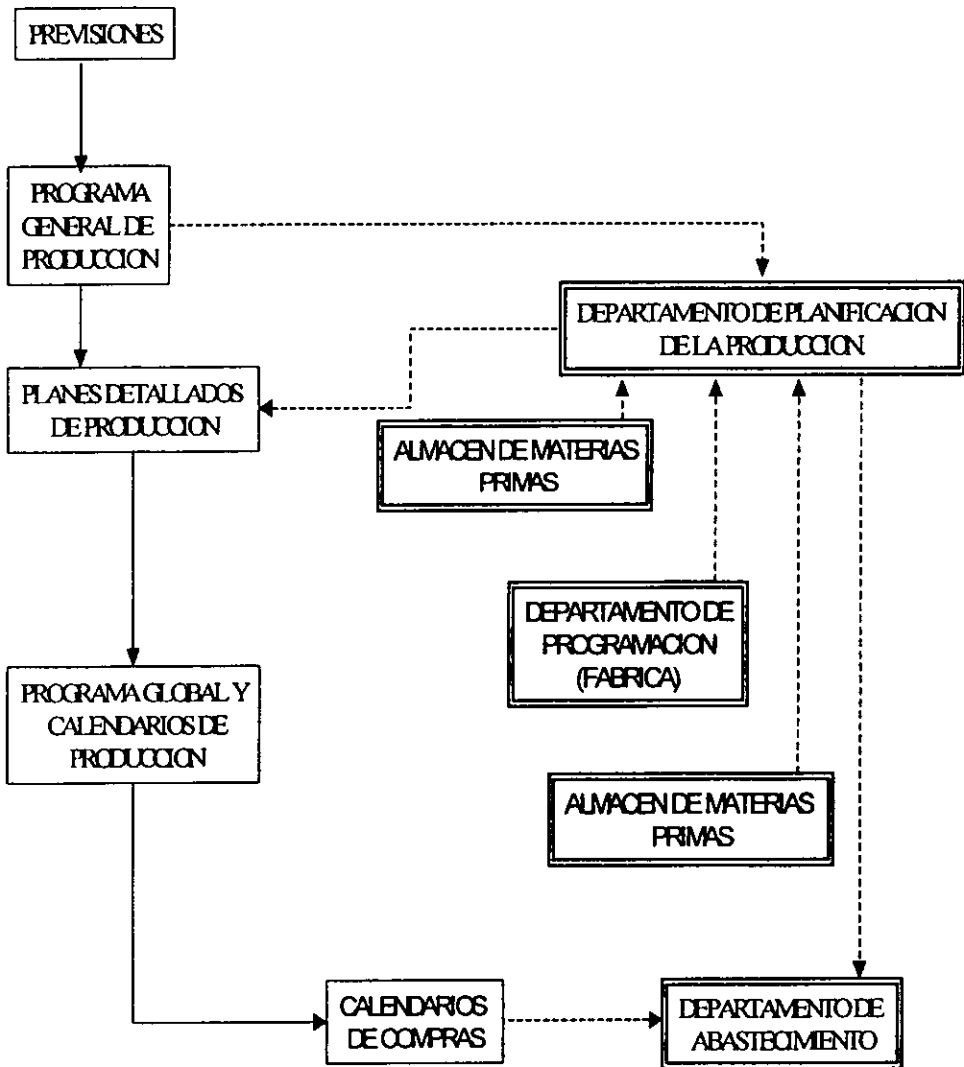


Figura 1.1.7.1. Etapas de la planificación de la producción en serie y del control de inventarios.

### **1.1.8. Planificación de los inventarios**

Los inventarios tienen gran importancia en el sistema de producción en serie, puesto que garantizan la continuidad de las operaciones, la utilización racional de los recursos disponibles y el mantenimiento de un nivel satisfactorio de servicio a la clientela, además de asegurar la independencia de las operaciones en cada etapa de la producción y de la distribución. En una línea de producción, los inventarios de productos en curso permiten a cada máquina o puesto de trabajo proseguir las operaciones hasta un cierto punto, aún si en la etapa precedente las máquinas se detienen.

De todo lo anterior podemos concluir que los inventarios producen costos. Para reducir estos costos es necesario determinar y controlar los niveles adecuados de inventarios para los diferentes estados de la producción y de la distribución.

### **1.1.9. Ventajas y desventajas de los inventarios**

Los sistemas de computación han revolucionado la contabilidad de los inventarios. Pueden proporcionar información actualizada de los inventarios, útil para la administración del negocio. Ayudan a rebajar los costos de la contabilidad al procesar grandes números de operaciones sin errores de cálculo. Los sistemas de computación, también refuerzan el control interno.

Es posible ejercer un alto grado de control administrativo, sobre el área de inventarios proporcionando elementos para que la disciplina necesaria en un buen sistema de control de inventarios, sea totalmente entendida y de manera cabal vigorizada. Muy a menudo se acumulan beneficios considerables, debido al empleo de mejores disciplinas y de procedimientos necesarios de soporte, para la implementación de un nuevo sistema. Un sistema de control de inventarios computarizado, puede ayudar a recobrar el capital invertido en la instalación de una computadora.

También, como la confianza del usuario en el sistema crece, los sistemas informales paralelos, mantenidos y usados por los empleados de la compañía en diferentes áreas funcionales, vienen a ser innecesarios.

Es posible tener registros exactos y hasta la fecha. Un programa comprensible, de validación de datos de entrada, puede verificar los datos antes de que los archivos sean puestos al día. Esto es particularmente cierto en el caso de los sistemas de procesamiento de transacciones en línea.

La disponibilidad de la potencia de una computadora, hace posible manejar grandes volúmenes de transacciones de inventarios y cualesquier reporte puede ser producido inmediatamente para ayudar a tomar decisiones.

Es posible valorar todos los inventarios mantenidos por la compañía sobre bases regulares.

El esfuerzo de oficina requerido para llevar a cabo un inventario manual es reducido, obteniéndose con esto menos horas hombre de trabajo y mayor confiabilidad, lo cual también se traduce en mayores utilidades para la empresa.

Proporcionar una herramienta amigable para el usuario, que le permita realizar operaciones en forma transparente

Implementación de distintos niveles de seguridad.

El problema de obsolescencia de los inventarios causado por niveles excesivos se puede superar o al menos reducirse.

Aun con niveles bajos de inventario y sus ahorros resultantes las demoras en la producción son reducidas por la revisión de los niveles de los lotes almacenados a intervalos frecuentes.



## 1.2. Teoría de programación estructurada

A finales de los años sesenta, Dijkstra y otros propusieron el uso de un conjunto de construcciones lógicas con las que podría formarse cualquier programa. Las construcciones reforzaban el "mantenimiento del dominio funcional". Esto es, cada construcción tenía una estructura lógica predecible; se entraba a ella por el principio y se salía por el final, facilitaba el seguimiento del flujo procedural.

Las construcciones son la secuencia, la condición y la repetición. La secuencia implementa los pasos de procesamiento esenciales de la especificación de cualquier algoritmo, la condición da la posibilidad de seleccionar un procedimiento basado en alguna ocurrencia lógica y la repetición proporciona iteración. Estas tres construcciones son fundamentales en la programación estructurada.

Las construcciones estructuradas se propusieron para limitar el diseño procedural del software a un pequeño número de operaciones predecibles. Las métricas de complejidad, indican que el uso de construcciones estructuradas reduce la complejidad de los programas y por tanto, facilitan la legibilidad, la prueba y el mantenimiento. Las construcciones estructuradas son trozos lógicos que permiten reconocer los elementos procedurales de un módulo, en vez de leer el diseño o el código línea a línea. Es difícil dar una definición de programación estructurada por no existir generalmente una definición que sea aceptada a todos los niveles. Podríamos enunciar la programación estructurada como una técnica de construcción de programas que utilizan al máximo los recursos del lenguaje, limita el conjunto de estructuras aplicables a leer y presenta una serie de reglas que coordinan adecuadamente el desarrollo de las diferentes fases de la programación.

Aunque, como decíamos antes, la definición anterior es una de las que más hemos visto, son diferentes las definiciones que se podrían hacer; sin embargo, con lo que sí hay un acuerdo casi total es que la programación estructurada utiliza en su diseño los siguientes conceptos o principios fundamentales recogidos esencialmente en la

definición anterior:

- Estructuras básicas.
- Recursos abstractos.
- Diseño descendente **top-down** (arriba-abajo).

El diagrama de flujo es la representación gráfica más ampliamente usada para el diseño procedural. Desgraciadamente, es también el método del que más se ha abusado.

Un diagrama de flujo es un gráfico muy sencillo. Para representar un paso de procedimiento se utiliza un cuadro, un rombo para representar una condición lógica y flechas de control.

### **Estructuras básicas de control**

El Teorema de la estructura (o antiguo teorema de Bohm y Jacopini) demostró que cualquier programa con un solo punto de entrada y un solo punto de salida puede resolverse con tres únicos tipos de estructuras de control: Secuencial, alternativa y repetitiva.

### **Estructura secuencial**

Una estructura es aquella que ejecuta las acciones sucesivamente unas a continuación de otras sin posibilidad de omitir ninguna y naturalmente sin bifurcaciones. Todas estas estructuras tendrán una entrada y una salida, figura 1.2.1.

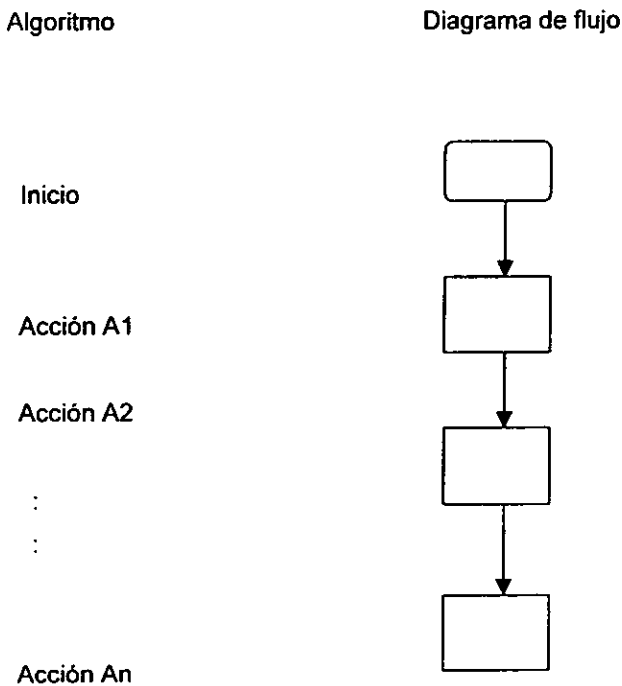
Si las acciones son A1, A2 .....An, la representación secuencial sería:

### Estructura alternativa

Es aquella estructura en la que únicamente se realiza una alternativa – una determinada secuencia de instrucciones – dependiendo del valor de una determinada condición o predicado.

Las estructuras alternativas, también llamadas condicionales, pueden ser de tres tipos:

- Simple
- Doble
- Múltiple



**Figura 1.2.1. Estructura secuencial.**

### Estructura alternativa simple

Es aquella en que la existencia o cumplimiento de la condición implica la ruptura de la secuencia y la ejecución de una determinada acción, figura 1.2.2.

### Estructura alternativa doble

Es aquella que permite la elección entre dos acciones o tratamientos en función de que cumpla o no determinada condición, figura 1.2.3.

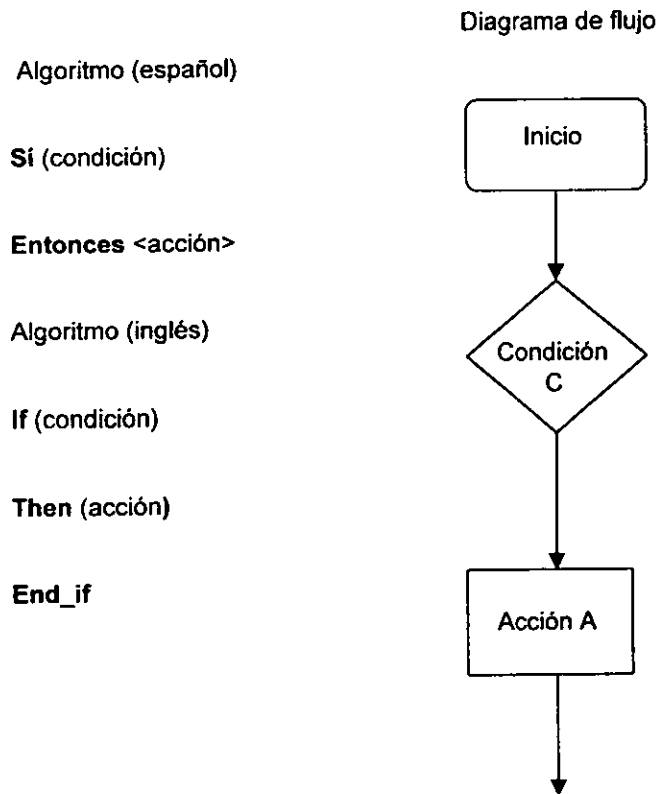


Figura 1.2.2. Estructura alternativa simple.

### Estructura alternativa múltiple

Las estructuras alternativas múltiples se adoptan cuando la condición puede tomar  $n$  valores enteros distintos:  $1, 2, 3, \dots, n$ . Según se elija uno de estos valores en la condición, se realizará una de las  $n$  acciones (cada vez sólo se ejecutará una acción), figura 1.2.4.

Algoritmo

Diagrama de flujo

**Si** (condición)

**Entonces** (acción A1)

**Si\_no** (acción A2)

**Fin\_si**

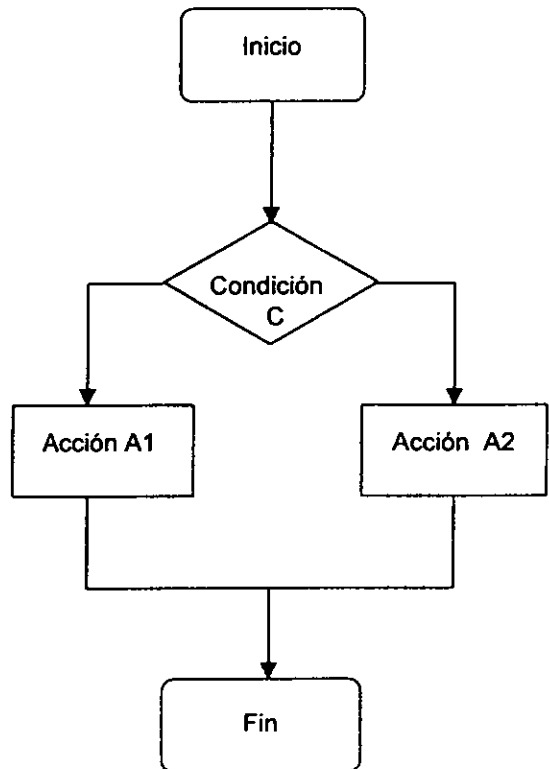
Algoritmo (inglés)

**If** (condición)

**Then** (acción A1)

**Else** (acción A2)

**End\_if**



**Figura 1.2.3. Estructura alternativa doble.**

## Estructura alternativa múltiple

Las estructuras alternativas múltiples se adoptan cuando la condición puede tomar  $n$  valores enteros distintos: 1,2,3,...,  $n$ . Según se elija uno de estos valores en la condición, se realizará una de las  $n$  acciones (cada vez sólo se ejecutan una acción), figura 1.2.4.

### Algoritmo

**Según(variable)**

Condición 1: (acción A1)

Condición 2: (acción A2)

Condición 3: (acción A3)

Condición  $n$ : (acción  $A_n$ )

**Otros**

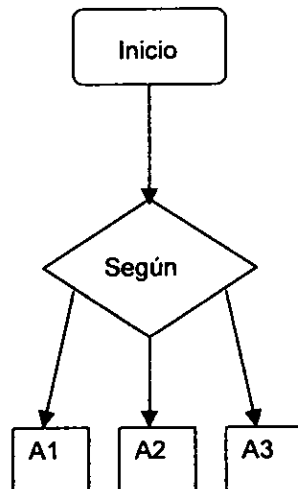
**Fin-según**

Algoritmo (inglés)

ASE

End\_case

### Diagrama de flujo



**Figura 1.2.4. Estructura alternativa múltiple.**

## **Estructuras repetitivas**

Las estructuras repetitivas o iterativas son aquellas en las que las acciones se ejecutan un número determinado de veces y depende de un valor predefinido o el cumplimiento de una determinada condición.

Las estructuras repetitivas permiten representar aquellas acciones que pueden descomponerse en otras subacciones primitivas. Una iteración de repetir la ejecución de una secuencia de acciones o de una acción. Un bucle o lazo es el conjunto de acciones iterativas.

Para describir una iteración, si se conoce el número de repeticiones, se puede escribir simplemente  $n$  veces la acción o secuencia de acciones a repetir. Sin embargo, si  $n$  es grande, las operaciones anteriores pueden resultar tediosas y la secuencia algorítmica difícil de leer. Con frecuencia, es difícil determinar el número de repeticiones.

En consecuencia, es preciso disponer de estructuras algorítmicas que permitan describir una iteración que permitan describir una iteración de forma cómoda. Las tres estructuras más usuales, dependiendo de que la condición se encuentre al principio o al final de la iteración, son:

-Estructura **mientras**

-Estructura **repetir\_hasta**

-Estructura **para (desde\_hasta)**

Algoritmo  
Mientras (condición) Hacer

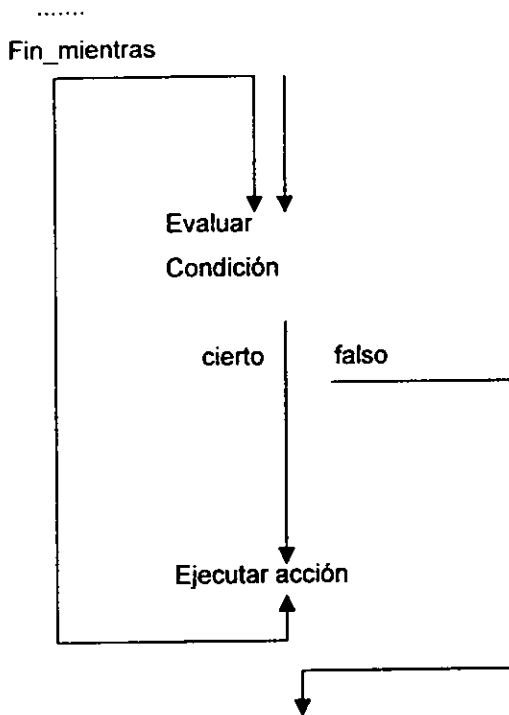


Diagrama de flujo

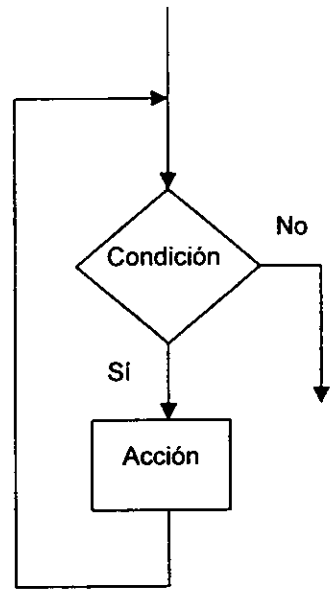


Figura 1.2.5. Estructura mientras.

Estructura **DOWHILE** (mientras)

El bucle **mientras** determina la repetición de un grupo de instrucciones mientras la condición se cumpla inicialmente, figura 1.2.5.

Estructura **DOUNTIL**(repetir\_hasta)

Existe la estructura en la que el número de iteraciones o repeticiones del grupo de instrucciones se ejecuta hasta que la condición deje de cumplirse. Esta condición se cumple al final, figura 1.2.6.



Algoritmo

Repetir (acciones)

Hasta que (condición)

Ejecutar (acciones)

Evaluar (condición)

cierto falso

Diagrama de flujo

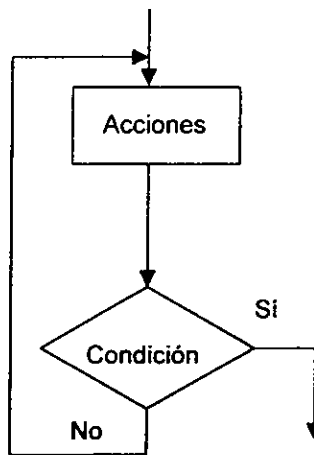


Figura 1.2.6. Estructura DOUNTIL.

Estructura para desde

La estructura **para es** aquella que se repite un número fijo de veces, n:

Algoritmo

Para (variable) de valor i a valor f incremento inc

Hacer (acciones)

Fin\_para

El incremento puede ser positivo como el caso anterior o negativo.

Para (variable) de valor i a valor i a valor f incremento dec.

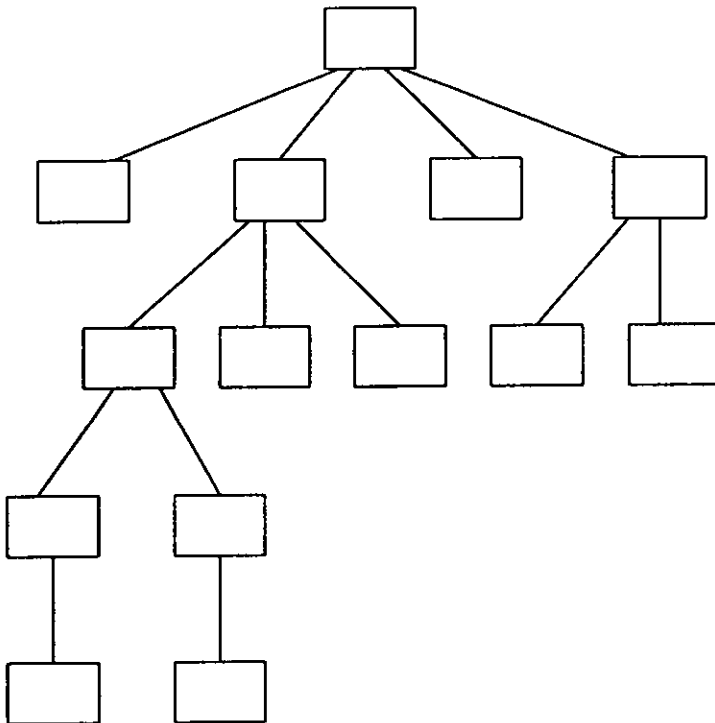
Hacer (acciones)

Fin\_para

**Metodología descendente "arriba-abajo"**

La metodología **top-down** (descendente), también conocida como arriba-abajo, consiste en dar solución al problema. En esencia, consiste en efectuar una etapa jerárquica y su inmediatamente inferior se relacionen mediante entradas y salidas de información.

Un programa estructurado tiene una representación en forma de árbol, figura 1.2.7.



**Figura 1.2.7. Metodología arriba-abajo.**

La marcha analítica de un proceso descendente estaría basado en dos características esenciales: representación en forma de árbol y descomposición funcional. El diseño se basa en la realización de diferentes niveles. El primer nivel resuelve totalmente el problema y el segundo y sucesivos niveles son refinamientos sucesivos del primero y se sigue siempre la metodología de recursos abstractos. Si el diseño y planteamiento son correctos, nunca será preciso volver atrás, ya que los niveles anteriores al que se esté situado en un momento dado ya habrán resuelto el problema en su totalidad.

### Diseño de programas estructurados

La realización del diseño estructurado de un problema se basa en la aplicación de los siguientes conceptos:

- Ir de lo general a lo particular, descendiendo en la estructura del programa y en su nivel de detalle.
- De la definición inicial del problema se pasa a un esquema de algoritmo descrito en pseudocódigo.
- Independencia inicial del lenguaje.
- Diseño por niveles, dejando los detalles para niveles posteriores. Verificar en cada nivel el esquema correcto.
- Finalizar con un trabajo de recomposición del algoritmo completo.

### Métodos de programación estructurada.

Existen diferentes métodos de programación estructurada. Entre las más conocidas se encuentran: Jackson, Bertini y Warnier.

### Método Jackson

Esta metodología creada por el inglés Michael Jackson se basa en que la estructura de un programa esta en función de la estructura de los datos que manipula. Jackson

empleo módulos según un orden jerárquico dentro de los diferentes niveles donde se encuentra. Cada módulo es un dato o conjunto de datos.

Las estructuras básicas en este método, son los siguientes:

**Secuencial:** Un número determinado de módulos se ejecutan una sola vez en el orden jerárquico preestablecido.

**Repetitiva:** Un módulo se ejecuta desde cero hasta  $n$  veces.

**Alternativa :** Se selecciona para la ejecución de un módulo entre varios posibles.

### **Método Bertini**

La metodología Bertini consiste en la descomposición de un problema en niveles, teniendo cada uno de ellos un inicio, un conjunto de procesos y un fin.

Esta metodología representa la estructura de los programas y no las operaciones del tratamiento. Según Bertini, las instrucciones se ejecutan de derecha a izquierda, pero el programador puede leerla al revés si le resulta más cómodo.

### **Método Warnier**

Se basa en una metodología matemática que establece un único lenguaje de comunicación entre usuarios, analistas y programadores, lo que permite la comprensión de forma sencilla por cualquier programador. La representación se puede hacer mediante llaves.

El método se basa en la descomposición por niveles del problema. En cada nivel se detallan los tratamientos que permiten la resolución del problema planteado.

### **1.3. Metodología de diseño**

Las metodologías de Yourdon, Ssadm y Merise que vamos a exponer, posee puntos comunes y técnicas similares en determinadas fases de su desarrollo.

Para establecer una comparación entre ellas hay que tomar como base esos puntos comunes que toda metodología debe contemplar y después como son abordados estos puntos desde cada una de las metodologías. Se tomará como referencias lo siguiente:

- Etapas del ciclo de vida.
- Modelización de datos.
- Modelización de procesos.

#### **Etapas del ciclo de vida**

En general todas ellas estructuran el desarrollo en diversas etapas. Si examinamos esta estructuración se puede observar una mayor precisión en cuanto a los pasos a seguir en la metodología de Yourdon. Resulta ser casi un recetario, con el cual podemos desarrollar el proyecto. Hay que resaltar que ninguna de las tres llega a determinar qué técnicas de programación estructurada se deben seguir, aunque todas recomiendan su uso.

#### **Modelización de datos**

Es común en todas las metodologías la definición de los niveles conceptual, lógico y físico en la definición de los datos.

A nivel conceptual, se admiten por parte de todas el modelo entidad/relación como el más adecuado para reflejar la información del sistema. También es una tendencia general del mercado la utilización de bases de datos relacionales como soporte del nivel lógico del modelo de datos.

## **Modelización de procesos**

Merise introduce una descripción más precisa de los mismos a través de las redes de Petri. En ellas la información acerca de los procesos, de qué hay que hacer, cómo, cuándo y quienes son muy completas. Esta técnica puede tener el inconveniente de la dificultad no sólo para el que la realiza sino también para la validación por parte del usuario por su complejidad y nivel de detalle.

Las otras metodologías realizan esta modernización a través de los DFD's de diferente nivel de detalle. Los DFD's resultan claros para el usuario y relativamente fáciles de realizar.

### **1.3.1. Yourdon**

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

- Diagramas de flujo de datos para la representación de procesos.
- Diagrama de transición de estados para la representación estructurada de las funciones a realizar en los procesos.
- Modelo Entidad/Relación para la representación conceptual de datos.
- Diccionario de datos como base o soporte de información del sistema.
- Diagramas o mapas de estructura para la representación modular de los procesos y las variables intercambiadas entre ellos.
- Especificaciones de programas basadas en lenguaje estructurado y tablas de decisión.

Las técnicas utilizadas por Yourdon al establecer el método para el desarrollo de cualquier sistema son:

### **Diagramas de flujo**

Se utilizan para la representación de gráficas de procesos y datos. Vamos a ver los formulismos de representación de los conceptos que adoptó Yourdon.

En los DFD's se usan los siguientes elementos.

### **Procesos**

Se representa 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.

### **Flujos de datos**

Representación del 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.

### **Entidades**









Son las personas o servicios que perciben o emiten algún flujo de información. Se representan como rectángulos en cuyo interior figura el nombre de la entidad.

### **Almacenamiento**

Su representación gráfica se corresponde con dos líneas paralelas en cuyo interior se pone el nombre del archivo o fichero. 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 que serán salida de los mismos.

Los símbolos empleados para la representación de los DFD existen dos variantes principales: la de Yourdon/DeMarco y la de Gane/Sarson, figura 1.3.1.1.

Concepto	Yourdon/DeMarco	Gane/Sarson
Proceso		
Entidad		
Flujo de datos		
Almacenamiento		

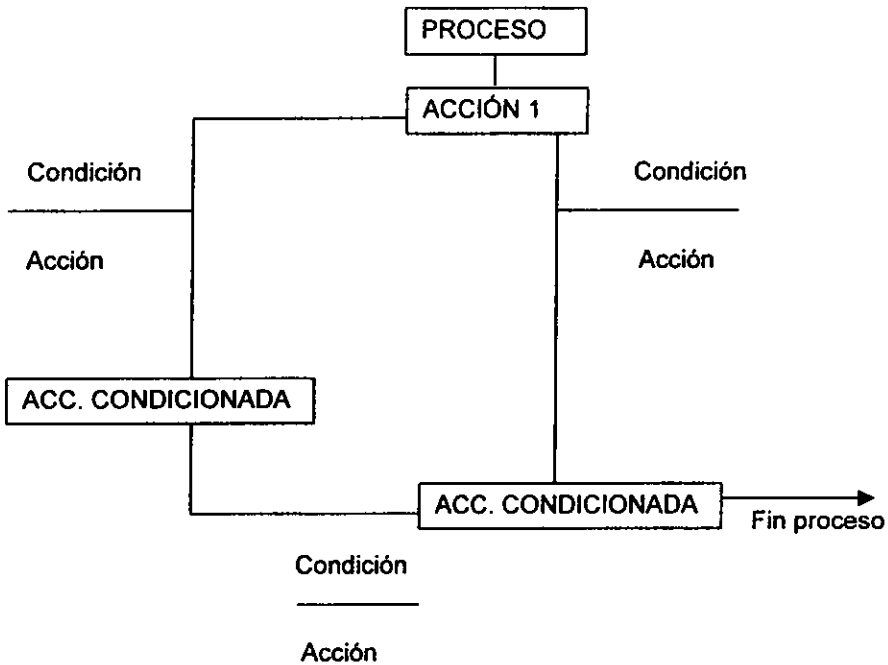
**Figura 1.3.1.1. Símbolos de los DFD's según Yourdon/DeMarco y Gane/Sarson.**

### 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 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 1.3.1.2.



**Fig. 1.3.1.2. Diagramas de transición de estados.**

### **Diccionario de datos (DD)**

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 o clave de un almacenamiento.

Mediante esta notación podría representar la sintaxis de los datos del sistema. Veamos un ejemplo:

Nombre	Tipo	Tamaño	Descripción
Cat	Texto	7	Número o clave de material
AllowZeroLength :	FALSO		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSO		
Tabla de origen:	sun_usu2		

## Diagrama de estructura

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

Podemos confeccionar, partiendo de este tipo de unidad, estructuras complejas que representen la secuencia de ejecución de funciones. Estas estructuras son similares al de técnicas estructuradas de análisis y diseño de programación, aunque en este tipo de diagramas se añaden los parámetros o variables que se intercambian los diferentes módulos.

Cada unidad de un diagrama de estructura se representaría de la siguiente manera, según la notación Yourdon, figura 1.3.1.3.

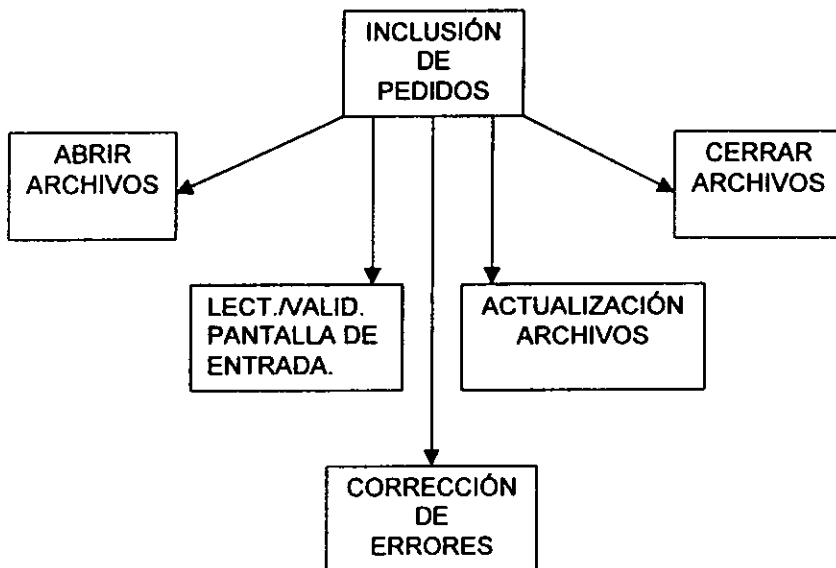


Figura 1.3.1.3. Unidad de un diagrama de estructura.

## Modelo Entidad/Relación

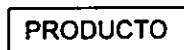
Para realizar el modelo conceptual de datos, Yourdon propone un esquema de entidades relaciones del sistema, pasando después a normalizar esta estructura. Veamos el ejemplo de la relación entre PROFESORES y ALUMNOS: un profesor pueden tener varios alumnos y un alumno varios profesores.

Los diagramas E/R son una técnica para representar gráficamente la estructura lógica de una base de datos. Como tal, ofrecen una forma sencilla de comunicar los rasgos prominentes del diseño de cualquier base de datos.

Básicamente en un diagrama del modelo E/R se deben representar los conceptos siguientes:

### Entidad

La componen una serie de datos que agrupados, tienen un cierto significado para la empresa. Se presenta por medio de un rectángulo en cuyo interior figura el nombre de la entidad, figura 1.3.1.4.



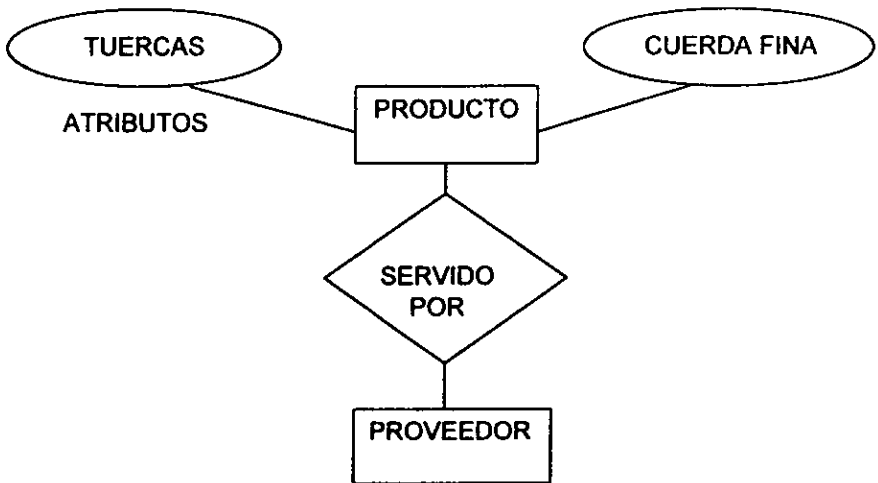
**Figura 1.3.1.4. Representación de una entidad.**

Cada entidad está compuesta por una serie de atributos que se representarán por una serie de círculos enlazados con la entidad.

### Relaciones

Constituyen el enlace entre las diferentes entidades del sistema. Se representan mediante líneas que unen las entidades o a través de un rombo en cuyo interior se

pone el nombre de la relación. Así, entre un PRODUCTO y un PROVEEDOR existirá la relación " ES SERVIDO POR". Figura 1.3.1.5.



**Figura 1.3.1.5. Relación entre entidades.**

### **Cardinalidad de una relación**

Es el número de ocurrencias de cada tipo de entidad que intervienen o pueden interervenir en la relación. Para un conjunto binario de relaciones R entre los conjuntos de entidades A y B, la cardinalidad de asignación debe ser una de las siguientes:

**Una a una.** Una entidad en A está asociada a lo sumo con una entidad en B, una entidad en B está asociada a lo sumo con una entidad en A, figura 1.3.1.6.

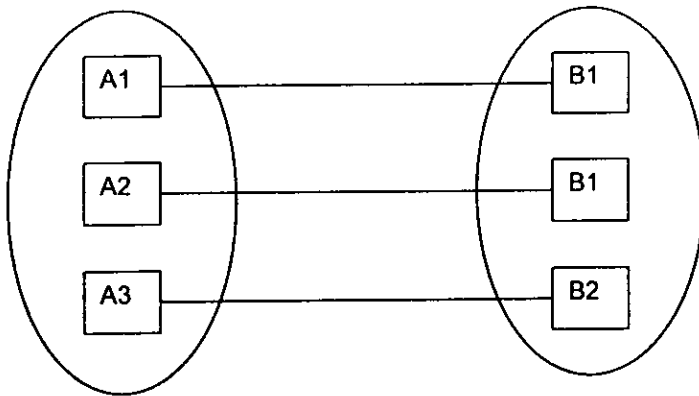


Figura 1.3.1.6. Relación una a una.

**Una a muchas.** Una entidad en A está asociada con un número cualquiera de entidades en B. Una entidad en B, sin embargo, puede estar asociado a lo sumo con una entidad en A, figura 1.3.1.7.

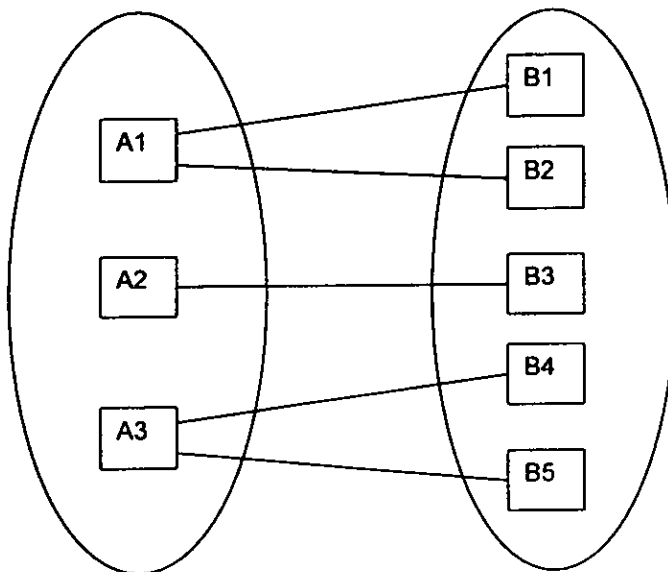
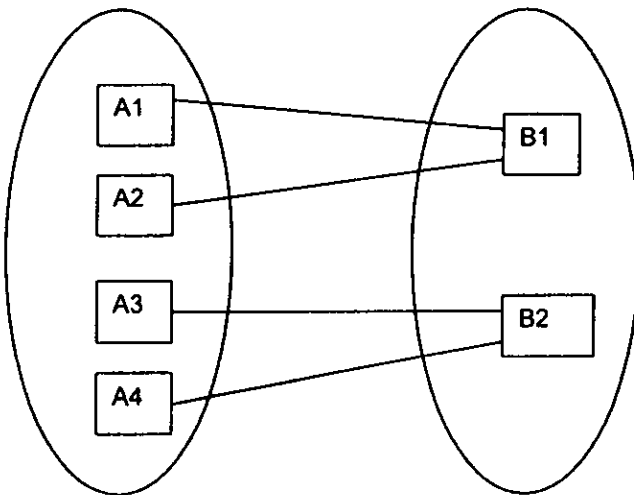


Figura 1.3.1.7. Relación una a muchas.

**Muchas a una.** Una entidad en A está asociada a lo sumo con una entidad en B. Una entidad en B, sin embargo, puede estar asociada con un número cualquiera de entidades en A, figura 1.3.1.8.

**Muchas a muchas.** Una entidad en A está asociada con un número cualquiera de entidades en B, y una entidad en B está asociado con un número cualquiera de entidades en A, figura 1.3.1.9.



**Figura 1.3.1.8. Relación muchas a una.**

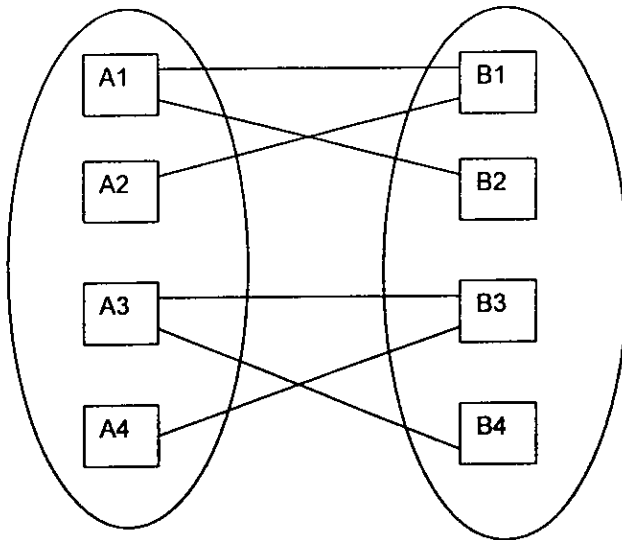


Figura 1.3.1.9. Relación muchas a muchas.

### Metodología

Yourdon define las siguientes etapas y niveles en el ciclo de vida de los sistemas informáticos, figura 1.3.1.10.

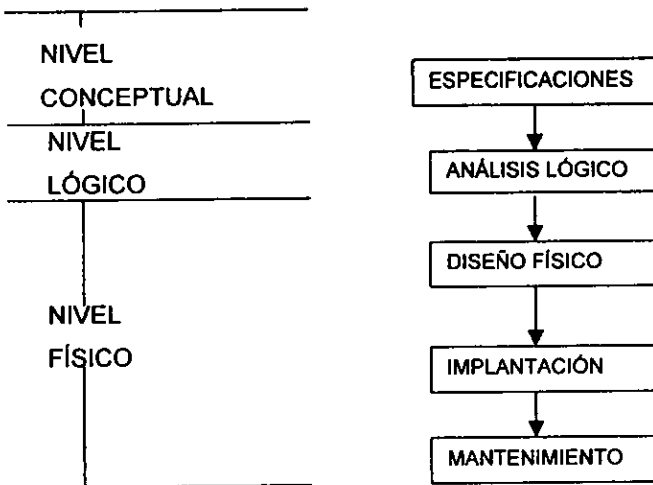


Figura 1.3.1.10. Ciclo de vida según Yourdon.



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

### **Estudio de viabilidad**

Debe hacer un estudio de la situación actual, representando la misma a través de DFD de primer nivel o diagramas de contexto en los que, de forma simple, se indiquen los procesos 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. El 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 prototipos 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 elegida (Codasyl, relacional, etc.) y una estructuración de los procesos utilizando diagramas de estructura de los mismos y generando las especificaciones de programa correspondientes.

### **Implementación ó producción**

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

### **Pruebas y test del sistema**

A 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 y de funcionamiento conjunto de programas y cadenas.

### **Control de calidad**

El objeto 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**

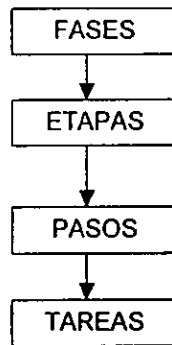
Comprende la puesta en marcha del sistema y en esta fase son de aplicación las consideraciones hechas de aspectos tales como formación y entrenamiento del usuario, entrega de manuales, procesos paralelos, etc.

#### **1.3.2. Ssadm**

La metodología consiste en una estructuración de los pasos a seguir en el desarrollo de un proyecto informático en las fases iniciales del ciclo de vida del mismo y en la descripción de una de las técnicas y formalismos sobre las que se basan los trabajos a realizar en cada fase.

En la metodología de SSADM usamos esquemas en los que se pueden ver la secuencia de los pasos y las técnicas utilizadas en cada uno de ellos, dentro de cada fase.

Así, según un carácter puramente jerárquico, podríamos distinguir:



**Figura 1.3.2.1. Estructura jerárquica de fases en SSADM.**

Aparecen abreviaturas de las diferentes técnicas:

DFD Diagrama de flujo de datos.

LDS Estructura lógica de datos.

ELH Historia de vida de la entidad.

LDD Diseño de diálogos lógicos de pantallas.

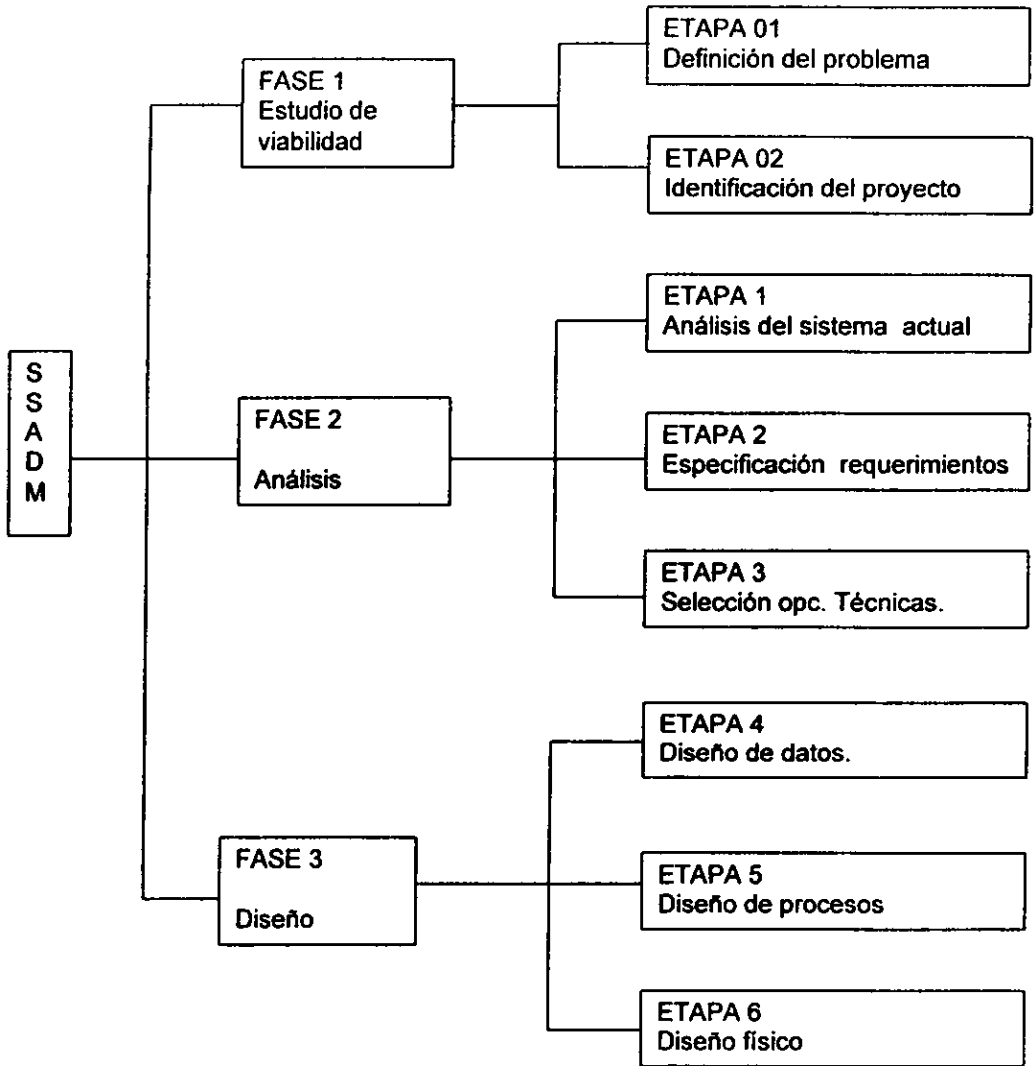
RDA Análisis relacionales de datos.

CLDD Diseño lógico completo de datos.

PS Especificaciones de programa.

PO Especificaciones de salidas del sistema.

El método contempla:



**Figura 1.3.2.2. Fases y etapas de SSADM.**

Las técnicas utilizadas por SSADM en cada una de las fases y etapas son:

### **Diagramas de flujo de datos (DFD)**

Son una forma de representación de los flujos de información en el interior del sistema y el exterior, es decir sus relaciones con otros.

### **Estructura lógica de datos (LDS)**

Mediante la representación de las entidades del sistema y las relaciones entre ellas. Para ello se utilizarán los formalismos y teorías del modelo entidad/relación.

### **Historia en la vida de la entidad (ELH)**

Representa la descripción de cómo las entidades descritas son afectadas por diferentes sucesos que ocurren en el sistema.

### **Tercera forma normal (TNF)**

La descripción de datos. Es un método matemático para la definición de datos que ayuda a evitar inconsistencias y ambigüedades en la estructura de los mismos.

## **FASE 1. Estudio de viabilidad**

Su objetivo es analizar el sistema actual, establecer una lista de requerimientos del nuevo sistema y una vez que se tiene claro lo que se quiere conseguir, evaluar la complejidad del proyecto y las diferentes opciones técnicas para abordarlo.

Esta fase se considera previa al inicio del desarrollo, es decir, a la fase de análisis, se divide en dos etapas:

1. -Definición del problema.
2. -Identificación del proyecto.

*Etapas:*  
*Etapas 1. Definición del problema.*

Iniciar el estudio de viabilidad

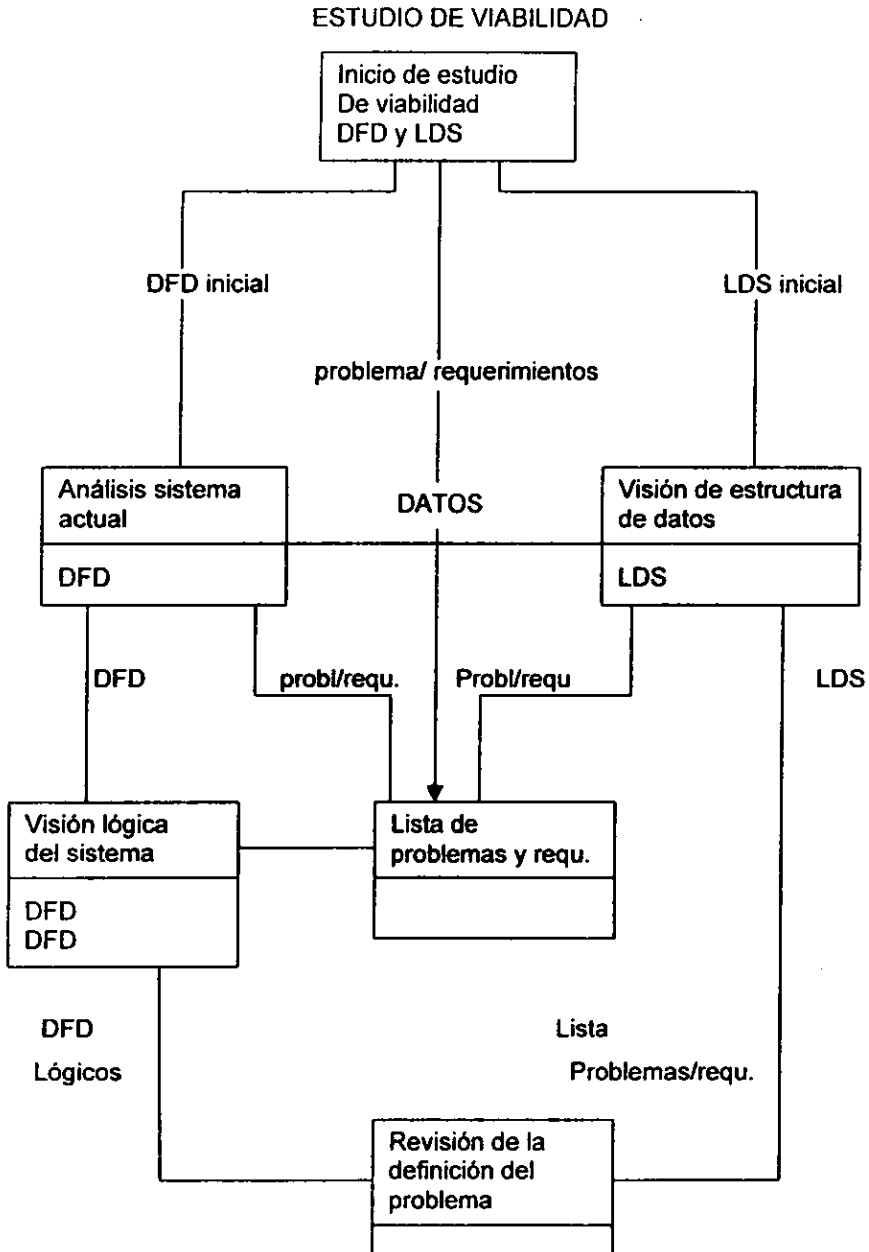


Figura 1.3.2.3. Definición del problema.

El objetivo de este paso es establecer la complejidad del proyecto, la duración aproximada de cada uno de los pasos y su entorno, si es que existe. En algún plan estratégico de la empresa.

### **Análisis del sistema actual**

Su objetivo es reflejar el funcionamiento del sistema actual a través de diagramas de flujo de datos. Se parte del primer nivel de DFDs (estudio de viabilidad) y se realizan los DFDs de segundo nivel, detallando o descomponiendo ya algunos procesos.

### **Creación de la estructura lógica de datos**

Su objetivo es reflejar la estructura de datos actual en el sistema después de un estudio inicial.

### **Desarrollo del sistema lógico**

Su objetivo es convertir la visión física del sistema actual reflejada en los DFDs de los pasos anteriores en una visión lógica del mismo.

### **Consolidación de la lista de problemas y requerimientos**

Su objetivo es revisar la lista después de las actualizaciones hechas en los pasos anteriores eliminando duplicidades e imprecisiones.

### **Revisión de la definición del problema**

En este punto el grupo de revisión debe analizar la documentación aportada y autorizar el paso a la siguiente fase. Esta documentación debe cumplir las normas especificadas en la metodología en el proyecto.

*Etapas 2. Identificación del proyecto.*



### **Identificación de opciones del proyecto**

Su objetivo es definir las diferentes opciones para la resolución del problema. Se trabaja con un número de opciones que debe estar entre 3 y 6.

### **Crear un esquema de las especificaciones del proyecto**

Ampliar el esquema que se tiene hasta ahora del proyecto con especificaciones lo suficientemente detalladas como para poder evaluarlo.

### **Evaluación de las opciones del proyecto**

Su objeto es la elección de una de las posibles opciones a desarrollar. De esta evaluación puede surgir el hecho de que ninguna de las opciones sea aceptada y por lo tanto el proyecto será considerado como inviable.

### **Informe del estudio de viabilidad**

Su objetivo es dar un informe en que figure toda la documentación de los trabajos realizados durante toda la fase previa del Estudio de Viabilidad.

## **FASE 2. Análisis**

### *Etapa 1 Análisis de la situación actual*

#### **Inicio del análisis**

Establecer el marco en que se va a desarrollar esta fase con sus tres etapas asociadas: análisis del sistema actual, especificación de requerimientos y selección de opciones técnicas.

### **Investigar el sistema actual**

Su objetivo es analizar el sistema actual, identificar sus procesos y realizar los diagramas de flujo correspondientes al mismo.

### **Investigar la estructura de datos del sistema**

Su objetivo es identificar las entidades mayores y sus relaciones en el sistema actual.

### **Desarrollo de la lista de problemas y requerimientos**

En este punto se debe revisar la lista y las actualizaciones hechas durante la fase de análisis del sistema actual.

### **Revisar los resultados del análisis**

Su objetivo es revisar que la documentación de toda la fase está realizada de acuerdo a los estándares y normas establecidas.

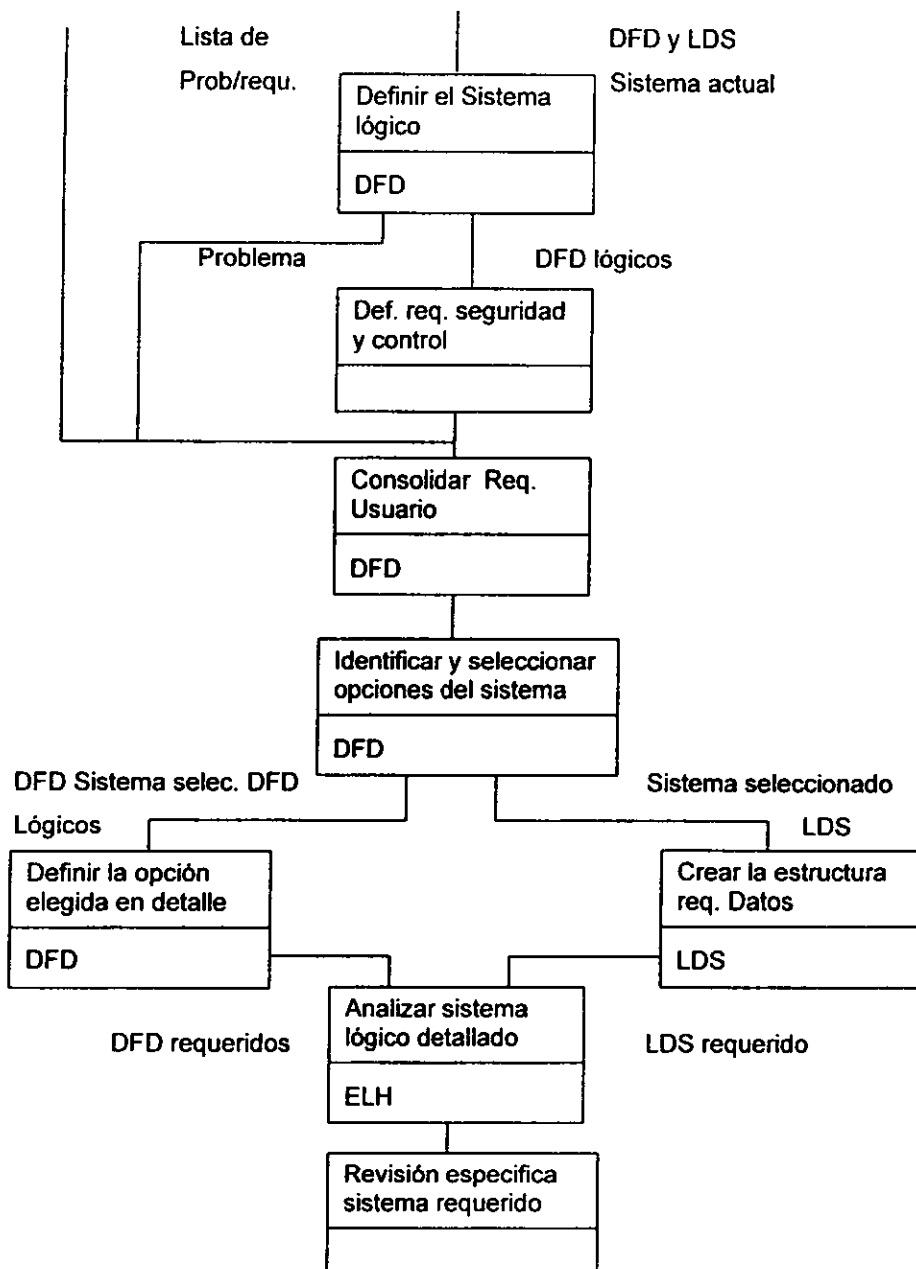
### **Definir el sistema lógico**

Su objeto es producir una visión lógica del sistema actual por medio de DFDs, simplifican flujos y procedimientos.

### **Definición de requerimientos de seguridad y control**

Su objetivo es hacer una lista inicial de requerimientos en estos aspectos.

Etapa 2. Especificación de requerimientos



**Figura 1.3.2.4. Especificación de requerimientos.**

**Identificar y seleccionar las opciones del sistema**

Su propósito es especificar el perfil del sistema deseado. Es uno de los pasos que siguen hasta seleccionar cual de las opciones es la mejor.

**Definir la opción elegida en detalle**

Su meta es hacer las especificaciones detalladas del sistema usando las técnicas de los flujos de datos.

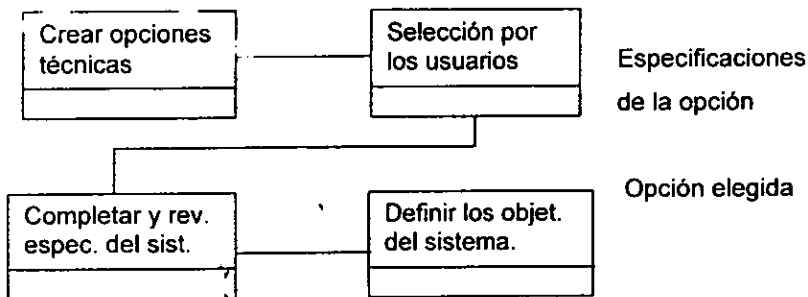
**Crear la estructura de datos requeridos**

Su objetivo, es crear la estructura lógica de datos necesarios para cumplir los requerimientos del sistema lógico.

**Investigar los detalles del sistema lógico**

Su fin es analizar la vida de cada entidad de datos del sistema, validando DFD y LDS, para sentar las bases para el diseño de procesos.

*Etapa 3. Selección de opciones técnicas*



**Figura 1.3.2.5. Selección de opciones técnicas.**

### **Crear opciones técnicas**

Su fin es identificar las posibles implementaciones físicas del sistema.

### **Selección por los usuarios de la opción**

Su finalidad es la elección de una de las opciones del paso anterior.

### **Completar y revisar las especificaciones del sistema requerido**

Debido a los cambios que han podido surgir, la opción final puede ser una de las propuestas pero con las inclusiones de parte de las otras, hay que fijar las especificaciones técnicas de esa opción final de cara a preparar el diseño físico.

### **Definir los objetos del diseño**

Fijando criterios de rendimiento del sistema para hacerlo efectivo. Serán requerimientos fijados por los usuarios y condicionarán el diseño físico.

### **Definir los procesos lógicos de actualización**

Describir los procesos asociados a cada suceso del sistema.

### **Revisar y validar el diseño lógico del sistema**

Para comprobar la integridad del sistema antes de pasar al diseño físico..

## **FASE 3. Diseño**

*Etapa 4. Diseño de datos y Etapa 5. Diseño de procesos*

### **Hacer un análisis relacional de datos**

Su meta es pasar del modelo de datos del sistema requerido a un conjunto de relaciones en tercera forma normal. Se partirá de LDS (estructura lógica de datos).

### **Hacer el diseño lógica de datos detallado**

Su objetivo es optimizar un diseño lógico de datos que sirva como base para el diseño físico, es decir, realizar el CLDD.

### **Definir los procesos de dialogo**

A partir del catálogo de lecturas y actualizaciones se comprobarán los formatos de entrada/salidas, diálogos lógicos y controles lógicos de diálogos.

## *Etapa 6. Diseño físico*

### **Crear las definiciones de archivos y bases de datos.**

Esto servirá de base a las siguientes fases de desarrollo. Hay que crear en el diccionario las entradas correspondientes a la descripción de archivos.

### **Completar las especificaciones de programas**

Las especificaciones serán diferentes dependiendo del entorno en el que se vayan a desarrollar los programas (lenguaje, máquina, etc.).

### **Hacer un plan de pruebas del sistema**

Su propósito es contemplar un conjunto de pruebas a realizar al sistema globalmente.

### **Realizar las instrucciones de operación**

Su fin es realizar informes o manuales con las instrucciones de operación para cada ciclo de explotación del sistema.

### **-Planificar la fase de implantación**

Teniendo en cuenta todas las tareas a realizar para la puesta en funcionamiento del sistema.

### **Realizar el manual de procedimientos**

Se trata en este paso de producir un manual que conjunte, resuma procedimientos automatizados con los manuales en la organización global de la empresa.

### **1.3.3. Merise**

La gran diversidad de lenguajes y formalismos empleados para la representación de sistemas informáticos hacen necesario un esfuerzo de unificación de criterios y métodos. En un momento en que hay que tratar de eliminar los problemas derivados de la rotación de personal.

Los principios en que se apoya MERISE son:

- Desglose del desarrollo en etapas.
- Definición de los documentos estándar de cada una.
- Uso del modelo entidad/relación y sus formalismos para la representación de datos.
- Uso de las redes de PETRI para la representación de procesos y tratamientos.
- Definición de grupos de trabajo y reparto de las responsabilidades y funciones a lo largo del desarrollo.
- Especificación del reparto de tareas y tratamientos entre los usuarios y el ordenador.
- Definición de los flujos de información entre las unidades del sistema.

El sistema se contempla desde diferentes niveles de abstracción y esto da lugar a una descripción del mismo a tres niveles: conceptual, lógico u organizativo y físico.

En la fase de concepción se trabaja básicamente sobre dos elementos: datos y tratamientos. La descripción de los datos reflejará la información existente en el

entorno y las relaciones entre ellas. La representación de los tratamientos reflejará los procesos a realizar con los datos así como su secuencia en el tiempo.

Estos dos elementos reflejan tanto el contenido del sistema como su funcionamiento. En la figura 1.3.3.1. se muestran los diferentes niveles de decisión como de descripción de datos y tratamientos:

NIVELES DE DECISION	NIVEL DE DESCRIPCION	
	TRATAMIENTOS	DATOS
Conceptual	Conceptual	Conceptual
Organizativo	Organizativo	Lógico
Técnico	Operativo	Físico

**Figura 1.3.3.1. Niveles de decisión.**

La descripción de estos niveles y cual es su contenido es el siguiente: el tratamiento, en nivel conceptual consiste en la descripción de lo que hay que hacer, es decir, es la descripción en términos de operaciones y resultados de la gestión que debe resolver el sistema independientemente de quién sea el que la realice, hombre o máquina, y de qué modo.

En el nivel organizativo, se desglosan las operaciones descritas a nivel conceptual en procedimientos funcionales, es decir, en tareas realizadas sucesivamente en un puesto de trabajo. A este nivel se concreta a quién, cuándo y dónde se han de realizar estas tareas.

En el nivel operativo se responde a la pregunta de cómo hay que hacer las cosas. Se detallan, tanto para procedimientos automatizados como para los manuales, las normas para realizarlos correctamente.



Se estructura la descripción de los datos desde diferentes niveles de abstracción.

En la descripción de datos contemplamos en primer lugar un nivel conceptual, en el que se observa la información del sistema en términos de objetos o entidades, se describen sus propiedades, la información de cada uno de ellos y las relaciones entre los mismos. Este modelo conceptual de datos es, en principio, bastante estable a lo largo del ciclo de vida del sistema.

En el nivel lógico se traduce el modelo conceptual en agrupaciones o estructuras lógicas de datos para su tratamiento por el sistema. Todavía este modelo debe ser independiente de la opción técnica en cuanto a soporte software del mismo.

En el nivel físico se concreta ya cual va a ser la estructura final de los datos de acuerdo al sistema gestor elegido. Una vez conocido esto se podrán hacer optimizaciones del modelo lógico para mejorar rendimientos.

En estos dos procesos, descripción de datos y de tratamientos, se hacen necesarias validaciones para comprobar la coherencia entre unos y otros. En este sentido se podrían estructurar los pasos a seguir en la siguiente figura 1.3.3.2. que propone un estudio paralelo de datos y tratamientos.

MERISE como todas las metodologías, propone una serie de etapas en el desarrollo de un proyecto.

- Estudio preliminar.
- Estudio detallado.
- Realización.
- Puesta en marcha o implantación.

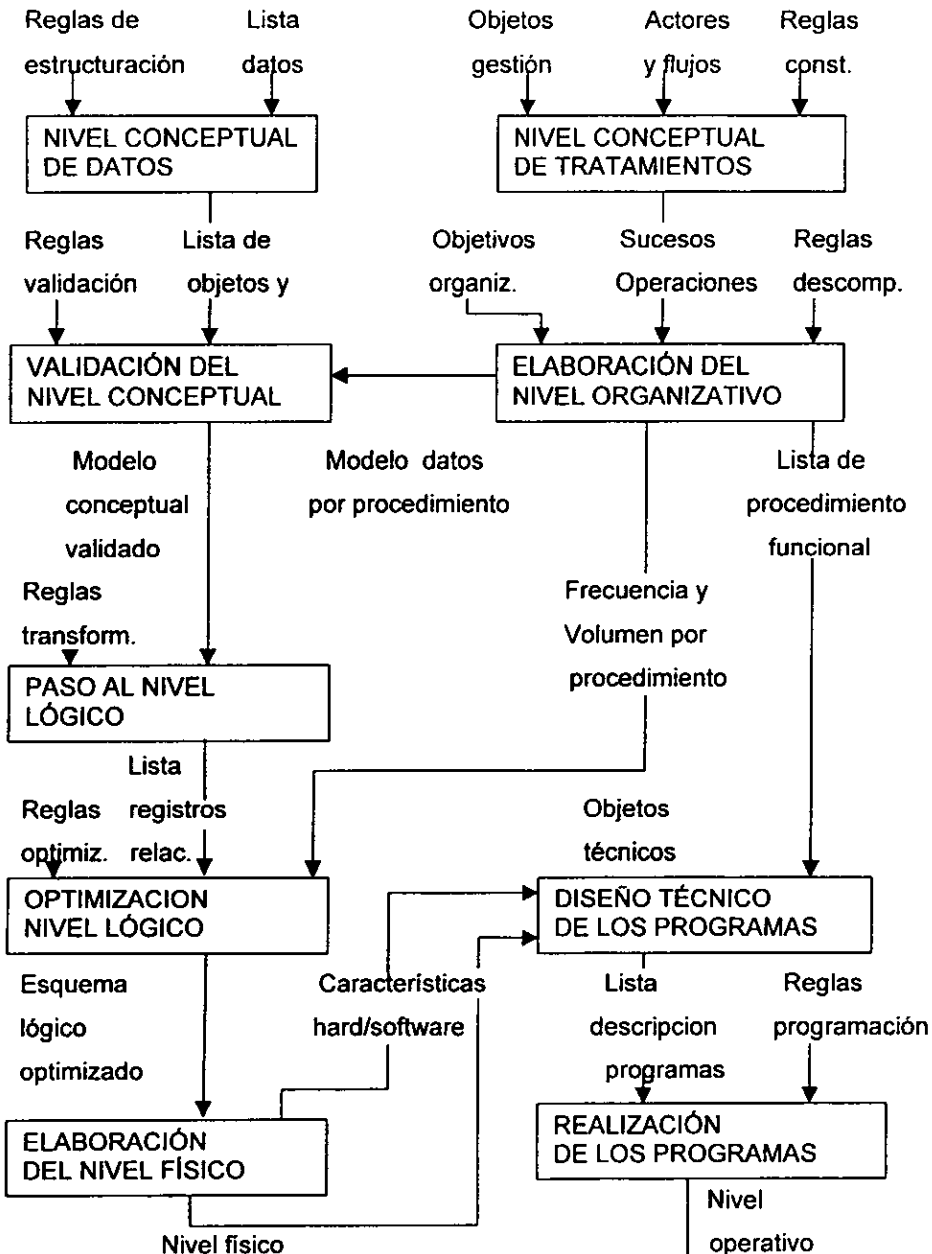


Figura 1.3.3.2. Niveles de descripción de datos y tratamientos.

Esto es lo que compone el ciclo de vida del sistema.

Otro punto contemplado por MERISE es la descripción de los diferentes grupos de trabajo, es decir, el personal implicado en el desarrollo del sistema a todos los niveles:

-Comité director que indicará los objetivos a cubrir por el sistema en las primeras fases de concepción.

-Comité de usuarios que aportará el punto de vista de los futuros utilizadores del mismo.

-Grupo de desarrollo o personal técnico encargado de satisfacer esos requerimientos a través del sistema informático.

#### Diagrama de flujos

Mediante esta técnica se representará gráficamente la organización de la empresa indicando los flujos de información entre los diferentes actores, entendiéndose por éstos a las personas, servicios o departamentos que generan actividad e información en la empresa.

Toda metodología, además de una serie de técnicas a emplear, propone una secuencia de etapas, fases y pasos a seguir en el desarrollo de un proyecto informático. Los pasos según la metodología de Merise cuentan con 4 etapas a lo largo de un proyecto.

Etapa 1. Estudio preliminar.

Etapa 2. Estudio detallado.

Etapa 3. Realización.

Etapa 4. Puesta en marcha.

Las acciones a realizar en estas etapas y su estructuración en las distintas fases. El índice general y fases sería el siguiente:

**Etapa 1. Estudio preliminar.**

**FASE 1. Recopilación de datos.**

Investigación inicial.

Estudio de la situación actual.

Síntesis y crítica de la situación actual.

**FASE 2. Concepción de la nueva solución.**

Objetivos a alcanzar.

Descripción de la solución.

**FASE 3. Evaluación y plan de desarrollo.**

Evaluación de la nueva solución.

Plan de desarrollo.

**Etapa 2. Estudio detallado.**

**FASE 1. Concepción general.**

**FASE 2. Concepción detallada de las fases.**

Realización de las especificaciones detalladas de los procesos.

**FASE 3. Plan de desarrollo.**

**Etapa 3. Realización.**

**FASE 1. Estudio técnico.**

**FASE 2. Producción.**

**Etapa 4. Puesta en marcha.**

**FASE 1. Preparación de los recursos físicos y humanos.**

**FASE 2. Recepción y lanzamiento del sistema.**

#### 1.3.4. Elección de la metodología óptima

Después de haber analizado las metodologías ya mencionadas llegamos a la conclusión que la de Yourdon es la más conveniente para la realización del proyecto ya que brinda las herramientas necesarias, entre las cuales tenemos la utilización de diagramas de flujo en la modelización de procesos ya que así obtenemos una mejor visión global del sistema y objetos a cubrir, en cambio la metodología de Merise no solo es inconveniente para la persona que lo realiza, también lo es para la validación del usuario por la complejidad que representa a nivel de detalle. Por lo tanto consideramos que a nivel conceptual el modelo entidad/relación es la más adecuada para reflejar la información del sistema, por la forma detallada que maneja este concepto en la metodología de Yourdon.

Así mismo esta metodología cuenta con un diccionario como soporte de información del sistema desarrollado haciendo especificaciones de programas que son basados en lenguaje estructurado y tablas de decisión. Hay que resaltar que ninguna de las tres llega a determinar que técnicas de programación estructurada se deben de seguir, aunque todas recomiendan su uso, ahora bien en cuanto a la metodología de SSADM no se logra determinar la técnica de programación que se va a utilizar y se llega solamente al diseño físico. La estructuración en la metodología de Yourdon tiene una mayor precisión en comparación con las demás y resulta más rentable para cualquier empresa.

Se podría decir que en el futuro el desarrollo de sistemas de información está ligado al uso generalizado de metodologías, gracias a su soporte sobre medios informáticos y herramientas para la automatización en la producción del software.

## 1.4. Sistemas operativos

Un sistema operativo es un conjunto de programas que sirve para controlar y organizar los recursos de un sistema de cómputo. Dichos recursos consisten en componentes de hardware, como son los terminales, las impresoras y los programas de software, los cuales le indican a la computadora que realice determinadas tareas.

La mayor parte de las computadoras personales cuentan con sistemas operativos para un usuario (o monousuarios); es decir, solamente una persona puede usar el sistema y éste sólo se ocupa de una tarea a la vez. En general, los sistemas mayores sirven para varios usuarios y realizan múltiples tareas; en estos sistemas varias terminales están conectadas a una sola computadora y el sistema operativo controla el acceso que puede tener a la computadora toda una comunidad de usuarios. Este tipo de computadoras tiene la capacidad para ejecutar varias tareas simultáneamente.

Como mayordomo, el sistema operativo ejecuta servicios que serían muy complicados o peligrosos si se dejara que otros programas los hicieran por sí mismos. Por ejemplo cuando un programa "quiere" escribir a un archivo en disco, el programa prepara la salida, y luego llama al sistema operativo para ejecutar la operación real de escritura.

De esta forma, el sistema operativo se encarga de todas las complicaciones de la selección de un área del disco que pueda guardar la salida, la transmisión de los datos, el mantenimiento de la organización del disco, etc.

Estos son algunos otros servicios que por lo general proporcionan los sistemas operativos.

**Ejecución de un programa:** cuando se tecléa un comando que uno quiere que se ejecute, es el sistema operativo el que hace todo el trabajo de cargar el programa en memoria y arrancarlo.

Llevar cuenta de la hora y la fecha: Algunos tipos de hardware tienen un reloj integrado, pero hasta en estos casos el sistema operativo ajusta el reloj y lee cuando un programa quiere saber la hora.

Ejecución de la mayoría de las operaciones relacionadas con la entrada y la salida y organización del disco, como el mantenimiento de la organización de archivos en disco, la recepción de entrada desde el teclado y la escritura de salida a la pantalla.

Como guardia de seguridad, el sistema operativo protege a los usuarios y a sus datos. Por ejemplo, el sistema operativo ejecuta todas las operaciones reales de escritura. Al hacerlo, el sistema operativo se asegura de que los datos escritos en un archivo no queden encima de datos que pertenecen a otro archivo. El sistema operativo evita que los archivos interfieran entre sí.

Si varias personas pueden usar la computadora simultáneamente, el sistema operativo también evita que esos usuarios interfieran entre sí. Por ejemplo, se asegura que ellos no puedan leer o escribir en los archivos de los demás. A menos que tengan permiso. También asigna a los usuarios su parte justa de los recursos de la computadora. Cada uno obtiene una parte de memoria, de espacio de disco, de tiempo de procesamiento disponible y de cualquier otro recurso que pueda ofrecer la computadora. Algunos sistemas operativos pueden negar a ciertos usuarios el acceso a ciertos recursos. Por ejemplo. A un usuario se le puede permitir solamente ejecutar un pequeño grupo de programas.

A veces los sistemas operativos restringen el acceso a la computadora a usuarios autorizados. Si uno trata de comunicarse con un sistema de éstos, se le pide que se identifique con un nombre o una contraseña. Si uno no es usuario autorizado, el sistema operativo no le permitirá trabajar en él.

### 1.4.1. Unix

Unix es un sistema operativo multiusuario y multitareas. Cuenta con programas para la edición de textos, envío de correo electrónico, la elaboración de tablas, ejecución de cálculos y muchas otras funciones especializadas, cada una de las cuales requeriría un programa de aplicación especial en el caso de otros sistemas operativos.

Son cada vez más los usuarios –escritores, científicos, programadores, administradores, etc.- que recurren a Unix para realizar tareas tales como:

- Edición de textos e impresión.
- Manejo y almacenamiento de documentos.
- Programación y diseño de software.
- Comunicaciones electrónicas.
- Instrucción asesorada por computadora.
- Control de procesos industriales.
- Administración de empresas.

Una de las características más poderosas de Unix es su potencial de personalización. Se dispone de gran cantidad de software popular en Unix. Antes y después del auge de las computadoras personales, el Unix se ha mantenido como el sistema favorito para el desarrollo de programas de punta. Muchos sistemas comerciales son particularmente buenos para una clase de trabajos, pero malos para otros. Debido a que el Unix es extremadamente personalizable, se le puede ajustar para que haga el trabajo que más se ocupa. Se puede encontrar al Unix para casi cualquier tipo de hardware. Esto hace posible que se usen las mismas utilerías en una amplia variedad de máquinas.

Hay varios "shells" (ambientes de trabajo) de amplio uso en los sistemas Unix, un shell es un programa que ejecuta otros programas, por lo general, cuando se ejecuta un programa, el shell tiene que encontrar el archivo que tiene ese programa, cargar el programa en memoria y luego iniciar la ejecución del mismo. Esto puede llevar una



cantidad significativa de tiempo, en especial en computadoras antiguas con procesadores y discos lentos.

Para ahorrar tiempo, algunos shells tienen comandos integrados, es decir, comandos que pueden ejecutar inmediatamente sin tener que encontrar y leer el archivo adecuado. Es común que los comandos usados más frecuentemente se encuentren integrados dentro del shell, para reducir el tiempo de espera para su ejecución.

El shell "habla" con el usuario a nombre del sistema operativo. Lee la línea de comandos que se teclea, determina lo que significa y hace que sea necesario para ejecutar esos comandos.

A veces los shells proporcionan facilidades adicionales más avanzadas. Por ejemplo, algunos shells tienen características que se pueden usar para reducir el tecleo.

Algunos shells pueden permitirle asignar formas cortas para comandos más largos. Los shells pueden llevar registro de todos los comandos que se ha dado recientemente, para que se les pueda editar o rejecutar en alguna forma simple.

Los shells pueden permitirle ejecutar un conjunto de comandos que estén guardados en un archivo. En esta forma se puede configurar un archivo que contenga una secuencia de comandos, y luego ejecutar los comandos del archivo con una sola instrucción.

Algunos sistemas operativos reconocen un solo shell. En tales sistemas, sólo una parte del software tiene la capacidad de ejecutar otros programas y no se puede usar ninguna otra cosa. Sin embargo, los sistemas operativos como el DOS y Unix no lo atan a un solo shell. Ellos permiten que se cree o se compre un shell diferente, y que se use ese shell en vez del estándar. Hay varios shells de amplio uso en los sistemas

Unix como el Bourne-Shell, el C-Shell y el Korn-Shell que tienen diferentes características:

**Bourne-Shell:** Es el intérprete de comandos básico.

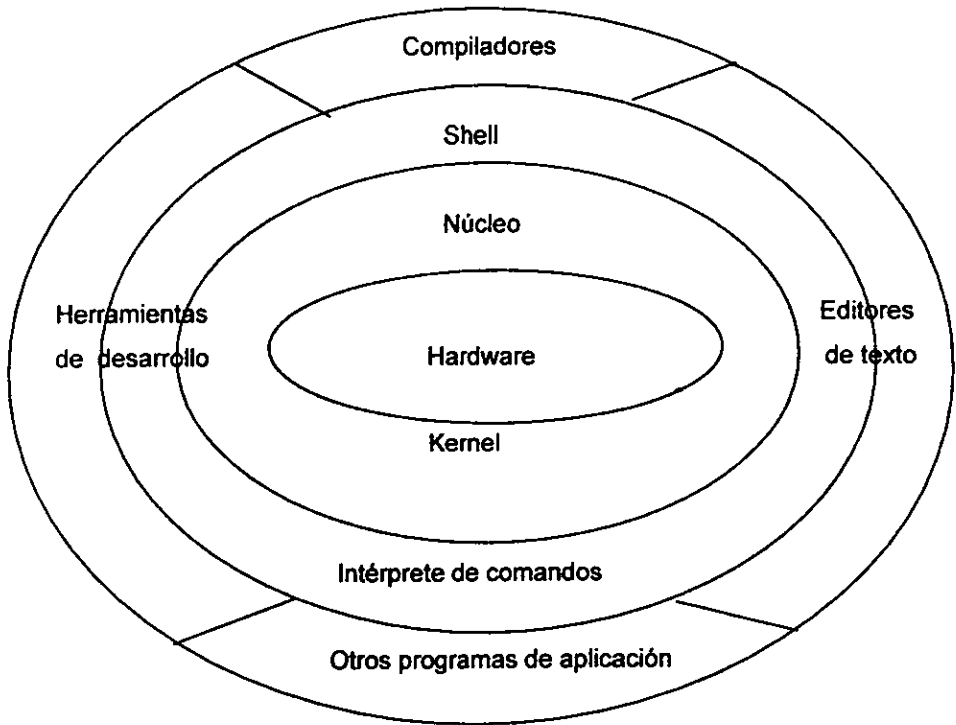
**C-Shell:** Es el intérprete de comandos creado en Berkeley para el sistema operativo BSD y para el Xenix un poco más completo que el anterior. Su programación es prácticamente "lenguaje C".

**Korn-Shell:** Se basa en los dos anteriores, siendo compatible con el Bourne en un 95%. Añade posibilidades de programación avanzada, facilidades aritméticas y mayor rapidez de ejecución, siendo el shell estándar para el Unix.

Unix es un sistema multiusuario. Los diseñadores del Unix anticiparon que diversas personas querían usar un sistema, y que varias personas desearían tener archivos personales que no quisieran compartir con los demás usuarios. La seguridad de los archivos es sólo una diferencia entre los sistemas multiusuarios y los sistemas de un solo usuario.

Si se quiere comenzar a usar un sistema multiusuario, se tiene que identificar usted mismo, para que el sistema sepa cual usuario es usted. Por lo tanto, los usuarios de Unix tienen que ejecutar un procedimiento de registro (login) en el cual le dicen al sistema su nombre y contraseña, para probar que son quienes dicen ser.

El sistema operativo Unix esta formado por una serie de elementos que se pueden representar en forma de capas concéntricas donde, en primer lugar, alrededor del hardware de la máquina se encuentra el núcleo (kernel), que interactúa, directamente con el hardware, aislando a éste de los usuarios, además de adaptar el resto del sistema operativo a la máquina a la portabilidad que existe en el mismo, figura 1.4.1.1.



**Figura 1.4.1.1. Estructura del sistema operativo UNIX.**

El sistema operativo Unix se compone de bloques funcionales que se encuentran representados en la figura 1.4.1.2.

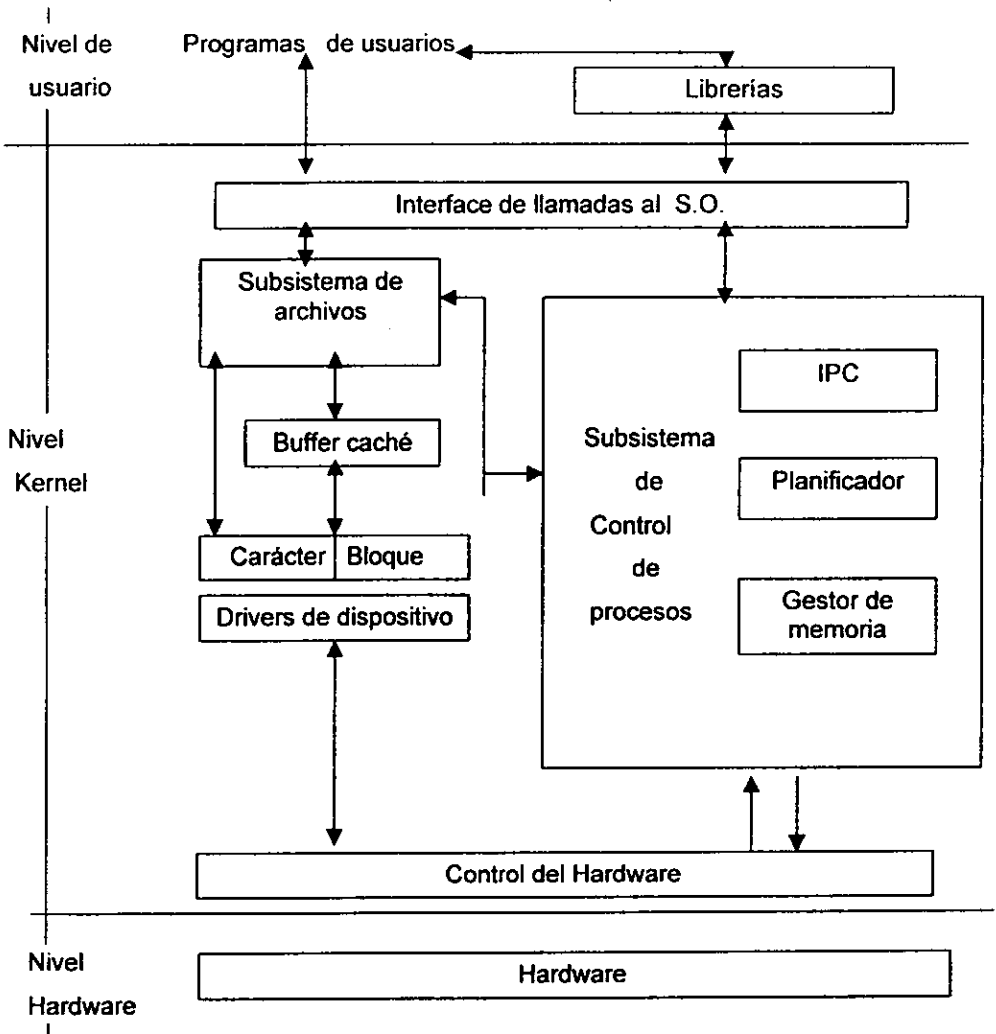


Figura 1.4.1.2. Bloques funcionales del sistema operativo UNIX.

### Archivos y directorios

La estructura de archivos y directorios es jerárquica y está constituida de forma arborescente, donde los nodos son directorios y las hojas son archivos normales. El árbol tiene un directorio raíz. El sistema de archivos Unix gestiona varios tipos de archivos:

-Archivos normales o regulares: Son los archivos de usuario que contienen programas, texto, etc.

-Directorios: Son particiones lógicas que a su vez son archivos que contienen la información necesaria para poder encontrar un archivo en el disco.

-Archivos especiales: Se utilizan para designar periféricos de entrada y salida.

-Pipes: Archivos que permiten la transferencia de datos entre procesos

En el sistema de archivos, cada archivo tiene asociado un conjunto de permisos que determinan qué puede hacerse con él y quiénes tienen acceso.

### Gestión de la memoria

La gestión de la memoria en el sistema operativo Unix se basa en intercambio y paginación. La paginación de la memoria se lleva a cabo si el hardware de la computadora la soporta. La política de carga y descarga de un proceso en la memoria depende del tiempo que lleve la misma, de su actividad y del tamaño.

### Portabilidad

La portación o portabilidad es el proceso de tomar un programa que se ejecute en un tipo de computadora y hacerlo que trabaje en otro tipo. Esto puede ser muy difícil, especialmente si las computadoras tienen diferente hardware y sistema operativo. Por otro lado, si las computadoras están ejecutando el mismo sistema operativo y el mismo programa no se apoya en idiosincrasias del hardware, la portación puede ser fácil.

Estandarizar el comportamiento de los sistemas operativos similares al Unix es el primer paso para la simplificación del trabajo de portación. Si los programadores saben que todos los sistemas similares al Unix proporcionan un juego de características estándar, pueden usar esas características en las aplicaciones que creen.

Al restringirse ellos mismos a las características que se garantiza que estarán disponibles en todos los similares al Unix, las aplicaciones llegan a ser más fáciles de portar. La portabilidad es importante por ejemplo: Si se adquiere una nueva computadora, por lo general se querrá mover todos los programas favoritos a la nueva máquina. Esto es particularmente importante para las compañías que tienen gran cantidad de programas hechos para llevar sus cuentas, sus pagos, la producción, etc. Las compañías no quieren reescribir todo su software simplemente porque compraron una nueva máquina. Los programas deben durar más que el hardware.

### **Control de procesos**

El núcleo del sistema operativo Unix conoce la existencia de un proceso a través de un bloque de control del proceso, donde se describe el proceso y su entorno, constituyendo un contexto consistente en:

-Espacio de direccionamiento y entorno de ejecución: Variables que utiliza el proceso.

-Contenido de los registros de hardware: Contador de programa, registro de estado del procesador, puntero de la pila registros de propósito general.

-Contenido de las estructuras del núcleo relacionadas con el proceso: Tablas de proceso, áreas, regiones, etc.

Si congelamos el proceso del procesador y el proceso que está en ejecución en un determinado momento, obtendríamos lo que se conoce como imagen estática del programa. En caso de producirse una interrupción o cambio de proceso, se almacena la imagen de la que está en ejecución en ese mismo instante.

Cada proceso se conoce dentro del sistema por un número que lo identifica unívocamente y que se conoce como Identificador del Proceso o "PID"

Todos los procesos, excepto el proceso 0, son creados por otro proceso; es decir, el sistema de creación y gestión de procesos en el sistema operativo Unix es jerárquico.

El proceso que se genera con el "PID 0" es un proceso especial creado en el momento de arrancar el sistema.

Se genera un proceso "init" que será el antecesor de todos los procesos que se generen en el sistema. A partir de aquí se generan tantos procesos como terminales existan.

Los procesos que atienden a las terminales crearán otros con el fin de identificar y controlar el acceso a los usuarios al sistema, los cuales, una vez que inicien la sesión, ejecutarán un proceso interprete de comandos Shell que será el que genere el resto de programas por el usuario.

### Ventajas y desventajas

Entre las ventajas que permiten al sistema operativo Unix gozar de la popularidad que en la actualidad tiene, se pueden citar las siguientes:

- Resulta de fácil lectura por estar escrito en lenguaje de alto nivel (C).
- Es un sistema jerárquico de procesos y archivos.
- Posee una interface con periféricos consistente y uniforme.
- Es un sistema multiusuario y multiproceso.

- Adapta los programas (fuente) a cualquier máquina con sistema operativo Unix.
- No está ligado a una marca comercial fabricante de computadoras.
- Permite la combinación de utilidades para producir otras nuevas.

Las desventajas que presenta este sistema son:

- Comandos poco claros y con demasiadas opciones.
- Sistema de archivo lento.
- En los SWAPS a memoria de procesos grandes, consume mucho y se alenta.
- Difícil, de sintonizar en situaciones no típicas.

## **Linux**

Una alternativa en UNIX: Linux

Linux es conocido como un sistema operativo UNIX (y así lo manejaremos en la mayoría de los casos), pero si nos vamos al significado estricto sobre ¿qué es Linux?, entonces la respuesta es que Linux es un kernel1, y para que sea un sistema operativo se necesita que el kernel sea complementado con un conjunto de utilidades para uso y configuración, así como de instalación, de esto último se derivan las distribuciones, las cuales cuentan con el mismo kernel pero con diferentes utilidades y lo mas importante es que es un sistema desarrollado en ambiente gráfico. Figura 1.4.1.3.





**Figura 1.4.1.3. Ambiente gráfico de Linux.**

Algunas de las características de Linux son:

Es multiplataforma (Existe para Intel desde 386, SPARC, PowerPC, Alpha, etc.).

Es multitarea (ejecución de varios programas al mismo tiempo).

Es multiusuario (soporta a varios usuarios conectados a la vez).

Maneja la protección de memoria (de manera que si un proceso falla, éste no afecte a todo el sistema).

Maneja la solicitud de ejecutables cargables (sólo lee y carga en memoria las partes que se utilizan de un programa).

Maneja memoria virtual usando paginación a disco (no se transfieren todos los procesos al disco, sino sólo los necesarios).

Manejo de banco unificado de memoria para programas de usuario y cache (de tal manera que la memoria libre puede ser utilizada como cache, y ser reducida cuando se ejecutan programas grandes).

Librerías ligadas dinámicamente (DLL).

Elaboración de archivos core (Para hacer un análisis del sistema no sólo cuando éste está funcionando sino también cuando falla).

Tiene código fuente disponible (el cual incluye al kernel en su totalidad y todos los controladores).

Soporta pseudoterminal (pty).

Maneja la emulación matemática a nivel del kernel (de manera que los programas no necesitan hacer la emulación matemática).

Soporta gran variedad de teclados.

Soporta múltiples consolas virtuales (con lo cual permite diferentes sesiones en la consola).

Soporta diferentes tipos de sistemas de archivos (entre los cuales están minix-1, Xenix, etc.).

Soporta el manejo transparente a particiones MS-DOS y OS/2 FAT (con lo cual tienes acceso a los archivos para lectura y escritura, que están en una partición de DOS y/o OS/2).

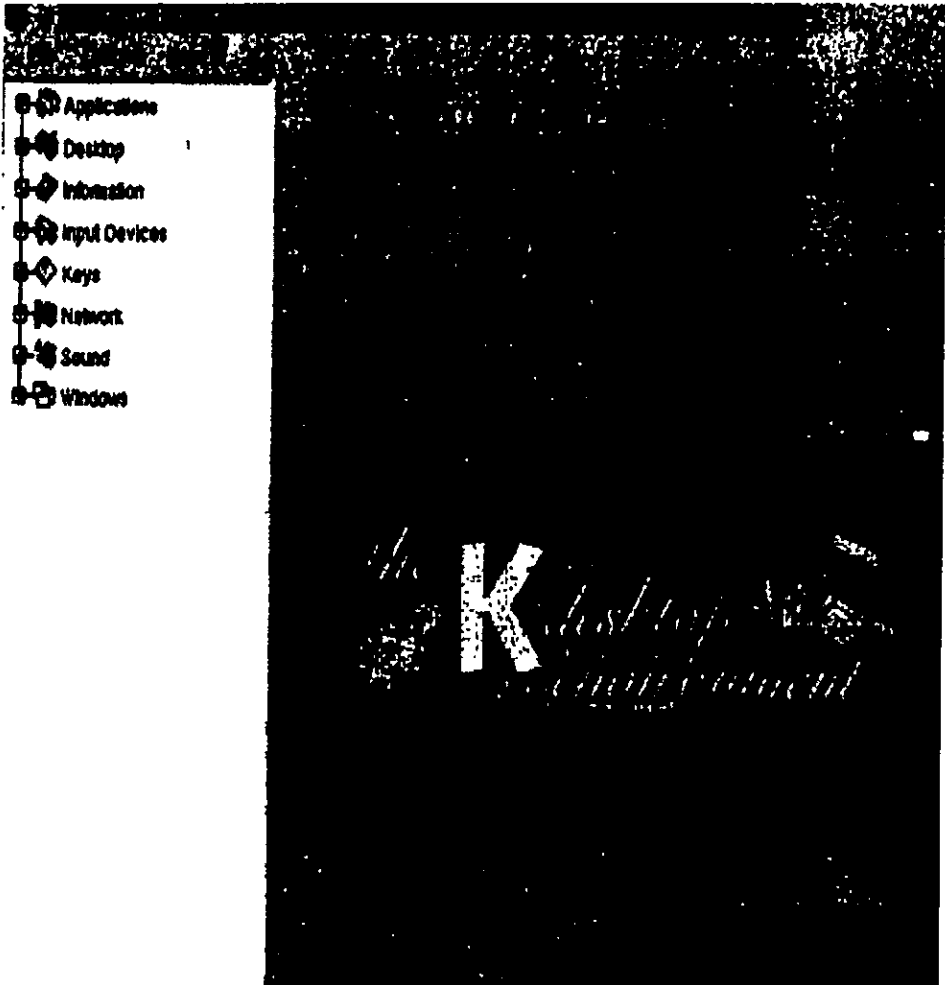


Figura 1.4.1.4. Control central de Linux.

Soporta varios protocolos de comunicación (TCP/IP, SLIP, X.25, etc.).

Es completamente libre (es decir, es ¡GRATIS!).

Una de las características más importantes de Linux es que trabaja en modo gráfico, por lo que es completamente amigable para el usuario, figura 1.4.1.4.

Para entender que es una distribución supongamos lo siguiente: un grupo de programadores deciden utilizar Linux (como kernel) para agregarle las utilidades de configuración y uso, herramientas de instalación, etc., de tal forma que hacen un sistema operativo, el cual deciden llamarlo EJEMPLINUX. Entonces EJEMPLINUX es lo que se llama una distribución. Como ya se mencionó todas las distribuciones tienen como base el kernel, por lo que son muy similares.

Algunas de las distribuciones de Linux son:

Craftworks (distribuida por Craftworks Solutions, Inc.).

Debian (distribuida por la Debian Linux Association).

DILINUX (distribuida por Kent Robotti).

Linux Pro (distribuida por Workgroup Solutions, Inc.).

Red Hat (distribuida por Red Hat Software).

Trans-Ameritech Linuxware (distribuida por Trans-Ameritech).

Walnut Creek Slackware (distribuida por Walnut Creek CDRom).

Etc.

La distribución más conocida es la de Slackware, aunque entre las más recomendadas están Red Hat y Debian, por el hecho de que son libres y ofrecen diferentes utilidades para trabajo. De entre las desventajas de Linux tenemos las siguientes:

### **Actualización Constante y Soporte**

La primera no es precisamente una desventaja, sin embargo se menciona mucho como tal. Esto se debe a que Linux siempre está en constante desarrollo, lo cual hace

suponer que si tengo una versión con la cual estoy trabajando, entonces si sale una nueva versión debería de actualizar mi versión. Lo cual no es obligatorio, porque la actualización sería necesaria si queremos utilizar hardware que no es soportado por la versión que estoy utilizando y que en la nueva versión si lo soporta. O también sería necesaria la actualización si queremos hacer uso de alguna característica que trae la nueva versión.

En cuanto al soporte existen algunas distribuciones que son completamente comerciales y sí ofrecen soporte, pero si adquieres una distribución como Debian no ofrece soporte, por ser completamente libre. Sin embargo, para solucionar de alguna forma esta desventaja, existe un proyecto que tiene por objetivo el documentar todos los temas relacionados con Linux (desde los manuales en línea hasta la instalación de software variado, entre ellos la instalación de Linux), este proyecto se llama LDP (Linux Documentation Project). De entre los documentos que están en el LDP, están los HOWTO's, los cuales nos explican paso a paso lo que puede ser una instalación de software, la configuración de un servicio o sólo la explicación de un tema. Para la instalación de Linux está el Installation-HOWTO, y si te interesa ver que hardware soporta está el Hardware-HOWTO.

Las formas de instalación son interesantes, pues existe variedad, ya que puedes instalar Linux obteniendo los archivos desde un servidor en Internet ( a través de FTP o NFS), hasta utilizando una partición DOS, que funcione como dispositivo de almacenamiento de los archivos de Linux.

Linux puede ser instalado en una PC junto con otro(s) sistema(s) operativo(s), por ejemplo DOS (por lo tanto con Windows) y OS/2, para lo cual es necesario particionar el disco duro, pero si ya tienes DOS instalado y no deseas hacer la partición, puedes instalar Linux en tu partición DOS y te quedará como un directorio, así, si deseas desinstalar Linux, sólo tienes que borrar el directorio (si te interesa esta opción puedes leer el documento UMSDOS-HOWTO). La desventaja de este tipo de instalación se refleja en el rendimiento, se puede ver un poco lento el desempeño.

Existe una gran variedad de software con el cual tú puedes explotar Linux, cuenta con editores de texto, hojas de cálculo, emuladores para DOS y Windows (aunque sólo puedes ejecutar algunas aplicaciones de estos ambientes), manejadores de bases de datos, compiladores, software de red, aplicaciones en ambiente gráfico, etc.

1.- Kernel. Es el núcleo del sistema operativo, el cual ejecuta tareas como el manejo de memoria, procesos, dispositivos, etc.

2.- Minix. Sistema operativo Unix para PC's XT, desarrollado por Andy S. Tanenbum  
S.u.S.E. Linux 5.3.

S.u.S.E. Linux contiene lo siguiente: la última versión del sistema operativo Linux y más de 850 paquetes de programas Linux en 5 CD-ROMs; nuestro propio libro de referencias de 400 páginas; "YaST", la herramienta de administración de un sistema Linux más extensa del mundo; y mucho más.

**Novedades en S.u.S.E. Linux 5.3 (respecto a S.u.S.E. Linux 5.2)**

A continuación la lista de los cambios y mejoras más importantes:

**Kernel 2.0.35:**

    Soporte de los sistemas de ficheros FAT32 y Joliet.

    Raid 1, Raid 4 y Raid 5 están soportados en software para todos los discos.

    Soporte mejorado para periféricos que utilizan el puerto paralelo.

**XFree86™ 3.3.2.3.**

**XFCom\_SIS** con soporte para: SiS 86c201, SiS 86c202, SiS 86c205, SiS 5597/5598 (beta).

**XFCom\_Cyrix:** soporta Cyrix MediaGX (no acelerado).

**XFCom\_Rendition:** soporte (no acelerado) para Rendition Verite chips (Rendition Verite 1000, 2100 y 2200).

**XBF\_NeoMagic:** Binary Free X-Server para NeoMagic 128XD y 128ZV (beta).

Configurar el sistema X Windows.

Nuevas características de la herramienta de administración e instalación yaz :

Instalación revisada. Ahora ofrece dos formas de instalación: Quickinstall o Expert Mode.

Más fácil de particionar y asignar sistemas de ficheros.

Nueva versión de fdisk (Win95-Fat32 y con compatibilidad Win95-extended).

Soporte FTP para instalación y update (¡ahora soporta full update via ftp!).

KDE 1.0

GIMP 1.0

### **Características**

Rutina de instalación por Menús. Según el Linux Journal, "de lo más simple".

Permite elegir libremente el tamaño de la instalación.

Susewm: herramienta para generar y actualizar menús basándose en los paquetes de siete importantes window managers (incluido KDE's kwm, Fwmm, Fwmm2, Bowman, Afterstep, Ctwm y Mwm!).

YaST, herramienta de S.u.S.E. para administrar el sistema Linux:

Permite la consolidación de administración de sistemas y redes.

Instala paquetes RPM y TGZ.

Enseña contenidos de paquetes y sus descripciones.

Hace búsqueda de contenidos de paquetes y descripciones.

Permite la verificación de permisos y sus dueños.

Administra grupos y usuarios.

Configura impresoras de redes.

Configura niveles de seguridad.

Actúa como consola - no es necesario configurar X Windows previamente. Está optimizado para tareas de administración remota.

Manual extensivo de 400 páginas que abarca desde lo más básico hasta tópicos avanzados como (Inglés):

Instalación y particiones.

Integrar Linux con otros sistemas operativos.

Los comandos básicos de Linux y procedimientos.

Utilizar YaST para maximizar su sistema.

Administración de Sistema y Redes.

Iniciar el sistema X Window y configurar los window managers.

Imprimir y escanear.

Utilizar aplicaciones útiles como LaTeX, emacs, emuladores, etc.

Conectar a Internet.

Servidores Apache y Samba.

Redes con S.u.S.E. Linux.

Concepto de arranque compatible con el System -V y accediendo al estándar del sistema de ficheros.

Poder elegir 11 windows managers diferentes.

Sistema de ayuda de hipertexto extensivo:

S.u.S.E. con soporte de teclado para búsquedas en las bases de datos.

Documentación para todos los paquetes.

Utiliza el language HTML, del World Wide Web (WWW).

600 MB live file system, sistema de ficheros en CD-ROM.

Graphical User Interface XFree86 3.3.2.3 (X11R6.3 "Broadway"), fácil configuración con XF86Setup.

Arranca desde el CD-ROM (sí el BIOS lo soporta).

Todas las herramientas necesarias para conectar su sistema a Internet como correo, noticias y WWW.

Código completo del sistema y todos los paquetes.



Modo demo ,directamente desde el CD-ROM o desde la partición del DOS.

Emuladores para: DOS, Atari ST, Amiga, C64, C128, VIC20, PET, ZX Spectrum, Gameboy, Nintendo Entertainment System, Atari VCS2600, Coleco Vision, ZX81.

### **¿Reemplazará Linux a los sistemas operativos comerciales?**

El enorme esfuerzo de desarrollo y amplia distribución del sistema operativo (SO) Linux asegurará que tome su lugar como una alternativa real, viable, y significativa a los sistemas operativos comercialmente restringidos. El modelo de desarrollo abierto, disponibilidad de la fuente y ausencia de restricciones de licencia son "características" del SO Linux que los desarrolladores de SO comerciales simplemente no pueden ofrecer. Los grupos de desarrollo de software que necesita este modelo incluyen grupos de organizaciones de investigación afiliados al gobierno, proyectos académicos de investigación y de enseñanza, desarrolladores de aplicaciones de software comercial.

El reciente rápido incremento en nuevas aplicaciones disponibles para Linux y la base de usuarios rápidamente creciente de estas tecnologías están causando incluso que las mayores organizaciones de la industria de la computación tomen Linux seriamente. Incluso Datapro (una compañía de McGraw-Hill) en su reciente examen de 1996 de la industria UNIX concluyó que: "Los programadores están considerando seriamente la viabilidad de Linux en plataformas de producción ahora que Linux cuesta menos que Microsoft y tiene los beneficios añadidos de UNIX, como gran rendimiento, conjunto inherente de potentes herramientas y capacidades de comunicación".

#### **1.4.2. Windows9x/Windows NT**

Windows NT es un sistema operativo completamente nuevo, que ha sido diseñado para sacar todo el partido de la gran potencia de los nuevos sistemas de escritorio. Windows NT representa un paso adelante muy importante en las características de Windows 3.1

y ofrece características que lo convierten en algo realmente único en el mundo de los sistemas operativos. Windows NT está dirigido a un amplio espectro de usuarios:

-Usuarios finales, que precisan de un alto rendimiento y la posibilidad de conmutar entre múltiples aplicaciones.

-Usuarios de grupos de trabajo, que han de compartir los sistemas, conectarse con otras computadoras también compartidas, intercambiar correo electrónico y mantener un registro actualizado de encuentros y citas.

-Desarrolladores de software, que desean crear aplicaciones para ser ejecutadas en sistemas que soporten Windows NT.

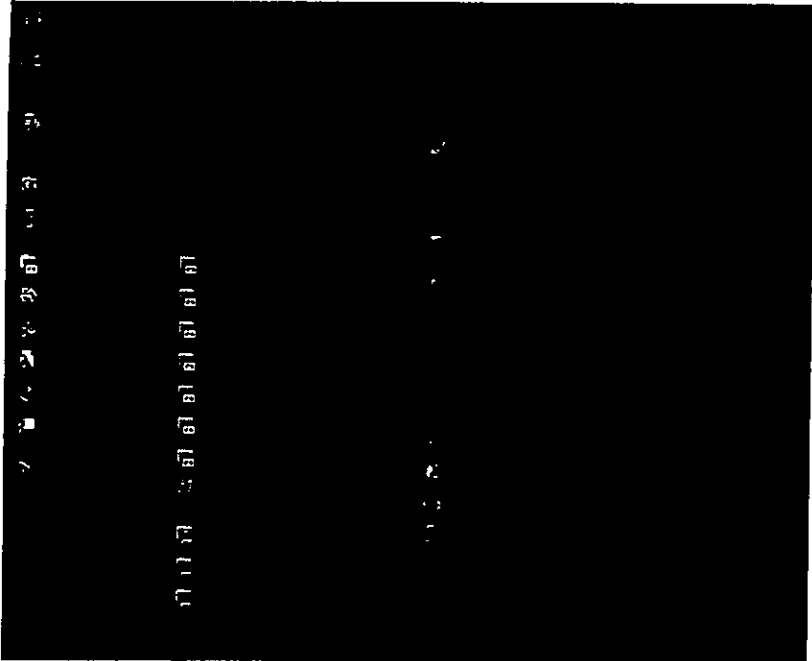
-Administradores de redes locales, que precisan de un entorno de red seguro, el cual puede sacar partido de los sistemas multiprocesador, figura 1.4.2.1.

### **Windows NT: Una vista al interior**

Windows NT es un sistema operativo de 32 bits con multitarea con derecho preferente y protección de memoria, que cuenta además con soporte para multiprocesamiento simétrico y trabajo en red, todo ello con una interfaz totalmente gráfico para el usuario.

El sistema operativo Windows NT está diseñado para operaciones de 32 bits. Esto permite sacar ventaja de los microprocesadores más avanzados, como el Intel 80486, así como de las computadoras con tecnología de conjunto reducido de intrucciones (RISC), como es el caso de MIPS 4000 o DEC Alpha. Los microprocesadores más antiguos utilizaban conjuntos de instrucciones de 16 e incluso 8 bits. Por ejemplo se quiere escribir un número de 32 dígitos en una hoja de papel muy estrecha. Si no hay espacio suficiente, deberá escribir los 16 primeros en una línea y otros 16 en una segunda línea. De forma análoga, los microprocesadores de 32 bits pueden trabajar con cantidades numéricas, direcciones de memoria e instrucciones de mayor tamaño,

sin necesidad de dividir las operaciones en bloques más pequeños para que puedan tener cabida en el microprocesador.



**Figura 1.4.2.1 Windows NT**

El rendimiento general, es decir, la combinación del rendimiento del microprocesador, transferencia de datos y acceso a memoria es, en definitiva, lo que ha venido a mejorar sensiblemente. La multitarea significa que el sistema operativo puede realizar varias tareas al mismo tiempo. El que sea con derecho preferente, significa que el usuario, o bien otra tarea, pueden interrumpir una tarea determinada si fuese necesario, sin tener que esperar obligatoriamente a que ésta termine. Conforme aumenta la velocidad de proceso, las actividades relacionadas con el hardware, como puede ser, por ejemplo, el acceso al disco, pueden resultar tremendamente lentas. Cuando un sistema sin derecho preferente accede al disco, el microprocesador ha de esperar hasta que tenga

lugar efectivamente el acceso mecánico al disco, malgastando en definitiva ciclos de procesamiento. En Windows NT existe la posibilidad de concurrencia de diversas tareas, de manera que si una de ellas queda parcialmente parada, como consecuencia, por ejemplo, de un lento acceso al disco, el microprocesador puede concentrar su atención en otras tareas. Básicamente, lo que se consigue es que no se malgasten ciclos de procesamiento. Puede verse una tarea individual como un corredor de velocidad. Si decae el ritmo de uno de ellos, los otros no frenen su marcha. El beneficio de esto es que los usuarios pueden realizar otras tareas mientras tiene lugar un acceso al disco en segundo plano, o bien se prepara un trabajo de impresión.

La protección de memoria asegura que los diversos programas van a ser ejecutados cada uno en su área individual de memoria, no pudiendo contaminar nunca las áreas correspondientes a otras aplicaciones. En el caso de que una determinada aplicación quede bloqueada, las aplicaciones restantes, así como el sistema operativo permanecerán inalterados, dando así la oportunidad al usuario de dar fin a su trabajo en la forma correcta.

El multiprocesamiento simétrico es una característica exclusiva de Windows NT, que permite sacar todo el partido de los procesadores múltiples. Sí bien es cierto que la vida de los sistemas multiproceso es relativamente corta. NT es el primer sistema operativo que está preparado para realizar un uso eficiente de este tipo de tecnología. Los sistemas operativos anteriores se limitaban a asignar tareas específicas a cada microprocesador individual, como pueden ser la entrada/salida en una red local, por ejemplo. El multiprocesamiento en donde un microprocesador se dedica exclusivamente a una tarea específica da lugar a que un microprocesador permanezca inactivo en cuanto finaliza su tarea. En el multiprocesamiento simétrico, el sistema operativo puede asignar diferentes tareas a un mismo microprocesador; así, si uno de ellos termina su trabajo antes que otro, el sistema operativo podrá ocuparlo en otra actividad. El multiprocesamiento simétrico es bastante más difícil de implementar, pero ofrece un rendimiento muy superior.

Las características de trabajo en red local con las que cuenta Windows NT permiten compartir archivos de un sistema con cualquier usuario de un entorno de red, así como conectarse a directorios compartidos de otros sistemas. Las computadoras donde se ejecuta Windows para grupos de trabajo pueden aprovechar las ventajas del trabajo en red. Además, Windows NT viene equipado con el software y los controladores necesarios para establecer las oportunas conexiones con otros tipos de sistemas operativos, como es el caso de Unix o las grandes computadoras IBM. Windows NT presenta externamente un aspecto muy similar a Windows 3.1. De hecho, el número de versión para Windows es precisamente el mismo 3.1. Windows NT utiliza la misma interfaz gráfica de su predecesor, la cual proporciona un modo cómodo y eficaz de gestión de las aplicaciones, figura 1.4.2.2. Además, puede ejecutar miles de aplicaciones Windows

### **3.1. y MS-DOS**

Otras de las características de Windows NT es que ciertamente fue diseñado con la idea del futuro siempre en mente. Se trata de un sistema operativo transportable, lo que quiere decir que puede ser modificado fácilmente para adecuarse a muy distintas plataformas de hardware. Sus posibilidades de procesamiento simétrico le proporcionan un amplio margen de rendimiento potencial en relación con otros sistemas operativos.

#### **Ventajas del multiprocesamiento simétrico**

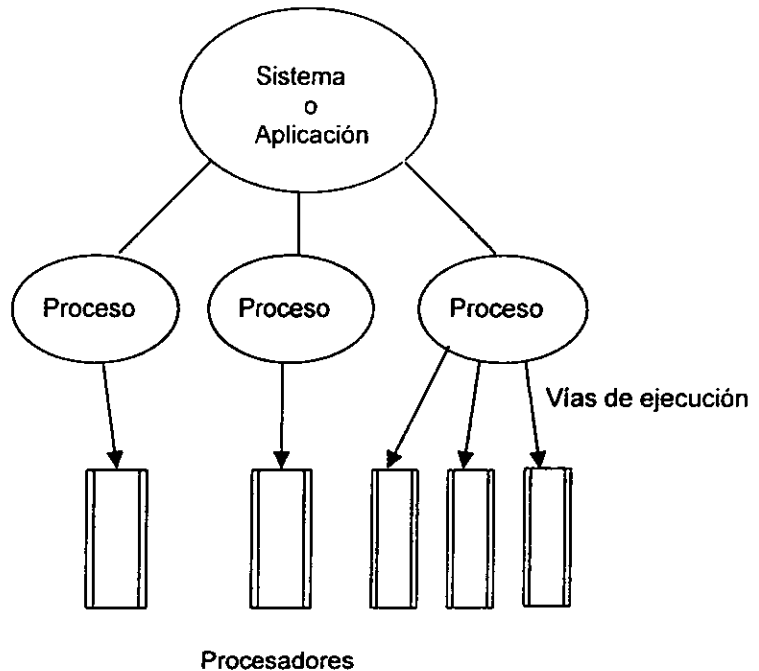
Un sistema operativo con multiprocesamiento simétrico (SMP), puede optimizar el rendimiento de los sistemas equipados con múltiples procesadores. En un sistema operativo multitarea, como es Windows NT, el trabajo se reparte entre distintos procesos, siendo asignado a cada uno de ellos una parte de la memoria, así como otros recursos del sistema. Cada proceso dispone, además, de al menos una vía para llevar a cabo su tarea. Existen múltiples ventajas al disponer del procesamiento simétrico. La más importante es, sin duda, el incremento notable en el rendimiento,

sobre todo en entornos en donde existe un servidor, y donde hay muchos usuarios que han de acceder al sistema, figura 1.4.2.3.



**Figura 1.4.2.2. Administrador Wizards de Windows NT.**

Los sistemas de escritorio pueden sacar partido también de los sistemas multiprocesador. Conforme aumenta la complejidad de las aplicaciones y se añaden nuevas características a las mismas, el rendimiento sobre los sistemas con un único procesador va viéndose cada vez más limitada. Con el soporte para el multiprocesamiento simétrico con que cuenta NT, es posible añadir procesadores en cuanto se precise un incremento en la velocidad de trabajo.



**Figura 1.4.2.3. Procesamiento simétrico.**

### **Soporte para otros entornos**

Windows NT cuenta con el entorno Win32, que es el encargado de dar soporte a las aplicaciones NT de 32 bits. Además, NT incluye algunos otros subsistemas de entorno. Estos subsistemas permiten que puedan ser ejecutadas en Windows NT aplicaciones diseñadas para otros sistemas operativos. Para acceder a los subsistemas de entorno se ha de iniciar la interfaz de comandos (Command prompt). Se cuenta con los siguientes subsistemas:

- El subsistema Virtual DOS Machine (VDM), que emula el entorno MS-DOS, para ejecutar aplicaciones diseñadas para este sistema operativo.

-El subsistema Win 16 Virtual Machine, que emula el entorno Windows(16 bits) para poder ejecutar aplicaciones Windows 3.1.

-El subsistema OS/2, permite ejecutar aplicaciones de tipo carácter escritas para MS OS/2 1.

-El subsistema POSIX , que permite ejecutar aplicaciones que se adecuan, a la norma definida por el Instituto de ingenieros eléctricos y electrónicos (IEEE).

### **Memoria virtual**

La memoria virtual proporciona un método para asignar más cantidad de memoria al sistema operativo y a las aplicaciones, de la que existe físicamente disponible en la computadora. Cuando comienza a escasear la memoria, partes de la misma pasan al disco para dejar espacio libre para otros procesos. Por ejemplo, si un determinado proceso cuenta con una información en la memoria que no está siendo utilizada en la actualidad, dicha información puede ser pasada al disco y liberar así una cantidad de espacio en la memoria que puede ser empleada para otra tarea.

### **Sistemas de archivos de NT**

Es preciso definir particiones y formatear un disco rígido para que puedan ser almacenados en el mismo. En el proceso de formateo se divide el disco en pistas y sectores, y se divide el método que se usará para almacenar y recuperar archivos y directorios.

El sistema primario de archivos utilizados en Windows NT es el nuevo NT File System (NTFS), si bien están contemplados igualmente otros más antiguos, como se describen a continuación:



**FAT (Sistema de tabla de asignación de archivos)** Es el sistema de archivos utilizados por Dos. Emplea un formato de denominación de archivos que permite un nombre base de ocho caracteres, y una extensión de un máximo de tres caracteres. Windows NT puede acceder a las unidades FAT, pero si se pone el sistema en marcha con Dos, no podrán utilizarse unidades NTFS.

**HPFS (Sistema de archivos de alto rendimiento)** Se trata del sistema de archivos diseñado para OS/2, en el cual se permiten nombres largos de archivos. Goza, además de algunas características que lo colocan en una posición ventajosa frente al sistema FAT. Windows NT puede acceder a unidades HPFS.

**NTFS.** Este es el nuevo sistema de archivos de NT. Permite el uso de nombres largos de archivos, protección y restauración de datos, así como diversos métodos de seguridad en base a permisos para archivos y directorios.

### **Ventajas del NTFS**

NTFS ha sido pensado para proporcionar la máxima velocidad cuando se trabaja en entornos de red local. Este sistema contempla el uso de discos de gran capacidad, así como el almacenamiento de archivos sobre múltiples discos (conocido como volúmenes repartidos).

NTFS cuenta con una serie de características intrínsecas de seguridad que permiten controlar el acceso y propiedad de los archivos. Los archivos de un volumen NTFS no son accesibles desde Dos ni otros sistemas operativos.

Esto forma parte del sistema de seguridad propio de Windows NT, pero tiene vigencia tan sólo cuando se utiliza NTFS.

NTFS permite nombres de archivo con una longitud de hasta 256 caracteres. Si bien no es posible para un usuario de Dos utilizar un volumen NTFS, estos pueden ser

copiados a volúmenes Dos. Cada archivo NTFS incluye un nombre de archivo que se ajusta al formato Dos estándar. Este nombre se genera de forma automática por NTFS a partir de los caracteres iniciales de su nombre largo original.

### **Comunicación con redes locales**

La transmisión de información en un entorno de red local se gestiona en base a tipos específicos de protocolos. Un protocolo es un conjunto de reglas que definen el modo en que dos o más sistemas informáticos pueden comunicarse a través de una red local. Básicamente, los protocolos vienen a determinar cuándo puede enviar datos a una computadora, cuándo puede recibir información, cómo se han de enviar los datos a través del cableado de la red y la forma de operar con aplicaciones de alto nivel, se gestionan los siguientes protocolos:

**NetBIOS (Sistema de entrada/salida Network).** Las aplicaciones NT emplean esta interfaz estándar para comunicarse con otras aplicaciones compatibles con NetBIOS. Se sitúa en la parte superior del protocolo de comunicaciones NetBEUI.

**NetBEUI (Interface de uso extensión de NetBEUI)** Este protocolo fue desarrollado inicialmente por IBM, y opera bajo la interfaz NetBIOS de Microsoft. Proporciona control de flujo y detección de errores cuando tiene lugar la conexión entre computadoras a través de cableado de red local.

**TCP/IP (Protocolo de Internet para control de transmisión).** Se trata de un protocolo, para la comunicación en red a través de largas distancias TCP/IP proporciona una forma mucho más amplia. Este protocolo se emplea frecuentemente en entornos UNIX. Los usuarios de NT pueden instalar TPC/IP si desean tener acceso a sistemas UNIX.

## El rendimiento de Windows NT

Windows NT exige bastante hardware para sus operaciones avanzadas. El sistema requiere por lo menos un CPU PENTIUM II, se recomienda por lo menos 64MB de RAM, aunque NT puede funcionar con 32 MB. Una consecuencia es la habilidad de NT de usar segmentos de 32 bits para tener acceso directo a 4 GB de RAM. Como opera en el modo protegido, Windows asigna la memoria de aplicación en segmentos pequeños según se necesite, y protege la memoria de cada aplicación contra el acceso de otras aplicaciones. Cuando no hay suficiente memoria disponible, NT usa memoria virtual, entrando y sacando el código de la aplicación de la memoria activa.



Figura 1.4.2.4. Rendimiento de Windows NT.

El acceso al disco duro puede ser un cuello de botella. Los requisitos de almacenamiento de disco son substanciales. NT requiere 30 MB de espacio libre en el disco duro para instalar los archivos de sistema, además del espacio de intercambio. En total el programa necesita 70 MB de espacio. NT apoya archivos de hasta 1.8 x 10 a la 10 GB, figura 1.4.2.4.

### **Windows NT portátil y escalable**

La mayor diferencia entre la arquitectura de Windows NT y la de sus predecesores es su módulo de diseño. Entre las más populares características de Windows NT está su portabilidad. Para lograr esta portabilidad se diseñó el sistema operativo por niveles. Para trasladar el sistema a diferentes plataformas se tuvo que refinar sólo las secciones que interactúan reciprocamente con el hardware para realizar tareas específicas. Dentro del ejecutivo de NT, el sistema operativo de Windows NT, sólo el nivel de separación de hardware ( HAL ) y porciones del núcleo y el administrador de memoria son específicas a la plataforma. El resto del sistema de Windows NT es independiente de la plataforma física. La portabilidad del sistema es sólo parte de las capacidades de la plataforma cruzada de Windows NT. No se ha prestado a la facilidad de escalar, así que muchas PCs y LAPTOPS encontrarán a Windows NT demasiado grande. No hay nada inherente en Windows NT que evite que pueda instalarse y correrse en una portátil o PAMMTOP cuaderno, pero los requisitos de memoria y de almacenamiento en el disco duro son grandes.

Los puntos más sobresalientes de los sistemas operativos de 32 bits tienen más puntos en común que diferencias. Mostramos las variables claves en la siguiente tabla 1.4.2.1.

	Windows 3.1	Windows NT	UNIX
Modelo de multitarea	Cooperativo	Prioridad	Prioridad
Capacidad de multiusuario	No	Si	Si
Núcleo	DOS, KERNEL	Ejecutivo NT	Mezclado
Sistema de archivo	FAT	FAT, HPFS, NTFS	HPFS,S5,
Apoyo de multiproc. simétrico.	No	Si	Si(NeXT)
Seguridad	No	Si (Nivel C2)	Si(nivelB1 )
Redes	No	Si (nivel,servidor)	Si nivel, serv.
Memoria máxima	32MB con mem.	4 GB	4 GB
Tamaño máx. de archivo	4 GB	17.000.000GB	2 GB
Plataforma de hardware	Intel PENTIUM.	Intel P-II sup.	Intel P-II sup.
Espacio en el disco duro	10 MB	70 MB	100 MB

**Tabla 1.4.2.1. Sistemas operativos de 32 bits.**

### **Puntos sobresalientes de Windows NT**

*Operaciones de 32 bits.* Pasa la información en bloques de 32 bits, con acceso directo a 4 GB de RAM, sin intercambio de archivos, ni penalidades de velocidad y no tiene límites en el tamaño de los datos (segmentos).

*Multitareas con prioridad y múltiples hilos de ejecución.* Como un policía de tránsito eficiente, Windows NT manipula los recursos del sistema entre y dentro de las aplicaciones. Incluso las aplicaciones de Windows 3.1 y DOS que operan dentro de una sesión virtual bajo Windows NT. Con el entorno de modo protegido, cada aplicación ejecuta un sistema individual y aislado y no puede tener acceso a más memoria de la que tiene asignada. Los múltiples hilos de ejecución le permite a una aplicación dividir las tareas en subtareas para mayor velocidad y eficiencia.

*Velocidad, con demandas serias de recursos.* Las indicaciones iniciales son que Windows NT es rápido, pero la velocidad y la potencia tiene su precio.

Windows NT requiere un mínimo de 32MB de RAM (se recomienda 64 MB); igualmente se recomienda una P-II y la instalación completa requiere 70 MB.

*Portabilidad entre plataformas.* Más eficiente que otras conversiones, Windows NT fue diseñado con niveles de funciones independientes del hardware y se comunica directamente con la plataforma de hardware solamente para algunas tareas específicas.

*Un sistema de archivos y memoria mejor.* Windows NT puede tener acceso a más memoria con más eficiencia, gracias a sus operaciones de 32 bits y el nuevo NTFS. NTFS trabaja con medios de almacenamiento grandes y añade apoyo para POSIX y servicios para la tolerancia de errores y recuperación de archivos.

*Sistema de redes integradas.* Los sistemas de redes se instalan con facilidad y su operación es virtualmente transparente.

*Seguridad incorporada.* Este es un derivado lógico a los sistemas de redes integrados. NT provee un servicio de conexión, funciones de resguardo, tolerancia de errores y recuperación.

*Manejo de la excepción de estructuras.* Esto hace la vida de los programadores más fácil al reemplazar el tedioso proceso de pruebas con estructuras más sencillas que señalan los errores y los problemas.

*Familiar.* Se parece a Windows, actúa como Windows... pero más poderoso. No se tiene que aprender una nueva forma de trabajar y (casi) todos sus programas favoritos funcionan igual que antes. Se puede personalizar la interfaz a su gusto compartir algunas aplicaciones y mantener las otras privadas.

## 1.5. Comunicaciones

### 1.5.1. Redes de Computadoras

**Introducción.** Los sistemas computacionales durante las dos primeras décadas de existencia estaban altamente centralizados, y con la combinación de las telecomunicaciones y computadoras ha sido la base para la organización actual de los sistemas computacionales.

**Origen.** La necesidad absoluta de pasar por medios electrónicos información entre computadoras y la facilidad de compartir recursos, da origen a lo que hoy en día conocemos como redes de computadoras.

**Definición.** Una red es un sistema de comunicación en las que las computadoras transfieren y comparten recursos. Las redes de computadoras se pueden clasificar por extensión geográfica en:

- **LAN** ("Local Area Network = redes de área local"). Estas redes cubren un área pequeña, como se muestra en la figura 1.5.1. Red LAN, con radio típico de no más de 1.5 km.
- **MAN** ("Metropolitan Area Network = redes de área metropolitana"). Cubren una ciudad, como se muestra en la figura 1.5.2. Red MAN.
- **WAN** ("Wire Area Network = redes de área vasta"). Son generalmente servicios internacionales como se muestra en la figura 1.5.3. Red WAN.

**Repetidores.** En las redes de computadoras las señales de comunicación se atenúan cuando pasan por algún medio de transmisión, esto debido a las características mismas del medio. Sin embargo, se puede amplificar la señal utilizando un dispositivo de amplificación llamado repetidor.

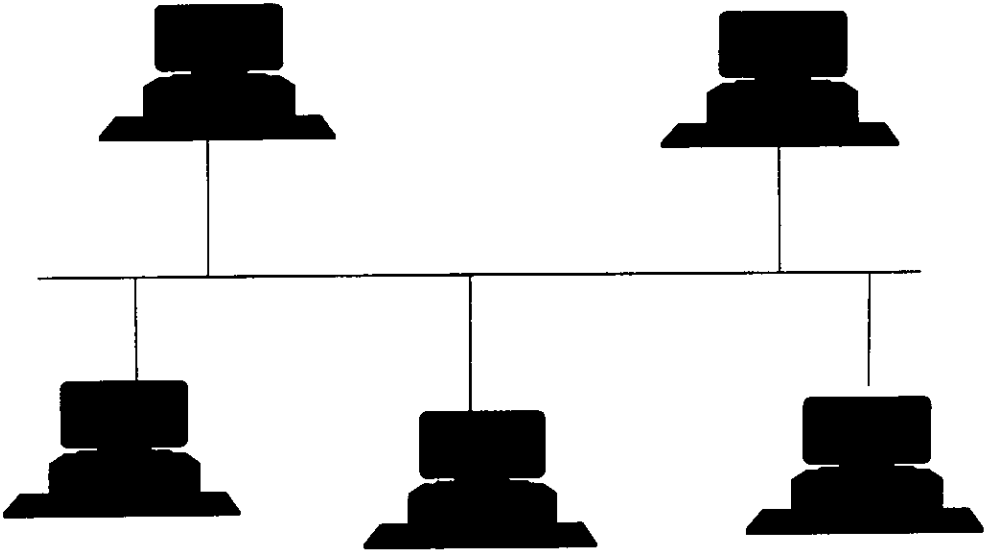


Figura 1.5.1. Red LAN.

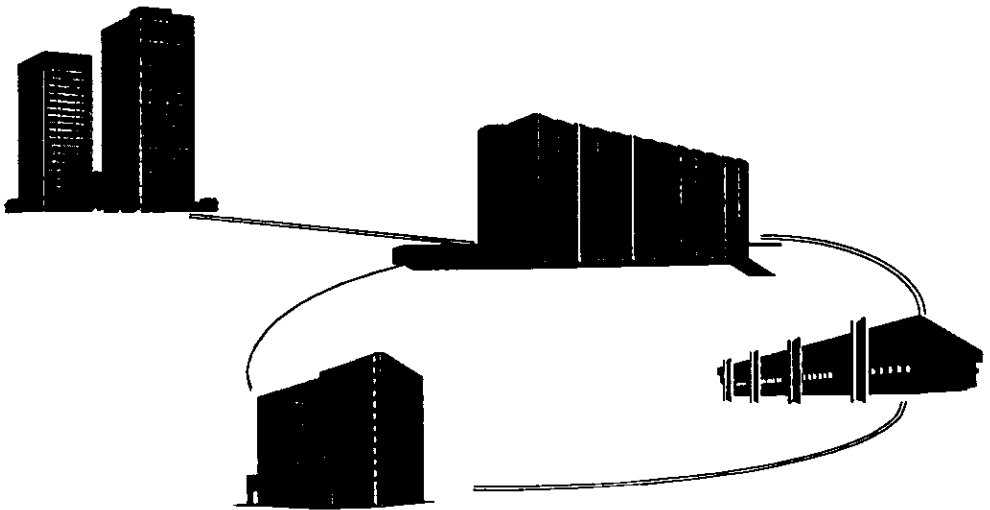
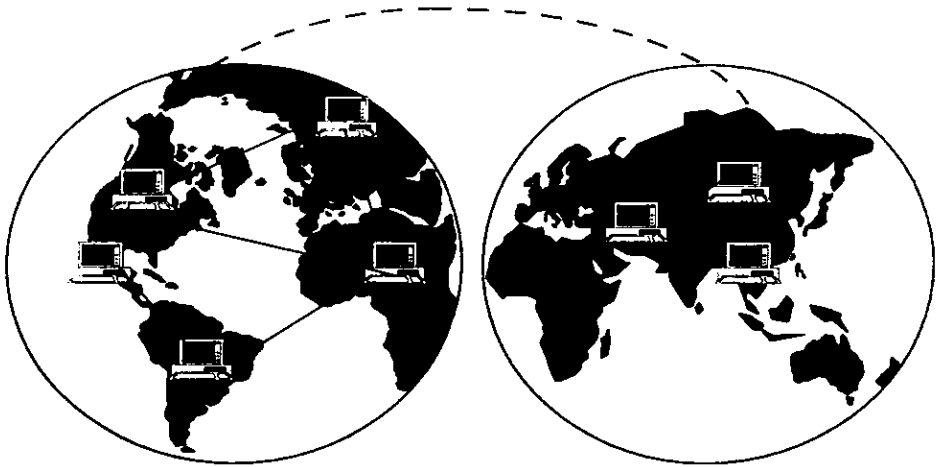


Figura 1.5.2. Red MAN.





**Figura 1.5.3. Red WAN.**

**Concentradores.** Se utilizan como un lugar central donde conectar las estaciones de trabajo y de este modo gestionar más fácilmente la red; para lograr esta gestión existen tres tipos de concentradores los cuales se describen a continuación:

**Concentrador pasivo.** Conecta entre sí varios segmentos de la red. Pero no se realiza amplificación de la señal, y cada computadora de la red recibe la información enviada por todas las demás computadoras.

**Concentrador activo.** Cumple con las funciones de un concentrado pasivo pero además en este sí se amplifica la señal.

**Concentrador inteligente.** Además de regenerar la señal, también realizan funciones como la selección de rutas de transmisión óptimas, y de esta manera contar con diferentes rutas para la entrega de información.

**Puentes.** Es un elemento de la red que permite el paso de señales entre segmentos de

red. Las funciones que realizan:

- a) Recibir todas las señales enviadas de un segmento de red.
- b) Filtrar las señales que están direccionadas a otros nodos en ese mismo segmento de red.
- c) Retransmitir todas las demás señales a través del puerto apropiado.
- d) Realizar las mismas funciones con los datos de otros segmentos conectados.

Los puentes cumplen con estas tareas mediante la determinación de la localidad física de las computadoras origen y destino en la red.

**Routers.** Los routers, o ruteadores son elementos de la red que sirven para conexión de dos o más redes separadas lógicamente (subredes).

Los ruteadores se utilizan cuando se debe mantener separada la información de distintas subredes, pero al mismo tiempo tener enlaces ocasionales entre ellas; cada subred tiene asignada una dirección de red diferente.

Los ruteadores realizan una función muy similar a la de un puente. Típicamente, los ruteadores tienen un procesamiento más intenso que los puentes. Como resultado, sus velocidades de procesamiento no son tan altas. Por otro lado, los ruteadores tienen una capacidad mucho más sofisticada de selección de rutas óptimas de transmisión.

**Brouters.** Los brouters son esencialmente ruteadores que también pueden hacer la conexión entre redes. Un brouter checa primero si puede soportar el protocolo que está siendo utilizado por la información de entrada, si el protocolo no es soportado el brouter transmite la información usando información de direccionamiento físico.

**Switch Digital.** En la actualidad para construir redes de área amplia se busca utilizar una infraestructura de conmutación de paquetes (Switches), con tecnologías como X.25, FrameRelay y actualmente con ATM ("Asynchronous Transfer Mode = Modo de

Transferencia Asíncrona”) de conmutación de celdas.

En este esquema todos los ruteadores funcionan como equipos de acceso a una red conmutada, y cada uno de los sitios se conectan al proveedor de servicio y de esta manera hacen uso de la infraestructura de telecomunicaciones para transmitir otro tipo de información.

Actualmente hay varios tipos de conmutación, que tienen diferentes grados de control de la red y por ende, un costo asociado:

- ◆ **Plataforma de Capa 2:** En esta plataforma los conmutadores toman decisiones de envío basándose en la dirección física destino contenida en cada paquete. Al contrario que el bridge (puente), el switch puede reenviar los datos con periodos de latencia muy bajos. Con esta tecnología conseguimos incrementar el ancho de banda tanto en segmentos de LAN compartidas como dedicadas, ya que segmenta la red en dominios de colisión más pequeños proporcionando mayor porcentaje de ancho de banda para cada estación, y elimina los cuellos de botella entre varias LAN.

El uso de ASIC (Circuitos Integrados Específicos de Aplicaciones) permite a los conmutadores enviar paquetes de forma simultánea a través de todos los puertos, proporcionando así un nivel de prestaciones muy superior al de los bridges tradicionales.

- ◆ **Plataforma de Capa 2/3:** El diseño de red se basa en la creación de redes LANs virtuales (VLANs), para minimizar la necesidad del routing. En esta plataforma conviven conmutadores y routers. Una red virtual no es tanto una tecnología desde el punto de vista físico, como un método o una fórmula de concepción de las redes que las hace más seguras, más fáciles de gestionar y más aptas para el suministro de aplicaciones. Las redes virtuales ocultan al usuario la complejidad de la red y permiten a su administrador centrarse en las aplicaciones y los servicios que proporciona y no en el cableado y los equipos.

♦ **Plataforma de Conmutación de capa 3:** En esta plataforma el conmutador lee la dirección de destino de la capa 3 (en redes TCP/IP la dirección de red o dirección IP). Para conseguir estas prestaciones se utilizan Circuitos Integrados Específicos de Aplicaciones o ASICs, en los que se integra la tecnología de ruteo. Junto con el control y la velocidad, las principales ventajas de la conmutación de capa 3 son su mayor rendimiento, simplicidad, seguridad, flexibilidad y su menor costo. Además se complementan con otro tipo de funcionalidades, por el momento en proceso de estandarización (VLAN, calidad de servicio con la que cuentan ya las redes de celdas).

**Arquitectura de los switches.** Los switches de nivel 3 son posibles gracias al diseño de los ASICs, cuyo desarrollo ha evolucionado ha tal grado que se pueda integrar en ellos la tecnología de enrutamiento. En la actualidad la tendencia de los fabricantes es el utilizar ASICs, como:

#### **Conmutación de Nivel 3 vs Router**

Además del control y la velocidad, las principales ventajas de la conmutación son:

- a) Rendimiento
- b) Simplicidad
- c) Seguridad
- d) Flexibilidad
- e) Costo

El mayor inconveniente de la solución de conmutación de nivel 3 es la latencia, en este caso, el switch, tardará más tiempo en determinar donde enviar cada paquete ya que ahora deben entender el protocolo que se está utilizando.

Los entornos más beneficiados serán aquellas redes complejas que en la actualidad utilizan routers para su segmentación, ya que en este caso el switch sustituye con ventaja al router por su bajo tiempo de latencia (hasta 10 veces menor en algunos casos) y con muy inferior costo por puerto.

### 1.5.2. Modelos de Interconexión

Existen diversas formas en las que podrían organizarse las redes, en cuanto a la conexión física y esto es la topología de la red.

#### Topología a Bus

- Todos los nodos de la red están conectados a un mismo medio de transmisión como se muestra en la figura 1.5.2.1. Topología de Bus.
- Sólo un par de usuarios de la red pueden comunicarse en un momento dado.
- Un paquete enviado a través del medio llega a todas las estaciones.
- Para recibir un mensaje cada estación monitorea continuamente el medio y copia sólo aquellos mensajes que le son dirigidos.

#### Ventajas de la topología bus

- Muy sencillo reconfigurar la topología al agregar y eliminar usuarios.
- Medio de transmisión muy confiable (el que una terminal no funcione, no afecta a la red).
- Permite velocidades elevadas de transmisión.

#### Desventajas de la topología bus

- Distancias limitadas.

#### Topología Anillo

- Nodos consecutivos están conectados con enlaces punto a punto formando un anillo como se muestra en la figura 1.5.2.2. Topología de Anillo.
- Los paquetes son transmitidos de nodo en nodo alrededor del anillo.
- La interfaz es un repetidor de paquetes.
- Los paquetes enviados son retirados del anillo por el transmisor, permitiendo así reconocimientos automáticos.

#### Desventajas de la topología anillo

- Dificultad para dar de alta nuevos nodos.
- La operación normal de la red se ve afectada si falla algún enlace o nodo.

### Topología a Estrella

- Nodos conectados a un equipo central de comunicaciones generalmente son concentradores como se muestra en la figura 1.5.2.3. Topología de Estrella.
- Los paquetes son transmitidos del nodo origen al nodo final por medio del concentrador.
- La interfaz en cada nodo es independiente de los nodos de la red, si falla un nodo no afecta a toda la red.

### Desventajas de la topología estrella

- Si el concentrador falla se pierde toda la red

Dependiendo del medio físico de comunicaciones se tienen limitantes en relación de las distancias de nodos.

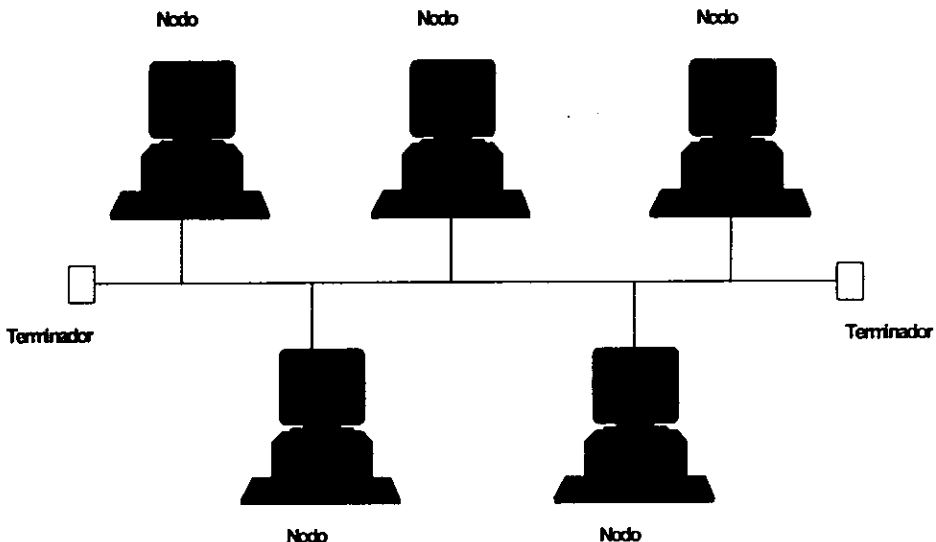


Figura 1.5.2.1. Topología de Bus.

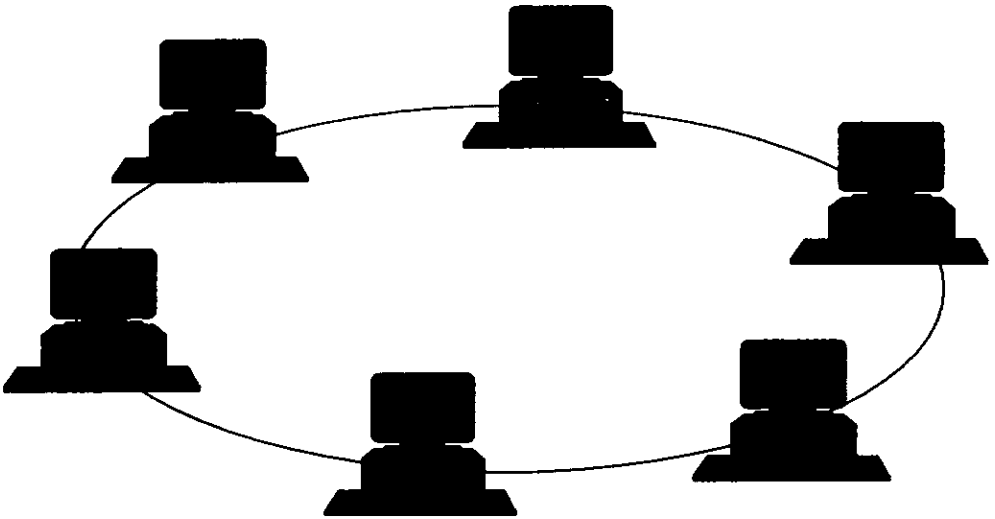


Figura 1.5.2.2. Topología de Anillo.

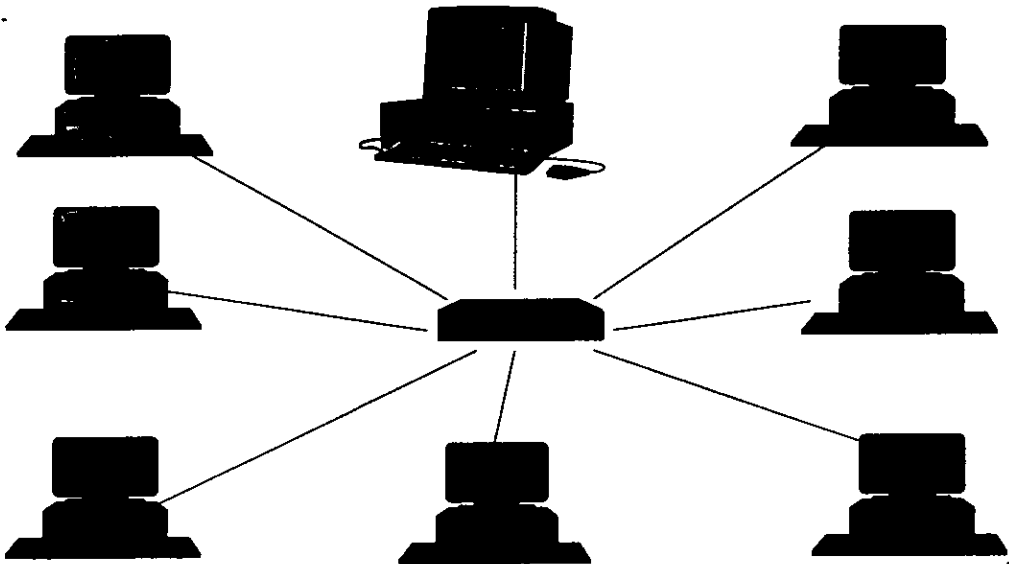
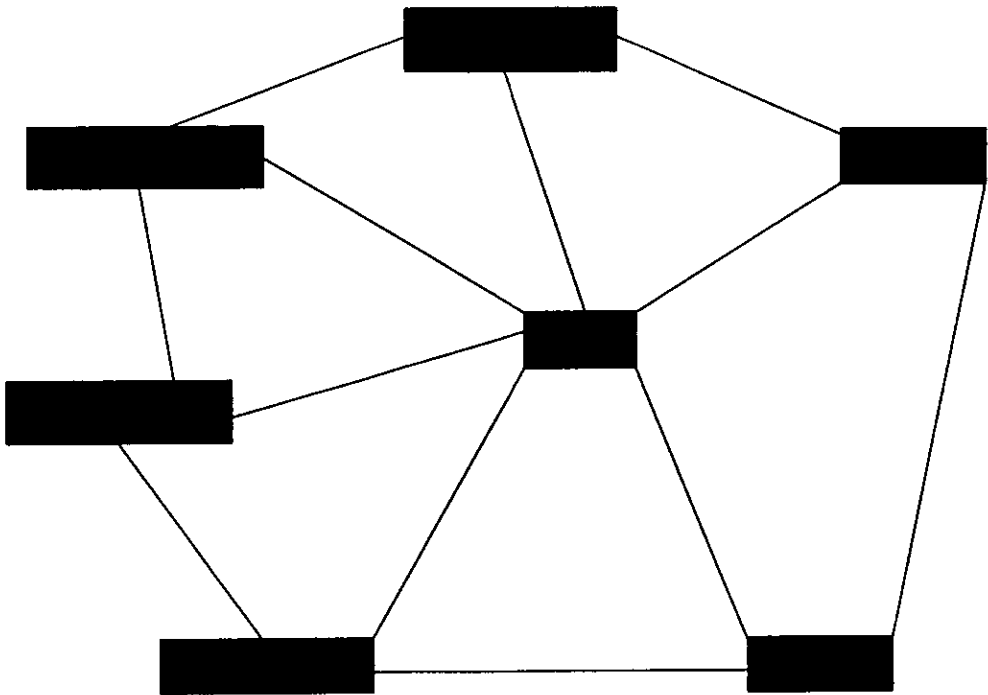


Figura 1.5.2.3. Topología de Estrella.

Una inter-red constan de LANs o de estaciones de trabajo que se interconectan con puentes y ruteadores. Para esta categoría tenemos las siguientes topología:

Red de malla. Los ruteadores se interconectan con otros ruteadores. La topología se puede configurar localmente, pero generalmente se encuentra en redes de WAN, que conectan oficinas remotas mediante enlaces de telecomunicaciones, como se muestra en la figura 1.5.2.4. Topología de Malla. Los ruteadores se utilizan para elegir la mejor trayectoria y la más eficiente de la fuente al destino a través de la malla. Los enlaces que fallan se evitan con el uso de otra trayectoria de la malla.



**Figura 1.5.2.4. Topología de Malla.**



### 1.5.3. Modelo OSI

#### Introducción

La necesidad de intercambiar información entre sistemas heterogéneos, ejemplo, entre sistemas cuyas tecnologías son muy diferentes entre sí, llevó a la ISO ("International Standard Organization = Organización de Estándares Internacionales"), a buscar la manera de regular dicho intercambio de información. El modelo de referencia OSI ("Open Systems Interconnection = Interconexión de Sistemas Abiertos"), surge en el año 1983 y es el resultado del trabajo de la ISO para la estandarización internacional de los protocolos de comunicación.

El modelo OSI consta de 7 capas o niveles. Las características generales de las capas son las siguientes:

- Cada una de las capas desempeña funciones bien definidas.
- Los servicios proporcionados por cada nivel son utilizados por el nivel superior.
- Existe una comunicación virtual entre 2 mismas capas, de manera horizontal.
- Existe una comunicación vertical entre una capa de nivel N y la capa de nivel N + 1.
- La comunicación física se lleva a cabo entre las capas de nivel 1.

Las 7 capas del modelo OSI y sus funciones principales son:

#### 1. Capa Física

- Transmisión de flujo de bits a través del medio. No existe estructura alguna.
- Maneja voltajes y pulsos eléctricos.
- Especifica cables, conectores y componentes de interfaz con el medio de transmisión.

#### 2. Capa Enlace de Datos

- Estructura el flujo de bits bajo un formato predefinido llamado trama.
- Para formar una trama, el nivel de enlace agrega una secuencia especial de bits al principio y al final del flujo inicial de bits.
- Transfiere tramas de una forma confiable libre de errores (utiliza reconocimientos y retransmisión de tramas).
- Provee control de flujo.
- Utiliza la técnica de "piggybacking".

### 3. Capa de Red (Nivel de paquetes)

- Divide los mensajes de la capa de transporte en paquetes y los ensambla al final.
- Utiliza el nivel de enlace para el envío de paquetes: un paquete es encapsulado en una trama.
- Enrutamiento de paquetes.
- Envía los paquetes de nodo a nodo usando ya sea un circuito virtual o como datagramas.
- Control de Congestión.

### 4. Capa de Transporte

- Establece conexiones punto a punto sin errores para el envío de mensajes.
- Permite multiplexar una conexión punto a punto entre diferentes procesos del usuario (puntos extremos de una conexión).
- Provee la función de difusión de mensajes (broadcast) a múltiples destinos.
- Control de Flujo.

### 5. Capa de Sesión

- Permite a usuarios en diferentes máquinas establecer una sesión.
- Una sesión puede ser usada para efectuar un login a un sistema de tiempo compartido remoto, para transferir un archivo entre 2 máquinas.

- Controla el diálogo (quién habla, cuándo, cuánto tiempo, comunicación unidireccional y bidireccional).
- Función de sincronización.

#### 6. Capa de Presentación

- Establece una sintaxis y semántica de la información transmitida.
- Se define la estructura de los datos a transmitir (define los campos de un registro: nombre, dirección, teléfono, etc.).
- Define el código a usar para representar una cadena de caracteres (ASCII, EBCDIC, etc.).
- Compresión de datos.
- Criptografía.

#### 7. Capa de Aplicación.

- Transferencia de archivos.
- Login remoto.
- Correo electrónico.
- Acceso a bases de datos, etc.

### 1.5.4. Arquitectura TC/IP

TCP/IP (*Transmission Control Protocol/internet protocol* = Protocolo de Control de Transmisión/ Protocolo Internet"), es junto con OSI una arquitectura de protocolos que ha sido determinante y básica en el desarrollo de los estándares de comunicación. Es la arquitectura más adoptada para la interconexión de sistemas. Al contrario de lo que ocurre con OSI, el modelo TCP/IP es software, es decir, es un modelo para ser implementado en cualquier tipo de red. Facilita el intercambio de información independientemente de la tecnología y el tipo de subredes a atravesar, proporcionando una comunicación transparente a través de sistemas heterogéneos. Por todo esto, TCP/IP no define una capa física ni de enlace; este protocolo define solamente tres capas que funcionarán en los niveles superiores a las capas físicas y de enlace para

hacerlo así un modelo independiente del hardware en el que se implemente.

La arquitectura de un sistema en TCP/IP tiene una serie de metas:

- La independencia de la tecnología usada en la conexión a bajo nivel y la arquitectura del ordenador
- Conectividad Universal a través de la red
- Reconocimientos de extremo a extremo
- Protocolos estandarizados

### **Estructura Interna**

El modelo básico en internet es el modelo Cliente/Servidor. El Cliente es un programa que le solicita a otro que le preste un servicio. El Servidor es el programa que proporciona este servicio.

La arquitectura de Internet esta basada en capas. Esto hace más fácil implementar nuevos protocolos. El conjunto de protocolos TCP/IP, al estar integrado plenamente en Internet, también dispone de este tipo de arquitectura. El modelo de capas de TCP/IP es algo diferente al propuesto por ISO para la interconexión de sistemas abiertos.

**Capa internet (IP):** En situaciones en las que los dos dispositivos estén conectados a redes diferentes, se necesitarán una serie de procedimientos para permitir que los datos atraviesen las diferentes redes interconectadas. Esta será la función de esta capa. El protocolo internet (IP) se utiliza en esta capa para ofrecer el servicio de encaminamiento a través de varias redes.

**Capa de origen-destino o de transporte (TCP):** Sería deseable asegurar que todos los datos llegaran a la aplicación destino y en el mismo orden en el que fueron enviados. Los mecanismos necesarios para ofrecer la seguridad son esenciales, independientemente de la naturaleza de la aplicación. El protocolo TCP es el más utilizado para proporcionar estas funciones.

**Capa de aplicación:** Contiene toda la lógica necesaria para llevar a cabo las aplicaciones de usuario. Para cada tipo específico de aplicación, como es por ejemplo la transferencia de un archivo, se necesitará un módulo particular dentro de esta capa.

### 1.5.5. Pasado y Presente

#### 1.5.5.1. Servicio en modo de conexión y en modo sin conexión

**Una red orientada a conexión** es aquella en la que inicialmente no existe una conexión lógica entre los elementos de la red. Es decir la conexión entre los elementos se encuentra en estado libre.

**Las redes no orientadas a la conexión** (también llamadas datagramas) pasan directamente del estado libre al modo de transferencia de datos, finalizado con el retorno al estado libre. La principal diferencia con la clase anterior es la ausencia de las fases de establecimiento y liberación. Además, las redes de este tipo no ofrecen confirmaciones, control de flujo ni recuperación de errores aplicables a toda la red, aunque estas funciones sí existen para cada enlace en particular. Evidentemente el costo de una red no orientada a conexión es mucho menor.

El compromiso entre una red orientada a la conexión y una red no orientada a la conexión radica en la sobrecarga frente a las funciones obtenidas. Una red orientada a conexión ofrece muchas funciones, aunque ello se traduce en un mayor costo del sistema. Por el contrario, una red no orientada a conexión resulta más económica, ya que las funciones de apoyo que ofrece al proceso de aplicación del usuario son limitadas.

#### 1.5.5.2. Direccionamiento

El Protocolo IP proporciona un sistema de distribución que es poco fiable incluso en una base sólida. El protocolo IP especifica que la unidad básica de transferencia de

datos en el TCP/IP es el datagrama. Los datagramas pueden ser retrasados, perdidos, duplicados, enviados en una secuencia incorrecta o fragmentados intencionadamente para permitir que un nodo con un buffer limitado pueda recibir todo el datagrama. Es la responsabilidad del protocolo IP reensamblar los fragmentos de datagrama en el orden correcto. En algunas situaciones de error los datagramas son descartados sin mostrar ningún mensaje mientras que en otras situaciones los mensajes de error son recibidos por la máquina origen. (esto lo hace el protocolo ICMP).

El protocolo IP también define cual será la ruta inicial por la que serán mandados los datos.

Cuando los datagramas viajan de unos equipos a otros, es posible que atraviesen diferentes tipos de redes. El tamaño máximo de estos paquetes de datos puede variar de una red a otra, dependiendo del medio físico que se emplee para su transmisión. A este tamaño máximo se le denomina MTU ("Maximum Transmission Unit = Máxima unidad de transmisión"), y ninguna red puede transmitir un paquete de tamaño mayor a esta MTU. El datagrama consiste en una cabecera y datos.

**Longitud de la Cabecera:** Este campo ocupa 4 bits, y representa el número de octetos de la cabecera dividido por cuatro, lo que hace que éste sea el número de grupos de 4 octetos en la cabecera.

**Versión:** El campo versión ocupa 4 bits. Este campo hace que diferentes versiones del protocolo IP puedan operar en la Internet.

**Tipo de servicio:** Este campo ocupa un octeto de la cabecera IP, y especifica la precedencia y la prioridad del datagrama IP. Los tres primeros bits del octeto indican la precedencia. Los valores de la precedencia pueden ser de 0 a 7. Cero es la precedencia normal, y 7 esta reservado para control de red. Muchos Gateways ignoran este campo.

Los otros 4 bits definen el campo prioridad, que tiene un rango de 0 a 15. Las cuatro

prioridades que están asignadas son: 0, (por defecto, servicio normal), 1 (minimizar el coste monetario), 2 (máxima fiabilidad), 4 (Maximizar la transferencia), 8 (El bit +4 igual a 1, define minimizar el retraso). Estos valores son utilizados por los routers para direccionar las solicitudes de los usuarios.

**Longitud Total:** Este campo se utiliza para identificar el número de octetos en el datagrama total.

**Identificación:** El valor del campo identificación es un número secuencial asignado por el host origen. El campo ocupa dos octetos. Los números oscilan entre 0 y 65,535, que cuando se combinan con la dirección del host forman un número único en la Internet. El número se usa para ayudar en el reensamblaje de los fragmentos de datagramas.

**Fragmentos Offset:** Cuando el tamaño de un datagrama excede el MTU, este se segmenta. El fragmento Offset representa el desplazamiento de este segmento desde el inicio del datagrama entero.

**Flags:** El campo flag ocupa 3 bits y contiene dos flags. El bit +5 del campo flags se utiliza para indicar el último datagrama fragmentado cuando toma valor cero. El bit +7 lo utiliza el servidor origen para evitar la fragmentación. Cuando este bit toma valor diferente de cero y la longitud de un datagrama excede el MTU, el datagrama es descartado y un mensaje de error es enviado al host de origen por medio del protocolo ICMP.

**Tiempo de Vida:** El campo tiempo de vida ocupa un octeto. Representa el número máximo de segundos que un datagrama puede existir en Internet, antes de ser descartado. Un Datagrama puede existir un máximo de 255 segundos. El número recomendado para IP es 64. El originador del datagrama manda un mensaje ICMP cuando el datagrama es descartado.

**Protocolo:** El campo protocolo se utiliza para identificar la capa de mayor nivel más cercana usando el IP. Este es un campo de 0 bits, que normalmente identifica tanto la

capa TCP (valor 6), como la capa UDP (valor 17) en el nivel de transporte, pero puede identificar hasta 255 protocolos de la capa de transporte.

**Checksum:** El checksum proporciona la seguridad de que el datagrama no ha sido dañado ni modificado. Este campo tiene una longitud de 16 bits.

El checksum incluye todos los campos de la cabecera IP, incluido él mismo, cuyo valor es cero a efectos de cálculo.

Un gateway o nodo que efectúe alguna modificación en los campos de la cabecera (por ejemplo en el tiempo de vida), debe calcular el valor del checksum antes de enviar el datagrama.

Los usuarios del IP deben proporcionar su propia integridad en los datos, ya que el checksum es solo para la cabecera.

**Dirección de Origen:** Este campo contiene un identificador de red (Netid) y un identificador de host (Hostid). El campo tiene una longitud de 32 bits. La dirección puede ser de clase A, B, C. (ver Direcciones IP).

**Dirección de Destino:** Este campo contiene el Netid y el Hostid del destino. El campo tiene una longitud de 32 bits. La dirección puede ser de clase A, B, C o D (ver Direcciones IP).

**Opciones:** La existencia de este campo viene determinada por la longitud de la cabecera. Si esta es mayor de cinco, por lo menos existe una opción.

Aunque un host no está obligado a poner opciones, puede aceptar y procesar opciones recibidas en un datagrama. El campo Opciones es de longitud variable. Cada octeto está formado por los campos Copia, Clase de Opción y Número de Opción.

- El campo Copia sirve para que cuando un datagrama va a ser fragmentado y viaja a través de nodos o Gateways: cuando tiene valor 1, las opciones son las



mismas para todos los fragmentos, pero si toma valor 0, las opciones son eliminadas.

- Clase de Opción es un campo que cuando tiene valor 0, indica datagrama o control de red; Cuando tiene valor 2, indica depuración o medida. Los valores 1 y 3 están reservados para un uso futuro.
- El Número de Opción indica una acción específica.

**Características de la Opción IP (Tabla 1.5.5.2.1.)**

Clase de Opción	Número de Opción	Ocetos	Descripción
0	0	1	Fin de alineamiento
0	1	1	Para alinear dentro de una lista de opciones
0	2	11	Seguridad (aplicaciones militares)
0	3	var	Ruteo del Origen
0	7	var	Grabar/trazar ruta
0	9	var	Ruteo estricto del Origen
2	4	var	Fecha y hora de Internet

**Tabla 1.5.5.2.1. Características de la opción IP**

**Padding** Cuando está presente el campo Pad, consiste en 1 a 3 octetos puestos a cero, si es necesario, para hacer que el número total de octetos en la cabecera sea divisible entre cuatro.

**Datos** El campo datos consiste en una cadena de octetos. Cada octeto tiene un valor entre 0 y 255. El tamaño de la cadena puede tener un mínimo y un máximo, dependiendo del medio físico. El tamaño máximo está definido por la longitud total del datagrama. El tamaño del campo Datos en octetos es igual a:

$$\text{Longitud Total del Datagrama} - \text{Longitud de la cabecera}$$

**Direcciones IP** Las direcciones IP hacen que el envío de datos entre ordenadores se

haga de forma eficaz, de un modo similar al que se utilizan los números de teléfono. Las direcciones IP tienen 32 bits, formados por cuatro campos de 8 bits separados por puntos. Cada campo puede tener un valor comprendido entre 0 y 255. Esta compuesta por una dirección de red, seguida de una dirección de subred y de una dirección de host.

Existen cinco clases de subredes, tal y como muestra el esquema.

### **Clases de Direcciones IP**

- ◆ La clase A contiene 7 bits para direcciones de red.
- ◆ La clase B contiene 14 bits para direcciones de red y 16 bits para direcciones de hosts.
- ◆ La clase C contiene 21 bits para direcciones de red y 8 para hosts.
- ◆ La clase D se reserva todas las direcciones para multidestino (multicast).
- ◆ La clase E se utiliza exclusivamente para fines experimentales.

### **Protocolos de Actualización de la Tabla de Direccionamiento**

Los protocolos que se describen a continuación se utilizan en el proceso automático de actualización de la tabla de direccionamiento.

**EGP** ("Exterior Gateway Protocol = Protocolo de compuerta exterior"): Un dominio de direccionamiento es un grupo de redireccionadores que usan un IGP ("Internal Gateway Protocol = Protocolo compuerta interior") común. Una forma de reducir el volumen de información intercambiado se basa en que un dominio de redireccionamiento utilice un Gateway seleccionado para comunicar información de direccionamiento con los Gateways seleccionados de otros dominios. El Gateway seleccionado se considera como un Gateway exterior, y el protocolo usado entre Gateways exteriores es el EGP.

El protocolo EGP se compone de tres partes:

- "Neighbor Acquisition Protocol = Protocolo de adquisición de vecino"
- "Neighbor Reachability Protocol = Protocolo de rechazo de vecino" (NR)
- "Network Reachability Determination = Determinación de rechazo de red"

El Neighbor Acquisition Protocol se utiliza simplemente para establecer comunicación. Consta de una solicitud y una respuesta.

El Neighbor Reachability Protocol se basa en un mensaje "Hello" ("Hello = Hola") (comando), y una respuesta "I heard you = Te escucho". Se utiliza para saber si la comunicación continua.

El mensaje Network Reachability se usa para comprobar si el siguiente "vecino" es un camino valido para llegar a un destino particular.

El principal inconveniente del protocolo EGP es que crea una estructura en forma de árbol, es decir que si hay problemas en Internet, los Gateways solo saben que hay problemas en el Gateway exterior.

**BGP-3** ("Border Gateway Protocol = Protocolo de compuerta de orilla"): El problema del protocolo EGP, fue el que impulso a diseñar e implementar el protocolo BGP.3

El protocolo BGP es un protocolo interno de sistema autónomo. Un sistema autónomo puede contener múltiples dominios de direccionamiento, cada uno con su propio protocolo interno de sistema autónomo, o IGP. Dentro de cada sistema autónomo pueden haber varios Gateways que se pueden comunicar con los Gateways de otros sistemas. También se puede elegir un Gateway para lograr un informe de la información de direccionamiento para el sistema autónomo. En cualquier caso, un sistema autónomo aparece ante otro sistema autónomo como un direccionador consistente. Esto elimina la estructura de árbol del protocolo EGP.

**GGP** ("Gateway-to-Gateway Protocol = Protocolo de compuerta a compuerta"): Los

primeros Gateways de internet utilizaban un IGP llamado Gateway-to-Gateway Protocol, que fue el primer IGP utilizado. Usando GGP cada Gateway manda un mensaje a todos los otros Gateways de su grupo autónomo, que contiene una tabla con las direcciones que el Gateway ha direccionado con su vector de distancia asociado.

**RIP** ("Routing Information Protocol = Protocolo de Información de enrutamiento"): El RIP es un IGP desarrollado mucho después del GGP, y está basado en el vector/distancia. Si un Gateway conoce varias rutas para llegar a un destino, asigna un coste a la ruta en función de los saltos de Gateways que deba realizar, cuantos más Gateways tenga que cruzar, más saltos deberá realizar.

Cada 30 segundos se envía un mensaje con su tabla de direccionamiento a los demás que actualizan sus tablas con los datos recibidos. (Esto produce un incremento del tráfico de red).

Este algoritmo tiene algún fallo, como por ejemplo no detecta bucles en la transmisión de la ruta. Esto daría un problema consistente en que dos rutas que se llamen entre ellas estarían emitiendo tablas de direccionamiento indefinidamente.

Otro error, es que no obliga a la autenticación de los intercambios, por lo que cualquier persona podría recibir información de las rutas enviadas por los Gateways. Existen dos versiones RIP I y RIP II (Soporta mascarar de subred).

**Hello Protocol:** Un IGP similar al RIP es el Hello Protocol. La diferencia básica es que el RIP cuenta los saltos de Gateways, y el Hello mide la distancia por el tiempo transcurrido. Este protocolo tiene un problema asociado al vector de distancia. El problema tiene dos etapas. La primera etapa es cuando los Gateways descubren una ruta más corta para llegar a un determinado destino. Esta ruta es más corta y accesible, lo que provoca que el tráfico de red pase a utilizar esta nueva ruta.

La segunda etapa empieza cuando los Gateways descubren que la nueva ruta es más lenta que la ruta vieja, debido a que al desviar el tráfico de red a la nueva ruta, esta se

satura, y todos los usuarios vuelven a la ruta vieja.

**OSPF** ("Open Shortest Path First = Protocolo de abrir primero la ruta más corta"): Uno de los protocolos IGP nuevos es el OPSF. Este protocolo ofrece un mayor grado de sofisticación con características como: Rutas basadas en el tipo de servicio, la distancia, nivel de carga, etc.

El formato del mensaje OSPF es más complejo que el RIP. Tiene una cabecera fija de 24 octetos, y una parte variable para especificar el tipo del mensaje. Existen cinco tipos de mensaje mostrados en la tabla 1.5.5.2.2.

Tipo	Significado
1	Hola (Utilizado para comprobar la accesibilidad)
2	Descripción de la Base de Datos
3	Solicitud del estado del enlace
4	Actualización del estado del enlace
5	Reconocimiento del estado del enlace

**Tabla 1.5.5.2.2 Mensajes de OSPF.**

### 1.5.5.3. Ruteo

#### Introducción

Los algoritmos de enrutamiento en IP son simples:

- Están basados en el camino más corto.
- Utilizan tablas.
- El único que debe efectuar operaciones de ruteo es el GW.
- Un host, (o multi-homed host), toma decisiones de ruteo *únicamente* para los datagramas generados por él.

**Enrutamiento directo (entrega de datagramas)**

- El emisor:
  - Encapsula el datagrama en una Trama.
  - Efectúa el mapeo de la dirección IP lógica en la dirección (ARP en el caso de ethernet).
  - Utiliza niveles inferiores para la entrega de la trama.
- Cómo sabe que se encuentra en la misma red local?
- La Entrega de Datagramas se refiere a la etapa final del enrutamiento.

**Enrutamiento indirecto**

- El encapsula el datagrama en una trama y la envía a al GW.
- Extrae el datagrama de la trama, analiza la dirección IP destino y decide enviarla al GW.
- Efectúa la entrega del datagrama mediante enrutamiento directo.

**La tabla de ruteo**

- Cada entrada es una pareja N, donde N es la dirección IP de la red destino y G es la dirección IP del siguiente GW en la ruta hacia la red N. (Una sola red entre los dos GW's.).
- El tamaño de la tabla depende del número de redes en la internet (red interconectada).
- Es independiente del número de host conectados a cada red.

Si la red destino no existe en la tabla, se utiliza un GW por default.

**Funcionamiento:**

1. IP pasa el datagrama y la "dirección IP siguiente" hacia la interfaz (nivel inferior a IP) de la red física a través de la cual el datagrama será enviado.

2. La interfaz realiza el mapeo de la dirección IP siguiente con su dirección.
3. Crea una trama usando la dirección e introduce el datagrama en la trama (campo de datos).
4. Envía la trama.
5. La dirección IP siguiente es descartada.

#### **1.5.5.4. Servicio de Nombres**

**DNS** ("Domain Name Server = Servidor de Nombre de Dominio"):

Muchos usuarios prefieren utilizar un nombre que sea más fácil de recordar que una dirección numérica. Para hacer esto, un servidor debe transformar el nombre en la dirección correcta, originalmente se hacía en Internet mediante una tabla única situada en un servidor central donde estaban contenidos todos los nombres de host, lo cual era posible debido a que solo existían unos cientos de servidores; debido a un gran aumento del número de servidores, fue necesario descentralizar el servidor de nombres y dividirlo en múltiples DNS (servidores de nombres de dominio).

Esto redujo el tiempo de respuesta del servidor, y disminuye el tráfico en la red. La estructura del sistema de dominios es similar a la estructura de directorios del DOS o del UNIX. Esto es una estructura en forma de árbol, y los archivos están identificados con una ruta de acceso. La diferencia es que en el DNS la ruta empieza con el nombre del nodo en vez del directorio raíz. Además, las rutas en un servidor DNS se escriben en sentido inverso a las del DOS.

Desde el punto de vista de un programa el funcionamiento de este servicio es muy simple. El programa proporciona un nombre de dominio, y el DNS la devuelve su dirección IP.

## **Nombres de dominio**

El programa de usuario proporciona el nombre de dominio como una secuencia de palabras. Las palabras están listadas de izquierda a derecha, y la que representa la zona más cercana al usuario es la primera.

Los programas DNS manipulan el nombre del dominio proporcionado por el usuario de manera que sea fácilmente interpretado por otros programas. Para los programas cada nombre de dominio contiene una secuencia de etiquetas, y cada etiqueta contiene un octeto de longitud seguido por una cadena de caracteres de un subconjunto de caracteres ASCII. Este subconjunto está formado por caracteres alfa (A-Z), dígitos (0-9) y un signo menos (-).

## **Arquitectura del DNS**

DNS es un protocolo de la capa de aplicación y está clasificado como una utilidad por convenio entre los usuarios y el administrador del sistema, en vez de una parte integrada en los servicios de usuario.

## **Elementos de programas de DNS**

Siguiendo el modelo Cliente/Servidor, DNS consiste en un usuario, un cliente, un servidor de nombres local y un servidor de nombres remoto. En términos de las especificaciones, DNS consiste en un programa de usuario, un cliente, un servidor de nombres, y un servidor de nombres remoto. Cada Host debe implementar un mecanismo utilizando el cliente DNS para convertir nombres de Host en direcciones IP.



## Elementos de Datos de DNS

Un nodo DNS se representa por una etiqueta en el interior del nombre de dominio, y todos los nodos tienen unos archivos de recursos (resource records (RR)) que contienen información que habilita el programa DNS para encontrar el nombre de dominio solicitado.

### Formato de un RR

- Nombre del propietario (Owner Name) o (SNAME) es el nombre del nodo al cual pertenece el Resource.
- Tipo (Type) es un entero de 16 bits que describe el tipo de Resource Record.
- Clase (Class) es un entero de 16 bits que define la clase del Resource Record. Un RR de Internet tiene el campo igual a 1.
- Tiempo de vida (Time-to-live) es un entero de 32 bits que especifica el intervalo de tiempo en el cual el RR debe ser almacenado en la memoria cache, antes de ser actualizado con la información del origen. El valor cero significa que el RR debe ser utilizado solo en la transacción en progreso, y no tiene que ser almacenado. El valor cero también se utiliza para datos muy volátiles.
- Longitud RD (RDLength) es un entero de 16 bits especifica la longitud en octetos del campo RDATA.
- RData es una cadena de longitud variable de octetos que describen el recurso. El formato de esta información varía según el tipo y clase del RR. Para el tipo A RR (Internet) , el campo RData contiene una dirección IP de 32 bits.

Otro elemento de datos del DNS es el SLIST. El SLIST es una estructura describiendo los servidores de nombres y la zona donde el cliente está intentando enviar una solicitud actualmente.

## Tipos de Resource Records

Valor	Código	Significado
1	A	La dirección de un Host
2	NS	Un servidor de nombres autorizado
5	CNAME	El nombre canónico de un alias
6	SOA	Inicio de la zona de autoridad
11	WKS	Descripción de un servicio conocido
12	PTR	Un puntero de nombre de dominio
13	HINFO	Información de un Host
14	MINFO	Información del Mailbox o de una lista de correo
15	MX	Intercambio de correo
16	TXT	Cadena de texto
22	NSAP	Cadena hacia un servicio de transporte OSI
23	NSAP-PTR	Puntero de nombre de dominio NSAP
252	AXFR	Solicitud de transferencia de un a zona entera
253	MAILB	Solicitud de los archivos del Mailbox
255	MAILA	Solicitud de todos los archivos

Tabla 1.5.5.4.1. Tipos de Registros de Recursos.

## Resource Record Format

Msb							Lsb
7	6	5	4	3	2	1	0
Owner name							
Type							
Class							
Time to live							
RDLength							
Rdata							
2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>

Tabla 1.5.5.4.2. Formato del Registro de Recursos.

## **Funcionamiento de DNS**

Un programa manda una solicitud a un cliente (resolver) que contiene un nombre de dominio para el cual se quiere la dirección IP asociada. La solicitud se suele hacer con una subrutina, o un puntero hacia el nombre de dominio en la pila del sistema. Los nombres de dominio en el cache del Resolver (cliente) están en un formato standard contenido en RRs.

Existen tres posibles respuestas de un Resolver al programa de usuario:

- Uno o mas RRs conteniendo la dirección IP solicitada. En el caso de que el nombre proporcionado fuera un alias, el Resolver simplemente devuelve el nombre de dominio al que hace referencia el alias.
- Un mensaje de error en el nombre, que significa que el nombre proporcionado no existe.
- Un error de datos no encontrado, que significa que el nombre proporcionado existe, pero no se refiere a ninguna dirección IP.

## **Formato de un mensaje DNS**

El Protocolo DNS utiliza mensajes enviados por el UDP para trasladar solicitudes y respuestas entre servidores de nombres. La transferencia de zonas completas la hace el TCP.

El formato de un mensaje DNS tiene cinco partes.

- Cabecera define el formato de las otras partes.
- Pregunta es el objetivo a resolver.
- Respuesta es la resolución del objetivo.
- Autoridad es la referencia a un servidor autorizado.
- Adicional es información relacionada, pero no la respuesta.

## Formato de la cabecera

La cabecera contiene los siguientes campos:

- ID es un campo de 16 bits utilizado para relacionar solicitudes y respuestas.
- QR es un campo de 1 bit que identifica el mensaje como una solicitud (0) o una respuesta (1).
- OPcode es un campo de 4 bits que describe el tipo de mensaje.

## Código de operación/Tipo de mensaje

Código	Descripción
0	Solicitud normal (nombre a dirección)
1	Solicitud Inversa (dirección a nombre)
2	Solicitud del estado del servidor

**Tabla 1.5.5.4.3. Códigos de operación.**

- A es un campo de 1 bit que cuando tiene valor 1 indica que la respuesta la ha hecho un servidor autorizado.
- T es un campo de 1 bit que cuando toma valor 1 indica que el mensaje ha sido truncado.
- RQ es un campo de 1 bit que cuando esta puesto a 1, indica la solicitud de un servicio recursivo por parte del servidor de nombres. Este servicio normalmente no esta disponible.
- RA es un campo de 1 bit que indica la disponibilidad del servicio recursivo.
- Z es un campo de 3 bits reservado para un uso futuro, y debe ser 0.
- RCode es un campo de 4 bits que lo rellena el servidor de nombres, y sirve para el estado de la búsqueda.
- QDCount es un campo de 16 bits que indica el número de entradas en la sección de Preguntas.
- ANCount es un campo de 16 bits que indica el número de Resource Records en

la sección de Respuesta.

- NSCount es un campo de 16 bits que define el número de Resource Records en la sección de Autoridad.
- ARCount es un campo de 16 bits que define el número de Resource Records en la sección de Archivos Adicionales.

### Estado de la búsqueda

Código	Descripción
0	Sin errores
1	Error de Imposible interpretar el formato de la búsqueda
2	Error de Imposible procesar el servidor
3	Error de nombre inexistente
4	Tipo de búsqueda no soportado
5	Solicitud rechazada

Tabla 1.5.5.4.4. Estados de la búsqueda,

### DNS Header Format

	Octet +0	Octet +1	Octet +2	Octet +3
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
+0	ID		QR Opcode	AA TR RA Z Rcode
+4	QDCOUNT		ANCOUNT	
+8	NSCOUNT		ARCOUNT	

Tabla 1.5.5.4.5. Formato del encabezado DNS.

### Formato de la Sección Preguntas

Esta sección la construye el cliente, y siempre esta presente. Contiene el nombre de dominio objetivo, seguido por los campos Qtype y Qclass. Esta sección es idéntica en

longitud y formato que la definida para los campos CName, tipo y clase de un Resource Record.

**Formato de la sección Respuesta** Esta sección contiene uno o mas RR.

**Formato de la sección Autoridad** La sección autoridad contiene uno o mas RR que apuntan hacia los orígenes de la información autorizada.

**Formato de la sección Adicional** Esta sección contiene uno o mas RR que proporcionan fuentes adicionales de información.

**Echo Protocol** El servidor eco utiliza el puerto de UDP número 7 para escuchar las solicitudes de eco del cliente. El cliente utiliza un número de puerto UDP libre para el número de puerto de origen y manda un mensaje por medio del UDP al servidor eco.

El servidor recibe la solicitud, intercambia las direcciones de origen y destino, intercambia las identificaciones de puertos y devuelve el mensaje al cliente.

#### **1.5.5.5. Suite de protocolo TCP/IP**

##### **1.5.5.5.1. Aplicaciones y utilidades**

- **Capa de Aplicación.** Esta capa corresponde a las aplicaciones que están disponibles para los usuario.

**BOOTP** ("Bootstrap Protocol = Protocolo detección de arranque")

##### **Información general**

En lugar de utilizar el protocolo ARP, una máquina que acaba de ponerse en funcionamiento por primera vez, puede utilizar el protocolo bootstrap para obtener la dirección IP e información sobre su sector de arranque. Este método tiene algunas

ventajas respecto al del protocolo ARP, por ejemplo, el BOOTP no requiere un acceso directo a la red.

### **Formato del mensaje**

Descripción de los campos:

- **Tipo (Type):** Este campo identifica si el mensaje es una solicitud o una respuesta
- **Cabecera (Header):** Este campo identifica el tipo de dirección de hardware.
- **Longitud-H (H-Length):** Este campo identifica la longitud de la dirección de hardware en octetos.
- **Contador de saltos (Hop count):** Se utiliza cuando el protocolo BOOTP se utiliza a través de varios Gateways. Cada paso por un Gateway aumenta en uno el contador.
- **ID de Transacción (transaction ID):** Lo utiliza la estación de trabajo para asignar las respuestas a las solicitudes.
- **Segundos (Seconds):** Se utiliza para calcular el tiempo transcurrido desde el envío de la solicitud hasta la recepción de la respuesta.
- **Dirección IP del Cliente (Client IP address):** Este campo lo completa el cliente, si la conoce. En otro caso se pone a cero.
- **Dirección IP del servidor (Server IP address):** Puede ser introducido por el cliente, si la conoce. Cuando el valor es diferente de cero, solo el servidor especificado puede contestar a la solicitud. Esta es una forma de forzar al servidor para que proporcione la información de arranque.
- **Dirección IP del Gateway (Gateway IP address):** Este campo lo pone a cero el cliente, y si la solicitud la obtiene un Gateway, este escribe su dirección en este campo.
- **Dirección de Hardware del cliente (Client Hardware Address):** Este campo lo completa el cliente.
- **Nombre del servidor Host (Server Host Name):** Este campo es opcional, y puede ponerlo a cero tanto el servidor como el cliente.

- Nombre del archivo de arranque (Boot File Name): Puede ponerlo a cero el cliente, o poner un nombre genérico. El servidor reemplazara este campo por la ruta completa del archivo completo.
- Area del Fabricante (Vendor-specific area): Puede tener un código escrito por el cliente.

### Formato del mensaje BOOTP

Octet +0								Octet +1								Octet +2								Octet +3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Type								Header Type								H-Length								Hop Count							
Transaction ID																															
Seconds																Zero															
Client IP Address																															
Response IP Address																															
Server IP Address																															
Gateway IP Address																															
Client Hardware Address (16 Octets)																															
Server Host Name (64 Octets)																															
Boot File Name (128 Octets)																															
Vendor-Specific Area (64 Octets)																															

**Tabla 1.5.5.1.1. Formato del mensaje BOOTP.**

**NTP** ("Network Time Protocol = Protocolo tiempo de red")

El NTP se utiliza para sincronizar los servidores en nanosegundos.

**Formato del mensaje**

El mensaje NTP esta formado por los siguientes campos:

- Indicador de Ajuste (Leap Indicator)(LI): Es un campo de 2 bits que indica el



ajuste debido al periodo de rotación de la Tierra.

Valor	Significado
00	Sin advertencias
01	-1 segundo
10	+1 segundo
11	Condición de alarma (Reloj no sincronizado)

**Tabla 1.5.5.5.1.2. Indicador de Ajuste.**

- **Número de Versión (Version Number) (VN):** Es un campo de 3 bits que indica el número de versión.
- **Reservado (Reserved):** Es un campo de 3 bits, que tienen valor cero.

**Estrato (Stratum):** Este campo tiene una longitud de 8 bits, y se utiliza para indicar el estrato local del reloj.

Valor	Significado
0	Sin especificar
1	Referencia primaria
2..n	Referencia secundaria (vía NTP)

**Tabla 1.5.5.5.1.3. Estrato.**

- **Poll:** Este campo tiene una longitud de 8 bits. Indica el intervalo máximo de tiempo entre mensajes.
- **Precisión:** Este campo tiene una longitud de 8 bits y indica la precisión del reloj local.
- **Distancia de sincronía (Sincronize distance):** Este es un campo de 32 bits, que indica el retraso aproximado de la primera ruta de sincronización.
- **Nivel de velocidad aproximado (Estimated Drift Rate):** Es un campo de 32 bits que indica el nivel de velocidad del reloj local.
- **Identificador del reloj de referencia (Reference Clock Identifier):** Campo de 32

bits que indica una reloj de referencia particular.

Valor	Código	Significado
0	DCN	Determinado por el algoritmo DCN
1	WWVB	Radio Reloj WWVB (60 KHz)
1	GOES	Reloj de satélite GOES (450 MHz)
1	Radio Reloj WWV	WWV (5/10/15 MHz)

Tabla 1.5.5.1.4. Identificador de reloj.

- Fecha y Hora (Timestamps): Existen 3 Timestamps (Fecha y hora) de 64 bits cada uno.

**SNMP** ("Simple Network Management Protocol = Protocolo simple de administración de red")

El protocolo SNMP se utiliza para administrar múltiples redes físicas de diferentes fabricantes, es decir Internet, donde no existe un protocolo común en la capa de Enlace.

La estructura de este protocolo se basa en utilizar la capa de aplicación para evitar el contacto con la capa de enlace.

**IGMP** EL IGMP ("Internet Group Management Protocol = Protocolo de administración de grupos internet") es un protocolo que funciona como una extensión del protocolo IP.

Se utiliza exclusivamente por los miembros de una red multicast para mantener su status de miembros, o para propagar información de direccionamiento.

Un Gateway multicast manda mensajes una vez por minuto como máximo. Un host receptor responde con un mensaje IGMP, que marca al host como miembro activo. Un host que no responde al mensaje se marca como inactivo en las tablas de direccionamiento de la red multicast.

#### **1.5.5.2. Protocolo de control de transferencia**

**ICMP** Internet es un sistema autónomo que no dispone de ningún control central. El protocolo ICMP ("Internet Control Message Protocol = Protocolo de Control de Mensajes Internet"), proporciona el medio para que el software de host y gateway intermedios se comuniquen.

El protocolo ICMP tiene su propio número de protocolo (número 1), que lo habilita para utilizar el IP directamente.

La implementación de ICMP es obligatoria como un subconjunto lógico del protocolo IP. Los mensajes de error de este protocolo los genera y procesa TCP/IP, y no el usuario.

#### **Formato del mensaje ICMP**

Cada Mensaje ICMP esta compuesto por los siguientes campos:

- Tipo
- Código
- Checksum
- Otras variables

#### **Solicitud de Eco**

Un host puede comprobar si otro host es operativo mandando una solicitud de echo El receptor de la solicitud la devuelve a su origen. Esta aplicación recibe el nombre de *Ping*. Esta utilidad encapsula la solicitud de eco del ICMP (tipo 8) en un datagrama IP y lo manda a la dirección IP.

El receptor de la solicitud de eco intercambia las direcciones del datagrama IP, cambia el código a 0 y lo devuelve al origen.

	Octet +0	Octet +1	Octet +2	Octet +3
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
+0	Type	Code	Checksum	
+4	Identifier		Sequence number	
	Optional Data			

**Tabla 1.5.5.2.1. ICMP Echo format.**

### Informes de Destinos Inalcanzables

Si un Gateway no puede enviar un datagrama a la dirección de destino, este manda un mensaje de error ICMP al origen. El valor del campo tipo es 3, y el tipo de error viene dado por el campo código.

Código	Descripción
0	Red no alcanzable
1	Host no alcanzable
2	Protocolo no alcanzable
3	Puerto no alcanzable
4	Necesaria fragmentación con la opción DF
5	Fallo de la ruta de origen
6	Red de Destino desconocida
7	Host de Destino desconocido
8	Fallo del Host de Origen
9	Red prohibida administrativamente
10	Host prohibido administrativamente
11	Tipo de servicio de Red no alcanzable
12	Tipo de servicio de Host no alcanzable

**Tabla 1.5.5.2.2. Códigos de Inalcanzable.**

bits que indica una reloj de referencia particular.

Valor	Código	Significado
0	DCN	Determinado por el algoritmo DCN
1	WWVB	Radio Reloj WWVB (60 KHz)
1	GOES	Reloj de satélite GOES (450 MHz)
1	Radio Reloj WWV	WWV (5/10/15 MHz)

Tabla 1.5.5.5.1.4. Identificador de reloj.

- Fecha y Hora (Timestamps): Existen 3 Timestamps (Fecha y hora) de 64 bits cada uno.

**SNMP** ("Simple Network Management Protocol = Protocolo simple de administración de red")

El protocolo SNMP se utiliza para administrar múltiples redes físicas de diferentes fabricantes, es decir Internet, donde no existe un protocolo común en la capa de Enlace.

La estructura de este protocolo se basa en utilizar la capa de aplicación para evitar el contacto con la capa de enlace.

**IGMP** EL IGMP ("Internet Group Management Protocol = Protocolo de administración de grupos internet") es un protocolo que funciona como una extensión del protocolo IP.

Se utiliza exclusivamente por los miembros de una red multicast para mantener su status de miembros, o para propagar información de direccionamiento.

Un Gateway multicast manda mensajes una vez por minuto como máximo. Un host receptor responde con un mensaje IGMP, que marca al host como miembro activo. Un host que no responde al mensaje se marca como inactivo en las tablas de direccionamiento de la red multicast.

FALTA PAGINA

No.

146

**Formato del mensaje:** El formato del mensaje de control de flujo es idéntico al mensaje de inalcanzable, excepto que el tipo es 4 y el código es 0.

**Cambio de ruta (redireccionamiento)**

Los Gateways en cualquier Internet contienen las tablas de redireccionamiento más comunes. Cuando la ruta por defecto no es la más adecuada, el Gateway puede enviar al Host un mensaje de redireccionamiento ICMP que contiene la ruta correcta.

**Formato del mensaje:** El formato del mensaje ICMP de control de flujo es igual al del mensaje de inalcanzable, excepto que el tipo es 5 y el valor del código es variable entre 1 y 3. Los motivos para la redirección y sus códigos son los de la tabla siguiente.

Código	Razón para la redirección
1	Por el host
2	Por el tipo de servicio y red
3	Por el tipo de servicio y host

Ta

**Tabla 1.5.5.5.2.5. Códigos de Redirección.**

**Tiempo de vida excedido:** Para prevenir bucles en la redirección, el datagrama IP contiene un tiempo de vida definido por el origen. A medida que cada Gateway procesa el datagrama, el valor del campo disminuye en una unidad. Posteriormente el Gateway verifica si el valor del campo es 0. Cuando se detecta un 0, el Gateway manda un mensaje de error ICMP y descarta el datagrama.

**Formato del mensaje:** El formato del mensaje de error es igual al del mensaje de inalcanzable, pero el tipo es 11, y el código es igual a 0 (contador sobrepasado), o 1 (tiempo de reensamblaje de fragmento excedido).

**Errores de parámetros** Un error de parámetros se produce cuando el que origina el datagrama, lo construye mal, o el datagrama está dañado. Si un Gateway encuentra un

- Nombre del archivo de arranque (Boot File Name): Puede ponerlo a cero el cliente, o poner un nombre genérico. El servidor reemplazara este campo por la ruta completa del archivo completo.
- Area del Fabricante (Vendor-specific area): Puede tener un código escrito por el cliente.

### Formato del mensaje BOOTP

Octet +0								Octet +1								Octet +2								Octet +3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Type								Header Type								H-Length								Hop Count							
Transaction ID																															
Seconds																Zero															
Client IP Address																															
Response IP Address																															
Server IP Address																															
Gateway IP Address																															
Client Hardware Address (16 Octets)																															
Server Host Name (64 Octets)																															
Boot File Name (128 Octets)																															
Vendor-Specific Area (64 Octets)																															

Tabla 1.5.5.1.1. Formato del mensaje BOOTP.

**NTP** ("Network Time Protocol = Protocolo tiempo de red")

El NTP se utiliza para sincronizar los servidores en nanosegundos.

Formato del mensaje

El mensaje NTP esta formado por los siguientes campos:

- Indicador de Ajuste (Leap Indicator)(LI): Es un campo de 2 bits que indica el



FALTA PAGINA

No. 148

LAN física, puede mandar una solicitud ICMP de máscara de subred.

Formato del Mensaje: El formato es igual a los primeros ocho octetos del ICMP Fecha y hora. El valor del campo tipo es 17 para la solicitud de máscara de subred y 18 para la respuesta. El código es 0, y el identificador y el número de secuencia se utilizan para identificar la respuesta

### **1.5.5.3. Protocolo de transferencia de archivos simple**

FTP ("File Transfer Protocol = Protocolo de Transferencia de Archivos") es un programa de transferencia de archivo en entornos TCP/IP. FTP es un componente fundamental en los sistemas TCP/IP y se encuentra ubicado en el nivel de aplicación del modelo OSI. su forma de trabajo se basa en el programa telnet y en TCP.

FTP se encuentra disponible para una gran variedad de sistemas y servidores como protocolo genérico de transferencia de archivos entre sistemas.

Las utilidades de FTP incorporan la siguientes:

- Soporte para archivos ASCII, EBCDIC, binarios y con formato.
- Capacidad de lectura y escritura para que los usuarios visualicen, creen y eliminen archivos y directorios.
- Protección de contraseñas.
- Compresión de datos.

Los archivos se transfieren como archivos de texto ASCII a menos que se elija un modo de transferencia diferente. Hay que escribir "binary" antes de escribir la orden para transferir un archivo binario, y escribir ascii para realizar la transferencia de un archivo de texto. Existen dos modos de transferencia utilizados: el modo de transferencia por flujo, envía las operaciones sin más operaciones; mientras que el modo de bloques realiza una verificación de errores en los bloques de datos.

#### 1.5.5.4. Protocolos TCP/IP

Aunque poca gente sabe lo que es TCP/IP todos lo emplean indirectamente y lo confunden con un solo protocolo cuando en realidad son varios, de entre los cuales destaca y es el más importante el protocolo IP.

Bajo este nombre (TCP/IP) se esconde uno de los protocolos mas usados del mundo, debido a que es el más usado por Internet y esta muy extendido en el sistema operativo **UNIX**.

Para que este protocolo pueda funcionar debe existir comunicación entre los distintos ordenadores conectados a Internet, deben usar el mismo protocolo de comunicaciones y deben fraccionar la información que se transmite en los que se insertarán las direcciones de los ordenadores origen y destino, asegurándose de que la información transmitida llegue intacta a su destino. Para ello elige las rutas más convenientes hasta el receptor y tras hacer comprobaciones de que la información original de forma de que esta quede como inicialmente se envió.

Este protocolo, en el momento de enfrentarse con un problema, el sistema que utiliza es el de dividir el problema en pequeñas porciones hasta que finalmente aborda cada uno de estos problemas hasta solucionarlo.

Algunos de los motivos de su popularidad son:

- Independencia del fabricante.
- Soporta múltiples tecnologías.
- Puede funcionar en máquinas de cualquier tamaño.
- Estándar de EEUU desde 1983

**Capa de Transporte:** Provee comunicación extremo a extremo desde un programa de aplicación a otro. Puede proveer un transporte confiable asegurándose de que los

datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente de tal manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota. En esta capa se encuentran los protocolos UDP y TCP.

**UDP** ("User Datagram Protocol = Protocolo de datagrama de usuario")

El protocolo UDP (User Datagram Protocol) proporciona aplicaciones con un tipo de servicio de datagramas orientado a transacciones. El servicio es muy parecido al protocolo IP, en el sentido de que no es fiable y no está orientado a la conexión. El UDP es simple, eficiente y ideal para aplicaciones como el TFTP y el DNS. Una dirección IP sirve para dirigir el datagrama hacia una máquina en particular, y el número de puerto de destino en la cabecera UDP se utiliza para dirigir el datagrama UDP a un proceso específico localizado en la cabecera IP. La cabecera UDP también contiene número de puerto origen, que permite al proceso recibido conocer como responder al datagrama.

**Formato del mensaje:** El datagrama UDP contiene cuatro campos, que son Número del Puerto de Origen, Número del Puerto de Destino, Longitud del mensaje y Checksum.

**Formato del mensaje UDP:** Números de Puerto de Origen y Destino. Estos números, junto con las direcciones IP definen el punto final de la comunicación. El número del puerto de origen, puede tener valor cero si no se usa. El número del puerto de destino solo tiene sentido en el contexto de un datagrama UDP y un a dirección IP en particular.

El número de puerto de origen es un campo de 16 bits. El puerto de destino tiene la misma longitud.

**Longitud del Mensaje:** Este campo tiene una longitud de 16 bits y contiene el número total de octetos que forman el datagrama, incluida la cabecera.

**Checksum:** El uso del checksum es opcional, y este campo debe ponerse a cero si no es utilizado. Mientras que es checksum del datagrama IP solo tiene en cuenta la cabecera del mensaje, el UDP tiene su propio checksum para garantizar la integridad de los datos. La longitud de este campo es de 16 bits, y esta formado por la suma de los campos del UDP, y algunos campos del IP. Para incluir los campos del IP, se construye una pseudo cabecera UDP. Esta pseudo cabecera de 12 octetos se utiliza únicamente a efectos de calcular la suma.

**TCP** El protocolo TCP proporciona un servicio de comunicación que forma un circuito. TCP proporciona un circuito virtual el cual es llamado una conexión.

Al contrario de los programas que utilizan UDP, los que utilizan el TCP tienen un servicio de conexión entre los programas llamados y los que llaman, un chequeo de errores, control de flujo y capacidad de interrupción.

**Interfaces TCP:** Existen dos tipos de interfaces entre la conexión TCP y los otros programas.

El primero es utilizar la pila de los programas de la capa de red. Como en esta capa solo esta el protocolo IP, la interface lo determina este protocolo. El segundo tipo es la interfaz del programa de usuario. Esta interfaz puede variar según el sistema operativo, pero en general tiene las siguientes características.

La interfaz envuelve el programa de usuario llamando a una rutina que introduce entradas en una estructura de datos llamada el bloque de control de transmisión (TCB). Las entradas se realizan inicialmente en la pila de hardware y transferidas al TCB por medio de una rutina de sistema. Estas entradas permiten al TCP asociar un usuario con una conexión particular, de modo que pueda aceptar comandos de un usuario y mandarlos a otro usuario en la otra parte de la conexión. TCP utiliza unos identificadores únicos para cada parte de la conexión. Esto se utiliza para recordar la asociación entre dos usuarios. Al usuario se le asigna un nombre de conexión para

utilizarlo en futuras entradas del TCB. Los identificadores para cada extremo de la conexión se llaman sockets. El socket local se construye concatenando la dirección IP de origen y el número de puerto de origen. El socket remoto se obtiene concatenando la dirección IP de destino y el número de puerto de destino.

El par de sockets de una conexión forman un único número en Internet. El UDP tiene los mismos sockets, pero no los recuerda. Esta es la diferencia entre un protocolo orientado a conexión y otro a no conexión. A continuación se explican los comandos más usuales:

- **Open:** Inicia una conexión o comienza a escuchar un socket. El usuario tiene un nombre de conexión local que actúa como un puntero dentro del TCB.
- **Send:** El comando Send manda datos del buffer especificado.
- **Recibir:** El comando recibir es un mensaje de error si el nombre local proporcionado no es utilizado antes con el comando Open.
- **Close:** El comando Close hace que se cierre una conexión. Se produce un error si la conexión especificada no ha sido abierta, o si no se tiene autorización para cerrar la conexión.
- **Status:** El comando Status solo tiene una variable asociada, que es el nombre de la conexión.
- **Abort:** El comando Abort hace que todos los comandos Send y Receive asociados al nombre de la conexión local se interrumpan. La entrada del usuario del TCB se elimina y se envía un mensaje especial de reinicio a la entidad del otro lado de la conexión.

El TCP recuerda el estado de cada conexión por medio del TCB. Cuando se abre una conexión, se efectúa una entrada única en el TCB. Un nombre de conexión se le asigna al usuario para activar los comandos de la conexión. Cuando se cierra una conexión se elimina su entrada del TCB.

**Control de Flujo:** El protocolo TCP puede controlar la cantidad de datos que debe

enviar mediante el campo "Window = Ventana". Este campo indica el número máximo de octetos que pueden ser recibidos. El receptor de un segmento con el campo window a cero, no puede enviar mensajes al emisor, excepto mensajes de prueba. Un mensaje de prueba es un mensaje de un solo octeto que se utiliza para detectar redes o host inalcanzables.

**Formato del segmento TCP:** El segmento TCP consiste en una cabecera y datos. A continuación se describen los campos del segmento TCP.

- **Número de puerto del Origen/destino (Source/Destination Port Numbers):** Este campo tiene una longitud de 16 bits.
- **Números de Secuencia (Secuence Numbers):** Existen dos números de secuencia en la cabecera TCP. El primer número de secuencia es el número de secuencia final (SSN). El SSN es un número de 32 bits. El otro número de secuencia es el Número de secuencia esperado de recepción, (También llamado Número de Reconocimiento (acknowledgement number).
- **Longitud de la cabecera (Header Length):** Este campo tiene una longitud de 4 bits y contiene un entero igual al número de octetos que forman la cabecera TCP dividido por cuatro.
- **Código de Bits (Code bits):** El motivo y contenido del segmento TCP lo indica este campo. Este campo tiene una longitud de 6 bits.
- **Window:** Este campo contiene un entero de 32 bits. Se utiliza para indicar el tamaño de buffer disponible que tiene el emisor para recibir datos.
- **Opciones (Options):** Este campo permite que una aplicación negocie durante la configuración de la conexión características como el tamaño máximo del segmento TCP. Si este campo tiene el primer octeto a cero, esto indica que no hay opciones.
- **Relleno (Padding):** Este campo consiste en un número de octetos (De uno a tres), que tienen valor cero y sirven para que la longitud de la cabecera sea divisible por cuatro.
- **Checksum:** Mientras que el protocolo IP no tiene ningún mecanismo para

garantizar la integridad de los datos, ya que solo comprueba la cabecera del mensaje. El TCP dispone de su propio método para garantizar dicha integridad. Como en el Checksum del protocolo TCP también se incluyen campos del protocolo IP, es necesario construir una pseudo-cabecera TCP que se considera únicamente a efectos de cálculo.

**Estados del TCP:** El inicio, mantenimiento y cierre de una conexión requiere que el TCP recuerde toda la información relativa a cada conexión. Esta información se almacena en una entrada para cada conexión dentro del TCB. Cuando se abre una conexión, la entrada en el TCB se realiza con todas las variables inicializadas con sus respectivos valores. Durante la conexión, la entrada del TCB es actualizada a medida que cambia la información.

- 0. CLOSED: No existe, solo para referencia.
- 1. LISTEN: Esperando solicitud de conexión de un TCP remoto.
- 2. SYN-SEN: Esperando un mensaje de solicitud de conexión después de haber enviado una solicitud de conexión.
- 3. SYN-RECEIVED: Esperando confirmación de una reconocimiento de solicitud de conexión, después de haber enviado y recibido una solicitud de conexión.
- 4. ESTABLISHED: Representa una conexión abierta. Los datos recibidos pueden ser enviados a un protocolo de una capa superior. Este es el estado normal de la fase de transferencia de la conexión.
- 5. FIN-WAIT-1: Esperando la solicitud de fin de conexión de un TCP remoto, o un reconocimiento de una solicitud de fin de transmisión enviada anteriormente.
- 6. FIN-WAIT-2: Esperando una solicitud de fin de conexión de un TCP remoto.
- 7. CLOSE-WAIT: Esperando una solicitud de fin de conexión de un protocolo de una capa superior.
- 8. CLOSING: Esperando el conocimiento de una solicitud de final de conexión de un TCP remoto.
- 9. LAST-ACK: Esperando el conocimiento de una solicitud de final de conexión enviada anteriormente al TCP remoto.



- 10. TIME-WAIT: Esperando el tiempo necesario para que el TCP remoto haya recibido el conocimiento de la solicitud del fin de conexión.

#### 1.5.5.5. Protocolos de Acceso de Red

**ARP** El protocolo ARP ("Address Resolution Protocol = Protocolo de resolución de dirección"), es el encargado de convertir las direcciones IP en direcciones de la red física.

El funcionamiento del protocolo ARP es bastante simple. Cuando una máquina desea enviar un mensaje a otra máquina que está conectada a través de una red ethernet se encuentra con un problema: la dirección IP de la máquina en cuestión es diferente a la dirección física de la misma. La máquina que quiere enviar el mensaje sólo conoce la dirección IP del destino, por lo que tendrá que encontrar un modo de traducir la dirección IP a la dirección física. Esto se hace con el protocolo ARP.

Este protocolo utiliza una tabla denominada Tabla de Direcciones ARP, que contiene la correspondencia entre direcciones IP y direcciones físicas utilizadas recientemente. Si la dirección buscada no está en la tabla el protocolo ARP envía un mensaje a toda la red. Cuando un ordenador reconoce su dirección IP envía un mensaje de respuesta que contiene la dirección física. Cuando la máquina origen recibe este mensaje ya puede establecer la comunicación con la máquina destino, y esta dirección física se guarda en la Tabla de direcciones ARP.

**Formato del mensaje ARP:** El mensaje ARP está formado por 28 octetos. En los campos que se describen a continuación se supone una Interfaz Ethernet.

**Tipo de Hardware:** El campo Hardware indica el tipo de interfaz de Hardware. Por ejemplo, el valor de una red Ethernet es 1.

Tipo	Descripción
1	Ethernet (10mb)
2	Experimental Ethernet (3 mb)
3	Amateur Radio X.25
4	Proteon ProNET Token Ring
5	Chaos
6	IEEE 802 Network
7	ARCNET
8	Hyperchannel
9	Lanstar
10	Autonet Short Address
11	LocalTalk
12	LocalNet

**Tabla 1.5.5.5.1. Tipos de Hardware.**

**Números de Protocolo:** El campo protocolo identifica el protocolo Ether usado. Por ejemplo el valor del interfaz Ethernet es 0800 hex.

**Longitud de la dirección Hardware:** El valor para Ethernet es 6, lo que proporciona 48 bits para una dirección Ethernet (12 semi-octetos)

**Longitud del Protocolo:** Este campo se usa para definir la longitud de la dirección de red. Para una red IP es 4.

**Operación:** Especifica el código de la operación. La solicitud ARP tiene valor 1, y la respuesta ARP tiene valor 2.

**Dirección Hardware del Origen:** Los campos Dirección Hardware del Origen, Dirección IP del Origen, y Dirección IP del Destino los completa el emisor (si los conoce). El receptor añade la Dirección Hardware del Destino y devuelve el mensaje al emisor con el código de operación 2. (El código de la Respuesta ARP).

La dirección Hardware de Origen (para Ethernet) esta formada por octetos que

representan una dirección Ethernet de 48 bits, o un número.

**Dirección IP de Origen:** La dirección IP de Origen puede ser una dirección de clase A, B o C. (Ver Direcciones IP para obtener una definición de estas clases).

**Dirección Hardware de Destino:** Este campo esta formado igual que el campo Dirección Hardware de Origen.

**Dirección IP de Destino:** Este campo es igual que el campo Dirección IP de Origen.

**RARP:** El protocolo RARP ("Reverse Address Resolution Protocol = Protodolo de resolución de dirección contraria") es el encargado de asignar una dirección IP a una dirección física.

**Formato del Mensaje RARP:** El formato del RARP es similar al del ARP. El valor del código de operación para una solicitud es 3, y el valor para una respuesta es 4.

### **Técnicas de acceso al medio**

Las técnicas de acceso al medio se clasifican en determinísticas como polling (poleo), y estocásticas o probabilísticas.

#### **Técnicas Determinísticas**

- *Polling Centralizado*
  - Una estación de control determina el orden en el que los nodos tendrán

acceso a la red.

- La estación de control secuencialmente pregunta a los diferentes nodos conectados en el bus, si tienen información a enviar.
- El tiempo que un nodo posee el canal puede estar determinado por un tamaño máximo de paquete o por un intervalo de tiempo asignado.
- *Polling Distribuido*
  - El método *token-passing* es un ejemplo de polling distribuido.
  - Un token es pasado de estación en estación.
  - Cuando una estación desea transmitir, remueve el token de la línea, adquiriendo acceso exclusivo a la red.
  - Al finalizar la transmisión, el token es regresado a la línea para que otras estaciones puedan obtener el acceso.

### **Técnicas estocásticas**

- Un ejemplo típico es la técnica CSMA/CD (*"Carrier Sense Multiple Access with Collision Detection = Acceso múltiple de percepción de portadora con detección de colisión"*).
- Cualquier nodo escucha el canal y transmite si éste está libre.
- Durante la transmisión, el nodo sigue escuchando el canal para verificar que otro nodo no inició una transmisión simultáneamente, en cuyo caso, la señal es alterada (colisión de mensajes).
- En caso de colisión cada nodo deja de transmitir, espera un tiempo aleatorio al final del cual intenta transmitir nuevamente.

## 1.6. Modelos de bases de datos

La información fue evolucionando y las necesidades de los usuarios fueron cada vez mayores. A los tratamientos masivos de archivos en los centros de cálculo sucedieron aplicaciones explotadas en tiempo real desde terminales conectadas al ordenador central a través de diferentes medios de comunicación.

Los tiempos de respuesta exigidos y el acceso concurrente de varios usuarios a un mismo archivo para su actualización, hicieron que las estructuras de los archivos fuesen insuficientes. Cuestiones como la integridad de los archivos, sistemas de seguridad de tensión o avería del ordenador, etc.

Empiezan a preocupar y la imposibilidad de cada programador se ocupe de estos temas en cada programa hace que aparezcan los primeros sistemas de gestión de bases de datos.

En este caso, el sistema se ocupa de controlar los accesos a la base de datos, de gestionar la posible concurrencia de varios usuarios a la misma, de realizar las correspondientes operaciones para la seguridad e integridad de los datos y, en resumen, una serie de trabajos que resultan transparentes al programador y que resuelven las necesidades planteadas por los usuarios.

Los diferentes tipos de sistemas de gestión de bases de datos, son tres que tenemos claramente diferenciados surgidos a lo, largo del tiempo:

-Jerárquico.

-En red.

-Relacional.

### 1.6.1. Bases de datos jerárquicas

En ellas los datos están estructurados en forma arborescente y las relaciones entre los diferentes tipos de registros se resuelven mediante punteros o enlaces entre ellos. Se establece una jerarquía de modo que las relaciones entre un registro y otro relaciona existe el registro "padre" asociado a él, figura 1.6.1.1.

La estructura sería del modo:

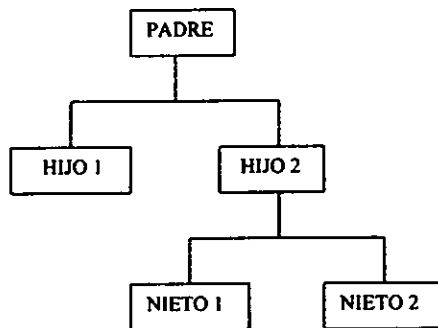


Figura 1.6.1.1. Estructura jerárquica.

La navegación a través de la estructura se resuelve mediante punteros que están incluidos en los registros de datos. Al conjunto formado por un "padre" y los "hijos" o registros asociados a él se denomina SET, existiendo en este SET un propietario (OWNER) que es el "padre" y unos miembros (MEMBER) que corresponden a los "hijos". En esta estructura de anillo se pueden establecer múltiples tipos de punteros:

-A1 NEXT: Une a un registro en el SET. El último MEMBER del SET señalaría al OWNER, cerrando el anillo.

-A1 PRIOR: Une a un registro con el anterior en el SET. El primer elemento del SET estaría ligado al OWNER por este tipo de puntero, cerrando igual que antes el anillo.

-A1 OWNER: Enlazaría cada registro con el propietario del SET, es decir, en todos los registros miembros del SET habría un puntero que señalaría al OWNER.

La estructura de la base de datos se define en el ESQUEMA. En éste hay una parte destinada a la definición de los registros y su modo de acceso y otra a la definición de los diferentes enlaces entre registros o SET. En cada SET. A su vez, hay que definir los tipos de punteros entre registros.

En cuanto a los modos de acceso a los registros, éste puede ser:

-Vía SET, es decir, siguiendo los enlaces entre los elementos del SET tal y como hemos visto.

-Mediante un ALGORITMO aplicado al campo clave del registro. La inclusión de sinónimos que teóricamente se deberían almacenar en la misma dirección según el algoritmo, se resuelve mediante el almacenamiento de éstos en una zona común de overflow.

-De forma DIRECTA en una dirección de la base de datos si se tiene la certeza de que la clave del registro nos indica esa dirección y no hay posibilidad de sinónimos,

Por medio de INDICES que nos permiten acceso a los registros de datos mediante el seguimiento de los mismos. Este sistema tiene el inconveniente de su progresiva degradación en base a las actualizaciones que se van haciendo a lo largo del tiempo. Esto puede requerir la reestructuración de los índices cada cierto tiempo.

Existe la posibilidad de tener vistas lógicas diferentes para los distintos usuarios que contemplan parte de la base de datos. Son los llamados SUBESQUEMAS.

Como conclusión, podemos decir que las características principales de este tipo de base de datos son su arborescente, la inclusión de los punteros en los registros de datos y la posibilidad de tener diferentes vistas o subesquemas de la base de datos.

#### VENTAJAS

Se tiene una estructura arborescente.

Inclusión de punteros en los registros.

Tiene la posibilidad de tener diferentes vistas o subesquemas.

#### DESVANTAJAS

Degradación progresiva del sistema.

Dependencia de un registro con otro.

Tiempo de respuesta alto.

Bases de datos rígidas.

### 1.6.2. Base de datos de red

Si en la estructura arborescente anterior, permitimos relaciones entre "hermanos", es decir, entre registros de un mismo padre lógico, tendremos una base de datos en red, figura 1.6.2.1.

En ellas se pueden construir esquemas del tipo siguiente:

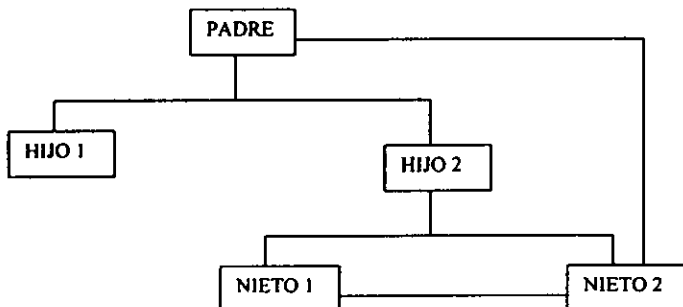


Figura 1.6.2.1. Estructura en red.



Debe considerarse a la base de datos jerárquico como un tipo de base de datos en red con ciertas restricciones.

Por tanto, es aplicable a este tipo de bases de datos todo lo dicho para las bases de datos jerárquicos en cuanto a tipos de enlaces, accesos, etc.

#### **VENTAJAS**

Estructura arborecente.

Relaciones entre registros del mismo padre.

Cuenta con subesquemmas.

#### **DESVENTAJAS**

Tiempo de respuesta alto.

Degradación del sistema.

Tiene restricción en los archivos.

### **1.6.3. El modelo relacional**

El modelo relacional por medio de una serie de reglas cuyo objetivo es lograr la independencia de la representación lógica de los datos de su almacenamiento físico.

Esta independencia física/lógica se refiere a tres aspectos:

-Independencia de la ordenación, es decir que el resultado obtenido en un acceso no dependa de cómo estén ordenadas físicamente.

-Independencia de la indexación, separando los índices de los datos haciendo que la creación y mantenimiento sean manejados por el sistema.

-Independencia de los caminos de acceso, haciendo que la navegación a través de los datos no tenga que estar previamente establecida consiguiendo así unas formas de acceso más flexibles.

Por ello, a través de las reglas pretende los siguientes objetivos:

- Independencia física/lógica.
- Eliminación de redundancia.
- Flexibilidad.
- Uniformidad.
- Sencillez.
- Sólido fundamento teórico.

### Conceptos fundamentales.

El modelo relacional se distingue por la reducción de los datos a estructuras planas o TABLAS con las filas y columnas. A estas tablas se les denomina RELACIONES y equivalen cada una de ellas a lo que en terminología tradicional se conocía como archivo.

Hay una correspondencia entre los términos utilizados al hablar de archivos, tablas y relaciones. Así, si un archivo corresponde a una tabla, cada registro del archivo se corresponde con una fila y cada campo dentro del registro con una columna, en el siguiente cuadro lo vemos con claridad, figura 1.6.3.1.

ARCHIVO	TABLA	RELACIÓN
Registro	Fila	Tupla
Campo	Columna	Atributo
No. de registros	No. de filas	Cardinalidad
No. de campos	No. de columnas	Grado

**Figura 1.6.3.1. Términos utilizados al hablar de archivos, tablas y relaciones.**

En terminología relacional cada fila de la tabla se conoce por TUPLA y cada columna por ATRIBUTO.

CARDINALIDAD de una relación será el número de tuplas que la componen y GRADO el número de atributos o columnas de la misma.

Las relaciones son tablas con las siguientes propiedades:

- No hay tuplas iguales.
- El orden de las tuplas no es significativo.
- El orden de los atributos no es significativo.
- En una tupla determinada, cada atributo puede tomar sólo un valor (no se admiten grupos repetitivos).

Dentro de los diferentes atributos de una relación podemos distinguir la llamada CANDIDATA como atributo o conjunto de atributos que identifican unívoca y mínimamente cada tupla.

Univocamente ya que no puede haber dos tuplas con la misma clave y mínimamente porque no se puede prescindir de ninguno de los atributos que forman la clave sin que éste deje de ser un identificador único. Dentro de una relación habrá una clave candidata que será la CLAVE PRIMARIA y el resto serán CLAVES ALTERNATIVAS ó FORÁNEAS.

VALOR NULO referido al contenido de un determinado atributo en una relación, será un símbolo convencional que se usará para representar una información desconocida o inaplicable.

Hay que diferenciar el contenido a blancos en un atributo de tipo alfabético o a 0 en un numérico de lo que sería el valor nulo en estos atributos. No es lo mismo que un

EMPLEADO no tenga hijos(NUM\_HIJOS=0) que ignoremos si tiene o no tiene (NUM\_HIJOS = VALOR NULO).

Una propiedad de las relaciones es la INTEGRIDAD DE ENTIDAD que supone que ningún atributo que forme parte de la clave primaria puede tomar un valor nulo, ya que el nulo podría ser cualquier valor y por tanto la tupla no queda identificada por la clave.

CLAVE FORÁNEA, es un atributo o conjunto de atributos de una relación que a su vez son clave primaria en otra relación. Veamos el siguiente ejemplo, figura 1.6.3.2.

N. EMPLEADO	NOMBRE	PROV. NAC.

Relación R1

PROV.NAC.	NOMBRE

Relación R2

Figura 1.6.3.2. Clave foránea.

## Relaciones

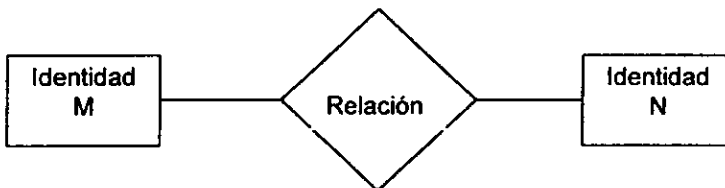
Constituyen el enlace entre las diferentes entidades del sistema. Se representa mediante líneas que unen las entidades o a través de un rombo en cuyo interior se pone el nombre de la relación, figura 1.6.3.3.

Podemos clasificar las relaciones según el número de entidades a las que asocia. Así tendremos:

**Relación binaria:** Entre dos entidades, por ejemplo, la relación entre un producto y proveedor.

**Relación N-aria:** Entre n entidades, por ejemplo, tendremos una relación terciaria entre las entidades conductor, coche y multa.

**Relación reflexiva:** La definida entre una entidad y ella misma.



**Figura 1.6.3.3. Relaciones.**

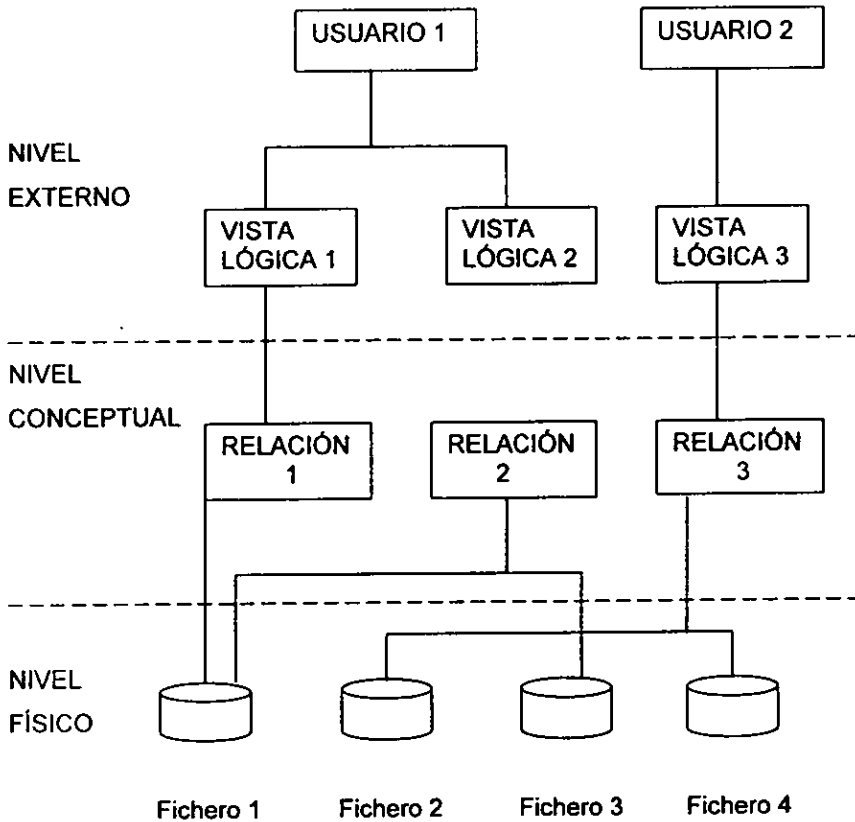
#### VENTAJAS

Flexibilidad e independencia de los datos.  
Reducción de datos a estructuras planas.  
Sólido fundamento teórico.  
Sencillez en los datos.

#### DESVENTAJAS

Rendimiento y tiempos de respuesta aceptables.

Gráficamente podemos identificar tres niveles en la arquitectura del modelo relacional, como se muestra en la figura 1.6.3.4 siguiente:



**Figura1.6.3.4. Arquitectura del modelo relacional.**

De esta forma se separa el nivel externo con las diferentes vistas lógicas que maneja, el nivel conceptual o estructura de relaciones con atributos, claves, etc. y por último el físico en el que las relaciones son almacenadas y ligadas por diferentes punteros e índices utilizados posteriormente para la localización de la información.

Los sistemas relacionales han mejorado mucho, con buenos rendimientos y tiempos de respuesta, aportan una flexibilidad y una independencia de almacenamiento físico que desbancan a las rígidas bases de datos jerárquicas.

## 1.7. Teoría de bases de datos relacionales

Una base de datos, es un conjunto de información electrónica que se almacena en una computadora con el propósito de ser explotada para atender las necesidades que se tengan dentro de un sistema; se pretende que en una base de datos, exista toda la información que es necesaria para el funcionamiento del sistema, por lo que debe tener características que faciliten el almacenamiento y la consulta de dicha información que en ella se guarda.

Las bases de datos relacionales presentan los datos como un conjunto de tablas con nombre único, las cuales se relacionan entre sí a través de un concepto de entidad-relación con el cual se entablan las ligas que pueden existir entre los datos de una tabla con los de otra.

Una tabla se puede entender como un conjunto de filas y columnas, la primera fila contiene la cabecera o título de las columnas o campos, puede tener cero o más filas adicionales, en las que se almacenarán los datos. Los datos en cada fila, conservan las mismas características por columna, es decir que todos los datos en una columna son del mismo tipo; el número de columnas en la tabla es fijo, no así el número de filas el cual es diferente en cada momento.

En la Figura 1.7.1., se tiene un ejemplo de una tabla, donde se indican las filas y las columnas, así como el tipo de datos que se almacenan en cada columna, para observar que estos conservan el mismo tipo.

El modelo relacional trata a los datos desde tres puntos de vista: su estructura, su integridad y su manipulación.



S#	SNOMBRE	SITUACIÓN	CIUDAD
S1	Salazar	20	Londres
S2	James	10	París
S3	Bernal	30	París
S4	Corona	20	Londres
S5	Aldana	30	Atenas
S6	Cortés	15	Inglaterra
S7	Carmona	40	Grecia

**Figura 1.7.1. Ejemplo de una tabla de datos.**

### **1.7.1. Elementos del modelo de bases de datos**

En la estructura de las tablas se tienen diferentes elementos los cuales se muestran en la Figura 1.7.1.1

- **Relación:** Una relación, en términos generales es una tabla, que es la base principal en las bases de datos relacionales.
- **Tupla:** Corresponde a una fila de datos en la tabla, como se indicó, el número de Tuplas es variable según la cantidad de datos.
- **Atributo:** Cada columna en una relación toma el nombre de atributo.
- **Cardinalidad:** Es el número de tuplas que tiene la relación.
- **Grado:** Es el número de atributos que tiene la relación.
- **Clave Primaria:** Es un identificador único de una tupla, es decir, el atributo o

combinación de ellos, que tienen la propiedad de no existir dos tuplas con el mismo valor en ese atributo o combinación de ellos.

- *Dominio*: Es el conjunto de valores que puede tomar un atributo o conjunto de ellos.

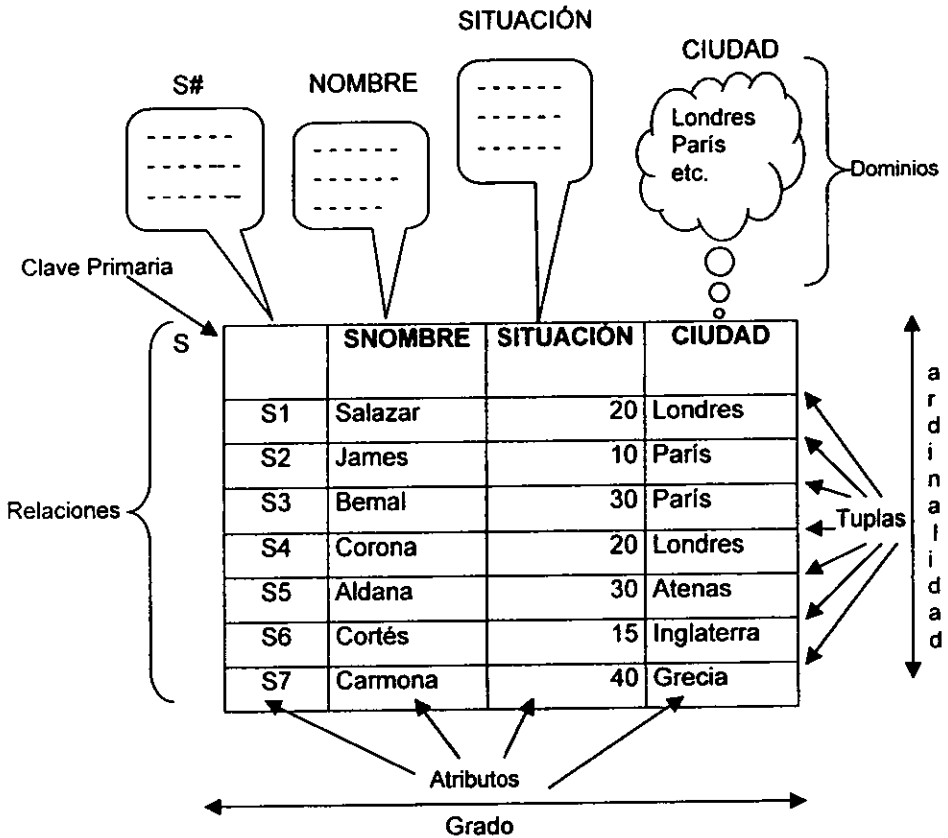


Figura 1.7.1.1. Elementos de una tabla.

Para poder identificar una tabla como una relación con los elementos mencionados, se deben cumplir las siguientes propiedades en dichos elementos que la conforman:

Hablando del dominio, el conjunto de valores que lo conforman deben tener la característica de *atomicidad*, es decir que no poseen estructura interna o que no se pueden descomponer en mas partes desde el punto de vista del modelo. Dentro de un dominio, todos los valores son del mismo tipo y un dominio puede ser utilizado en diferentes relaciones, esto para poder facilitar las comparaciones que se puedan requerir en la base de datos, es decir: si se tienen 2 relaciones (ver Figura 1.7.1.2.), una llamada catálogo de materiales y otra que represente los proveedores que surten los materiales, en ambas existirá un atributo llamado MATERIAL, el cual tendrá el mismo dominio en ambas relaciones.

**Materiales**

<b>Material</b>	<b>Descripción</b>
M001	Caja de cartón
M002	Lápices de colores
M004	Arco con segueta

**Proveedores**

<b>Material</b>	<b>Proveedor</b>
M002	Papelería Juanita
M004	Ferretera El Aguila

**Figura 1.7.1.2 Tablas con un dominio en común.**

Una comparación entre ambas tablas, es congruente cuando se realiza entre campos de igual característica, en este caso, la comparación sería entre los campos MATERIALES.MATERIAL y PROVEEDORES.MATERIAL, los cuales tienen el mismo dominio y los datos son del mismo tipo, por lo que el resultado sería congruente.

Aunque el tipo de dato en los atributos de las tablas sea texto, el dominio de cada uno es diferente, y este debe estar bien claro para poder realizar las comparaciones y/u operaciones que se requieran.

Pueden existir dominios complejos como una fecha, la cual se compone de Día, Mes y Año, para lo cual se requiere tener la posibilidad de descomponerlos en el momento que se requiera, y tener las validaciones necesarias para garantizar valores correctos, pero en general cuando se requiera descomponer un dominio, se tendrán que manejar varios atributos dada la atomicidad de los datos.

Para definir correctamente lo que es una relación, se deben mezclar los términos que la componen, estableciendo las siguientes reglas que se deben cumplir entre ellos.

Una relación se compone de dos partes: cabecera y cuerpo. La cabecera es la primera fila o tupla que contiene un número fijo de atributos ( $A_j$ ) y cada atributo corresponde a solo uno de los dominios, ( $D_j$ ) esto se puede representar de la siguiente forma:

$$\{ (A1:D1), (A2:D2), \dots, (An:Dn) \}$$

( $n = 1, 2, \dots, n$  donde  $n$  es el número de atributos o grado)

Al nombrar los atributos, estos deben ser diferentes, ya que no pueden existir dos atributos iguales aunque contengan el mismo dominio. El cuerpo es el conjunto de tuplas que contienen los datos regidos por los dominios de la cabecera, el número de tuplas puede variar con el tiempo y cada una está formada por una pareja de atributo-valor, representado como sigue:

$$\{ (A1:V11), (A2:V12), \dots, (An:V1n) \}$$

$$\{ (A1:V21), (A2:V22), \dots, (An:V2n) \}$$

$$\vdots$$

$$\{ (A1:Vi1), (A2:Vi2), \dots, (An:Vin) \}$$

( $i = 1, 2, \dots, m$  donde  $m$  es el número de tuplas o cardinalidad)

Cada una de estas parejas tiene un valor del dominio único  $D_j$  asociado al atributo  $A_j$ .

### **Propiedades de las relaciones**

- 1) Las tuplas de una relación, son un conjunto de elementos en el cual no existen elementos repetidos, o sea que cada una de ellas es diferente y se considera diferente, para lo cual se les asigna una clave primaria única en uno o varios atributos, por lo que se puede decir que la combinación de todos los atributos en cada tupla es única y sin repetidas.
  
- 2) Aunque las tuplas en las relaciones se presenten con un cierto orden, estas carecen de un ordenamiento real por posición, la relación en sí hace referencia al conjunto de tuplas y no a la forma como aparecen. No importa en que orden se presenten las tuplas de una relación, esta sigue siendo la misma y no se puede nombrar primera tupla, segunda tupla, etc.
  
- 3) La cabecera de las relaciones, está también definida como un conjunto de atributos y como conjunto no guarda un orden, en una relación los atributos pueden ser presentados en diferente orden y sigue siendo la misma relación, por lo que no hay primer atributo, segundo atributo, etc.
  
- 4) Como se explicó anteriormente, los valores de los atributos son atómicos, esto nos da la característica en las relaciones de contener en cada posición de tupla y atributo solo un valor y no múltiples valores o listas de valores es decir que no tienen grupos repetitivos. El cumplimiento de esta condición, lleva a decir que la relación esta normalizada existiendo varios niveles de normalización que se explicarán mas adelante.

### 1.7.2. La estructura de las bases de datos

Una base de datos es una colección de relaciones normalizadas las cuales tienen inter-relación entre ellas, dentro de una base de datos existen varios tipos de relaciones:

- *Relaciones base:* o reales, son aquellas que tienen un nombre específico y son autónomas, almacenan en ellas información específica e independiente de otras relaciones y se considera son principales o medulares en la base de datos.
- *Vistas:* Una vista es una relación que aunque tiene nombre, se deriva de una o varias relaciones base, por lo que no posee datos almacenados propios, estos datos están en las relaciones base.
- *Instantáneas:* Es también una relación derivada, que se genera con los datos de relaciones base, pero ésta sí almacena los datos como propios y se actualiza en forma periódica, un ejemplo es cuando se desean datos estadísticos en función de los datos reales.
- *Consultas:* Cuando se realiza una consulta a una o varias relaciones, esta se presenta como una relación, solo que no lleva nombre y no se almacena físicamente la información en la base de datos, desaparece una vez terminada la consulta.
- *Resultados intermedios:* Un resultado intermedio es una relación sin nombre que se utiliza anidada para obtener una expresión relacional mayor; no se almacena físicamente en la base de datos, es como una consulta.
- *Relaciones temporales:* Es similar a la relación base, pues posee un nombre y también almacena datos, solo que ésta se destruye en un momento dado, generalmente cuando termina el proceso que la utiliza.



Con la operación seleccionar, se escogen filas en una tabla.

**PROYECTAR:** Esta operación sirve para seleccionar un conjunto de atributos en una relación (columnas en la tabla), pudiendo ser cualquiera de ellos; esta operación se puede representar como sigue:

$$\pi\langle\text{Lista de atributos}\rangle(\langle\text{nombre de la relación}\rangle)$$

La lista de atributos puede ser desde 1 hasta  $n$ , donde  $n$  es el grado de la relación.

En la figura 1.7.2.1., se ejemplifica esta operación:

Alumnos	$\pi\text{Nombre, Sexo}(\text{Alumnos})$					
Clave	Nombre	Edad	Sexo	Grado	Nombre	Sexo
001	Juan Mendoza	23	M	5°	Juan Mendoza	M
002	Ofelia Pérez	21	F	3°	Ofelia Pérez	F
003	Felipe García	23	M	4°	Felipe García	M
004	Juan Mendoza	25	M	6°		

**Figura 1.7.2.1. Ejemplo de la operación PROYECTAR.**

Se puede observar que la proyección que se hace de la relación Alumnos, genera una nueva relación con los atributos seleccionados; pero como una relación no puede tener tuplas repetidas, la proyección elimina estas y genera la nueva relación sólo con las tuplas válidas, esto es una eliminación implícita de duplicados.





dos relaciones que se están operando. La representación de esta operación es de la siguiente forma:

$$\langle \text{relación 1} \rangle \cap \langle \text{relación 2} \rangle$$

Un ejemplo de esta operación lo podemos ver en la Figura 1.7.2.3., en la que se puede observar como es el resultado de la intersección:

Alumnos  $\cap$  Maestros

Nombre	Edad	Estatura
Jesús	40	1.73
Rogelio	38	1.65

Alumnos

Nombre	Edad	Estatura
Juan	23	1.70
Heriberto	21	1.72
Jesús	40	1.73
Pedro	35	1.69
Rogelio	38	1.65

Maestros

Nombre	Edad	Estatura
Pablo	53	1.68
Jesús	40	1.73
Rogelio	38	1.65
Raúl	23	1.68
Raquel	32	1.56

Alumnos — Maestros

Nombre	Edad	Estatura
Juan	23	1.70
Heriberto	21	1.72
Pedro	35	1.69

Figura 1.7.2.3. Ejemplo de INTERSECCIÓN y DIFERENCIA de Relaciones.

Se puede observar que el resultado de esta intersección, son solo las tuplas que tienen los mismos valores en ambas relaciones.

**DIFERENCIA:** La operación de diferencia entre dos relaciones, es contraria a la intersección, es decir, se van a obtener solo las tuplas que están en la primera relación, pero que no están en la segunda. La representación de esta relación es como sigue:

$$\langle \text{relación 1} \rangle - \langle \text{relación 2} \rangle$$

En la figura 1.7.2.3. anterior, se puede observar el resultado de aplicar la función de diferencia a las mismas relaciones que se trataron con intersección, obteniéndose las tuplas de la relación Alumnos, que no están en Maestros.

Esta operación no es conmutativa, es decir, si se invierten las relaciones, no se obtiene el mismo resultado, por lo que el resultado de las siguientes expresiones no el mismo:

$$\text{Alumnos} - \text{Maestros} \quad \neq \quad \text{Maestros} - \text{Alumnos}$$

Esto, al contrario de las dos operaciones anteriores en las que no importa cual relación esté primero, el resultado siempre es el mismo.

**PRODUCTO CARTESIANO:** también se le conoce como REUNIÓN CRUZADA; esta operación, no requiere que las relaciones involucradas sean compatibles con la unión, como las anteriores. Se puede representar como sigue:

$$\langle \text{relación 1} \rangle \times \langle \text{relación 2} \rangle$$

El producto cartesiano entre dos relaciones, da como resultado una relación que combina las dos relaciones en la cual se tendrán  $n + m$  atributos, donde  $n$  es el número de atributos de la relación 1 y  $m$  es el número de atributos de la relación 2. Para obtener las tuplas de esta relación resultante, se combinan cada tupla en la relación 1 con cada tupla en la relación 2, por lo que la cardinalidad de la nueva relación va a ser

la multiplicación del número de tuplas en la relación 1 por el número de tuplas en la relación 2, por lo que tenemos:

Dado:

$R(A_1, A_2, A_3, \dots, A_n)$  con cardinalidad  $nr$  y  
 $S(B_1, B_2, B_3, \dots, B_m)$  con cardinalidad  $ns$

$R \times S$  será:

$Q(A_1, A_2, A_3, \dots, A_n, B_1, B_2, B_3, \dots, B_m)$  con cardinalidad  $nr * ns$

En la figura 1.7.2.4, se tienen dos relaciones, NOMBRESEMP y DEPENDIENTES. El producto cartesiano entre las dos tablas, representado por:

$DEPENDIENTES\_EMP \leftarrow NOMBRESEMP \times DEPENDIENTES$

Da por resultado la relación  $DEPENDIENTES\_EMP$ , en la que se puede observar que los atributos de esta nueva relación, son todos los de  $NOMBRESEMP$  y todos los de  $DEPENDIENTES$ ; también se puede ver que la primera tupla en  $NOMBRESEMP$  se repite tantas veces como tuplas tiene la relación  $DEPENDIENTES$ , y así la segunda tupla y la tercera.

Dado que cada relación, al ser una relación válida, no tiene tuplas duplicadas, el producto cartesiano, aunque repite cada una de las tuplas de la relación  $NOMBRESEMP$ , al combinarlas con las tuplas de la relación  $DEPENDIENTES$ , se convierte cada una en única cumpliendo con el principio de "no-duplicidad".

DEPENDIENTES

NSSE	NOMDEP	SEXO	FECHAN	Parentesco
333445555	Alicia	F	05-ABR-76	Hija
333445555	Teodoro	M	25-OCT-73	Hijo
333445555	Jobita	F	03-MAY-48	Cónyuge
987654321	Abdiel	M	29-FEB-52	Cónyuge
123456789	Miguel	M	01-ENE-78	Hijo
123456789	Alicia	F	31-DEC-78	Hija
123456789	Elizabeth	F	05-MAY-57	Cónyuge

NOMBREEMP

NOMBREP	APELLIDO	NSS
Alicia	Zapata	999887777
Jazmín	Valdés	987654321
Josefa	Esparza	453453453

DEPENDIENTES\_EMP

NOMBREP	APELLIDO	NSS	NSSE	NOMDEP	SEXO	FECHA	Parentesco
Alicia	Zapata	999887777	333445555	Alicia	F	05-ABR-76	Hija
Alicia	Zapata	999887777	333445555	Teodoro	M	25-OCT-73	Hijo
Alicia	Zapata	999887777	333445555	Jobita	F	03-MAY-48	Cónyuge
Alicia	Zapata	999887777	987654321	Abdiel	M	29-FEB-52	Cónyuge
Alicia	Zapata	999887777	123456789	Miguel	M	01-ENE-78	Hijo
Alicia	Zapata	999887777	123456789	Alicia	F	31-DEC-78	Hija
Alicia	Zapata	999887777	123456789	Elizabeth	F	05-MAY-57	Cónyuge
Jazmín	Valdés	987654321	333445555	Alicia	F	05-ABR-76	Hija
Jazmín	Valdés	987654321	333445555	Teodoro	M	25-OCT-73	Hijo
Jazmín	Valdés	987654321	333445555	Jobita	F	03-MAY-48	Cónyuge
Jazmín	Valdés	987654321	987654321	Abdiel	M	29-FEB-52	Cónyuge
Jazmín	Valdés	987654321	123456789	Miguel	M	01-ENE-78	Hijo
Jazmín	Valdés	987654321	123456789	Alicia	F	31-DEC-78	Hija
Jazmín	Valdés	987654321	123456789	Elizabeth	F	05-MAY-57	Cónyuge
Josefa	Esparza	453453453	333445555	Alicia	F	05-ABR-76	Hija
Josefa	Esparza	453453453	333445555	Teodoro	M	25-OCT-73	Hijo
Josefa	Esparza	453453453	333445555	Jobita	F	03-MAY-48	Cónyuge
Josefa	Esparza	453453453	987654321	Abdiel	M	29-FEB-52	Cónyuge
Josefa	Esparza	453453453	123456789	Miguel	M	01-ENE-78	Hijo
Josefa	Esparza	453453453	123456789	Alicia	F	31-DEC-78	Hija
Josefa	Esparza	453453453	123456789	Elizabeth	F	05-MAY-57	Cónyuge

Figura 1.7.2.4. Ejemplo de un producto cartesiano.

**REUNIÓN:** Esta operación es de las más importantes dentro de una base de datos relacionales, con ella se combinan dos relaciones a través de tuplas que tienen relación entre sí, combinándolas. Para esta operación, debe haber por lo menos un atributo común en las dos relaciones que se operan y el único operador de comparación que se utiliza es el igual (=); este atributo o atributos, deben ser del mismo tipo. La reunión se representa de la siguiente forma:

$$\langle \text{relación 1} \rangle \otimes_{\text{atributo 1} = \text{atributo 2}} \langle \text{relación 2} \rangle$$

El resultado de la reunión es la combinación de todos los atributos de las dos relaciones, pero a diferencia del producto cartesiano, aquí solo se presentarán las parejas de tuplas en las que los atributos comparados, tengan el mismo valor. Un ejemplo de esto se puede ver en la figura 1.7.2.5, donde se presenta la reunión de las relaciones DEPARTAMENTO y EMPLEADO dando como resultado la relación GTE\_DEPTO. Esta reunión se representa como sigue:

$$\text{GTE\_DEPTO} \leftarrow \text{DEPARTAMENTO} \otimes_{\text{NSSGTE} = \text{NSS Empleado}}$$

Se puede observar en esta figura (1.7.2.5), que las combinaciones de tuplas que se presentan, son solo en las que los atributos NSSGTE y NSS tienen el mismo valor, ya que fue la condición de la reunión.

Los atributos comparados, tienen el mismo valor y por lo tanto uno de ellos es superfluo, ya que con solo ver uno de los involucrados en la condición, se sabe el otro.

Empleado

NOMBREP	INIC	APELLIDO	NSS	FECHAN	DIRECCIÓN	SEXO	SALARIO
José	B	Silva	123456789	09/01/55	Fresnos 731, MX	M	30000
Federico	T	Vizcaya	333445555	08/12/45	Valle 638, MX	M	40000
Alicia	J	Zapata	999887777	18/07/58	Castillo 3321, MX	F	25000
Jazmín	S	Valdés	987654321	20/06/31	Bravo 291, MX	F	43000
Ramón	K	Nieto	666884444	15/09/52	Espiga 875, MX	M	38000
Josefa	A	Esparza	453453453	31/07/62	Rosas 5631, MX	F	25000
Ahmed	V	Habbar	987987987	29/03/59	Dalias 930, MX	M	25000
Jaime	E	Botello	888665555	10/11/27	Sorgo 450, MX	M	55000

DEPARTAMENTO

NOMBRED	NUMEROD	NSSGTE	FECHAINICGTE
Investigación	5	333445555	22/05/78
Administración	4	987654321	01/01/85
Dirección	1	888665555	19/06/71

GTE\_DEPTO

NOMBRED	NUMEROD	NSSGTE	...	NOMBREP	INIC	APELLIDO	NSS	...
Investigación	5	333445555	...	Federico	T	Vizcaya	333445555	...
Administración	4	987654321	...	Jazmin	S	Valdés	987654321	...
Dirección	1	888665555	...	Jaime	E	Botello	888665555	...

Figura 1.7.2.5. Ejemplo de la operación REUNIÓN.

Un caso particular de la reunión, es cuando los atributos que se comparan tienen el mismo nombre en las dos relaciones, en este caso, los atributos superfluos son eliminados debido a que no puede haber en la nueva relación, dos con el mismo nombre, quedando el valor en uno solo que tiene el nombre común; a este tipo de reunión se le llama **REUNIÓN NATURAL**.

## Reglas que debe cumplir una base relacional

Existen 12 reglas que deben cumplir las bases de datos relacionales, estas reglas son las siguientes:

- 1) *Información*: Indica que toda la información en la base de datos debe estar presentada de la misma forma mediante relaciones con filas y columnas.
- 2) *Acceso garantizado*: esta regla indica que para tener acceso a un dato en la base de datos, debe lograrse conociendo la tabla y la columna que lo contiene además del valor de la llave primaria de la fila que lo contiene, por lo que todas las tablas deben tener una llave primaria que garantice este acceso.
- 3) *Manejo sistemático de valores nulos*: Cuando en un campo no debe existir valor, esta deberá ser tratada sin poner un valor que pueda ser validado según el tipo de campo, es decir se pondrá un "valor nulo" para esta representación. Dado que este valor implica la ausencia de datos, no podrá ser usado en una llave principal, porque ésta representa siempre valores existentes.
- 4) *Catálogo activo basado en el modelo relacional*: El catálogo del sistema, es una base de datos que contiene la descripción de los diferentes objetos del mismo; estos objetos pueden ser las tablas base, vistas, índices, usuarios, planes de aplicación, privilegios de acceso, etc.; las principales relaciones o tablas que contiene el catálogo son las siguientes:
  - a) *SYSTABLES*: Contiene la información de todas las tablas que tiene el sistema, incluyendo el usuario que la creo, el número de columnas que contiene cada una, entre muchos otros datos.



- b) **SYSCOLUMNS:** En esta tabla se almacenan los nombres de todas las columnas que tiene cada tabla, conteniendo el nombre de ambas (tabla y columna), tipo de datos, etc.
  
- c) **SYSINDEXES:** Esta tabla contiene una fila por cada índice que tiene el sistema con los datos de nombre del índice, nombre de la tabla a la que indexa, usuario que lo creo, etc.

Dependiendo del controlador de base de datos, las tablas que contiene el catálogo pueden variar incluyendo diferentes datos referentes a las tablas, usuarios, permisos, datos, etc.

Este catálogo es accesible al usuario, como cualquier otra tabla de su sistema, el cual puede ser consultado para crear consultas, vistas o nuevas tablas con dicha información.

- 5) ***Sublenguaje de datos completo:*** Un sistema de base de datos, debe tener un lenguaje de manipulación que contenga una sintaxis lineal, que pueda ser interactivo o dentro de un programa de aplicación; este lenguaje debe contener operaciones de definición de datos que permitan crear la base con la estructura que corresponde. Algunas instrucciones que permiten la creación de elementos en una base de datos son las siguientes:

- a) **CREATE TABLE:** Sirve para crear una tabla dentro de la base de datos, en esta instrucción se definen las columnas que va a tener, el tipo de datos que lleva cada columna y las llaves primaria y secundarias que va a tener. Un ejemplo de la sintaxis se puede ver en la figura 1.7.2.6.

```
CREATE TABLE proveedores
( Proveedor      CHAR(5)   NOT NULL,
  Snombre        CHAR(20)  NOT NULL,
  Situación      SMALLINT NOT NULL,
  Ciudad         CHAR(15)  NOT NULL,
  PRIMARY KEY ( Proveedor ) );
```

**Figura 1.7.2.6. Ejemplo de una instrucción CREATE.**

- b) **CREATE INDEX:** Sirve para crear algún índice en una tabla, esto es cuando ya existe la tabla y se desea agregarle algún índice que se requiera.
- c) **CREATE VIEW:** Cuando se requiere crear una vista de una tabla, la cual se puede definir como una tabla virtual que solo existe porque existe la tabla principal de la cual fue creada.

En forma paralela, tal como existen instrucciones para crear elementos, existen las respectivas que permiten eliminar dichos elementos. Estas instrucciones son las siguientes:

- d) **DROP TABLE:** Al indicarle el nombre de una tabla, esta instrucción lo que hace es borrarla, desapareciéndola del sistema.
- e) **DROP INDEX:** Permite eliminar un índice existente en una tabla, el indicar el nombre de dicho índice.
- f) **DROP VIEW:** En forma análoga a las anteriores instrucciones, esta borra la vista o tabla virtual que se le indique.

En el lenguaje de una base de datos relacional, también se deben incluir instrucciones que permitan la manipulación de los datos que se almacenan en las tablas del sistema. Las instrucciones básicas de manipulación de datos son las siguientes:

- g) **SELECT**: Esta instrucción permite aplicar el álgebra relacional a las tablas de una base de datos; su función es la de seleccionar los datos que se requieran ver a través de la sintaxis que presenta. Ejemplo:

```
SELECT nombre, grado
      FROM alumnos
      WHERE edad < 25;
```

En este ejemplo se están solicitando dos columnas (NOMBRE y GRADO) de la tabla ALUMNOS, realizando una Proyección; pero solo los que cumplen la condición (WHERE), tal condición indica que solo se deben mostrar los que sean menores de 25 años (EDAD < 25), realizando con esto la Selección.

- h) **UPDATE**: Esta instrucción permite actualizar algún valor dentro de la tabla que se opera. Ejemplo:

```
UPDATE alumnos
      SET grado = 5
      WHERE grado = 4;
```

Se puede observar en el ejemplo anterior, que para ejecutar la instrucción UPDATE, se hace una selección de una o varias filas a las cuales se les va alterar el valor indicado con el sufijo SET. Con esta instrucción se puede alterar cualquier dato de cualquier tabla del sistema, con cualquier valor que cumpla con su dominio. Si no se

indica una condición, se modifican todas las filas de la tabla.

- i) **DELETE:** Esta instrucción permite eliminar filas en la tabla que se indique, obedeciendo también a una condición de selección, es decir, se eliminan las filas seleccionadas. Ejemplo:

**DELETE alumnos**

**WHERE nombre = "Juan";**

Esta instrucción elimina todas las filas de una tabla, que cumplen con la condición nombre = "Juan". De igual forma que la anterior, si se omite una condición se eliminan todas las filas de la tabla, quedando esta vacía.

- i) **INSERT:** Esta instrucción permite agregar filas a una tabla, y debe cumplir con asignar un valor a todos los campos que conforman una fila en la tabla.

Esta quinta regla, también indica que deben tenerse restricciones de seguridad e integridad así como administración de transacciones:

**Seguridad:** Es la protección que deben tener los datos de una base de datos, a ser mostrados, alterados o eliminados sin la autorización debida es decir, los usuarios deben tener autorización para realizar dichas acciones.

**Integridad:** Es la exactitud y validez de los datos almacenados, implica asegurar que lo que se trata de hacer sea lo correcto, por ejemplo, si un campo debe tener solamente valores numéricos, este debe aceptar solo valores numéricos y rechazar cualquier otro valor.

*Transacción:* Una transacción se define como una unidad lógica de trabajo, este concepto se utiliza para poder atender a los problemas de recuperación de información y concurrencia. Cada actualización a la base de datos, se define como una transacción en ella; es posible agrupar estas modificaciones para mantener la integridad de la información en diferentes tablas ejecutando una transacción que contenga el grupo de actualizaciones que se requieren para ello. Las instrucciones que se utilizan para ello son:

*COMMIT:* Esta instrucción, compromete a que se realicen todos los cambios desde que se inicia la transacción, hasta este punto. De igual forma, una vez que se terminó la transacción con esta instrucción, se inicia la siguiente.

*ROLLBACK:* Esta instrucción, también termina una transacción e inicia otra, la diferencia con la anterior, es que con esta, no se realizan los cambios a la base de datos, sino que se desechan.

- 6) *Actualización de vistas:* Todas las vistas que existan en el sistema deberán poderse actualizar si es que en teoría son actualizables.
- 7) *Inserción, modificación y eliminación de alto nivel:* Esta regla se relaciona con la quinta, solo que globaliza las instrucciones de manipulación, justificando lo que se vio en los ejemplos, de poderse manipular un conjunto de datos.
- 8) *Independencia física de los datos:* Los usuarios y los programas no deben depender de la estructura física de la base de datos, esta existe y siempre deberá ser vista de la misma forma aunque físicamente cambie; las aplicaciones que utilicen la base de datos, deberán entonces enfocarse a la solución del problema o necesidad y no a la administración de los datos.

- 9) *Independencia lógica de los datos*: La independencia lógica de los datos, indica que cuando exista un cambio lógico en la base de datos, éste no deberá afectar a los usuarios y programas que la utilizan; podrá crecer la base en cuanto al número de campos o número de tablas, y esto será transparente para la aplicación. Esto también permitirá realizar alguna reestructuración de la base de datos, sin necesidad de hacer cambios en las aplicaciones.
  
- 10) *Independencia de la integridad*: Esto quiere decir, que la integridad de una base de datos no debe depender de la aplicación o programa que la trabaje o utilice, es decir, que debe tener inherente a sí toda la integridad de datos posibles, la cual deberá ser cargada en una tabla dentro del catálogo del sistema.
  
- 11) *Independencia de la distribución*: Cuando exista la necesidad de distribuir la base de datos en diferentes equipos, las aplicaciones existentes que la utilicen, deberán seguir funcionando sin cambio alguno, es decir, estos no deben sufrir cambios para poder tener acceso a los datos.
  
- 12) *No Subversión*: Independientemente de la interface que se ofrezca, esta no debe pasar por alto ninguna restricción de la base de datos cuando se intente subvertir el sistema.

### 1.7.3. Normalización

Como se comentó anteriormente, una relación es normal cuando para cada atributo en una tupla, existe un solo valor atómico. La teoría de normalización, nos lleva a que no solo las relaciones estén normales, sino toda la base de datos, evitando redundancias en la información que exista en las relaciones que contiene.

En esta teoría existe el concepto de formas normales las cuales podemos ver en el diagrama de la Figura 1.7.3.1 en la que se representan los diferentes niveles a los que se puede llegar en el proceso de normalización.

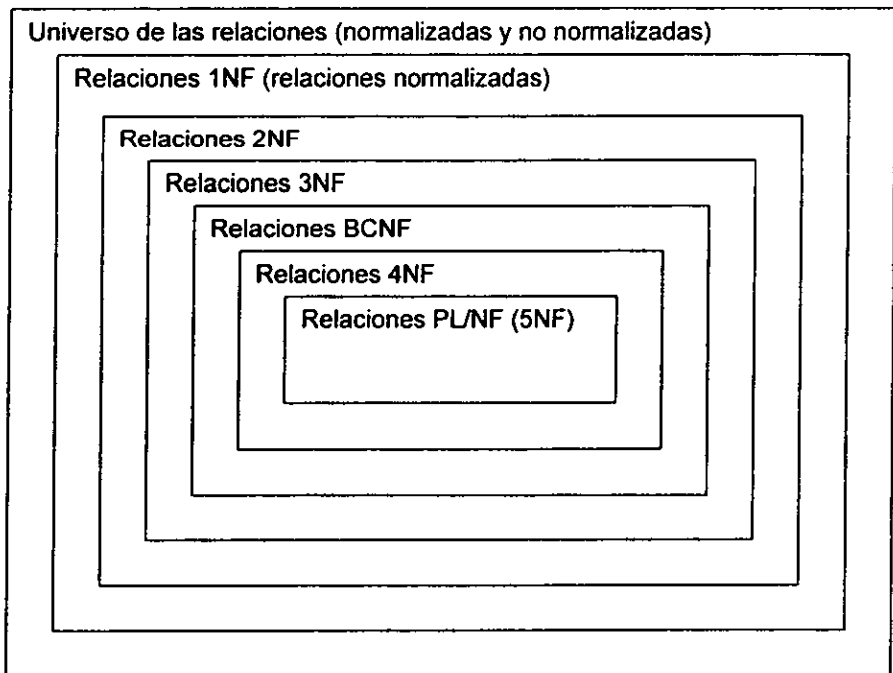


Figura 1.7.3.1. Formas Normales.

Estas formas normales se van alcanzando gradualmente cuando se aplica esta teoría, siendo deseable llegar al 5NF aunque no forzoso. Como se puede observar en la Figura 1.7.3.1., el hecho de que una relación esté en una forma normal avanzada, implica que ya está en las formas normales primarias; es decir, cuando un conjunto de relaciones está en 3NF, ya están en 2NF y en 1NF, pero no en forma inversa, una relación que está en 1NF no está en 2NF ni en 3NF.

### **Primera forma normal (1NF)**

La primera forma normal es cuando todas las relaciones de la base de datos cumplen con la propiedad de tener un valor atómico en cada uno de sus atributos.

### **Segunda forma normal (2NF)**

Se dice que una relación está en segunda forma normal (2NF) si, y solo si, está en 1NF y todos los atributos que no son llave, dependen completamente de la llave primaria.

Para llegar a esta forma normal es importante aplicar la teoría de dependencias funcionales, la dependencia funcional indica que un atributo o grupo de atributos dependen de otro que se define como llave primaria, pero son independientes entre sí. Por lo que para llevar una relación a segunda forma normal, se tienen que identificar los atributos que dependen del atributo llave, en caso de no ser así, se deberá dividir la relación en tantas que se pueda asegurar esto.

Ejemplo:

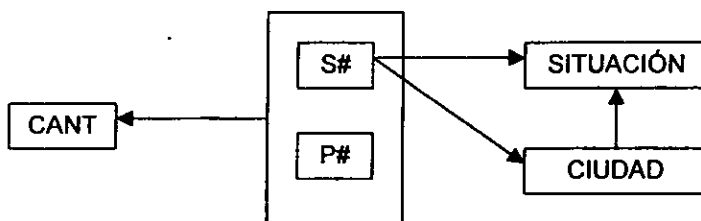
Teniendo la relación PRIMERA de la Figura 1.7.3.2., se puede observar que no hay una dependencia funcional entre ninguno de los atributos, por lo que no está en 2NF. Para llevarla a 2NF, se puede observar que los atributos Situación y Ciudad son



dependientes de S#, por lo que se separan en una relación aparte quedando dos relaciones como se muestran en la Figura 1.7.3.3.

PRIMERA	S#	SITUACIÓN	CIUDAD	P#	CANT
	S1	20	Londres	P1	300
	S1	20	Londres	P2	200
	S1	20	Londres	P3	400
	S1	20	Londres	P4	200
	S1	20	Londres	P5	100
	S1	20	Londres	P6	100
	S2	10	París	P1	300
	S2	10	París	P2	400
	S3	10	París	P2	200
	S4	20	Londres	P2	200
	S4	20	Londres	P4	300
	S4	20	Londres	P5	400

Muestra de tabulación de la tabla PRIMERA



Dependencias funcionales en la relación PRIMERA

Figura 1.7.3.2. Muestra de una tabla sin dependencia funcional.

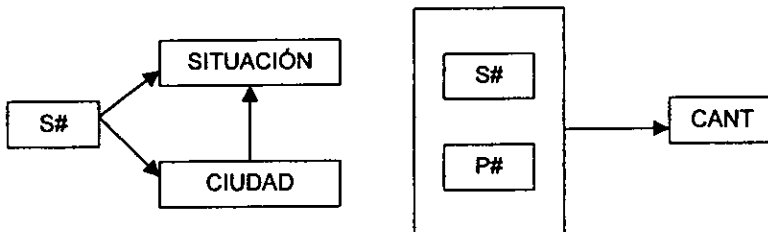
SEGUNDA

S#	SITUACIÓN	CIUDAD
S1	20	Londres
S2	10	Paris
S3	10	Paris
S4	20	Londres
S5	30	Atenas

SP

S#	P#	CANT
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Muestra de tabulación de las relaciones SEGUNDA y SP resultantes



Dependencias funcionales en las relaciones SEGUNDA y SP

Figura 1.7.3.3. Dependencia funcional en 2NF.

En esta descomposición, se debe evitar el perder datos, es decir que las nuevas relaciones van a tener todos los atributos que tenía la original, solo que repartidos en

las relaciones resultantes. Además es importante establecer la liga entre las relaciones resultantes, en el ejemplo se puede ver que la relación SEGUNDA tiene los atributos que son funcionalmente dependientes, y se determinó que la llave primaria es S#, en la relación SP, se sustituyen los tres atributos por la llave principal de SEGUNDA para que quede establecida la inter-relación entre ambas relaciones.

### Tercera forma normal (3NF)

Se dice que una relación esta en tercera forma normal (3NF) si, y solo si está en 2NF y todos los atributos que no son llave principal, son mutuamente independientes.

Regresando al ejemplo anterior, se puede observar en la Figura 1.7.3.3. que en la relación SEGUNDA, los atributos SITUACIÓN y CIUDAD, dependen mutuamente uno del otro, esto se observa también en el respectivo diagrama de dependencias funcionales incluido en la misma figura. La relación SP ya está en 3NF puesto que no hay una dependencia funcional entre sus atributos que no son llave, mostrado en el diagrama correspondiente.

SC	S#	CIUDAD	CS	CIUDAD	SITUACIÓN
	S1	Londres		Atenas	30
	S2	París		Londres	20
	S3	París		París	10
	S4	Londres		Roma	50
	S5	Atenas			

Muestras de tabulación de las relaciones SC y CS



Dependencias funcionales en las relaciones SC y CS

Figura 1.7.3.4. Dependencia funcional en 3NF.

Para llevar esta relación a 3NF, es necesario realizar una separación de la relación SEGUNDA para obtener las relaciones SC y CS que se muestran en la Figura 1.7.3.4., donde se puede ver también su diagrama de dependencias funcionales en el que sólo existe una flecha de salida de la llave principal hacia el atributo respectivo. En este caso las llaves principales son S# y CIUDAD respectivamente.

### **Forma normal Boyce/Cood (BCNF)**

"Una relación está en forma normal Boyce/Codd (BCNF) si y solo si todo determinante es una clave candidata"<sup>1</sup>. Se entiende como clave candidata, a una llave que no es principal, que es compuesta (varios atributos) y que tienen en común algún atributo.

Esta forma normal es similar a la 3NF, solo que considera que pueden haber llaves compuestas o alternas, que ocasionan problemas de actualización aun estando en 3NF. Para mostrar la aplicación de esta forma normal, tomemos como ejemplo la relación SSP que se muestra en la Figura 1.7.3.5., en la que S# es la llave principal (número de proveedor) y que SNOMBRE es el nombre del proveedor S#, en este caso, solo puede haber un nombre para cada proveedor y cada proveedor tiene un solo nombre, por lo que SNOMBRE puede ser una llave alterna y se ve claro la falta de normalización.

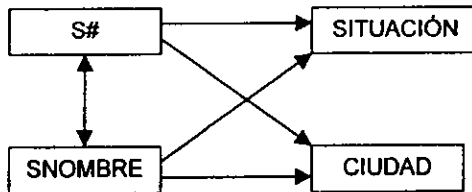
El diagrama de dependencia funcional es más complejo ya que hay múltiples relaciones. La descomposición de esta relación, se realiza separando por un lado S# y SNOMBRE en la relación SS y dejando S#, P# y CANT en una tabla SP.

---

<sup>1</sup> Sistemas de bases de datos, C. J. Date, Addison-Wesley Iberoamericana, Quinta edición, pp 533

SSP	S#	SNOMBRE	P#	CANT
	S1	Salazar	P1	300
	S1	Salazar	P2	200
	S1	Salazar	P3	400
	S1	Salazar	P4	200
	•	•	•	•

Muestra de tabulación de la relación SSP



Dependencias funcionales en la relación S, si SNOMBRE es clave candidata (y no se cumple CIUDAD -> SITUACIÓN)

Figura 1.7.3.5. Ejemplo de relación que requiere BCNF.

#### Cuarta forma normal (4NF)

Esta forma normal se basa en el hecho de que puedan existir dependencias multivaluadas en una relación, es decir que haya dependencia entre A y B y entre B y C, siendo A, B y C atributos de una relación y que no son llaves; esto implica que haya una dependencia indirecta entre A y C que pueda llevar a problemas de actualización de los datos; al no ser atributos llave, una relación de este tipo ya está en BCNF, pero aun no es óptima para su manejo. En este caso la 4NF consiste en descomponer las dependencias multivaluadas, creando dependencias funcionales con llaves principales.

Con esto, una relación está en 4NF cuando esta en BCNF y además, todas las dependencias multivaluadas son o se representan como dependencias funcionales.

### Quinta forma normal (5NF)

“Una relación R está en quinta forma normal (5NF) –llamada también forma normal de proyección-reunión (PJ/NF)- si y solo si toda dependencia de reunión en R es una consecuencia de las claves candidatas de R<sup>2</sup>. Una dependencia de reunión es cuando en una tabla existen por lo menos 3 llaves, y no existe una dependencia ni funcional ni multivaluada entre ellas, por lo que es necesario descomponer dicha relación a fin de eliminar dicha dependencia, la cual debe cumplir la restricción de que al descomponer las tres llaves (A, B, C), se puedan tener las relaciones (A, B), (A, C) y (B, C).

Estas tres últimas formas normales son muy similares, la diferencia radica en el tipo de dependencia que está causando las anomalías de actualización de la relación, ya sea por dependencias funcionales (BCNF), multivaluadas (4NF) o de reunión (5NF).

En conclusión decimos que el proceso de normalización tiene como objetivos:

- Eliminar los diferentes tipos de redundancia de datos que pueda haber en una base de datos.
- Evitar anomalías de actualización de las relaciones, es decir que al modificar un dato solo se tenga que actualizar una tupla y múltiples de ellas para mantener la integridad de la información.
- Que el diseño de la base de datos sea una representación acercada a la realidad de los datos, para facilitar su entendimiento y tener una base para el crecimiento futuro.

---

<sup>2</sup> Sistemas de bases de datos, C. J. Date, Addison-Wesley Iberoamericana, Quinta edición, pp 546

Estos lineamientos de normalización, no son aplicables en todos los casos, habrá ocasiones en que el llevar una relación a una forma normal 5NF o incluso BCNF no sea lo más práctico dada la naturaleza de los datos, proyecciones de este tipo pudieran entrar en conflicto o incluso dejar de ser claras. Ya que no todas las redundancias se pueden eliminar con esta técnica, va a depender del diseño que desde inicio se haga de los datos requeridos.

## **1.8. Características, Ventajas y Desventajas de Visual Basic 5.0**

### **Características Generales**

Es un sistema de desarrollo diseñado especialmente para crear aplicaciones con interfaz gráfica, en una forma rápida y sencilla. Para soportar este tipo de desarrollos, Visual Basic utiliza fundamentalmente dos herramientas, una que permite realizar los diseños gráficos y un lenguaje de alto nivel.

Visual Basic esta centrado en dos tipos de objetos, Ventanas y Controles, que permiten diseñar sin programar, una interfaz gráfica (mecanismo de comunicación entre el usuario y la aplicación) para una aplicación. Para realizar una aplicación, se crean ventanas, llamadas formularios y sobre ellas se dibujan otros objetos llamados controles, tales como cajas de texto, botones de ordenes, listas desplegadas, etc.

Quiere esto decir que cada objeto (ventanas y controles) están ligados a un código que permanece inactivo hasta que se da el suceso que lo activa. Por ejemplo, se puede programar un botón de órdenes (objeto que se puede pulsar) para que responda a un clic del ratón.

Visual Basic también incluye, como características más sobresalientes:

Controles 3D, además de una interfaz para múltiples documentos que permite crear una aplicación con una ventana y múltiples ventanas de documento, tal como, por ejemplo, Microsoft Word, soporte para intercambio de datos con otras aplicaciones (DDE Dynamic Data Exchange).

Soporte para establecer enlaces, con Windows y con rutinas escritas en otros lenguajes (DLL Dinamic Link Libraries).



Soporte para objetos incrustados y vinculados (OLE Object Linking and Embedding).

Esto permite la construcción de aplicaciones utilizando un Software autocontenido en pequeñas y potentes unidades o componentes de Software.

Visualización y manipulación de datos de otras aplicaciones Windows utilizando controles OLE.

Acceso y manipulación de objetos incrustados y vinculados suministrados por otras aplicaciones utilizando la automatización de OLE.

Un compilador para añadir ayuda en línea; esta herramienta permite la creación de ayudas estilo Windows (HC-Help Compiler).

Biblioteca para SQL que permite manipular bases de datos relacionales, tales como Microsoft Access (SQL- Structure Query Lenguaje).

#### **Un sistema de ayuda en línea.**

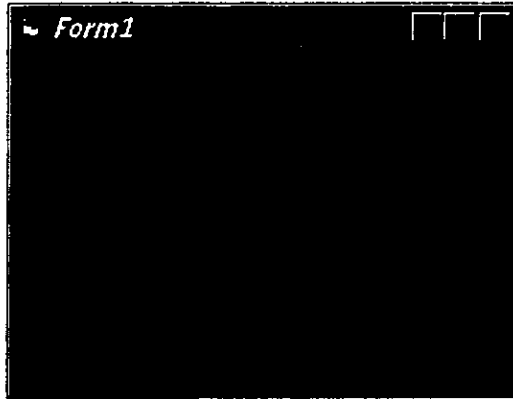
Una biblioteca de iconos para trabajos profesionales, acceso a Base de Datos por medio del control (Data Control). A través de objetos de acceso a datos. Ambos pueden utilizar el acceso del motor de Access o de controladores ODBC.

Manipulación de base de datos en varios formatos (Microsoft Access, DbaseIII y IV, etc.) con el administrador de datos (Data Manager).

Un control que permita utilizar las cajas de diálogo más comúnmente utilizadas (abrir, guardar como, imprimir, color y fuentes).

Menús desplegables flotantes.

Un asistente, Setup Wizard, para crear los discos de distribución para sus aplicaciones.



**Figura 1.8.1. Formulario en Visual Basic.**

Un generador de informes (Crystal Reports), para la creación de informes, listados y documentos a medida utilizando datos de una base de datos.

Creación de aplicaciones de 16 bits (ediciones profesional y empresarial) y de 32 bits (ediciones estándar profesional y empresarial).

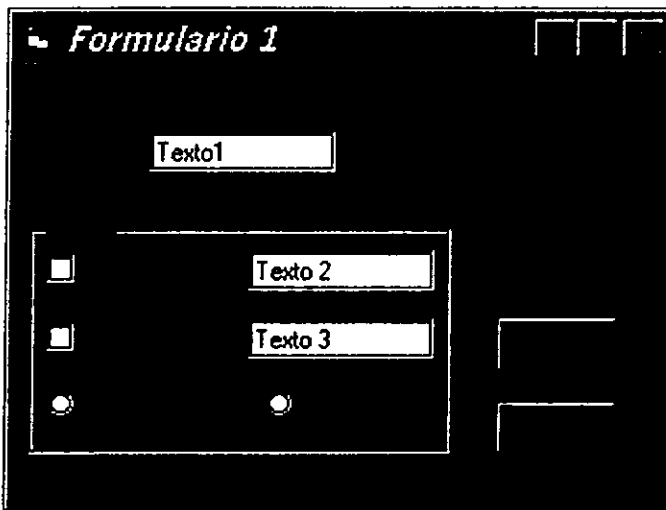
Cuando se combinan estas características, algunas de ellas sólo disponibles en la versión profesional y empresarial, se dispone de un sistema de desarrollo que permite diseñar rápidamente aplicaciones sofisticadas.

Con Visual Basic se pueden crear formularios sin escribir nada de código. Para crear un nuevo formulario se ejecuta la orden formulario del menú Insertar (figura 1.8.1.).

Un formulario es una ventana que sirve de fondo para los controles y para los gráficos situados sobre la misma. Se pueden utilizar tantos formularios como se necesiten, y dependiendo de la utilidad que se les dé, éstos serán de diferentes tipos. Así, se pueden crear un formulario para que contenga un gráfico, para visualizar información o

para aceptar datos.

Los controles son objetos que dibujamos sobre el formulario, tales como etiquetas, cajas de texto, marcos, casillas de verificación, botones de opción, botones de pulsación, etc.; con fines aceptar, ejecutar o visualizar datos (figura 1.8.2.).



**Figura 1.8.2. Formulario con Controles (Objetos).**

Para dibujar los controles se utiliza la caja de herramientas de Visual Basic (Toolbox). También es posible incluir cualquier imagen diseñada a medida o dada por un icono suministrado por Visual Basic, un mapa de bits o un archivo Windows.wmf.

Además, podemos añadir menús desplegables y menús desplegables flotantes, para proporcionar al usuario un amplio rango de órdenes seleccionables.

Visual Basic permite crear aplicaciones de muy diversos tipos. Por ejemplo, un archivo representado como un montón de archivos convencionales; un formulario para introducir información para una base de datos; un formulario que permita introducir una

palabra de paso, como medida de seguridad, para entrar en la aplicación, un juego, un diagrama de barras, etc.

### **Nuevas características en Visual Basic**

La característica más importante que incorpora Visual Basic es el nuevo motor del lenguaje - Visual Basic para aplicaciones - totalmente compatible con versiones anteriores de Visual Basic y con el Visual Basic utilizado en las aplicaciones Microsoft Excel y Microsoft Project. Visual Basic para aplicaciones facilita mucho la programación de objetos OLE automatizados. Otras características nuevas son las siguientes:

**Soporte para desarrollo de aplicaciones en sistemas de 32 bits.**

Posibilidad de ejecutar varias instancias de Visual Basic (VB). Esto permite abrir varios proyectos a la vez facilitando el traspaso de código entre ellos, así como la depuración de su interacción, cuando ésta exista.

La ventana Opciones (orden Opciones del menú Herramientas), cuyos parámetros permiten especificar la configuración general de VB, presenta ahora cuatro diálogos que permiten cambiar las características del entorno, del proyecto, del editor y otras características avanzadas.

Posibilidad de ver todos los procedimientos y funciones, uno tras otro, de un mismo objeto o de un determinado módulo. Para ello, basta activar la opción Ver módulo completo en el diálogo Editor de la ventana de Opciones.

**Utilización del botón derecho del ratón para acceder a un menú rápido.**

Posibilidad de fijar los controles en el formulario (orden Bloquear controles de menú Edición).

Las propiedades de los objetos se pueden editar ahora directamente en sus celdas, sobre las ventanas de propiedades.

Posibilidad de crear objetos incrustados y vinculados – clases (OLE – Object Linking and Embedding). Esto permite la construcción de aplicaciones utilizando un software autocontenido en pequeñas y potentes unidades o componentes de software reutilizables.

Visual Basic soporta, a la vez, dos tecnologías de componentes software: VBX y OCX. Los controles OCX (controles OLE “automatizados”) representan un nuevo estándar en componentes de software que sustituirán gradualmente a los controles VBX.

Compilación condicional (directriz # If... Then ... # Else).

Archivos de recursos. En una aplicación Windows escrita en C/C++ los mapas de bits (cursores, iconos, imágenes, etc.), textos y cuadros de diálogo se guardan en Archivos separados del archivo ejecutable, denominados archivos de recursos. Ahora Visual Basic permite introducir una aplicación a otros lenguajes y guardar los recursos en archivos de estos. Las funciones LoadResString, LoadResPicture y LoadResData nos permitirán obtener esos recursos y utilizarlos en la aplicación. Esto exige disponer de un compilador de recursos externo, como por ejemplo el de Visual C++.

Posibilidad de construir aplicaciones VB que amplíen o modifiquen el entorno de trabajo VB. Las aplicaciones construidas con este objetivo serán añadidas al menú Complementos con el Administrador de Complementos.

Construcción y reutilización de nuestros propios objetos utilizando módulos que definen clases. Un módulo que define una clase de objetos es muy similar a un formulario, excepto que un objeto de esa clase no es visible durante la ejecución. Los procedimientos y las funciones definidos en el módulo son los métodos de la clase y las variables definidas a nivel del módulo y los procedimientos Property son las

propiedades o atributos de la clase. Desde el punto de vista de la programación orientada a objetos, es importante dejar claro que no se trata de objetos vistos desde el punto de vista de C++; por ejemplo, están presentes las características de encapsulamiento y polimorfismo, pero no la herencia. Los procedimientos Property permiten añadir propiedades personalizadas a los formularios, a los módulos estándar y a las clases.

Ahora, una sentencia Visual Basic puede ser escrita en varias líneas físicas utilizando el carácter de continuación de línea (un espacio en blanco seguido del carácter de subrayado).

Nuevas sentencias For Each ... Next y With ... End With.

Colecciones propias de objetos. Podemos añadir cualquier objeto a una colección propia mediante el método Add, eliminarlo mediante Remove y acceder al mismo mediante Item.

Examinador de objetos(object browser). Muestra los objetos que se pueden crear en el proyecto y las propiedades y métodos asociados con esos objetos.

Esto permite navegar rápidamente de un punto a otro en el código, lo que hace posible copiar y pegar código de una forma rápida y reduciendo el riesgo de cometer errores.

Objetos para acceso a datos (DAO). Visual Basic incorpora la versión más nueva del motor de base de datos, junto con la nueva interfaz de objetos de acceso a datos. Con respecto a la versión anterior de VB, hay muchas características nuevas como un lenguaje de definición de datos totalmente orientado a objetos, nuevos controles ligados a datos (DBList, DBCombo y DBGrid), nuevos objetos Recordset, mejoras en el Data Control y mejoras en los controladores ODBC.

Construcción de aplicaciones que soporten la automatización de OLE. La

automatización de OLE permite a una aplicación controlar objetos contenidos por otras aplicaciones (servidores OLE).

Creación de DLLs relativas a objetos OLE.

Nuevos controles: carpetas (TabStrip), barra de herramientas (ToolBar), barra de estado (StatusBar), árbol de directorios (TreeView), lista de archivos (ListView), matriz de imágenes (ImageList), barra de desplazamiento en forma de un control de volumen (Slider), etc.

Entorno de desarrollo de Visual Basic

Visual Basic provee de varias herramientas para facilitar el diseño de cualquier aplicación gráfica. Cuando se arranca Visual Basic aparece en pantalla una interfaz como la que se puede observar en la siguiente figura (figura 1.8.3.).

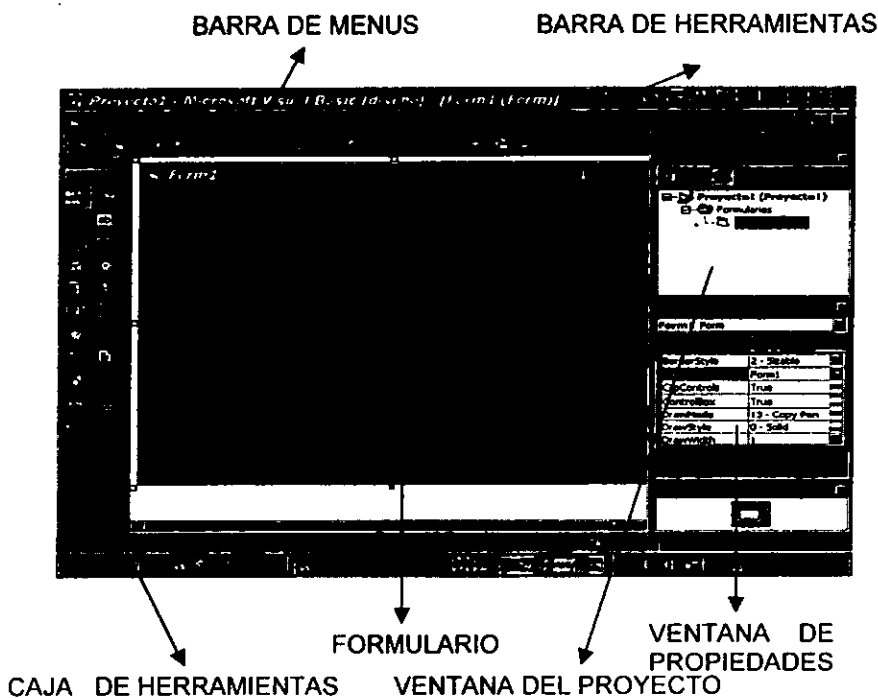


Figura 1.8.3. Interfaz de Visual Basic.

En la interfaz que aparece en pantalla se distinguen los elementos siguientes:

Barra de menús. Visualiza las órdenes a utilizar para desarrollar una aplicación.

Barra de herramientas. Facilita un acceso rápido a las ordenes más comúnmente utilizadas. Las órdenes correspondientes a estos iconos son las siguientes:

Crear un nuevo formulario.

Crear un nuevo módulo.

Abrir un proyecto existente.

Guardar el proyecto actual.

Fijar los controles en el formulario activo.

Visualizar la ventana de diseño de menús.

Visualizar la ventana de propiedades.

Examinador de objetos.

Visualizar la ventana del proyecto.

Ejecutar la aplicación en modo de diseño.

Introducir una pausa en la línea actual.

Visualizar el valor del elemento seleccionado en la ventana de código.



Visualizar la estructura de las llamadas activas.

Ejecutar una sentencia cada vez (ejecución paso a paso).

Ejecutar un procedimiento o una sentencia cada vez.

Ventana del proyecto. El conjunto de todos los archivos (formularios, módulos, clases y recursos) que forman una aplicación se denomina aplicación o proyecto. Si observamos la figura 1.8.3. la ventana titulada Proyecto1; esta ventana contiene la lista de los archivos que componen el proyecto actual.

Para visualizar un formulario de un proyecto, primero hay que seleccionarlo en la ventana del proyecto, haciendo clic sobre el nombre del mismo y después se hace clic sobre el botón Ver Formulario.

Para ver el código de un formulario, de un módulo o de una clase, primero hay que seleccionarlo en la ventana del proyecto y después se hace clic sobre el botón Ver Código.

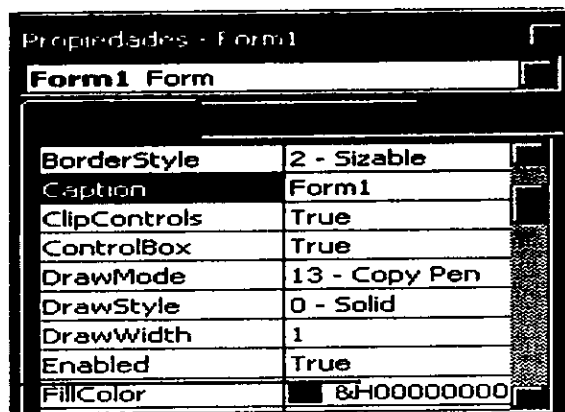
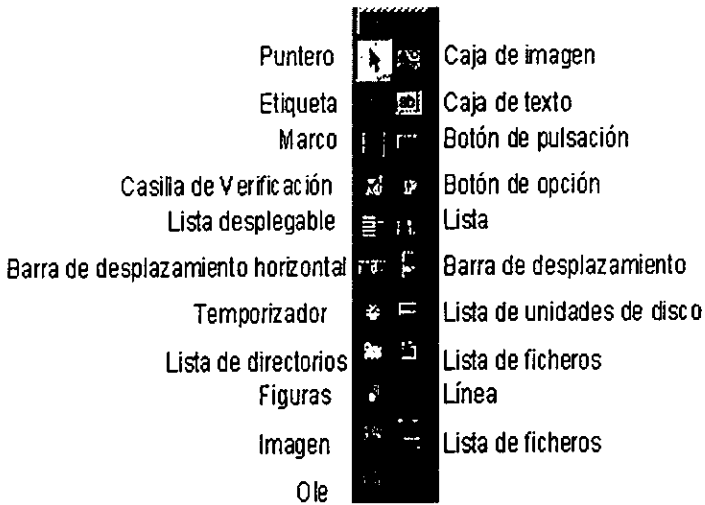


Figura 1.8.4. Ventana de propiedades.

Ventana de propiedades. Cada objeto lleva asociado un conjunto de propiedades (nombre, posición, tamaño, color, etc.). Para especificar las propiedades de cada uno de ellos, utilizaremos la ventana de propiedades (figura 1.8.4.).

Caja de herramientas. Provee de un conjunto de herramientas (fig. 1.8.5.) que permiten colocar los controles en el formulario durante el diseño de la interfaz gráfica del usuario.



**Figura 1.8.5. Elementos de la caja de herramientas.**

Se expone a continuación el significado de los controles más comunes:



**El Puntero.** Se utiliza para manipular los controles existentes sobre el formulario. Con el puntero se puede seleccionar, mover y ajustar el tamaño de los objetos.



**La caja de imagen.** Se utiliza cuando queremos visualizar mapas de bits, iconos o archivos .wmf también se puede utilizar para visualizar texto o como contenedor de otros controles.



**La etiqueta.** Se utiliza cuando se quiere un texto, de una o más líneas, que no pueda ser modificado por el usuario. Una etiqueta se utiliza para dar instrucciones al usuario.



**La caja de texto.** Es un área dentro del formulario en la que el usuario puede escribir o visualizar texto.



**El marco.** Se utiliza para realzar el aspecto del formulario. A veces utilizamos los marcos para agrupar los objetos relacionados entre sí, los marcos tienen propiedades propias, como por ejemplo, color.



**El botón de pulsación.** Tiene asociada una orden con él. Esta orden se ejecutará cuando el usuario haga clic sobre el botón.



**La casilla de verificación.** Se utiliza para seleccionar una opción. De esta forma se pueden seleccionar varias opciones de un grupo.



**Botón de opción.** Se utiliza para seleccionar una opción de entre varias. De esta forma sólo se puede seleccionar una opción de un grupo de ellas.



**Lista desplegable.** Combina una caja de texto y una lista. Permite al usuario escribir lo que desea seleccionar o elegir un elemento de la lista.



**Lista.** Contiene una lista de elementos de la que el usuario puede seleccionar uno.



**La barra de desplazamiento horizontal y la barra de desplazamiento vertical.** Permiten seleccionar un valor dentro de un rango de valores. Estos controles son utilizados independientemente de otros objetos, y no son lo mismo que las barras de desplazamiento de una ventana.



**El temporizador.** Permite activar procesos a intervalos regulares de tiempo.



**La lista de unidades de disco.** Se utiliza para visualizar la lista de unidades disponibles, con el fin de seleccionar una.



**La lista de directorios.** Se utiliza para visualizar los directorios a los que el usuario puede moverse.



**La lista de ficheros.** Se utiliza para visualizar los archivos de un determinado directorio a los que el usuario puede acceder.



**El control figuras.** Se utiliza para añadir rectángulos, cuadrados, elipses o círculos a un formulario.



**El control Línea.** Se utiliza para añadir líneas rectas a un formulario.



**El control imagen.** Se utiliza cuando queremos visualizar mapas de bits, iconos o archivos .wmf. Cuando se hace clic sobre este control, su comportamiento es el mismo que el de un botón de pulsación.



**El control OLE.** Permite incrustar datos en una aplicación.

**Formulario.** Es la ventana sobre la que colocaremos los controles de la interfaz gráfica que el usuario identificará para comunicarse con la aplicación.

Además cuenta con un asistente para aplicaciones de VB (VB Application Wizard), fig.1.8.6.; que automatizan tareas basándose en preguntas y respuestas. La aplicación generada por el asistente, se conoce como estructura del programa, su propósito es crear una estructura de código, la cual se puede ejecutar pero no lleva consigo alguna funcionalidad específica.

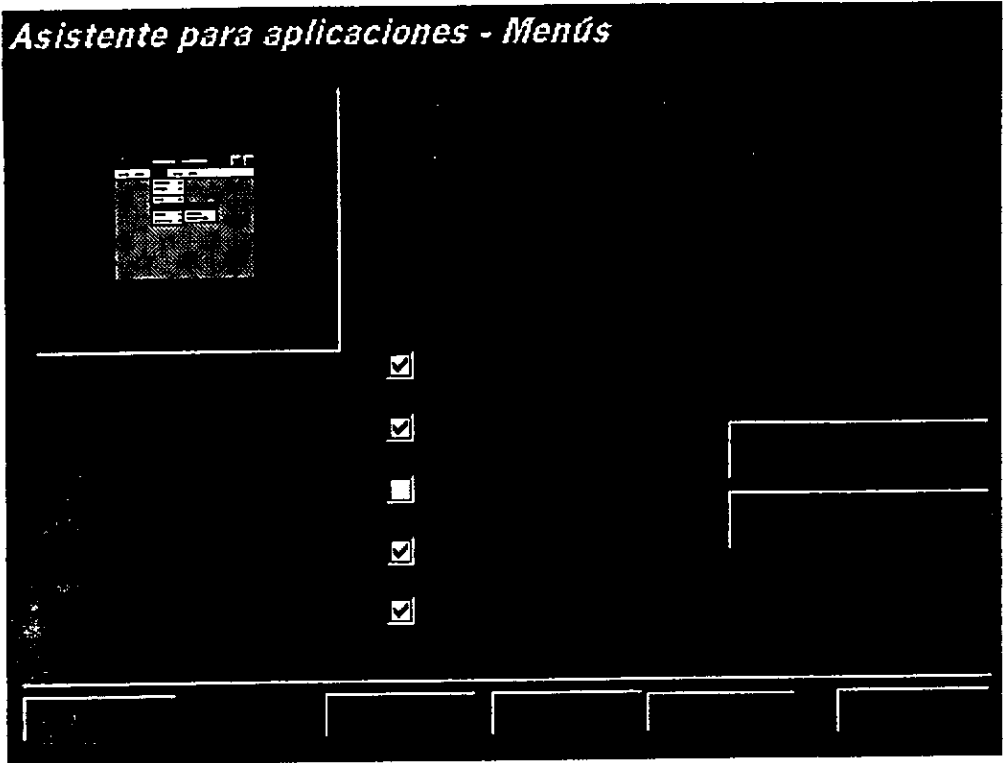


Figura 1.8.6. Asistente para aplicaciones de Visual Basic.

Esta estructura contiene un menú, una barra de herramientas y algunas otras características, pero tales características podrían tomarse más como puntos que le indican dónde agregar código para ejecutar una tarea específica.

#### Recursos mínimos de Hardware que requiere Visual Basic 5.0

Microprocesador pentium o superior.

Un disco duro con un espacio mínimo disponible de 50Mb para poder realizar una instalación completa.

Una unidad de diskette (3 ½" o 5 ¼") o una unidad de CD-ROM.

Un Mouse.

Una tarjeta de Vídeo que soporte Windows 3.1.

16 Mb de memoria o más. Con Windows 95 se requieren 32Mb.

### **Ventajas y Desventajas de Visual Basic**

Con Visual Basic podemos manipular los datos de una base de datos a nuestro gusto de forma que resolvamos problemas concretos mediante aplicaciones que sean fáciles de usar por cualquier persona que no necesita tener ningún conocimiento de bases de datos ni de programación.

VB es el lenguaje de programación visual con mayor soporte a nivel mundial. Se puede obtener ayuda e información por diversos medios y es relativamente sencillo encontrar desarrolladores especializados. Su facilidad de aprendizaje y su interfaz amigable al usuario.

Algunas de sus desventajas: no cuenta con apuntadores estándar, ni con programación orientada a objetos (OOP, sus siglas en inglés) real. En ocasiones, el manejo automático de la memoria (con datos de cadena y Variant) puede producir pequeños y muy ocasionales retrasos en los programas. Microsoft ha alterado en varias ocasiones la estrategia de acceso a datos (DAO, RDO y ADO), lo que ha traído serias consecuencias a los desarrolladores. A pesar de esto, la firma que han desarrollado las ventanas ha dicho que ADO es la estrategia a seguir y que los desarrolladores ya no deberán preocuparse al respecto.

## **1.9. Características, Ventajas y Desventajas de Access 97**

### **Access 97**

Microsoft Access no es mas que una aplicación informática (o programa informático) y por tanto, esta diseñada para su utilización en un computador.

Mientras otras aplicaciones se utilizan para escribir textos (los procesadores de texto) o para realizar cálculos repetitivos (las hojas de cálculo), Access se utiliza para gestionar información, ya que es un gestor de base de datos relacionales.

Como gestor Microsoft Access proporciona el conjunto de herramientas necesarias para llevar a cabo la gestión completa de nuestros datos.

### **Gestión de bases de datos**

El objetivo de la informatización de los datos es su gestión. El objetivo de introducir datos en una computadora es poder gestionarlos en forma eficaz y segura (evitando errores).

La gestión de base de datos abarca cualquier operación que se pueda llevar a cabo con los datos contenidos en las bases de datos o con las bases de datos en sí. De este modo, la gestión de bases de datos tiene un doble objetivo: el continente (la base de datos) y el contenido (los datos). Son muchas las operaciones que se pueden llevar a cabo con las bases de datos y con la información que contienen. Al crear una tabla se esta gestionando la base de datos, al introducir datos en dicha tabla también se le esta gestionando e incluso al eliminar una tabla también se realiza un proceso de gestión.



En resumen, cualquier operación que se lleve a cabo con una base de datos o con sus datos entra dentro de la gestión de bases de datos. Las siguientes operaciones muestran algunos ejemplos de esta gestión.

Añadir información a la base de datos. Por ejemplo introducir un nuevo registro de empleado.

Modificar la información ya existente en la base de datos. Un ejemplo, cuando un empleado contenido en la base de datos cambia de domicilio o teléfono.

Eliminar la información que ya no sea útil. Si un empleado abandona la empresa, sus datos dejarán de ser útiles en un momento dado.

Buscar datos concretos en la base de datos. Si se necesita el teléfono de un empleado, se puede extraer de la base de datos.

Clasificar los registros de la base de datos según distintos criterios (que puede ser el contenido de un campo en concreto). Si se desea enviar cartas a todos los empleados de una empresa, sería útil ordenar sus datos por código postal.

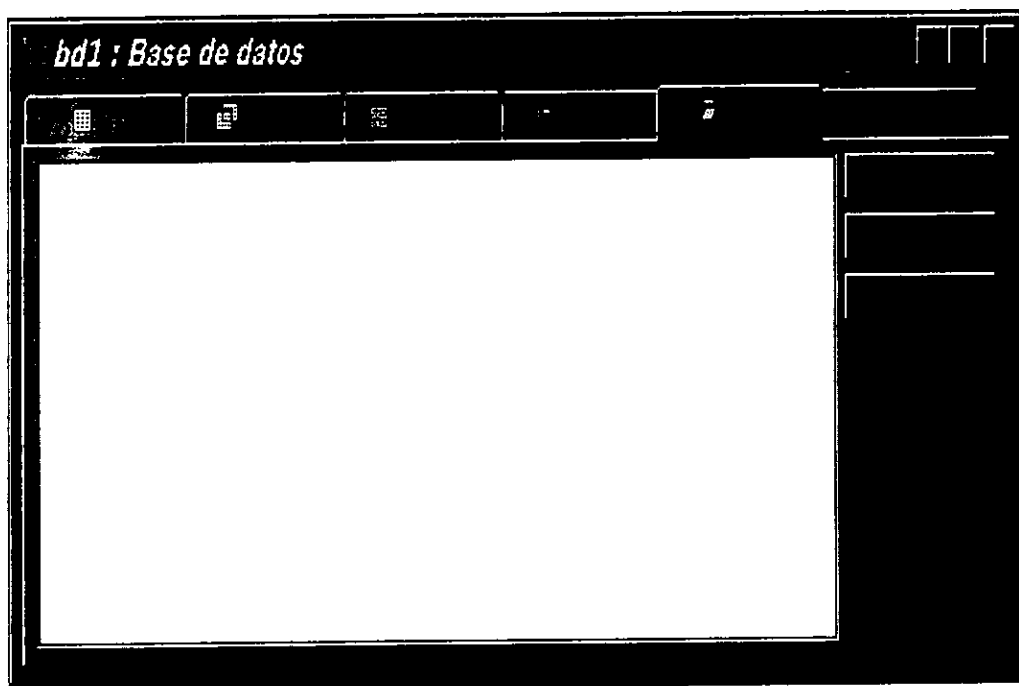
Copiar el contenido de una base de datos en otra distinta. Para tener una copia de seguridad de la misma.

Realizar consultas a la base de datos. Esta es una de las operaciones más importantes, ya que el fin de este tipo de programas suele ser la extracción de

información de forma rápida y precisa. Las consultas consisten en pedir a la base de datos que proporcione información sobre los datos que cumplan ciertas condiciones. Un ejemplo es pedir a la base de datos que proporcione el nombre de los empleados que vivan fuera de la ciudad en la que tengamos asentada nuestra oficina.

Calcular valores basándose en ciertos datos existentes en la base de datos.  
Imprimir los datos existentes en las tablas en un formato determinado, normalmente utilizando un informe.

A continuación se describirán las características de las herramientas que Access nos proporciona para llevar a cabo las tareas anteriormente descritas figura 1.9.1.



**Figura 1.9.1. Ventana de aplicación Access (Ventana de documento).**

## **Las tablas**

Las tablas son los objetos de Access que contienen los datos y, por tanto, se pueden considerar lo más importante. Están estructuradas en filas (registros) y columnas (campos).

En la figura 1.9.2., se muestra la ventana de diseño de tablas, en la que se procede a la definición de los campos que contendrán los registros de cada una de las tablas.

Además de crear las tablas, la ventana anterior permite también llevar a cabo la modificación de los campos o de sus propiedades. Una vez creada una tabla (la estructura de nuestra tabla sin datos), el siguiente paso consiste en introducir los datos que se desea que contenga.

## **Las consultas**

Las consultas, como su nombre lo indica, tienen la misión principal de “preguntar” a Access sobre el contenido de una o varias tablas (figura 1.9.2.). Además de “preguntar” sobre el contenido, hay otros tipos de consultas que también nos permiten modificar los datos existentes en las tablas.

## **Los formularios**

Otras herramientas muy potentes son los formularios. En la figura 1.9.3. se presenta un formulario que permite la modificación (o introducción).

Las ventajas principales que se pueden percibir, son su mayor claridad, su mejor presencia y la posibilidad de introducir datos en varias tablas a la vez.

## Los informes

Los informes se utilizan con el fin de obtener una salida (normalmente impresa, aunque se pueden ver en pantalla) de los datos existentes en las tablas de forma elegante. Además se pueden aprovechar las consultas o formularios ya creados para facilitar el proceso y presentar los datos en un orden preestablecido (figura 1.9.3.).

Categorías : Tabla	
IdCategoría	AutoNumérico
NombreCategoría	Texto
Entero largo	
Incrementalmente	

Figura 1.9.2. Ventana para el diseño de una tabla.

Ventas de productos para 1994 : Consult	
<b>Categorías</b>	<b>Productos</b>
*	*
IdCategoría	IdProducto
NombreCatego	NombreProduct
Descripción	IdProveedor
NombreCategoría	NombreProducto
Categorías	Productos
Agrupar por	Agrupar por
Ascendente	Ascendente
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 1.9.3. Consulta en Access.

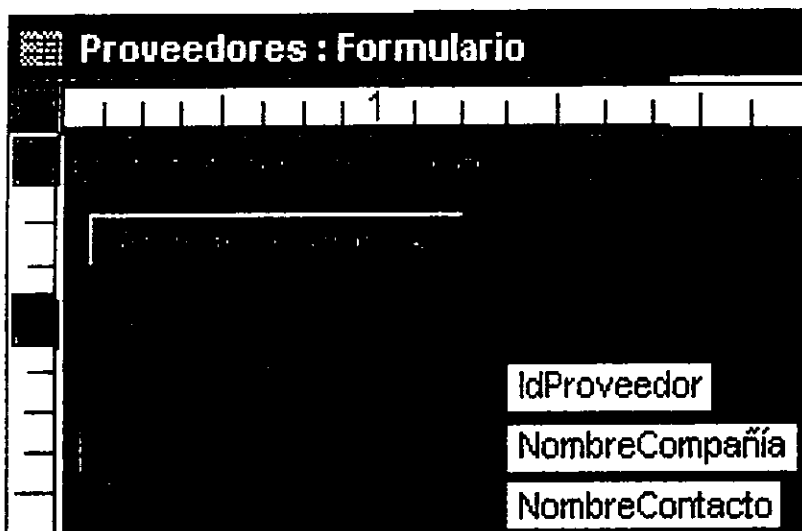


Figura.1.9.4. Formularios en Access .

<b>Resumen de ventas</b>			
<i>21-May-95</i>			
<b>Año:</b>	<b>Trimestre:</b>	<b>Pedidos:</b>	<b>Ventas:</b>
<b>1994</b>			
	1	95	\$121,824
	2	93	\$150,596
	3	107	\$169,458
	4	137	\$204,694
<b>Totales para 1994:</b>		432	\$646,572

Figura 1.9.5. Ejemplo de un informe.

La creación y modificación de los distintos objetos de una base de datos puede ser mas o menos complicada según se trate de una tabla, una consulta o un informe. Sin embargo, la mayoría de estos objetos disponen de dos herramientas, que facilitan las operaciones de creación y diseño; los asistentes y los generadores.

Los generadores tienen la misión de simplificar una tarea concreta, como puede ser la creación de una expresión o la inclusión de un control.

Los asistentes también simplifican algunas tareas, pero lo hacen interactuando con el usuario, mediante una serie de preguntas y respuestas. Entre los asistentes hay que destacar los que sirven para crear tablas, consultas, formularios e informes (Figura 1.9.6. en la que se muestra el acceso al asistente para crear bases de datos).

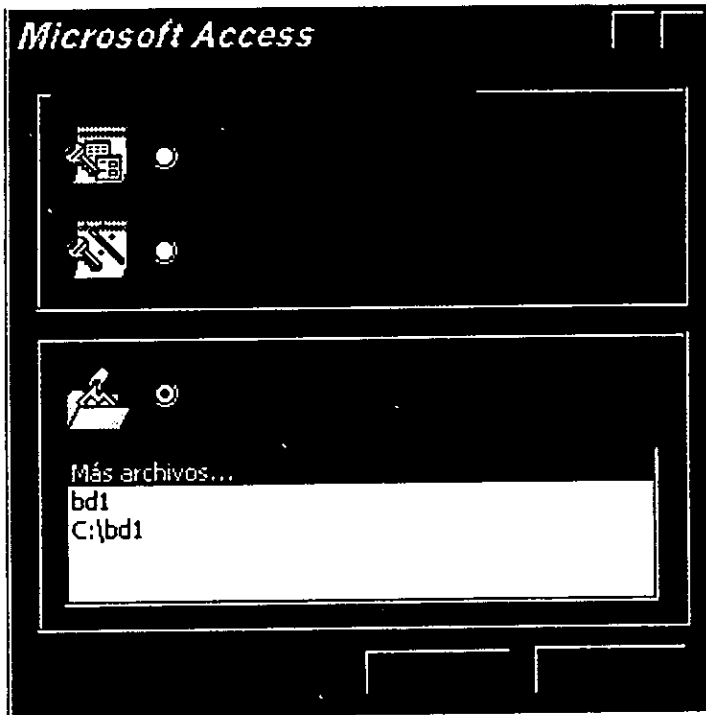


Figura 1.9.6. Ventana de acceso al asistente de bases de datos.

## **Asistentes, complementos y generadores en Microsoft Access 97**

<b>Asistente</b>	<b>Descripción</b>
Asistente para gráficos	Agrega un gráfico a un formulario o informe basado en los datos en una tabla o consulta.
Asistente para cuadros combinados	Crea un control de cuadro combinado en un formulario.
Asistente para botones de comando	Crea un control de botón de comando en un formulario.
Asistente para resolver conflictos	Resuelve conflictos entre bases de datos replicadas en el momento de la sincronización.
Asistente para consultas de tablas referencias cruzadas	Crea una consulta que resume datos en de un formato compacto parecido a una hoja.
Asistente para dividir bases de datos	Divide bases de datos en las partes de datos e interfaz, de forma que uno o varios usuarios puedan tener copias locales de la interfaz conectadas a los datos en un servidor.
Asistente para bases de datos	Crea una base de datos completamente nueva para diversos usos basándose en 22 modelos incorporados.
Asistente para exportación de texto	Exporta datos a un archivo de texto.

<b>Asistente para consultas de buscar duplicados</b>	Crea una consulta que busca registros de valores de campo duplicados en una única tabla o consulta.
<b>Asistente para consultas de buscar no-coincidentes</b>	Crea una consulta que busca registros en una tabla que no tienen registros relacionados en otra tabla.
<b>Asistente para formularios</b>	Crea un nuevo formulario.
<b>Asistente para importación de HTML</b>	Importa tablas y listas HTML desde Internet o una intranet a una tabla de Microsoft Access.
<b>Asistente para importación de una hoja de cálculo.</b>	Importa una hoja de cálculo de Microsoft Excel o otro tipo a una tabla de Microsoft Access
<b>Asistente para importación de texto</b>	Importa un archivo de texto a una tabla de Microsoft Access.
<b>Asistente para máscaras de entrada</b>	Crea una máscara de entrada para un campo que elija en una tabla.
<b>Asistente para etiquetas</b>	Crea etiquetas postales en tamaños estándar y personalizados.
<b>Asistente para vinculación de HTML</b>	Vincula una tabla o lista HTML en Internet o una intranet a una tabla de Microsoft Access.
<b>Asistente para vinculación de hojas de cálculo.</b>	Vincula datos de una hoja de cálculo a una tabla de Microsoft Access.



Asistente para vinculación de texto	Vincula un archivo de texto a una tabla de Microsoft Access.
Asistente para cuadros de lista	Crea un control de cuadro de lista en un formulario.
Asistente para búsquedas	Crea una columna de búsqueda en una tabla, que presenta una lista de valores entre los que el usuario puede elegir.
Asistente para combinar correspondencia con Microsoft Word	Administra operaciones de combinación de correspondencia utilizando cartas almacenadas en Microsoft Word y direcciones almacenadas en Microsoft Access.
Asistente para grupos de opciones	Crea un grupo de botones de opción en un formulario.
Analizador de rendimiento	Analiza la eficiencia de una base de datos y produce una lista de sugerencias para mejorar su rendimiento.
Asistente para tablas dinámicas	Coloca una tabla dinámica de Microsoft Excel en un formulario de Microsoft Access.
Asistente para publicar en el Web	Crea documentos HTML estáticos o dinámicos a partir de su aplicación de Microsoft Access para ser colocados en Internet o una intranet.
Asistente para informes	Crea un informe basado en una tabla o consulta.

Asistente para consultas sencillas	Crea una consulta de selección a partir de los campos que usted elige.
Vinculador de campos de un subformulario	Vincula los campos en un formulario principal, en un informe principal y en un subinforme.
Asistente para subformularios y subinformes	Crea un nuevo subformulario o subinforme en un formulario o informe.
Asistente para analizar tablas	Toma una tabla con muchos datos duplicados y la divide en tablas relacionadas para un almacenamiento más eficiente.
Asistente para tablas	Crea una nueva tabla.
Asistente para seguridad a nivel de usuario	Crea una nueva base de datos codificada, con acceso de usuarios regulado a partir de una base de datos existente.
Asistente para publicación en el Web	Este asistente es llamado por el Asistente para publicar en el Web y coloca los documentos HTML en un servidor de Internet intranet.

## **Generador**

## **Descripción**

Generador de colores	Presenta una paleta para crear colores personalizados.
----------------------	--

Generador de expresiones	Crea expresiones para macros, consultas y hojas de propiedades.
Generador de campos	Crea campos en tablas.
Generador de cadenas de conexión	Crea la sintaxis correcta para la conexión ODBC a una base de datos ODBC.
Generador de imágenes	Crea imágenes de mapa de bits para formularios e informes.
Generador de consultas	Crea la sintaxis correcta para una consulta.

### **Complemento**

### **Descripción**

Administrador de complementos	Instala y desinstala asistentes, generadores y complementos.
Administrador de tablas vinculadas	Administra los adjuntos a tablas entre bases de datos.
Administrador de Paneles de control	Crea y administra formularios del Panel de control para aplicaciones.

### **Las macros y los módulos**

Microsoft Access está también pensado para aquellos que quieren realizar programas para ellos mismos o para terceros. Las dos herramientas que proporciona Access para llevar a cabo esta programación son las macros y los módulos.

Una macro tiene la función de repetir de forma automática un conjunto de operaciones que se usen con frecuencia, de este modo se ahorra gran cantidad de tiempo.

Un módulo es un conjunto de procedimientos escritos en Visual Basic para aplicaciones. Visual Basic para aplicaciones es un lenguaje de programación visual que se puede utilizar con Access y con el resto de aplicaciones de Microsoft Office.

### **Diferencia entre tabla y base de datos**

Para Access una tabla es una base de datos simple, sin embargo, llama base de datos al conjunto de tablas que están relacionadas entre sí y a todos los "objetos" (consultas, formularios, etc.) relativos a dichas tablas.

### **Otras herramientas**

La creación y modificación de los distintos objetos de una base de datos puede ser algo complicada, según se trate de una tabla, una consulta o un informe. Sin embargo la mayoría de estos objetos disponen de dos herramientas que facilitan las operaciones de creación y diseño: los asistentes y los generadores.

Los generadores tienen la misión de simplificar una tarea concreta, como puede ser la creación de una expresión o la inclusión de un control (un campo) en un formulario.

Los asistentes también simplifican algunas tareas, pero lo hacen interactuando con el usuario mediante una serie de preguntas y respuestas, entre los asistentes que hay destacan los que sirven para crear tablas, consultas, formularios e informes.

## **Características de Access 97**

El asistente para bases de datos genera automáticamente tablas, consultas, formularios e informes con mas de 20 tipos de plantillas completas, que le dan la opción de personalizarlas para su uso posterior para que encajen en sus necesidades.

El asistente para consulta sencilla, ordena la información de la base de datos, incluyendo datos de varias tablas, y después determina como se agrupan juntos para responder a sus preguntas.

Filtro por selección, filtra sus datos, de acuerdo a los datos que se resalten, presentando solo la información que corresponda a sus elecciones.

Filtro por formulario, le permite escribir el criterio de búsqueda en formularios con los que ya esta familiarizados, y después presenta solo los datos que coinciden con sus parámetros.

Tipo de dato hipervínculo es una de las primeras bases de datos de escritorio que admite el almacenamiento de hipervínculos como tipo de dato nativo. Por ejemplo, un nombre de candidato que se almacena en una base de datos de candidatos a un trabajo, se puede convertir en un hipervínculo permanente al resumen del candidato.

Hipervínculos en documentos de Microsoft Office admite hipervínculos que van a direcciones URL, así como a ubicaciones de otros documentos de Microsoft Office 97.

Se pueden construir bases de datos completas mediante los hipervínculos como mecanismo para conectar objetos.

Guardar en formato HTML, los usuarios pueden compartir vistas estáticas de sus datos en el WEB, haciendo clic en el comando guardar como formatos HTML/WEB del menú archivo. Microsoft Access proporciona la salida de una tabla, consulta, hoja de datos de formulario así como informes con formato directamente en formato HTML.

Salida en formato IDC/HTX influido por la funcionalidad nativa de conector a base de datos de Internet de Microsoft Internet, Information Server y personal Web Server, Access 97 proporciona una manera fácil para que los usuarios compartan sus datos estructurados en un grupo de trabajo o sobre Internet.

Consultas con parámetros extiende este paradigma a los resultados de Web de un origen de datos que no es solo dinámico, sino también interactivo. Los usuarios pueden solicitar solo la información relevante para ellos y se puede estar seguro de que reciben las respuestas mas actualizadas a sus consultas.

Asistente para publicar en el Web, el asistente para publicar en el Web, permite a los usuarios publicar cualquier objeto de su base de datos de forma estática o dinámica.

Permite el formato HTML personalizado mediante plantillas y permite a los usuarios guardar la configuración que se utiliza para mostrar los objetos.

Importación y vinculación de HTML, los usuarios pueden apuntar a un documento HTML que contiene una tabla de datos y el asistente para importar/exportar lee la información y la importa directamente a una nueva tabla o agrega los registros a una tabla ya existente.

### **Características mejoradas**

Asistente para analizar tablas, identifica automáticamente las relaciones y datos estructurados, como por ejemplo una base de datos en formato de archivo plano, y después reorganiza la información en una base de datos relacional.

Asistentes para formularios e informes mejorados, genera formularios e informes, adaptando específicamente las opciones que ofrecen para acomodar los datos que se planea incluir.

Rendimiento de 32 Bits mejorado, hace que Microsoft Access 97 sea mucho más rápido y más sensible con formularios más pequeños, mejor compilación y manipulación de datos más rápida con 32 bits.

Microsoft Visual Basic for applications, proporciona un completo entorno de desarrollo integrado que incluye arrastrar y soltar código, sintaxis codificada en colores, y ventana de depuración mejorada, exploración de objetos en su sitio, junto con soporte para controles ActiveX, y vínculos fáciles a las aplicaciones de Microsoft Office.

Asistente para analizar el rendimiento, analiza su base de datos, sugiere las mejores formas de optimizar su velocidad y rendimiento, y después puede realizar automáticamente, los cambios necesarios conforme se requieran.

Bandas de comando programable, le proporciona un control completo y personalizado de la interface de Microsoft Access permitiéndole combinar las barras de herramientas y menús que necesite en un nuevo tipo de objeto que se comparte en todas las aplicaciones de Microsoft Office 97.

Código fuente removible, le permite eliminar código fuente de las soluciones empresariales que cree y reduce su tamaño, haciéndolas funcionar mas rápidamente y proteger su propiedad intelectual. Exploración jerárquica de objetos Access 97, incluye un explorador de objetos que permite a los programadores explorar rápidamente una jerarquía de objetos para buscar información.

El explorador distingue entre las propiedades incorporadas, las propiedades personalizadas, los métodos, los controladores de eventos y los procedimientos definidos por el usuario.

Módulos de clase, los nuevos módulos de clase de Microsoft Access 97 pueden contener la definición de un nuevo objeto. Cuando los usuarios crean una nueva instancia de una clase, crean el nuevo objeto. Cualquier procedimiento definido en el

módulo se convierte en las propiedades y métodos del objeto.

Formularios con varias instancias, los programadores pueden crear con código varias instancias del mismo formulario, permitiendo la ramificación dentro de las aplicaciones.

Cada formulario mantiene su propio registro subyacente y puede actualizar sus datos de manera independiente a los demás datos.

Compilación condicional, los programadores pueden establecer etiquetas de compilación dentro de su código para controlar el comportamiento resultante de la aplicación. Esto permite a los programadores crear compilaciones de prueba y venta al menudeo de sus aplicaciones estableciendo etiquetas en su código.

Control de imagen, un control de imagen proporciona una manera sencilla de incluir información gráfica en formularios o informes y mejora el rendimiento en la visualización de la imagen. Anteriormente, las imágenes se mostraban mediante OLE, que incluía pasos innecesarios.

Conversión automática, opción que permite a los usuarios migrar desde bases de datos ya existente de Microsoft Access a la versión mas reciente. Conforme los usuarios abren una base de datos creada en una versión anterior, se les pregunta si desean convertirla a formato actual.

Reaplicación parcial de tablas, pionera en la reaplicación de base de datos de escritorio, en la cual la reaplicación de objetos de base de datos a través de una red corporativa es fácil. Los usuarios pueden especificar subconjuntos de sus datos para replicar. Proporciona extensiones para la reaplicación parcial de tablas mediante objetos de acceso a datos.

Objeto y controlador de automatización, para automatizar la tarea de bases de datos comunes de unidades de Access o que se dirijan desde cualquier aplicación, que



admita Automation. Los programadores pueden tomar el control mediante código de la funcionalidad del producto, como por ejemplo el motor de informes, desde fuera de Microsoft Access.

Control calendario, incluye un control de calendario OLE de 32 bits que se pueden usar para seleccionar fechas rápidamente, sin ningún código. El usuario arrastra este control a un formulario, como una manera fácil de seleccionar fechas. Los cambios realizados, mediante el calendario se almacenan en la tabla subyacente.

Controles Active Xnns de tipo Databound, el contenedor se ha actualizado para que admita los recientes controles ActiveX producidos con el sistema de desarrollo Control development Kit de Microsoft Visual C++.

Cuadros de lista con selección múltiple, los controles nativos de cuadro de lista de Microsoft Access, se pueden seleccionar de manera sencilla o múltiple.

Control de código fuente, admite la integración de varios proveedores de control de código fuente, incluyendo Microsoft Visual Source Safe. Este soporte esta disponible como un complemento de la versión Office 97 Developers Edition.

### **Creación de relaciones con Access 97**

Access 97 cuenta con la opción de establecer las relaciones. Al tener ya abierta una base de datos, se ejecuta el comando Herramientas-Relaciones (figura 1.9.7.).

### **Visual Basic 5.0 con Access 97**

En cuanto al volumen de datos se puede trabajar con bases de datos con casi un millón de registros, y el rendimiento es muy bueno.

En Access se debe tener en cuenta que los campos por los que se elaboren los Querys deben estar indexados para acelerar las consultas.

Los Querys se pueden crear en Access y luego pegar en Visual, ligándolos bien sea a un Data o a un Recordset. Los resultados son muy buenos, ya que solo se hace un llamado al Jet de Access y no es necesario cargar el Access pues resulta muy pesado. Al trabajar con Visual 5.0/Access 97, los programas corren mucho más rápido, todo esto aunado a las características, ventajas y desventajas anteriormente descritas de ambos productos.

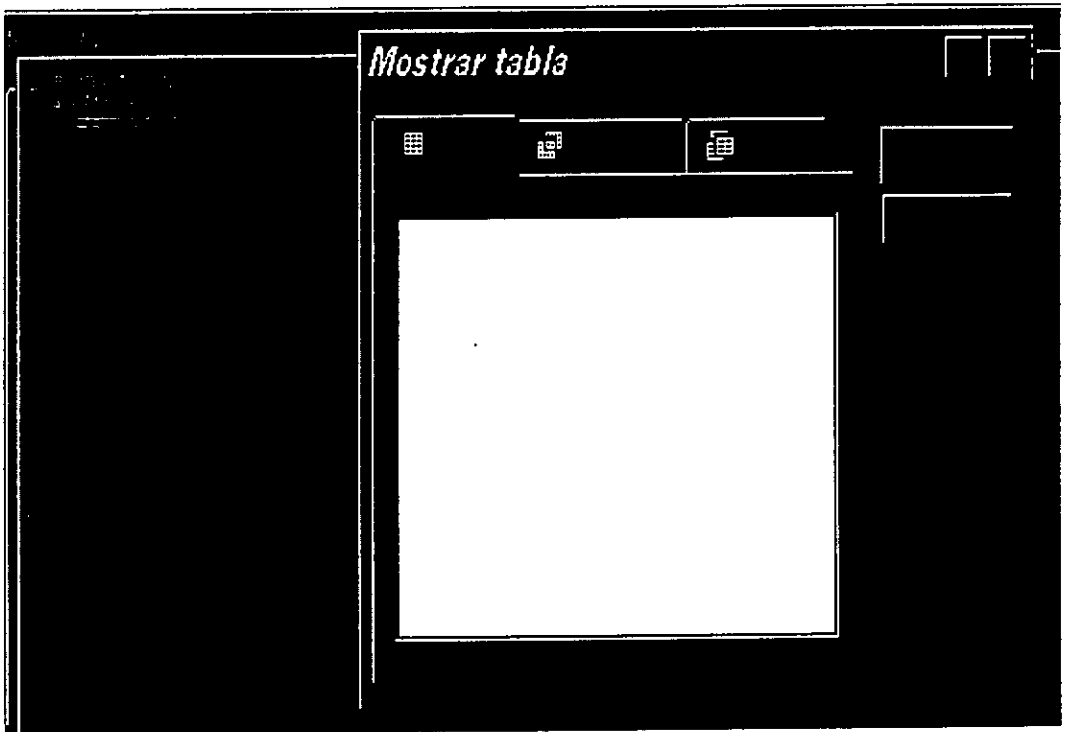


Figura 1.9.7. Establecer relaciones con Access 97.

## El diccionario de datos en Access 97

El diccionario de datos se puede generar también por medio de un asistente de Access (figura 1.9.8. y 1.9.9.).

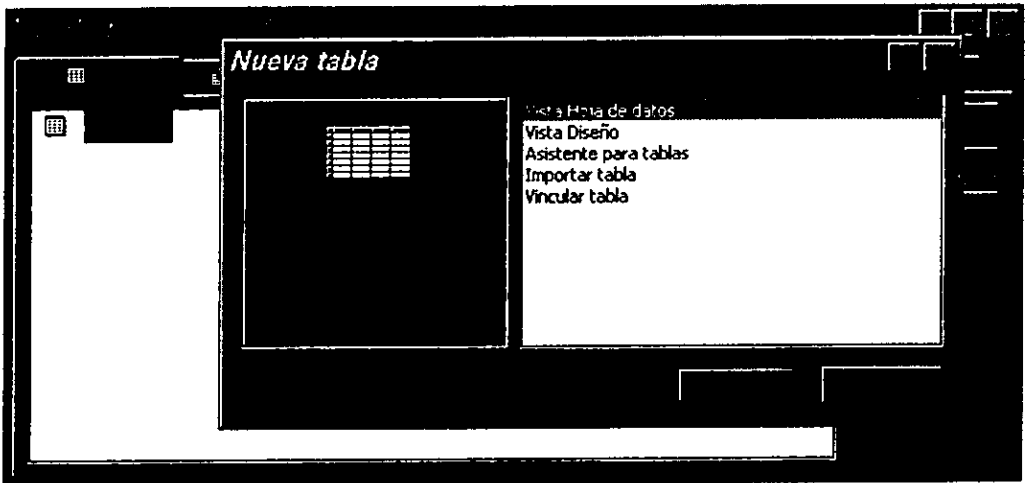


Figura 1.9.8. Creación del Diccionario de datos en Access 97.

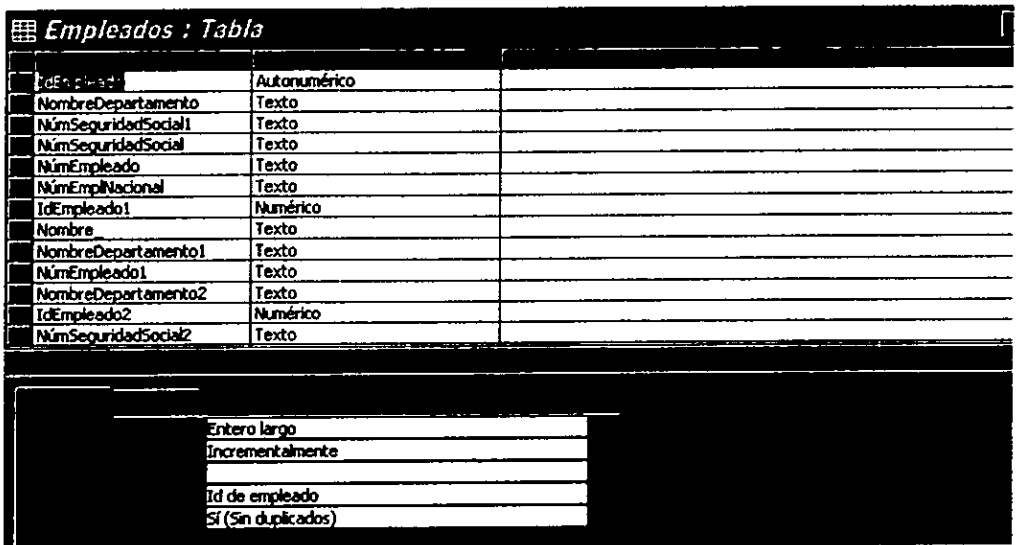


Figura 1.9.9. Creación del diccionario de datos en Access 97.

## **CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN**

### **2.1. Situación actual**

La empresa telefónica, está dividida geográficamente en 10 direcciones divisionales, cada una de ellas, se encarga de atender las necesidades de su región. Dentro de las funciones básicas de una dirección, esta la de atender las solicitudes de los clientes que requieren la instalación del servicio, así como mantener el adecuado funcionamiento de la planta telefónica instalada para reducir al mínimo las fallas que puedan existir.

Para esta operación, es necesario tener materiales suficientes y a la mano para proporcionarlos al personal encargado de realizar estas tareas. Esta situación, la empresa la ha solventado, creando almacenes en los centros de trabajo estratégicos; en la actualidad cuenta con 154 de estos almacenes llamados auxiliares repartidos en las 10 divisiones. Estos almacenes auxiliares, son abastecidos de material a través de 10 grandes almacenes concentradores denominados distribuidores, que son los que reciben el material directamente de los proveedores. Los almacenes se encuentran localizados geográficamente en las divisiones tal como se muestra en la Figura 2.1.1. En la tabla de la figura 2.1.2. se indican los almacenes distribuidores que atienden a cada división.

Los materiales se encuentran divididos en 2 grandes grupo: materiales misceláneos y materiales estratégicos; estos últimos (Cables y Postes) son los que implican mayor costo y son de uso exclusivo de la empresa telefónica, los otros son todos los demás materiales que pudieran requerirse en la planta telefónica.



**Figura 2.1.1. Ubicación de Almacenes Distribuidores.**

División	Almacén Distribuidor
Noroeste	22 Hermosillo
	23 Mazatlán
Norte	13 Chihuahua
Noreste	11 Monterrey
Centro	32 Celaya
Occidente	21 Guadalajara
Metro Oriente	07 La Perla (material misceláneo)
Metro Poniente	
Metro sur	08 Viveros (material estratégico)
Pacífico Golfo	41 Puebla
Sureste	43 Villahermosa

**Figura 2.1.2. Almacenes Distribuidores que atienden a las divisiones.**

Adicional a las direcciones divisionales, existen áreas corporativas que dan servicio de administración a estas divisiones y dictan las normas para una operación estandar de toda la empresa. En esta área corporativa, se encuentra el área de Control de Inventarios, el área de Compras así como la Contabilidad de Planta y Auditoría, entre otras, este esquema se muestra en la figura 2.1.3.

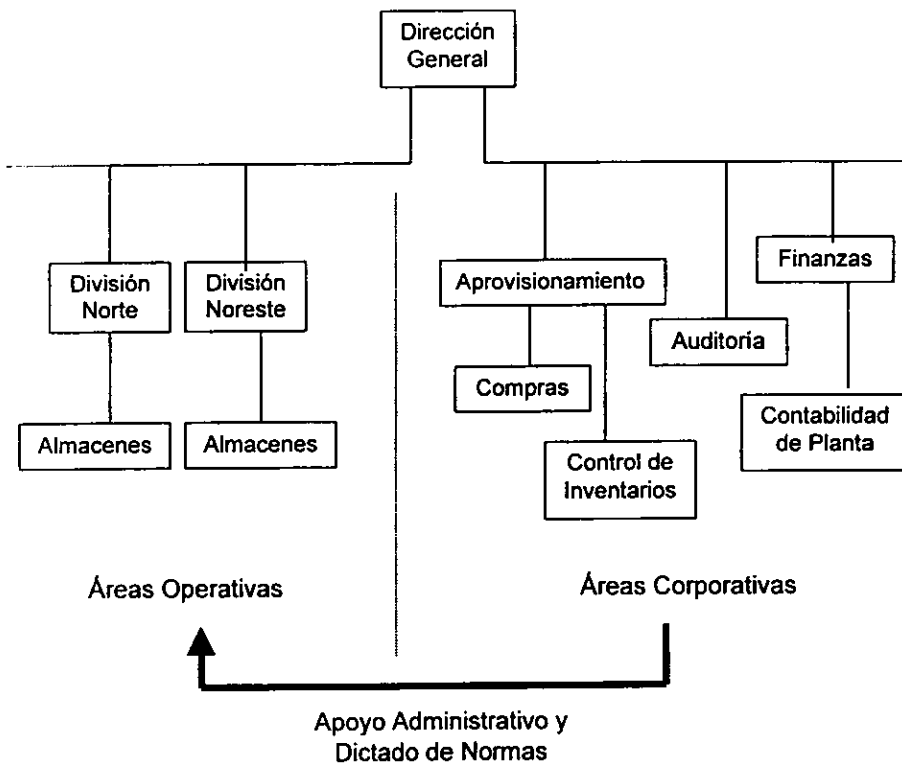


Figura 2.1.3. Esquema de relación entre Áreas Operativas y Corporativas.

El área de Control de Inventarios es responsable de determinar la cantidad de material que debe comprarse de cada artículo, de tal forma que se tenga el material en forma oportuna para el funcionamiento de la planta, cuidando que los niveles de inventario no provoquen excesivos costos de almacenaje; los costos los determina y administra el área de Contabilidad de Planta y el área de Auditoría es quien supervisa estas tareas para que se realicen con transparencia y según procedimientos. Las cantidades que se determinan comprar para cada almacén distribuidor son informadas al área de compras para que esta elabore los pedidos correspondientes; estas deben de tener una proyección de varios meses para poder aprovechar las ventajas de compra por volumen, pero con entregas parciales programadas para no sobre-inventariar a las divisiones.

El nivel de inventario de toda la empresa, se mide tomando en cuenta la existencia en los almacenes de todas las divisiones, tanto auxiliares como distribuidores; el área de Control de Inventarios, es responsable de mantener un equilibrio óptimo entre las existencias de las divisiones, determinando que material debe ser traspasado de una división a otra para evitar compras innecesarias. El nivel de inventario de cada división lo determina la misma área de control de inventarios y se mide tomando en cuenta la existencia en todos sus almacenes, tanto auxiliares como distribuidores, los coordinadores de logística de cada división, son los encargados de mantener un nivel de existencias óptimo entre los almacenes de su división, retirando material que no se utiliza en ciertos auxiliares para concentrarlo en el distribuidor y tenerlo disponible no solo para otro almacén en su división, sino también disponible para las demás divisiones, pero existen casos en los que el material que se tiene en los almacenes auxiliares, no se puede considerar como disponible dado que están bajo la supervisión y control de las áreas operativas por lo que no se puede concentrar en un distribuidor ni tomar como disponible para otras divisiones; esto debe ser tomado en cuenta por Control de Inventarios, para no solicitar traspasos de estos materiales, ya que la respuesta será negativa y por consiguiente una falta de material en otra división. En la figura 2.1.4. se muestra gráficamente esta situación.

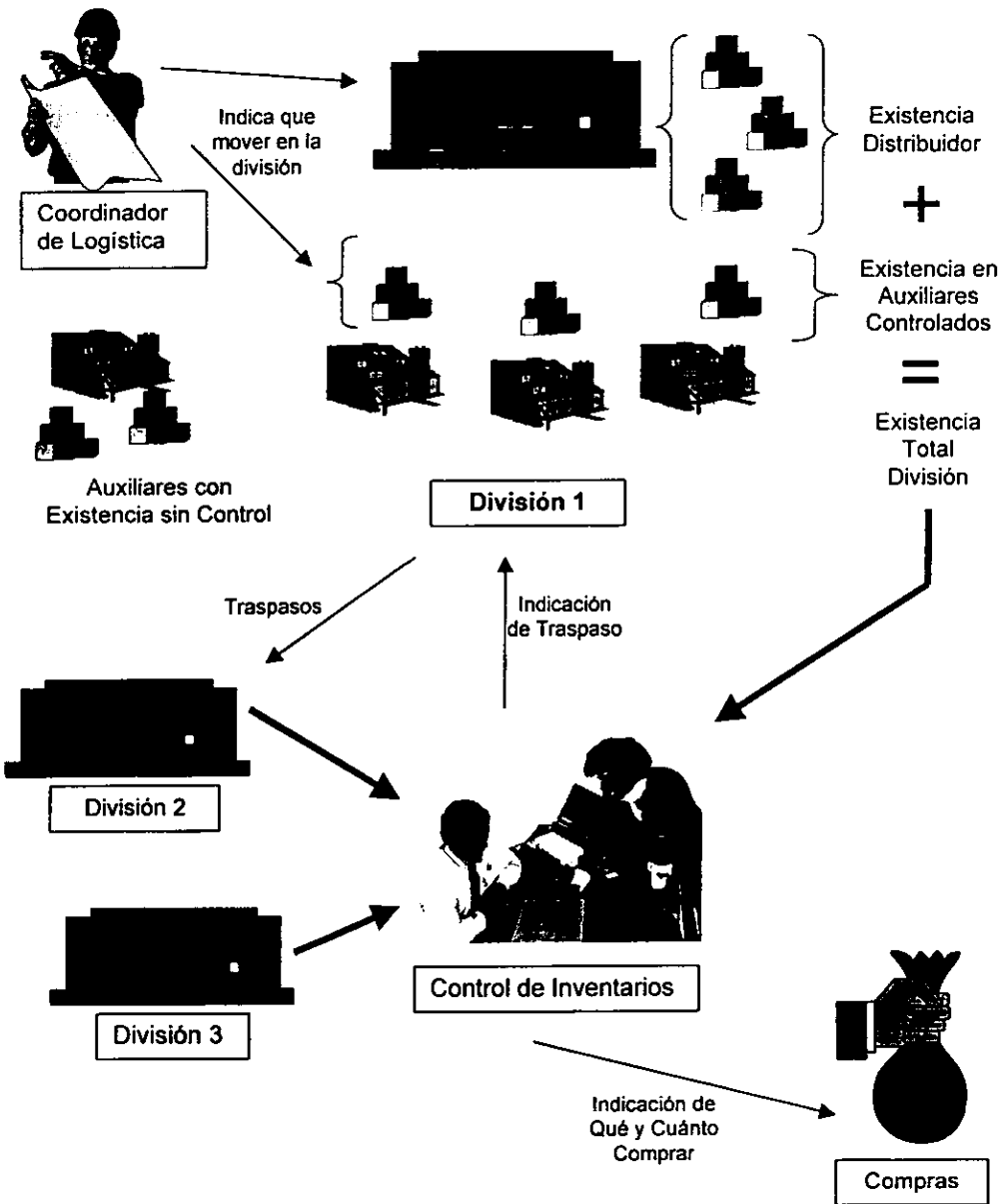


Figura 2.1.4. Gráfica de la disponibilidad de Existencias en las Divisiones.



Los trasposos que determina Control de Inventarios, deben ser planeados y programados, de tal forma que no se incurra en gastos indebidos de transporte o paquetería, es decir, solicitar todo lo que se tenga que mover de una división, en un solo evento, para que se mueva de la misma forma, porque si se solicita un día un material y tres días después otro material al mismo destino, se pagaría doble flete y los camiones solo se llenarían parcialmente.

Actualmente para realizar estas funciones, el área de Control de Inventarios, cuenta con varios procesos aislados el diagrama de la situación actual con la que se realiza la tarea, se muestra en la Figura 2.1.5.

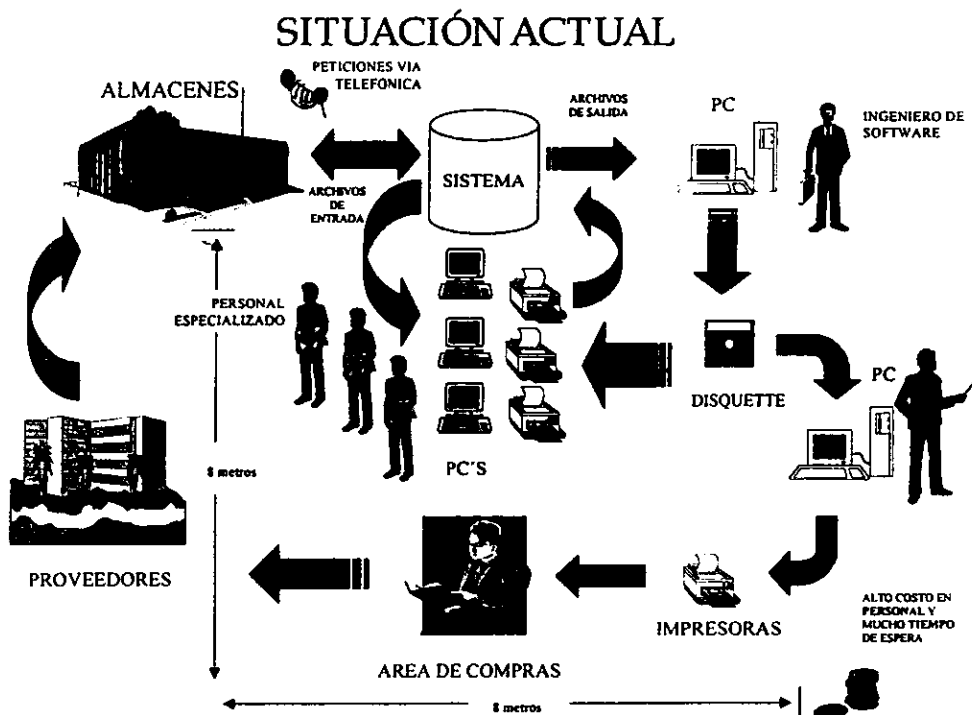


Figura 2.1.5. Diagrama de las Actividades actuales en Control de Inventarios.

En esta figura 2.1.5. se observa que las operaciones de entradas y salidas de los almacenes se registran diariamente en un Sistema Central de Recursos Materiales (SIREM), en este mismo sistema el área de compras captura los pedidos fincados para que puedan ser registrados por los almacenes, teniéndose un control de los saldos de estos pedidos. Este sistema da la facilidad de extraer la información de existencias, pedidos y movimientos, cada mes; la información es extraída hacia una PC en formato dbf y a través de programas de Dbase III, se actualiza un archivo histórico del cual se extrae toda la información de consumos para generar un consumo promedio por almacén y por división, esta actividad es desarrollada por una persona especialista, que prácticamente dedica todo su tiempo a este proceso y a mantener actualizado el catálogo de materiales.

Para generar los consumos promedio, se clasifican los movimientos de los almacenes, ya que existen algunos que no deben ser tomados en cuenta; los movimientos que pueden ser realizados en los almacenes son los siguientes:

- *Entradas por proveedor:* Se registran las entregas que hacen los proveedores a los almacenes, dado que es un ingreso debido a una demanda, **NO AFECTA LOS CONSUMOS.**
- *Entradas por devolución de usuario:* Estos movimientos son por material que el usuario regresa al almacén por no ser utilizado, como previamente salió y se registró como consumo, **SE RESTA DE LOS CONSUMOS.**
- *Entradas por devolución de venta:* Las ventas que realiza la empresa, son derivadas de un excedente de material, por lo que no se consideran consumos, cuando una de estas ventas es devuelta al almacén, de igual forma **NO AFECTA LOS CONSUMOS.**
- *Entrada por traspaso:* Los movimientos inter-almacenes, **NO SE CONSIDERAN CONSUMOS,** ya que lo único que se realiza es el movimiento de un material de un

almacén a otro, sin que esto implique una disminución o aumento en la contabilidad de la empresa.

- *Salida a usuario:* Esta es la principal actividad que se tiene en los almacenes, y es para la operación normal de la empresa, por tanto SE CONSIDERA UN CONSUMO.
- *Salida a venta:* Como se indicó en las entradas por devolución de venta, estas ventas se realizan para deshacerse del material que se tiene en exceso, si se tomaran como consumo se obligaría a comprar mas material, por tal motivo NO AFECTAN LOS CONSUMOS.
- *Salidas por traspaso:* Las salidas por traspaso, son los movimientos de material que se realizan internamente en la empresa, sin que se descargue el material del inventario, dado que no está saliendo realmente este material, NO AFECTAN LOS CONSUMO.
- *Ajustes:* Para mantener una conciliación entre las existencias físicas con los registros contables, se realizan ajustes que pueden ser por merma del material, pérdida del mismo, entregas inexactas, etc. estos movimientos NO AFECTAN LOS CONSUMOS.

En la figura 2.1.6. se representan todos los movimientos que registra un almacén y que afectan el inventario del mismo, en ella se observa cuáles sí se consideran para los cálculos de consumo y cuáles se omiten.

Una vez obtenido el archivo de consumos promedio, proceso que tarda una semana en realizarse, este es turnado a tres analistas especializados en manejo de hojas de cálculo y bases de datos junto con los archivos de existencias y pedidos, para que determinen las cantidades a comprar.

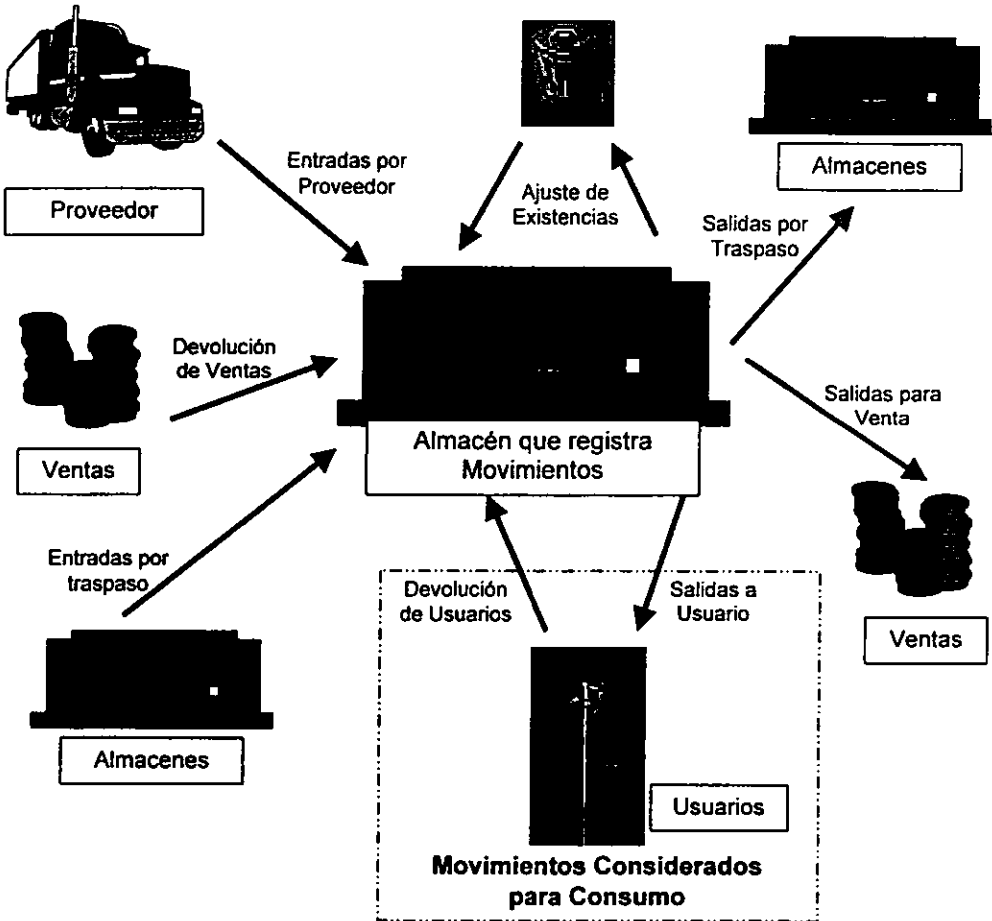


Figura 2.1.6. Movimientos que realiza un Almacén y los que son Consumos.

En el catálogo de materiales, existen mas de 5000 registros, de los que se tienen vigentes 2000, los cuales pueden tener necesidades y/o existencia en cualquiera de los 164 almacenes; para simplificar el trabajo, y poder tener respuestas a tiempo, se tienen

clasificados los materiales de la siguiente forma:

- Refacciones de Telefonía Pública: 100 materiales.
- Cables telefónicos: 250 materiales
- Misceláneos con alto valor: 200 materiales
- Menos costos: el resto de los materiales.

Cada uno de los primeros 3 grupos, los atiende cada analista, generando diferentes procesos en dBase III, para obtener archivos que posteriormente se pasan a una hoja de cálculo de Excel, en esta hoja de cálculo, se obtiene la información por diferencia, de las necesidades o sobrantes por mes una vez sumados los pedidos de cada uno de esos meses y las existencias que se tienen en él, o los almacenes que son tomados en cuenta en cada división.

En forma manual, se revisa cada material, para determinar si el excedente que se tiene, puede ser traspasado a otra división que tenga un faltante, y si el movimiento es entre almacenes relativamente cercanos, para no encarecer el traslado.

Una vez detectado el material que se puede traspasar, se recalcula la hoja de cálculo, para obtener un nuevo faltante por mes. Estas cantidades resultantes como faltantes, se trabajan en una hoja aparte para obtener una relación de solicitudes, la cual se envía físicamente al área de compras para que elabore el pedido.

Periódicamente el encargado de cada grupo de materiales, revisa en el SIREM que estén los pedidos, a fin de detectar posibles retrasos en el área de compras que pudieran traducirse en falta de material.

El último grupo de materiales, se trata de una manera más general, uno de los 3 analistas, genera un reporte semestral que contiene los consumos promedio comparados con las existencias de cada división, para comunicar a compras los meses de consumo que se tienen, el área de compras es la encargada de generar los pedidos para todo el semestre, según se requiera tomando los meses de consumo reportados.

Cuando existe algún requerimiento adicional de todos estos materiales, es entonces cuando otra persona, verifica las existencias en las demás divisiones, y solicita los trasposos necesarios o genera solicitudes de pedido a compras, para cubrir el resto del semestre.

Esta última actividad, lejos de ser esporádica, es muy frecuente (múltiples llamadas diarias), y se cae en problemas tales como la no atención del trasposo respectivo, y el retraso en el surtido del proveedor dada la premura como se hace la solicitud de pedido.

Estos materiales, no tienen un costo individual muy alto, pero en conjunto significan un porcentaje elevado del inventario.

También en forma esporádica, se generan listados de sobre inventario, los cuales generan una carga de trabajo extra para determinar que pedidos deben ser cancelados o que trasposos deben ser generados para nivelar las existencias; también se traduce en un excesivo trabajo para los almacenes distribuidores, ya que además del surtido normal que no debe detenerse, deben realizar estos trasposos de nivelación.

## 2.2. El usuario y sus requerimientos

Los requerimientos que presenta el usuario para la atención de estas necesidades, son los siguientes:

- Desea que la información proporcionada por el SiREM, sea concentrada en una sola computadora, en la que se realicen todos los procesos necesarios para la generación de las listas de solicitudes de pedido, a través de un sistema interactivo.
- El sistema debe ser simple, para que no sea necesario tener expertos en computación que lo operen, sino que cualquier persona con conocimientos básicos de computación lo pueda hacer.
- El ambiente que más se maneja en el área es un ambiente Windows (de ventanas), por lo que el sistema debe tener una interface gráfica que permita al usuario operarlo con facilidad.
- Dado que se requiere una interface gráfica, las pantallas del sistema, deben poderse operar con un mouse y mediante la selección de iconos en cualquiera de las opciones que tenga.
- Dado que el sistema podrá ser operado por una persona no especializada, debe presentar opciones de configuración exclusivas para un administrador que sea el que prepare el ambiente en el que el operador deba trabajar.
- El sistema debe ser capaz de almacenar la suficiente información histórica de hasta un año, para poder seleccionar el número de meses con los que se quiera hacer el pronóstico o cálculo de consumos requerido. Es decir, debe proporcionar la facilidad de escoger cuantos meses hacia atrás se requieren tomar para el cálculo del promedio futuro.

- Independientemente de la información que tenga en línea, debe dar opciones de respaldar en archivos la información que se desee borrar o que tenga mas del tiempo requerido para los cálculos.
- También debe ser capaz de generar diferentes bases de datos, para trabajar en forma separada la información o realizar pruebas sin afectar los datos reales.
- Como la información fuente está en un equipo independiente, y los archivos se deben extraer a través de la red de datos, es necesario que el sistema proporcione la opción de hacer la transferencia automática sin la necesidad de que el operador conozca de redes o de programas de transferencia de datos.

El objetivo que se persigue es el de disminuir los tiempos de proceso, personal especializado, optimización del proceso para abarcar el 100% de los materiales con los que se cuenta y con esto una disminución sustancial en el costo, tanto del proceso, como del inventario de la empresa. Esto se esquematiza en la figura 2.2.1. que al compararla con la 2.1.3. se observa que se lograrían estos objetivos.

Otros requerimientos que tiene el usuario para el sistema que se propone, se describen a continuación:

- La información básica que se requiere tener de salida, (reporte de necesidades), debe poder ser completamente configurable, los materiales que presente deben poderse seleccionar una vez para que se pueda ejecutar cuantas veces se requiera sin necesidad de tener que volverlos a seleccionar.
- Dado que la información que se requiere puede ser variable, según el análisis o decisión que se necesite tomar, el sistema debe proporcionar la opción de presentar o no algunas columnas: a veces se requerirá solo ver los pedidos, o solo las existencias, o ver determinado número de meses hacia el futuro.





Figura 2.2.1. Situación Propuesta.

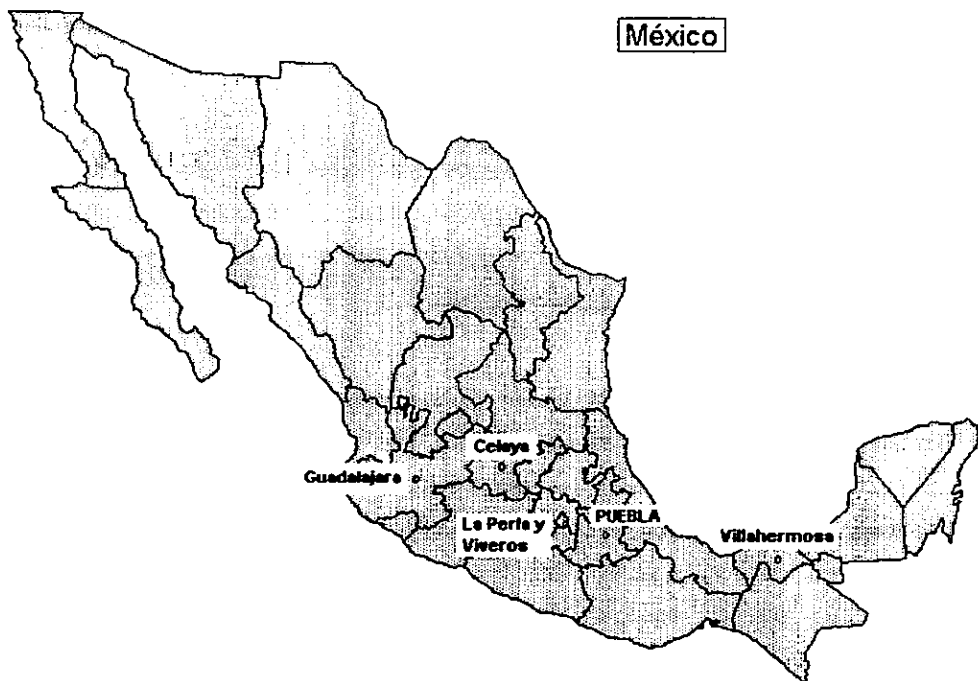
- El reporte principal debe presentar el panorama de los materiales seleccionados en un determinado tiempo hacia el futuro, ya sea que falte o que sobre material. Deberá ser posible determinar cuantos meses hacia delante se requieren analizar.
- Para atender los requerimientos adicionales planeados o no, el sistema debe proporcionar la facilidad de capturar la información que se requiera adicional al cálculo de consumo promedio y de igual forma permitir tomarla en cuenta o no para los reportes que se configuren.
- La información que llega del SIREM con respecto a las existencias y pedidos,

deberá ser registrada por versiones, es decir, conservar las anteriores hasta que se decida eliminarla o respaldarla, para dar la facilidad de generar reportes con diferentes fechas de análisis.

- Adicionalmente, se deberán presentar una serie de consultas y reportes sobre la información cargada o capturada.
- En algunas ocasiones, se tienen varios materiales que cubren una misma necesidad, por lo que el sistema deberá ser capaz de considerar estos como un conjunto de materiales, de tal forma que consolide la información presentándola como una sola necesidad de un solo material.
- No todos los consumos que se tienen en la planta telefónica son constantes, se tienen proyectos específicos de construcción, que requieren una lista de materiales en cantidades específicas según un factor estadístico; el sistema deberá calcular las necesidades de material para construcción generando una explosión de materiales en función a los factores de esta lista. Para esto, el reporte podrá configurarse de tal forma que permita escoger entre atender necesidades de mantenimiento, construcción o ambas.
- Un proceso posterior que se requiere en el sistema, es el de determinar una vez obtenido el reporte de faltantes y sobrantes, es el de determinar los trasposos factibles entre divisiones, para ello, debe permitir que se le indique en que orden pueden hacerse trasposos de una división a otra, y cuales de estos no son factibles. Con esta información, debe generar un reporte de las cantidades que se puede traspasar en cada mes a fin de nivelar las existencias según el resultado obtenido. Este reporte le indicará al usuario que trasposos de todos los materiales seleccionados se deben realizar en una misma emisión para optimizar los recursos de transporte.

En la figura 2.2.2. se observa un mapa en el que se especifican los distribuidores

que son factibles de traspasar a la división Pacífico-Golfo, esto se da por la distribución geográfica de la misma y la facilidad de acceso a ella.



**Figura 2.2.2. Regiones Factibles para traspasar a PUEBLA.**

- Al momento de generar los traspasos, se deberá generar el reporte de lo no cubierto por cada mes en cada material, en cada división, el cual servirá para comunicar al área de compras las cantidades que deben comprarse, en que fecha, a que destino y de que material.
- Al generar los reportes, de deberá poder escoger sobre que versiones de cada uno de los archivos se desea trabajar.
- Con la información que se tiene de los pedidos, el sistema también podrá ser capaz

de generar una lista de los pedidos que se encuentran con fecha de entrega atrasada, a fin de dar una herramienta al área de compras para que presione a los proveedores respectivos.

Todos estos requerimientos, son para que el sistema que se implante, facilite el trabajo optimizando los recursos, de tal forma que se requiera de menor especialización en el personal ya que el sistema debe realizar todos los procesos mediante una selección de iconos; menos equipos de cómputo dedicados a procesos, al estar en una sola máquina todo el proceso, se evite estar pasando archivos con diskettes a otras máquinas. Esto también permite que toda la información y resultados del sistema se encuentren en una misma base de datos teniendo a la mano cualquier consulta de cualquier resultado que se haya generado.

Se pretende también reducir los tiempos de proceso para que sea más oportuno el abasto del material, tomando todo el universo de materiales si es necesario y no solo ciertos grupos como hasta ahora se está realizando.

Una ventaja que se espera tener con este sistema, es el control del inventario de toda la empresa, afín de que solo se tenga lo necesario de todos los materiales que se manejan y no solo de algunos grupos, como se lleva hasta el momento. Esto se traduce en un ahorro sustancial para la empresa.

### 2.3. Recopilación y análisis de la información

Como se definió en el punto 2.1. (Situación actual) en la empresa existen direcciones divisionales, las cuales son las responsables de atender todas y cada una de las demandas de materiales de la misma, para lograr esta función, en cada una de las divisiones existen las siguientes áreas:

- ◆ Área de Control de Inventarios
- ◆ Área de Compras
- ◆ Área de Contabilidad de Planta
- ◆ Área de Auditoría
- ◆ Área Almacenes

**Área de Control de Inventarios.** En esta área se realizan todos los procesos necesarios para determinar la cantidad de material que debe comprarse, para los diferentes almacenes.

**Área de compras.** En esta área, una vez realizados los procesos y reportes de compra para los diferentes almacenes, estos reportes son analizados para realizar los pedidos de compra, y la asignación a proveedores.

**Área de contabilidad de planta.** La función principal en esta área es analizar los consumos que se tienen en los diferentes almacenes y de esta manera tener el control de los gastos realizados en los mismos.

**Área de auditoría.** En esta área se realizan las funciones de supervisar todas y cada una de las tareas realizadas por las áreas antes mencionadas, con el fin de que todo se realicen en orden y procedimientos definidos por la empresa.

**Área de Almacenes.** Estas son las entidades de la empresa donde se encuentran los diferentes materiales, en ellas se generan las peticiones de compra y en otros casos el

traspaso de materiales.

**Área de Control de Inventarios.** La información que se utiliza en esta área se deriva de las bases de datos del sistema SIREM.

**SISTEMA SIREM (Sistema de Recursos y Materiales).** En este sistema se registran todos los movimientos de materiales (salidas, entradas, existencias, excedentes, solicitudes de material, etc.) que se efectúan en los almacenes de la empresa, en cada uno de los almacenes existen terminales que accesan al sistema central vía la red interna de datos de la empresa y este sistema opera a nivel nacional.

En la empresa hay una área de sistemas responsable de la operación y control de este sistema, y en esta área se genera información de vital importancia para el área de control de inventarios, esta información esta constituida en tres archivos y son:

**Archivo de Movimientos,** en este archivo se tiene toda la información registrada por los almacenes, este registro es desde salida, cambio, traspaso, y entrada de material, fechas de registro, personal que se asigna a cada uno de los recursos, categoría, tipo de material, zona, etc.

El contenido de este archivo es tipo texto y es un archivo plano como se muestra en la tabla 2.3.1. Contenido de archivo de movimientos y esta organizada de la siguiente manera:

<b>Campo</b>	<b>Descripción</b>
UN	Unidad.
AR	Área.
PO	Población.
NAF	Nivel de afectación.

CTA	Cuenta.
SCTA	Subcuenta.
AS	Almacén surtidor. En caso de ser traspaso.
AR	Almacén receptor. En caso de se traspaso.
ALM	Almacén que genera el movimiento.
CAT	Catálogo del material.
PED	No. de Pedido en caso de ser entrada por pedido.
SER	Serie o sea grupo de materiales al que pertenece el material.
DEPTO	Departamento a donde sale el material en caso de salida.
T_MOV	Tipo de movimiento.
MOT_S	Motivo de salida.
F_CAP	Fecha del movimiento.
F_U_C	Fecha de última compra en caso de ser entrad por pedido, para un mejor control de compras.
ZONA	Zona del departamento al que sale el material.
CDO	Código (no se usa).
PROV	Proveedor.
UNI	Unidad de medida.
DOC	Número de documento interno del SIREM que registra el movimiento.
ADM	En caso de salida a una obra en específico, el número de la obra ( Autorización de Desembolso Mayor).
REM	Número de remisión del proveedor cuando es entrada por pedido.
EXPED	Expediente del empleado al que se le surtió el material en caso de salida.
MOVTO	Movimiento.
C_MAT	Clase de material, clave que se usa para contabilidad si es inversión o gasto.
CATEGO	Categoría.
CANT	Cantidad del material que se movió en la entrada o salida.
SALDO	Saldo del pedido en caso de entrada por pedido.





**El archivo de Existencias.** En este archivo se tiene información referente al material que existe en cada uno de los almacenes, contando con la clave del almacén, clave de material, la cantidad de material en existencia y si hay material en reserva; el contenido del archivo (en formato de dBase IV) es como se muestra en la tabla 2.3.2.

ALMACEN	NUM. MAT.	CANTIDAD	RESERVADOS	DISPONIBLE
01	0110207	3776.000	0.000	3776.000
01	0110403	100.000	0.000	100.000
01	0110458	60.000	0.000	60.000
01	0110707	222.000	0.000	222.000
01	0110725	340.000	0.000	340.000
01	0110458	60.000	0.000	60.000
01	0110707	222.000	0.000	222.000
01	0110403	100.000	0.000	100.000
01	0110458	60.000	0.000	60.000
01	0110725	340.000	0.000	340.000
01	0110743	237.000	0.000	237.000

**Tabla 2.3.2. Contenido archivo de existencias.**

MATERIAL	DESCRIPCION	ALMACEN	DESTINO	COMPR.	PROGRAMES
1225001			06		4000.000
1656597			07		975.000
4291632			30		630.000
4291632			02		630.000
8183668			67		3.000
8183971			07		2043.000
1381004			07		98.000
1592007			21		1550.000
0144801			07		12699.000
0144801			07		12699.000
0144801			15		12699.000
0144801			80		12699.000
0144801			90		12699.000

**Tabla 2.3.4. Contenido archivo de peticiones de compra de materiales.**

BO STATUS	N MATERIAL	ALMACEN	N PEDIDO	F PEDIDO	FE REQ	CANTIDAD	RECT	C PROG	F PROG	NUM_PROV
CANCEL	1225001	07	5007596	20/01/98	20/01/98	4000.000	0.000	2000.000	2/03/98	45725
PRINTED	1656597	07	5010505	1/12/98	1/12/98	975.000	960.000	975.000	30/12/98	83375
PRINTED	4291632	07	5010750	4/12/98	4/12/98	630.000	0.000	315.000	17/05/99	15349
PRINTED	4291632	07	5010750	4/12/98	4/12/98	630.000	0.000	315.000	16/04/99	15349
PRINTED	8183668	07	5010732	26/11/98	26/11/98	3.000	0.000	3.000	23/12/98	34400
PRINTED	8183971	07	5010443	25/11/98	25/11/98	2043.000	0.000	2043.000	29/01/99	51440
PRINTED	1381004	07	17327	22/01/97	24/09/96	98.000	78.000	98.000	27/09/96	42400
PRINTED	1592007	07	14939	12/06/96	12/06/96	1550.000	1540.000	348.000	30/09/96	56467
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	27/08/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	29/05/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	29/04/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	30/03/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	29/05/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	29/04/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	30/03/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	137.000	28/02/97	88236
PRINTED	1381004	07	17327	22/01/97	24/09/96	98.000	78.000	98.000	27/09/96	42400
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	27/08/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	29/05/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	29/04/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	1142.000	30/03/97	88236
PRINTED	0144801	07	5002012	19/02/97	19/02/97	12699.000	0.000	137.000	28/02/97	88236
PRINTED	1381004	07	17327	22/01/97	24/09/96	98.000	78.000	98.000	27/09/96	42400
PRINTED	1592007	07	14939	12/06/96	12/06/96	1550.000	1540.000	348.000	30/09/96	56467
PRINTED	1013118	07	16257	12/08/96	12/08/96	26500.000	26340.00	15000.000	6/09/96	96611

Tabla 2.3.3. Contenido de Pedidos.

Descripción de los campos que conforman el archivo de existencias, que se muestra en la tabla 2.3.2.:

<b>Campo</b>	<b>Descripción</b>
ALMACEN	Define la clave asignada a cada uno de los almacenes.
NUM_MAT	Define la clave de material, esta clave es la que se tiene definida en el catálogo de materiales.
CANTIDAD	Define la cantidad de material que existe en el almacén.
RESERVADOS	Define la cantidad de material que se tiene en reserva por las necesidades de algún proyecto en especial o la demanda del mismo.
DISPONIBLE	Es la cantidad de material que en un momento dado se puede disponer para uso interno del almacén o en su defecto para traspaso a otros almacenes.

**El archivo de Pedidos.** Este archivo tiene la información de petición de materiales de los almacenes, este archivo tiene la siguiente información como se muestra en la tabla 2.3.3.:

<b>Campo</b>	<b>Descripción</b>
PO_STATUS	Estado del pedido.
N_MATERIAL	Clave del material.
ALMACEN	Clave del almacén.
N_PEDIDO	Número de pedido.
F_PEDIDO	Fecha del pedido.
F_REQ	Fecha requerida del material.
CANTIDAD	Cantidad de material solicitado.

C_RECT	Cantidad de material recibido por pedidos.
C_PROG	Cantidad de material programado recibido.
F_PROG	Fechas programa de recepción de materiales.
NUM_PROV	Número de proveedor asignado.
LINE_STATU	Estado del pedido.

**Área de compras.** En esta área se utiliza la información generada por el área de control de inventarios, que es la petición de materiales a comprar para cada almacén, como se muestra en la tabla 2.3.4.

<b>Campo</b>	<b>Descripción</b>
MATERIAL	Clave del material según catálogo.
DESCRIPCION	Descripción del material.
ALMACEN_DESTINO	Clave del almacén donde se requiere el material.
COMPRA_POR_MES	Las compras que deben realizarse por mes.

Cuando el área de compras crea necesario información adicional para la generación de algún pedido en particular, genera un archivo en Excel como se muestra en la tabla 2.3.5., donde se contemplan todos los materiales que requieran ser analizados. Para efectuar este análisis es necesario tener cierta información que el área de control de inventarios debe proporcionar y es:

- Cuál es el consumo.
- Cuál la existencia.
- Cuál es la existencia Nacional.
- Qué pronósticos se tienen adicionales.
- etc.

MATERIAL	ALAMCEN_DESTINO	COMPRA_POR_MES
1225001	01	4000.000
1656597	02	975.000
4291632	10	630.000
4291632	08	630.000
8183668	12	3.000
8183971	30	2043.000
1381004	07	98.000
1592007	08	1550.000
0144801	03	12699.000
0144801	107	12699.000

Tabla 2.3.5. Contenido de archivo de análisis de compras.

**Área de contabilidad de planta.** En esta área la información que se maneja la podemos dividir en dos grupos, el primero sería el de verificar y evaluar la cantidad de material que se está consumiendo en cada almacén, la existencia y los pedidos de los mismos; para efectuar esta tarea se utiliza el sistema SIREM, para poder proporcionar a el área de control de inventarios los niveles de inventarios en cada uno de los almacenes. El segundo grupo es la responsabilidad de generar el archivo de precios por material, este archivo es generado en dBase IV y contiene la información como se muestra en la tabla 2.3.6., este archivo es utilizado por las áreas de compras y control de inventarios.

MATERIAL	DESCRIPCION	PRECIO PROMEDIO	PRECIO REEXPRESADO
1225001		3.75	3.33
1656597		10.30	10.20
4291632		23.40	23.00
4291632		47.80	47.00
8183668		1345.00	1300.00
8183971		765.30	750.00
1381004		23.00	20.00
1592007		50.00	45.00
0144801		330.00	300.00
0144801		430.00	400.00
0144801		11.00	10.00
0144801		5.20	5.05

Tabla 2.3.6. Contenido de archivo precios por material.

**Área de auditoría.** Esta área como se mencionó al principio, la función principal es de verificar que todas las tareas de las áreas antes mencionadas, se realicen conforme a las normas establecidas por la empresa; la información que utiliza esta área, es un compendio de información de las otras áreas como puede ser: listado de precios, catálogos de materiales, contratos, normas de materiales, garantías, etc.

Cuando a esta área se le solicita que efectúe una auditoría en algún otra área en particular referente a materiales y que pueden ser en: uso de los mismos, contratos, garantías, cantidades asignadas, etc.; la información que se utilizará va a ser muy variable y depende del alcance de la misma.

**Área de Almacenes.** En estas entidades es donde se hace la recepción y entrega de materiales, para realizar esto se utilizan los siguientes formatos internos de la empresa que son conocidos como F-1106 y F-1107 respectivamente, estos formatos utilizan la siguiente información:

<b>Campo</b>	<b>Descripción</b>
Expediente del empleado	Clave de empleado que la empresa le asigno.
Clave material	Clave del material según catálogo.
Cantidad	La cantidad de material solicitado.
Unidad de medida	La unidad de medida (kg, m, pieza, lts, etc.).
Cuenta de aplicación	Cuenta donde se carga el gastos del material.
No. de Firma autorización	La firma que autoriza la salida o ingreso del material
Teléfono	Número de teléfono del responsable.
Proyecto	Nombre del proyecto.
Fecha	Fecha de salida o entrada de material.

En la siguientes paginas se muestran los formatos de entrada (figura 2.3.1.) y salida (figura 2.3.2.) de materiales en que se utilizan en los almacenes.

ENTRADA DE MATERIAL			
NOMBRE PROVEEDOR/ALMACÉN		CÓDIGO	
DOMICILIO		CÓDIGO	
NUMERO DE PEDIDO	FECHA DE PEDIDO	FECHA DE RECEPCION	FECHA DE VENCIMIENTO
ORIGEN DEL MATERIAL	DESCRIPCION DEL MATERIAL	CANTIDAD	VALOR
NACIONALIDAD	TIPO DE MATERIAL	UNIDAD DE MEDIDA	VALOR UNITARIO
AUTORIZACION DEL DIRECTOR GENERAL			

Figura 2.3.1. Forma F-1106, Entrada de materiales.





Otro tipo de información que se ve involucrada en el proceso, son las peticiones de materiales proyectadas a ciertos periodos por parte del usuario. Es decir, en función de los proyectos asignados, el usuario puede programar las necesidades de materiales que puede necesitar para cumplir con las expectativas de los proyectos y que por falta de los mismos no se cumplan. Para lograrlo, el usuario organiza sus requerimientos y los hace llegar al área de control de inventarios para que, cuando se realicen las peticiones de pedidos al área de compras, se contemplen dichas peticiones; esta información es generada en diversos formatos que pueden ser archivos en:

- Excel
- Word
- dBase
- Escritos a mano
- etc.

Estos archivos generalmente están integrados por:

<b>Campo</b>	<b>Descripción</b>
Clave material	Clave material según catálogo.
Descripción	Descripción del material.
Cantidad	Cantidad de material.
Unidad de medida	La unidad de medida (kg, m, piezas, lts, etc.).
Fechas	Fechas de requerimientos, entrada, salida, etc.

El área de control de inventarios genera un formato de traspaso de materiales, como se muestra en la figura 2.3.3., este formato se manda al área de logística y a los almacenes involucrados para que realicen en movimiento. La información que contiene este formato es:

<b>Campo</b>	<b>Descripción</b>
Clave material	Clave material según catálogo.
Descripción	Descripción del material.
Unidad de medida	La unidad de medida (kg, m, piezas, lts, etc.).
Almacén origen	Clave del almacén donde esta el material.
Almacén destino	Clave del almacén a donde se pasa el material.
Cantidad	Cantidad de material.
Consumo promedio	Consumo promedio del material que se traspasa en el almacén origen.
Existencias promedio	Existencia promedio del material que se traspasa en el almacén origen.
Excedente	Excedente de material que se traspasa en el almacén origen.

Debido a la cantidad de materiales que se manejan en la empresa se requiera de un área que permita organizar y centralizar la información de todos los materiales que en un momento dado se requiera transportar entre los diferentes almacenes. Es decir, cuando en un almacén se requiera de algún material que exista en otro almacén y del cual se pueda traspasar a los almacenes que lo requiera debe de existir una logística para realizar esta tarea.

El área de control de inventarios genera los reportes de todos los materiales que en un momento dado se requiera traspasar, el área de logística utiliza esta información para la organización y control de los diferentes movimientos de materiales que se tengan que realizar entre los diferentes almacenes.

Cabe mencionar que la función que desempeña esta área es de vital importancia para que los materiales necesarios se entreguen en tiempo y orden a las diferentes entidades donde sean requeridos.



## **2.4. Planteamiento del problema**

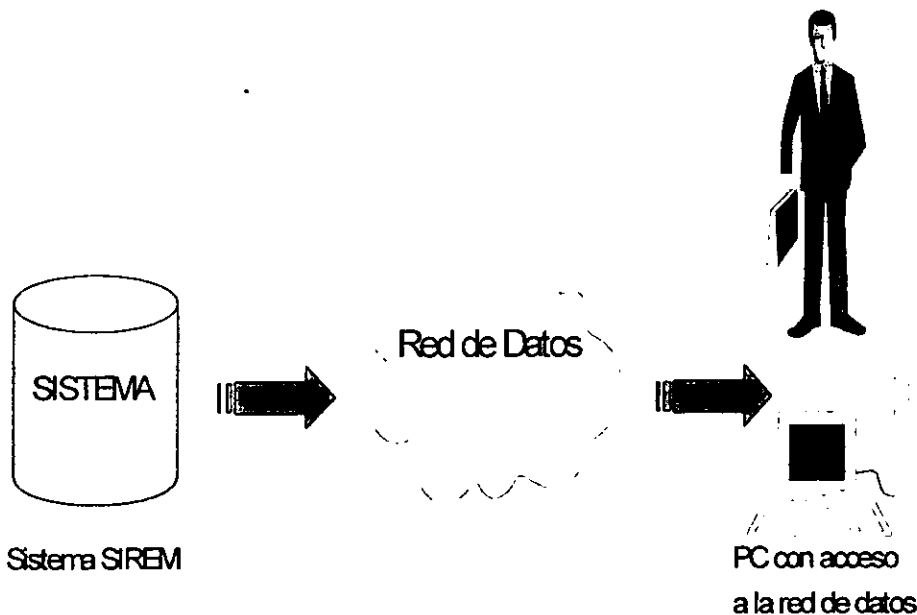
En este punto analizaremos cada uno de los problemas que existen en las diferentes áreas, como se mencionó en el punto anterior (2.3.), en la empresa se tienen las siguientes:

- Área de Control de Inventarios
- Área de Compras
- Área de Contabilidad de Planta
- Área de Auditoría
- Área Almacenes

### **Problemática para el área de control de inventarios**

Para el área de control de inventarios, se sabe que la información es generada en el sistema SIREM, y está agrupada en tres archivos, la obtención de estos archivos se tiene que hacer de manera manual, es decir se utiliza una PC con acceso a la red de datos de la empresa y con software de comunicaciones bajo protocolo TCP/IP, dado que el sistema esta en ambiente UNIX, se tiene la necesidad de hacer uso del programa de transferencia de archivos FTP, con este programa se transfieren los archivos del sistema SIREM a una PC del área como se muestra en la figura 2.4.1.

Es evidente que este proceso no esta integrado y que debe de estar una persona con conocimientos tanto del uso del programa FTP, como saber en que dirección IP se tiene que conectar, también debe conocer los archivos que se van a transferir; esto crea un problema, si la persona que conoce esta información no se encuentra y existe la necesidad de transferir los archivos, en ese momento no se podría realizar y con ello el área no contaría con los datos para realizar sus funciones; el detener un proceso de compra de insumos en una empresa de servicios puede ser muy costoso.



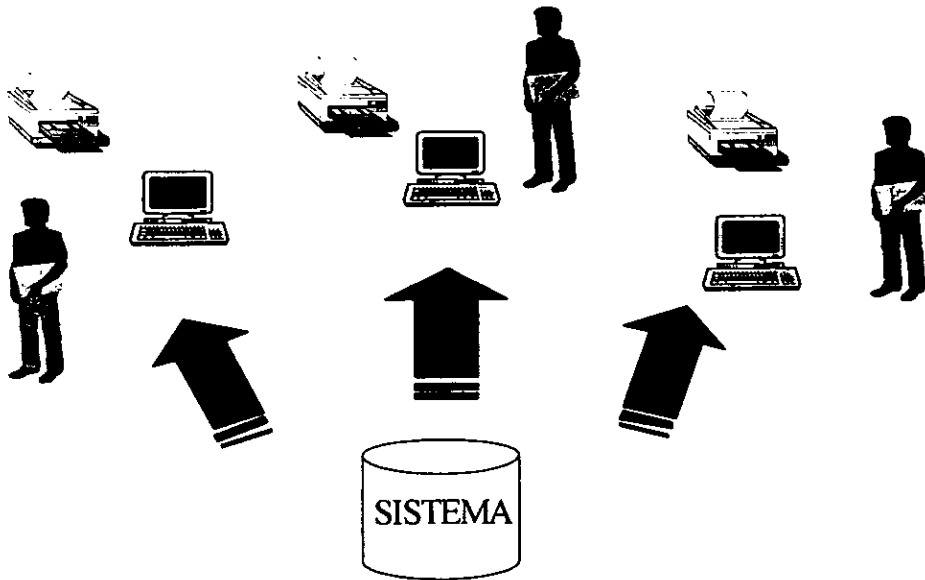
**Figura 2.4.1. Transferencia de archivos.**

Los archivos que se transfieren, como se mencionó anteriormente son tres, los cuales están en diferente formato; para el caso del archivo de movimientos, es un archivo en formato texto y trae demasiada información en blanco, es decir, en cada línea del archivo existen 31 campos, de los cuales solo traen información los que son referentes al movimiento que en su momento se efectuó. Esto implica mucho almacenamiento en disco y cuando se corre el proceso de clasificación de movimientos, el tiempo de ejecución es excesivo, si en esta archivo no existiera tanta información innecesaria se optimizarían los tiempos.

Esto sin duda es un problema en el tiempo de entrega de resultados por parte del área, primero porque tiene que depurar mucha información y exportarla a un formato de archivo de bases de datos en este caso dBase, y esto implica que exista una persona especialista en la ejecución de este proceso, haciendo uso de varias herramientas de software.

Otro problema que hay es que debe de existir personal especializado para realizar el análisis del contenido del archivo de pedidos, dado que se requiere verificar la información para que sea la correcta; es decir, que estén bien las características de los materiales solicitados, tales como: tamaño, longitudes, marcas, tipos, capacidades, diámetros, pesos, etc. Este proceso sin duda requiere de mucho tiempo y con ello la dependencia de recursos especializados.

Como existe personal especializado en diferentes materiales, como se muestra en la figura 2.4.2., el análisis de información lo realizan de manera independiente, utilizando las herramientas que para cada persona le sean prácticas, con ello se pierde el control de procedimientos y orden en dicho proceso. Esto sin duda es un problema, ya que no existe un control en la validación de la información y se produce una variedad de reportes y formatos.



**Figura 2.4.2. Personal especializado para el análisis de materiales.**

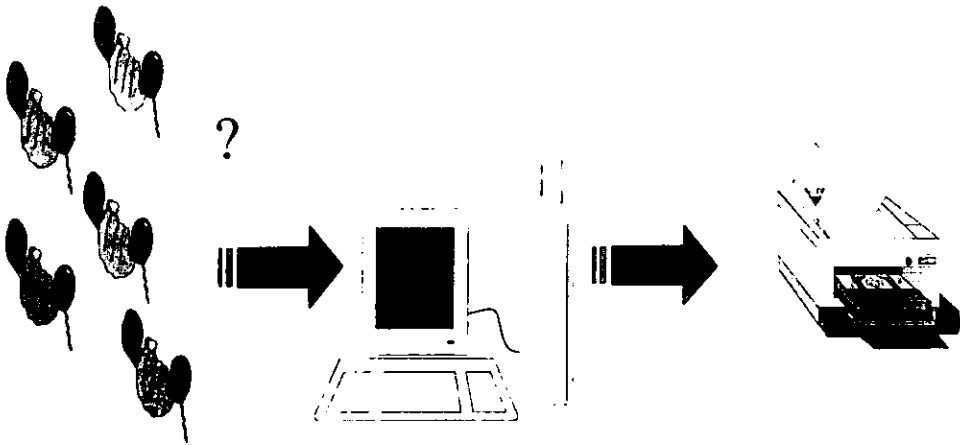
Dado que existen varios procesos manejados de manera independiente, el tiempo de

respuesta y entrega de información a las demás áreas no es el esperado u óptimo, y con ello se tengan tiempos de entrega de materiales fuera de fechas requeridas por parte del usuario final, esto afecta directamente los intereses de la empresa, dado que no se puede dar una respuesta oportuna a los servicios esperados por los clientes.

Dentro de las funciones que realiza esta área, existe la generación de reportes estadísticos, los reportes solicitados no son genéricos, dado que son pedidos por personal de diversas áreas y de niveles superiores, esto implica que existe una gran variedad de informes con requerimientos de información, lo que provoca mucho tiempo de proceso, esto debido a que la información con la que cuenta el área (control de inventarios) son archivos en dBase y no se tiene una base de datos completa y organizada; con lo cual no se cumplen las entregas en un tiempo razonable y oportuno establecido por las áreas de interés ocasionando un problema que impacta en la toma de decisiones.

Dado que en esta área existen un sin número de procesos, estos son ejecutados por personal especializado de manera independiente, y es por ello que a la falta de un sistema que pueda controlar y administrar la información, es evidente que el manejo de la misma sea procesada con diversas herramientas de software; estos procesos en su conjunto están consumiendo muchos recursos y el tiempo de respuesta no es el esperado afectando a las áreas que hacen usos de esta información.

Un problema adicional, es sin duda la gran cantidad de llamadas que se atienden diariamente, como se muestra en la figura 2.4.3., solicitando información respecto al estado de sus pedidos y en ocasiones apoyo para generación de pedidos que en un momento dado se requieren por alguna petición especial o por algún cambio de prioridad en los proyectos.



## CONTROL DE INVENTARIOS

**Figura 2.4.3. Información de pedidos.**

Dado que los procesos se ejecutan en diferentes equipos PC, cuando se conjunta la información, mucha es manejada en discos flexibles y esto sin duda son puntos de falla, de los cuales podemos mencionar:

- ◆ El riesgo de no tener la información correcta, es decir la última versión.
- ◆ Posibilidad de contaminación de virus.
- ◆ Riesgo en la integridad de la información.
- ◆ Información no normalizada.
- ◆ No contar con la información agrupada.



### Problemática para el área de compras

El área de compras recibe un formato de solicitud de materiales, que es generado por el área de control de inventarios, como se muestra en la figura 2.4.4., pero la información que se requiere no es generada con oportunidad para que el área realice los pedidos en tiempo.

El reporte entregado por parte del área de control de inventarios, en algunos casos lo requiere con más análisis de la información; es decir, para comprar algún material requiera saber: existencia en almacenes, consumo promedio del material por almacén solicitante, verificar existencia en algún otro almacén. Esto desde luego, es un trabajo que el área de control de inventarios (por la falta de información, organización y estructura de la misma) le implica mucho tiempo de proceso para su obtención, ocasionando tiempos muertos en est área de compras.

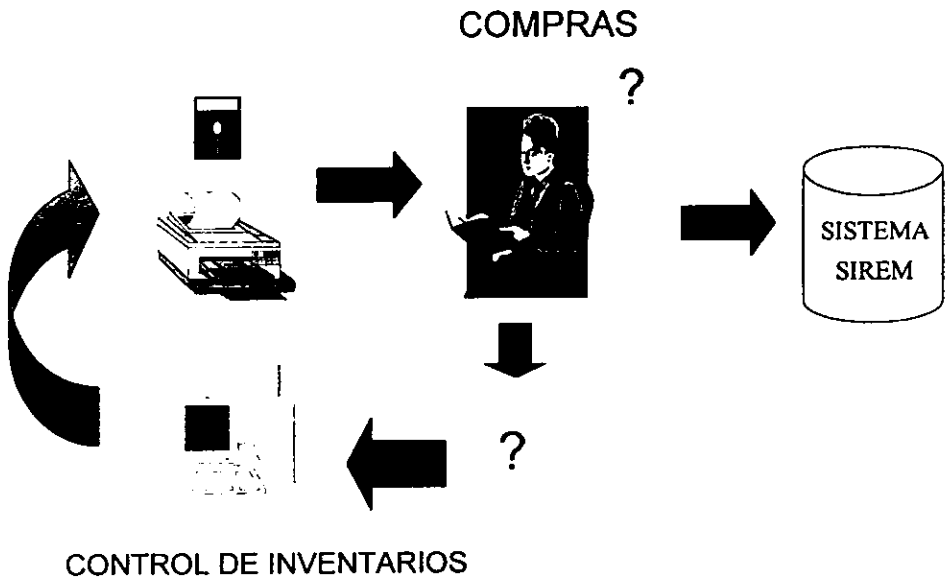


Figura 2.4.4. Flujo de formato de compra.

El reporte que el área de control de inventarios entrega al área de compras, es un archivo en Excel, esto es un problema por que no existe un sistema que permita que esta información pueda ser consultada directamente por los responsables del área y en su defecto poder consultar información adicional directamente y no depender del área de control de inventarios una vez procesada la información para pedidos de compra.

#### **Problemática del área de contabilidad de planta**

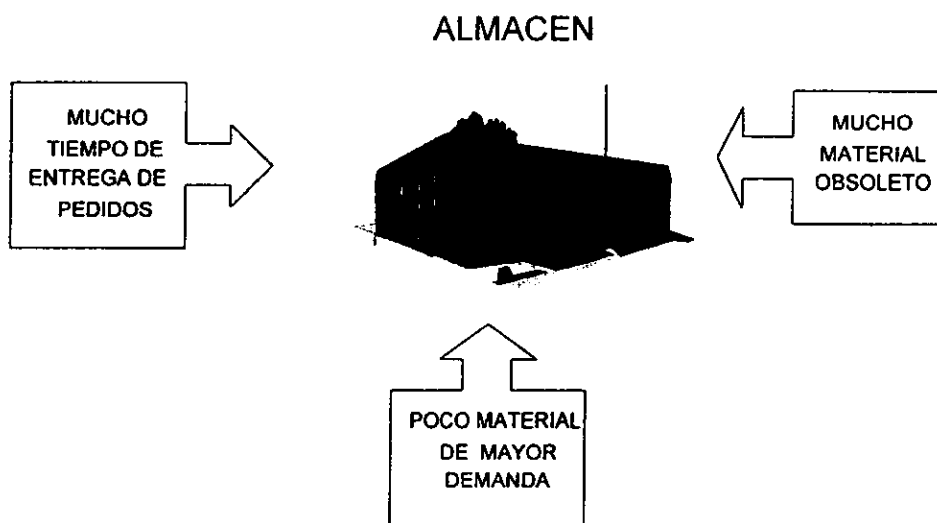
Esta área es la responsable de verificar los consumos en los almacenes, creación de catálogos de materiales, precios de los mismos y niveles de inventarios en los almacenes; el principal problema que se tiene una gran dependencia de la información existente en el sistema SIREM, de este sistema genera sus reportes de manera independiente. Esto implica que la información para las demás entidades no esté en línea y haya que esperar que se termine de verificar toda la información en el sistema y posteriormente procesarla para después hacerla llegar a las áreas interesadas (control de inventarios, compras, almacenes).

#### **Problemática del área de auditoria**

Esta área se respalda mucho en la información del sistema SIREM, información de catálogos, contratos de proveedores, información de los proyectos, etcétera. Para esta área sería conveniente que por lo menos los procesos que se involucran en los procedimientos de la generación de pedidos de materiales, se concentrarán en un sistema centralizado donde se tenga registrado todos los procesos de las diferentes áreas; dado que existe la necesidad de validar información en los procesos, es necesario contar con información real y confiable, en la actualidad se tiene la información en diversos formatos. Esto desde luego dificulta las funciones de esta área, que sin duda es una de las más importantes.

### Problemática en el área de almacenes

En esta área existe el problema de que en los almacenes se tienen niveles de materiales muy altos y estos en su mayoría son obsoletos, como se muestra en la figura 2.4.5.; esto sin duda es la causa de una mala administración con los procedimientos actuales, lo anterior es debido a que en las áreas involucradas no se cuenta con la infraestructura y los sistemas apropiados para el traspaso y baja de materiales, o estos no son realizados con tiempo y orden.



**Figura 2.4.5. Problemas en los almacenes.**

En los almacenes se tiene el problema de que el tiempo de entrega de materiales que realmente se necesitan, es mucho y en la mayoría de los casos los empleados no cuentan con los elementos necesarios, para realizar su trabajo y con ello se detiene la atención al usuario final.

En esta área se tiene además el problema de que por la mala administración de proyectos surgen las famosas peticiones express, es decir cuando un proyecto por algún motivo cambia de prioridad y con ello las necesidades de los recursos materiales también, con esto, los usuarios se ven en la necesidad de mandar peticiones al área de control de inventarios directamente, esta información el usuario la genera en diversos medios, los cuales por mencionar algunos son (archivos en excel, correo electrónico, word, vía fax, vía telefónica, etcétera), y con ello se tiene una problemática más para el área de control de inventarios.

### **Problemática en el área de Logística**

Esta área participa de manera directa con el área de control de inventarios, es decir cuando el área de control de inventarios, tiene los formatos de solicitud de traspaso de materiales entre almacenes, esta área realiza la logística para el traspaso de estos materiales, pero debido a la mala administración de los recursos materiales, en esta área se tiene el problema de que en muchos casos los transportes no son aprovechados a su máxima capacidad, dado que las peticiones de traspaso de materiales no se pueden organizar por falta de información centralizada, para que se pudiera utilizar el transporte a su máxima capacidad.

### **Problemática de costos**

En este proceso del control de inventarios, se tiene la participación por un lado de personal especializado y por otro mucho equipo de cómputo para poder efectuar dicho proceso, como se muestra en la figura 2.4.6. Se ocasiona un costo muy elevado para los resultados que se están obteniendo, ya que se tienen muchos problemas con la entrega de pedidos de materiales y con esto se afecta directamente a los intereses de

la empresa, que tiene como fin brindar un mejor servicio a sus clientes. Es por ello que se tiene la necesidad de mejorar los tiempos de respuesta de este proceso, así como, la organización y estructuración de la información para poder controlarla.

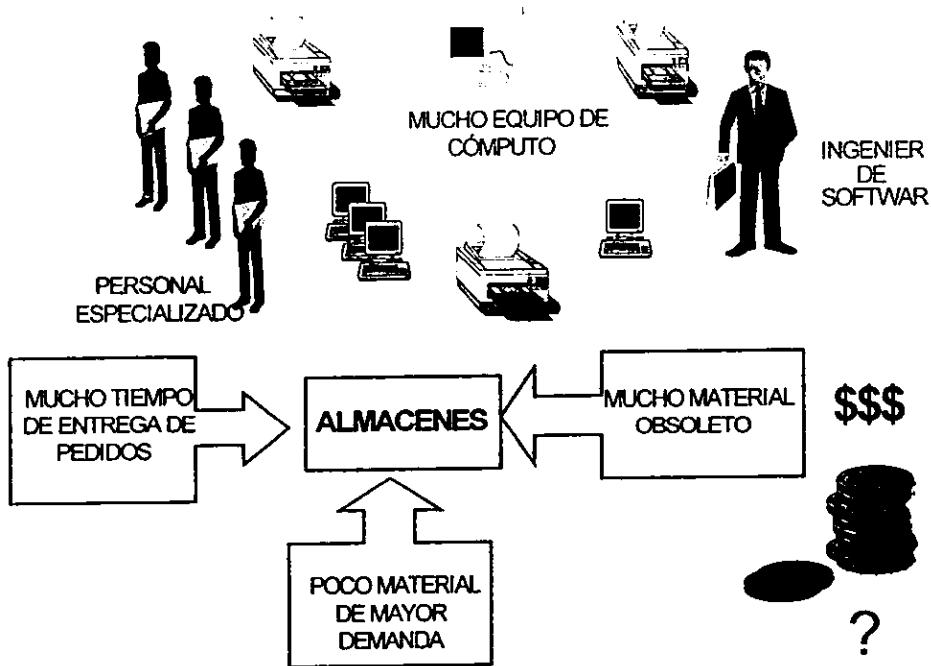


Figura 2.4.6. Elementos y problemas que se tienen en el proceso de pedidos.

## 2.5. Descomposición funcional

De acuerdo a los incisos 2.1. al 2.4. en los que básicamente se plantea el problema, en este inciso se analizara en que áreas hay problemas.

Se pueden detectar 6 áreas en las que se presenta algún tipo de problemática, que están relacionadas con el área de inventarios (figura 2.5.1.) y estas son el área de:

- Área de Control de Inventarios
- Área de Compras
- Área de Contabilidad de planta
- Área de Auditoria
- Área de Almacenes
- Indirectamente Área de Usuarios.

Todas las áreas comparten 2 puntos en común, los cuales son la falta de información y de un sistema de consulta en línea, lo cual es muy importante para tener la mayor información en el tiempo requerido, esta debe ser confiable según las necesidades de cada uno de ellos. Esto lleva de la mano el hecho que en todos los casos la información generada por cada una de estas áreas está en formatos diferentes: archivos de Excel, Word, Dbase, etc. provocando pérdidas de tiempo y una integración de dicha muy dudosa de dicha información.

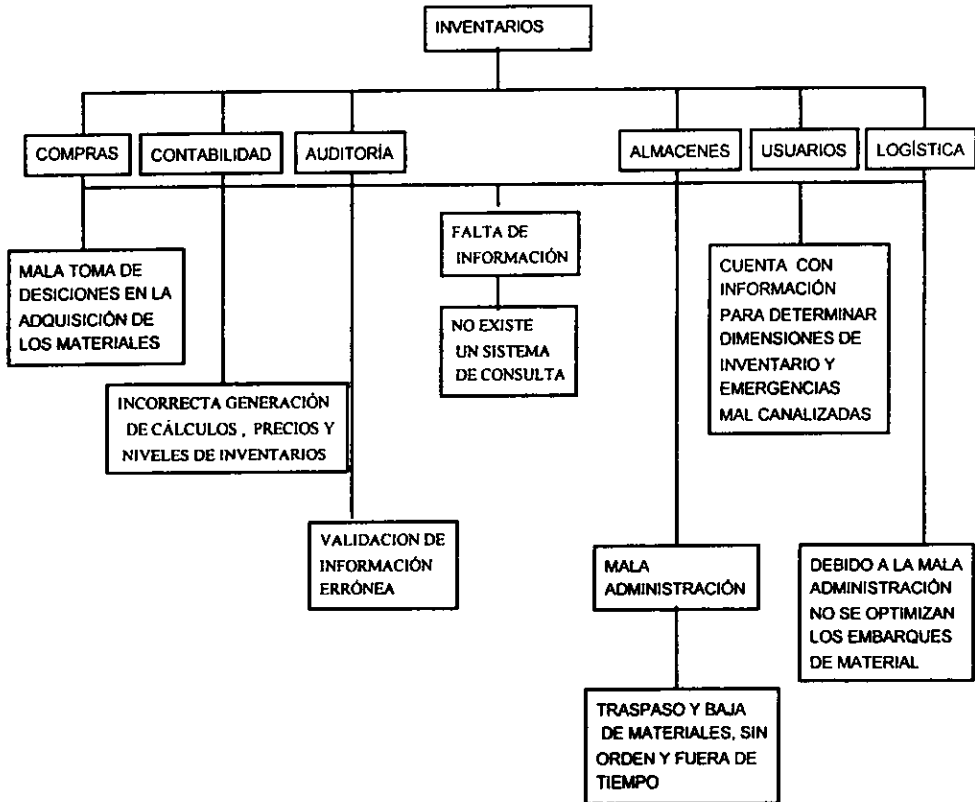
El área de compras es la encargada de autorizar la compra de los materiales, siempre y cuando el costo no sea excesivo; estos materiales son los que componen el inventario. La falta de información genera que se realicen incorrectamente cálculos, precios y niveles de inventario, lo cual impacta de manera directa en la decisión de autorizar la compra de los materiales que forman el inventario, ya que esto puede generar compras excesivas o compras insuficientes de los mismos.

Si antes de adquirir un lote de mercancías (materiales) el pedido de los mismos, no fue

planificado, se pueden generar a futuro pérdidas para la empresa telefónica, tanto como si se compran materiales en exceso (ya que si se cubrirán las necesidades, pero se tendrá en almacén cantidades excesivas de materiales que no se utilizaran, y esto representa pérdidas) o en cantidades inferiores a las que se necesiten (ya que probablemente se cubran las necesidades previstas, pero siempre surgen emergencias de necesidades de materiales y esto también significa pérdidas para la telefónica, al no dar una respuesta inmediata al usuario).

El área de almacenes, también presenta una problemática, ya que su función principal es el abasto oportuno de material a las diferentes áreas que lo demanden; y si en este caso el área de almacenes no da un abasto en tiempo, ya sea por la decisión del área de compras de no adquirir el lote de materiales solicitados o solo adquirir una parte de los mismos, los compromisos adquiridos y las emergencias, no podrán llevarse a buen término en las fechas establecidas. Lo anterior también genera que el traspaso y baja de materiales se lleve a cabo sin orden y fuera de tiempo

Por su parte el área de logística también esta involucrada en esta problemática, ya que es la encargada de embarcar los materiales a quien lo demande. En este caso, si al involucrar al área de compras, la cual ha autorizado la compra de un lote de materiales que sólo se ajuste al presupuesto y no a las necesidades reales de la empresa; dicha área de logística ya tiene un plan de rutas a seguir para el abastecimiento a los usuarios en el tiempo demandado; sólo se embarcarían los materiales que en ese momento estén en existencia y dado la demanda de materiales que no hayan sido adquiridos en el primer pedido autorizado, se tengan que adquirir en un segundo pedido. Esto significa que se tendría que enviar otro embarque de materiales a donde no se haya abastecido de los mismos, siendo que quizá si la planeación, tanto de lo que se iba a adquirir, como de lo que se va a abastecer hubiese sido la óptima, no se habría tenido que enviar en diferentes embarques los materiales o hacer diversos pedidos para subsanar lo faltante. Lo cual degenera en pérdidas de tiempo y dinero para la empresa.



**Figura 2.5.1. Diagrama de bloques de problemática por áreas.**

La problemática detectada en el área de auditoría, parte del hecho de que, como ya se mencionó, el flujo de información no sea el adecuado y no se cuente con un sistema de consulta en línea, generen una validación de información errónea, ya que esta área es la encargada de la generación de los pedidos de materiales y es muy importante que todos los procesos de las áreas esté centralizada, para que dicha validación de esta información sea real y confiable.



La última parte involucrada que también presenta una problemática es el área del usuario, ya que es el que puede pronosticar las necesidades que se tienen, así como las que se puedan presentar como emergencias. Esta información es muy importante, ya que permite calcular o predecir las necesidades de la compañía.

Por último estos seis factores como se puede percibir están relacionados entre sí, para la toma de decisiones en cuanto al inventario, pero a su vez el flujo de información entre los mismos no es la adecuada, podemos observar que la información que se requiere para la asignación de materiales no es hecha a tiempo, y la misma debe de recolectarse de diferentes áreas, después procesarla y completarla de tal forma que se obtenga un reporte que permita realizar la labor más rápida y sistemática.

Actualmente para determinar la cantidad de material que debe comprarse, se extraen los archivos del sistema central una vez por mes, y se realizan varios procesos en forma aislada utilizando bases de datos de Dbase III y hojas de cálculo en diferentes computadoras; con esto se determina en forma manual qué materiales pueden ser traspasados de una división a otra para evitar la compra excesiva. Dado el volumen de información, se ha dividido el catálogo de materiales para controlar en forma mas precisa aquellos que se consideran más importantes por su alto costo, restando atención a los demás, que por volumen representan también un alto costo para la empresa y careciendo de oportunidad en el análisis de los mismos.

Actualmente en esta empresa telefónica, el surtido de materiales se hace realizando una serie de procesos aislados, tomando la información del sistema central de almacenes. Este proceso es muy tardado dada la plataforma con la que se realiza (hojas de cálculo y pequeños programas en DBase III). Con estos tiempos de proceso, solo se tienen en observación algunos materiales que se consideran muy importantes por su costo, teniendo una velocidad de respuesta de un mes para la determinación de compras.

## 2.6. Opciones de solución

Para evitar estos problemas de falta de inconsistencia de la información, se evaluaron diferentes alternativas para desarrollar un sistema de control de inventarios, el cual elimine toda la problemática anteriormente descrita.

El objetivo de este punto de la tesis es evaluar herramientas de bases de datos y herramientas visuales que mejor se adapten para crear las aplicaciones específicas, en torno al desarrollo de este proyecto del Sistemas de Control de Inventarios.

Estas opciones deben de contar con una funcionalidad modificada y también deben de tener herramientas interactivas para los usuarios finales y que puedan usarse en construir rápidamente aplicaciones más sencillas y fáciles, sin tener que programar.

Para el desarrollo de este sistema se evaluaron las siguientes opciones de software.

En cuanto a bases de datos relacionales se evaluaron las siguientes opciones: Access, Microsoft Fox para Windows y Paradox para Windows.

### **Microsoft FoxPro para Windows**

Los productos FoxPro, cuentan con un rico ambiente de programación, lo que lo coloca como la herramienta principal de la compañía para los programadores.

Los productos de Fox lograron prominencia en el mercado de base de datos mediante una combinación de velocidad y compatibilidad estricta con el lenguaje de Dbase. Hoy la compatibilidad con Dbase, no es tan importante, pero la velocidad sigue siendo el objetivo principal.

FoxPro siempre ha sido uno de los sistemas de base de datos más rápidos. Una de las razones es porque el código es compilado en vez de interpretado. El código original

modificado para programas, pantallas, reportes, etc.; se compila de forma transparente antes de ejecutarse. Otra razón para su velocidad es el uso extensivo de caches internos para evitar acceso innecesario al disco. El acceso al disco es minimizado aun más por el uso de índices, comprimidos y combinados (archivos.CDX).

### **Herramientas interactivas para usuarios finales**

La velocidad de FoxPro no es la única razón para su popularidad; sus herramientas interactivas fáciles de usar también han contribuido a su atractivo. La ventana de vista (View Window) puede abrir las bases de datos, fijar relaciones, e inspeccionar y modificar sus contenidos, todo con unos pocos pulsos al Mouse. Cuando inicia cada acción hay una ventana de comando (Command Windows) que muestra las instrucciones de programación que realizan la misma tarea.

El enfoque de aprender sobre la marcha se usa en la herramienta RQBE de Fox Pro.

Se puede reconstruir consultas complejas con enlaces de tabla múltiple.

De tal manera, RQBE genera una declaración SQL SELECT y envía los resultados de la consulta a una Browse Windows (ventana de vista), un reporte impreso, una base de datos nueva, un cursor SQL en memoria o una rutina para crear diagramas.

Si se pulsa el botón ver SQL (See QL) en cualquier momento, se mostrará como se construye la instrucción Select.

Un generador de aplicación automático le deja crear instantáneamente una aplicación de múltiples tablas con habilidades de consulta, actualización y reporte.

Pero el hacer cualquier mejora importante a la aplicación puede requerir tanta programación como la construcción de la aplicación desde cero con las herramientas

mecánicas (Power Tools).

### **Herramientas mecánicas para crear aplicaciones**

Las herramientas mecánicas de FoxPro incluyen el creador de pantalla (Screen Builder), El creador de menú (Menú Builder), el escritor de reportes (Report Writer), y el transportador entre plataformas (Cross Platform Transporter) para moverse entre las versiones de DOS y la de Windows de FoxPro.

El creador de pantallas de FoxPro, que es visual y basado en objetos, le deja colocar y dar tamaño rápidamente a objetos como texto, campos, y botones de apretar en una ventana. Puede combinar las ventanas para formar aplicaciones completas.

### **FoxPro para Windows**

FoxPro para Windows también ofrece un número de opciones que incluyen barras de herramientas para objetos, mapas de bits, imágenes en pantalla y el escritor de reportes, un "GraphWizzard = Asistente gráfico" para la especificación de diagramas sobre la marcha en RQBE y un corrector ortográfico para campos de texto y anotaciones.

La versión de Windows también le deja especificar fuentes y tamaños de éstas para objetos generados por el creador de pantalla. Una caja de diálogo Tab Order (Orden de Tabulador) facilita la organización de los objetos en la pantalla. La pantalla rápida (Quick Screen) genera "pantallas instantáneas" como un punto de partida para modificaciones futuras. El reporte rápido realiza una función similar para el escritor de reportes.

El editor de texto de FoxPro para Windows apoya el movimiento de arrastrar y soltar y puede darle mas o menos sangría a un bloque de texto con una sola operación.

### **Access.**

Access es un potente administrador de base de datos con características atractivas para los usuarios de bases de datos, los profesionales y los programadores. Hasta ahora los programadores de bases de datos y sus usuarios han estado claramente definidos.

Access logra colocarse en el mercado, combinando un lenguaje de programación completo, capacidad de multiusuarios y enlaces de SQL a datos remotos con la facilidad de uso gráfico, consulta por ejemplo, mediante arrastrar y soltar, y características de reporte y creación de modelos elegantemente automatizados.

Access es lo suficientemente potente, tanto para los usuarios experimentados y los programadores y lo suficientemente fácil para los usuarios finales.

Frecuentemente, los productos de base de datos en la PC no logran comunicar la realidad global, que una base de datos es una colección de tablas, índices y programas, y que estas piezas están relacionadas como parte de una unidad mayor que se usa en combinación para solucionar un problema.

Muchos productos ignoran esta realidad y confunden a los usuarios al aplicar términos como base de datos, tablas y archivos de bases de datos e intercambiarlos entre sí.

Al construir una base de datos lo primero que se nota es un recipiente vacío en el que se puede crear y colocar objetos de la base de datos: tablas, consultas, modelos, reportes y módulos. No se puede ignorar el hecho de que todas estas herramientas trabajan juntas, Access no lo permite. Esta vista combinada le da un acceso rápido a todos los componentes de su base de datos y simplifica enormemente la organización conceptual y física, ya que todos los objetos están almacenados juntos en un solo archivo MDB.

Access es un producto nativo de Microsoft Windows y tiene mejoradas técnicas de Windows como el arrastrar y soltar, barras de herramientas dinámicas y MDI.

Aunque Access almacena la información en su propio formato exclusivo, también puede leer y actualizar datos almacenados en archivos de Btrieve, Dbase y Paradox y sus índices asociados. Esto significa que se puede hacer una transición a Access sin cambiar los datos actuales al formato de Access. De hecho, como las herramientas de Access son más fáciles de usar que las que vienen con muchos productos, se puede usar como un frente para la información existente, la cual permanece almacenada en el formato nativo.

Access tiene apoyo para los tipos de campos usuales de texto, número, si/no, fecha/hora y añade apoyo para moneda, anotaciones, contador y objetos de OLE. El tipo de objeto de OLE le permite incluir o enlazar en su tabla, tales como anotaciones de más de 32,000 caracteres, documentos de Word for Windows, y archivos de sonido o gráficos.

El tipo de contador es un valor entero de lectura solamente que Access incrementa automáticamente para cada registro nuevo, una gran conveniencia es que se puede usar como el campo de clave primaria en muchas tablas. De hecho, puede dejar que Access cree automáticamente una clave primaria en su lugar, en cuyo caso también añade un campo de contador.

Access también le permite definir una clave primaria de uno o múltiples campos y crea automáticamente un índice apropiado y previene los valores de clave duplicados.

Cuando se define un campo, puede especificar el criterio de validación y un valor prefijado que se debe incluir al añadir un registro. Estos criterios de tablas pueden ser erróneos, ya que Access no aplica la validación a nivel de tabla. Los criterios especificados solo sirven para valores prefijados que se llevan a sus diseños de modelos o reportes.

En realidad, Access válida la información a nivel de los modelos, y la validación sólo se realiza si se cambia el valor de un campo. Si un usuario entra en un campo en modelo y sale sin cambiar nada, la validación no se realiza. Puede superar esta limitación validando los datos antes de entregarlos a la base de datos, pero en general esta es un área donde Access necesita mejorar.

El usuario puede definir relaciones entre tablas y hacer que Access coloque ciertas restricciones en lo que se puede hacer con las tablas.

Si el usuario selecciona la opción de "Enforce Referential Integrity = hacer cumplir la integridad de referencias", Access no le permitirá borrar un registro si existen referencias al mismo. Por último, no se puede hacer referencia a un registro que no existe.

El servicio de consultas de Access es fácil de usar. Primero, se tienen que arrastrar las tablas que necesita del recipiente de la base de datos a la ventana de consulta (Query), luego se debe arrastrar los campos que quiere de las tablas al retículo de la QBE. La razón por la que se llama un QBE gráfica es que debe especificar las relaciones de unión arrastrando un campo en una tabla y soltándolo sobre un campo en otra tabla. El efecto es el de arrastrar una línea entre las dos tablas gráficamente.

Access es lo suficientemente inteligente para crear las uniones automáticamente si ha definido relaciones entre las dos tablas.

También puede especificar los criterios de consulta y ejecutar o guardar la consulta.

Internamente, Access convierte su consulta en una declaración SQL SELECT.

Puede llamar la declaración SELECT, y si sabe lo que esta haciendo, editar la declaración directamente. Los cambios se reflejan automáticamente en el retículo QBE. Esta puede ser una herramienta de aprendizaje fantástica y es un gran bono para los

expertos en SQL. El producto también apoya a UPDATE, INSERT INTO y DELETE aunque no a UNION.

Las consultas de Access son optimizadas, y también se pueden ejecutar en un segundo plano, lo que significa, que se podrá comenzar a ver y a trabajar con los resultados de la consulta, antes de que ésta haya terminado.

La salida de una consulta, una vista actualizada llamada un "Dynaset = grupo dinámico", es probablemente el mayor logro técnico de Access. A diferencia de los resultados de consultas en otros productos, un Dynaset es un objeto activo.

Se pueden cambiar los datos en el Dynaset, y en la mayoría de los casos los cambios se ven reflejados en las tablas originales. Además un Dynaset puede ser la fuente de modelos y reportes. Esto le da una gran flexibilidad en la manera en que trabaja con su información.

Access automáticamente suministra los enlaces correctos si define las relaciones de las tablas involucradas. Los modelos traen controles que le permiten navegar hacia adelante o hacia atrás en la tabla. La creación de modelos fue facilitada por Wizard, asistentes de Access que lo guían por una sesión de preguntas y respuestas sobre el trabajo y luego generan, modelos, reportes y gráficos.

El servicio de macros de Access permite automatizar las tareas y no se confunde con un grabador de pulsos de teclas.

Puede crear aplicaciones sofisticadas simplemente usando el servicio de modelos y el servicio de macros, sin tener que usar el Access Basic para nada.

Puede llamar los macros asignándolos a eventos en un modelo (de "on click = pulsar botón", el evento de "Before Undate = antes de actualizar", etc.). Además define bibliotecas de macros, pero no puede seleccionar un macro desde una biblioteca.



Access solo muestra el nombre de la biblioteca y no las macros.

### **Paradox para Windows**

Con Paradox todo es un objeto, de manera que la construcción de una aplicación solo requiere unir los bloques apropiados. Las tablas, los reportes y las consultas, son todos objetos. Hasta las cuadrículas en su tabla de revisión son objetos.

Los objetos tienen propiedades y métodos. La propiedad de los datos en una tabla, por ejemplo, podría incluir su fuente, color y alineación de texto. Los métodos son porciones de código (en el caso de Paradox, es código de ObjetPAL) que determina lo que hace el objeto. El método le dice al objeto como responder al evento. Los objetos y los métodos se pueden volver a utilizar. El hecho de que Paradox este tan arraigado en objetos, brinda dos características importantes en él: es constante y es visual.

La "DataBase Desktop = área de trabajo de base de datos" de Paradox es la interfaz gráfica para usuario que administra los componentes, los objetos, de su aplicación de base de datos.

El propio Desktop, que es un objeto, le provee de herramientas gráficas que son intuitivamente fáciles de usar, hasta cuando se crean aplicaciones complejas.

Una de las herramientas que caracterizan a Borland, juega una parte importante en el diseño de la base de datos: los inspectores de Objetos (Object inspectors), le dejan pulsar con el botón derecho del ratón sobre un objeto en la pantalla e inspeccionar sus métodos y propiedades subyacentes. Adelantándose a otros productos de Borland, los inspectores de objetos de Paradox tienen una nueva característica de "alfiler" que los deja colocados en el área de trabajo para tener acceso fácil a los mismos.

Hay también poderosas herramientas de diseño visual para crear y editar tablas, modelos y reportes. Puede representar gráficamente las relaciones entre los datos si

selecciona archivos y crea enlaces desde los campos de una tabla a otra. Los enlaces complejos se hacen algo tan simple como apuntar y pulsar con el ratón.

Hay una poderosa característica de QBE que deja primero describir visualmente la consulta (que es también un objeto que se puede volver a emplear), y optimizar automáticamente la consulta.

Una vez que se ha definido una consulta, es fácil definir modelos complejos o reportes basado en los resultados.

Paradox colocara automáticamente los campos de consulta en un modelo, aun en una relación "de uno a muchos".

Las relaciones "muchos" se colocaran en formato de tabla al pie del modelo.

Aunque los modelos prefijados son meramente funcionales, puede modificar el esquema existente para producir la salida con calidad de presentación.

También tiene un vasto juego para opciones de diagramación y es fácil combinar los datos y los gráficos en un reporte.

El núcleo de Paradox es la máquina Interbase de Borland, que usa la ODAPI (Interfaz abierta para programación de base de datos de Borland) para proveer acceso a archivos en diferentes formatos como si fueran los propios.

Los archivos en formatos diferentes pueden enlazarse de forma transparente para lograr consultas de multiarchivos.

Los manejadores de ODAPI para Paradox y Dbase vienen con el producto.

La ODAPI no es una solución del mínimo común denominador. Por ejemplo, los

archivos .DBF que administra Paradox pueden incluir expresiones en sus archivos de índice, algo que ellos no pueden realizar normalmente.

La opción de integridad de referencia asegura que un campo o grupo de campos en una tabla "hija" se refiere a una clave de índice de la tabla "madre". Cuando se activa la comprobación de integridad de referencia, Paradox aceptará solo valores en la tabla "hija" que ya existen en el índice de la tabla "madre".

Paradox guarda los enlaces de integridad de referencia en un archivo con el nombre de la tabla y una extensión .VAL. Se crea un índice de los campos de integridad de referencia cuando se guarda el archivo de definición.

Puede definirse la integridad de referencia solo entre tablas de Paradox y las tablas deben ubicarse en el mismo directorio.

Como Paradox apoya directamente los archivos .DBF, no hay opciones de importación/exportación para ellos.

Paradox le brinda la capacidad de usar campos tradicionales de anotaciones, así como campos de anotaciones que mantienen un rico formato de texto.

Los campos de gráficas pueden contener cualquiera de los formatos gráficos populares de Windows y DOS, como .PCX, .BMP, .EPS, .GIF y .TIF.

Paradox puede importar y exportar datos desde la mayoría de las hojas de cálculo populares y delimitar modelos de texto, también cuenta con apoyo para importar los archivos de longitud fija que usualmente producen los MainFrames.

Paradox crea nuevas tablas para almacenar los archivos importados.

ObjectPAL es un superconjunto de PAL (Lenguaje de Aplicaciones de Paradox), el cual

es mucho más que un lenguaje activado por eventos, basado en objetos. Los programas de Object PAL se escriben añadiendo porciones de código (snippets) a objetos de interfaz, de forma similar a la de Microsoft Visual Basic. Cada objeto tiene un conjunto de métodos incluidos para modificar su comportamiento, y los programadores pueden crear sus propios métodos para manejar situaciones específicas.

El trazador de errores de Paradox, solo le permite al usuario ejecutar el código paso a paso. Los programadores de Paradox, pueden seguir la ejecución del programa, fijar puertos de parada e inspeccionar variables. El trazador de errores puede entrar en niveles anidados de código, o saltar por encima de secciones que sabe que no tienen problemas. También puede seleccionar cuales métodos incorporados quiere inspeccionar o ignorar.

Paradox incluye un servicio de ayuda en línea sobre el lenguaje, pero este muestra solo la sintaxis de llamada a los métodos. Depende de los manuales para buscar los ejemplos y descripciones de las funciones del lenguaje.

## **En la parte del Front End se evaluaron Visual C++, Visual Basic y Delphi**

### **Visual C++**

Visual C++ cuenta con herramientas de programación visuales impresionantes, pero todavía se tiene que saber programación de Windows.

Sus herramientas visuales App Wizard, Class Wizard y el editor de recursos App Studio, de diseño excelente están disponibles dentro del WWB (Wizard Workbench).

Parte de la integración se logró a través de un manejador de dispositivo virtual.

Todas las herramientas de visualización de Visual C++ son para trabajos en Windows.

Puede generar aplicaciones para DOS si se tiene la edición profesional, pero no es en eso en lo que Visual C++ se destaca. Las herramientas App Wizard y Class Wizard dependen de la biblioteca Microsoft Foundation Class (MFC), que incluye la mayor parte del juego de programación de Windows (SDK) en una estructura de clase de C++. Si se quiere programar solamente en C o escribir directamente a la API (Interfaz de programación de aplicaciones) de Windows, los Wizards no le serán útiles.

El App Wizard le permite seleccionar funciones básicas como "Printing = impresión", "Previewing = vista anticipada" y "Context Sensitivity Help = ayuda sensible al contexto" y luego generar código basado en MFC.

Es fácil generar la estructura de una aplicación, personalizarla requiere un amplio conocimiento del lenguaje C++ y de MFC. Los programas Wizard son herramientas buenas para el aprendizaje y pueden aumentar considerablemente la productividad de aquellos que tienen una buena base de C++ y MFC, pero no se protege al usuario de los peligros internos de Windows.

La herramienta más importante del C++ es App Studio, que es un editor de recursos maravilloso. Puede manejar cualquier recurso y es completamente Visual. Con el App Studio, la creación de un menú cambia totalmente, ya que es más amigable para el usuario.

### **Visual Basic**

Visual Basic 5 (versión sólo de 32 bits) concluye toda una generación en este lenguaje. Con su remozada interfaz (que puede regresarse a la de las versiones anteriores) incluye muchas más opciones que VB4. La principal novedad, por la que es buscado, es la creación de controles ActiveX, que debió incluirse debido a las presiones

comerciales recibidas por Delphi (la versión visual de Pascal) de Borland, que captó muchos desarrolladores desilusionados con el atraso de VB4. También es muy promocionada el aumento en el performance, debido a la posibilidad de generar código compilado en vez de código interpretado como en las anteriores versiones. Aclaremos que, a pesar de tratarse de código compilado, es necesario seguir incluyendo el "runtime". En aplicaciones de distribución normal (vía diskettes, o CD-ROM) no se nota el tamaño de este archivo, pero si se desea crear controles ActiveX para incluir en páginas HTML, ocupando 1.5 MB a veces es necesario pensar en otras herramientas de desarrollo. Microsoft aclaró que se incluirá en la próxima versión de su explorador (Internet Explorer 4.0) y en la actualización de Windows 95 (de nombre-código Memphis) y NT (de nombre código Cairo).

Denominado "Dream Release = la versión soñada" por el equipo de desarrolladores, incluye muchas novedades, como la posibilidad de armar otra clase de componentes, ActiveX Documents, que cierran la historia del MDI, permitiendo aplicaciones realmente centralizadas en los documentos OLE, que pueden incluirse en cualquier contenedor de objetos OLE (como por ejemplo una página HTML).

Este lenguaje ha crecido conforme aparece una nueva versión. Se orienta, sobre todo, al desarrollo empresarial y de redes, como Intranet, Extranet e Internet. Además, incluye diversas utilerías que la integran, incluso, una serie de herramientas de programación.

Cuenta con cuatro diferentes ediciones que se adaptan a los requerimientos del usuario:

**Working Model**, que es gratuita y puede obtenerse desde la Red. Esta edición es más apta para conocer el entorno y ver algunas de sus características, que para utilizarse en un ambiente de producción.

**Aprendizaje** (antes Estándar), aquella que se consigue junto con una serie de libros y que sirve para aprender programación en Visual Basic. Incluye varias funciones completas (se pueden generar ejecutables), aunque carece de suficientes características para generar aplicaciones de tipo profesional, está es la más utilizada en el medio de los desarrolladores. Incluye gran parte de las herramientas necesarias para generar aplicaciones comerciales y profesionales aptas para cubrir casi cualquier necesidad.

**Empresarial**, la más completa. Incluye herramientas que se utilizan en el entorno empresarial, como acceso a datos en redes LAN. También cuenta con la capacidad de crear grupos de desarrollo controlados (con Visual SourceSafe) y algunos otros aspectos del desarrollo empresarial, como la conexión directa a bases de datos SQL Server y Oracle.

Las tres últimas pueden encontrarse en el idioma español, así como la documentación, herramientas de ayuda, mensajes y utilerías diversas. (El lenguaje de programación permanece en inglés.)

## **Delphi**

Este lenguaje nació grande y, más que crecer, se ha pulido en cada versión. Se apoya en Object Pascal, con lo que es posible acceder a estructuras de datos y a la programación orientada a objetos (OOP).

Permite, además, generar aplicaciones en modo de consola, así como aplicaciones visuales. Con algunas adaptaciones, se pueden ejecutar programas generados en versiones anteriores de Pascal que no hagan uso de TurboVision (con grandes adaptaciones, estas aplicaciones también pueden compilarse).

Desde que apareció, ofreció la generación de ejecutables autónomos (que no requieren de archivos en tiempo de ejecución) y un modelo de objetos basado en VCL ("Virtual Component Library = Componente de librerías virtuales"), aunque también soporta VBX (en Delphi 1), OCX y ActiveX sin mayor problema, así como los esquemas OLE y COM, y otras propuestas como CORBA. Además, permite el desarrollo de aplicaciones que tienen que ver con Internet. Ofrece tres suites:

**Estándar**, que es el entorno básico, orientado al aprendizaje y a su uso por parte de desarrolladores que no requieren de muchas características.

**Profesional**, dirigida al grueso de los desarrolladores (la que más se distribuye). Permite la generación de controles OCX, ActiveX y VCL, así como la emulación de entornos de datos distribuidos mediante el Interbase.

**Cliente-Servidor**, dirigida a usuarios poderosos y de entorno empresarial. Incluye el control de código fuente y grupos de desarrollo, así como el acceso a bases de datos distribuidas, muy en especial de Oracle.

Entre las desventajas de Delphi se encuentra el no tan extenso soporte a nivel mundial y la relativa dificultad para encontrar desarrolladores especializados. A su vez, no existen versiones en español (aunque algunos pueden dar una importancia mínima a este detalle).



## 2.7. Elección de la solución óptima

Después de haber evaluado las alternativas anteriormente descritas se tomo la decisión de llevar a cabo el desarrollo del sistema de control de inventarios en Access con Visual Basic por las siguientes razones:

### Tablas comparativas Back-End :

Acceso a información	Access	FoxPro	Paradox
Habilidad para trabajar con varios formatos de datos	Si Requiere instalación adicional de Office 97	Si	Si
Completa conectividad con todos los formatos de datos	No permite el control total de la administración de datos, a menos que se importen a Access	Si	Capacidades muy restringidas
Crea Nuevas tablas en cualquier formato	Si	No	No
Salva las tablas en cualquier formato de base de datos	Si	No	No

Tabla 2.7.1. Comparativo de acceso a información.

<b>Productividad</b>	<b>Access</b>	<b>FoxPro</b>	<b>Paradox</b>
Smart Masters para facilitar el desarrollo inicial de aplicaciones	Si	Si	Si Selección Limitada
InfoBox Para personalización de todos los objetos	Si Altamente Intuitivo	Si	Si Cambios No aplicados Inmediatamente
Asistentes Expertos para la simplificación de tareas complicadas	Si Menos Flexible	Si	Si Menos Poder
Asistente paso a paso con consultas complejas	Si Menos Intuitivo	No	No
Consultas preprogramadas (encontrar duplicados, valores máximos, etc.)	Si Pocas Ocasiones	No	No
Asistente de reportes paso a paso.	Si Menos Flexible	No	Si Layouts y Estilos Pequeños
Ambiente paso a paso para la creación de tablas cruzadas.	Si Pocos campos por dimension	No	No

Tabla 2.7.2. Comparativo de productividad.

Producto	Compañía
Visual FoxPro	Microsoft Corporation
Paradox	Borland Inc.
Access	Microsoft Corporation

Tabla 2.7.3. Compañía del producto.

Internet/Intranet	Access	FoxPro	Paradox
Publica visitas estáticas al Internet	Si	No	No
Crea páginas Web interactivas que acceden dinámicamente a datos.	Si	No	No
Búsqueda de texto en Internet de bloques de texto seleccionados	No	No	No
Conectividad a un sitio Web con un simple click.	Si	No	No

Tabla 2.7.4. Comparativo Internet/Intranet.

<b>Personalización</b>	<b>Access</b>	<b>FoxPro</b>	<b>Paradox</b>
"InfoBox = Caja Información" para una fácil personalización de todos los objetos.	Si Altamente Intuitivo	Si	Si  Los cambios no se hacen automáticamente
Asistentes/expertos para construir rápidamente formas, reportes y demás.	Si  Menos Flexible	No	Si
Un verdadero reporteador para la edición rápida y precisa de los reportes.	Si	No	No
Variedad de validaciones de campos y valores de default.	Si	Si	No
Soporte de lenguaje de programación visual.	Si	Si	Si

Tabla 2.7.5.Comparativo de personalización.

Otras	Access	FoxPro	Paradox
Compatibilidad con Dbase para dos	Si	No	No
Ayuda en línea	Si	No	No
Expertos/Wizzard	Si	Si	Si
Uso de controles	Dll, Vbx	No	No
Acceso transparente a datos locales y SQL.	Si	Si	Si

Tabla 2.7.6. Comparativo de otras características.

**Tablas Comparativas del Front -End.**

Producto	Compañía
Visual Basic 5.0	Microsoft Corporation
Visual C++	Microsoft Corporation
Delphi	Borland Inc.

Tabla 2.7.7. Compañía del producto.

Producto	Facilidad de Aprendizaje
Visual Basic 5.0	Bueno
Visual C++	Regular
Delphi	Regular

Tabla 2.7.8. Evaluación de Facilidad de uso.

Producto	Evaluación de seguridad
Visual Basic 5.0	Bueno
Visual C++	Bueno
Delphi	Regular

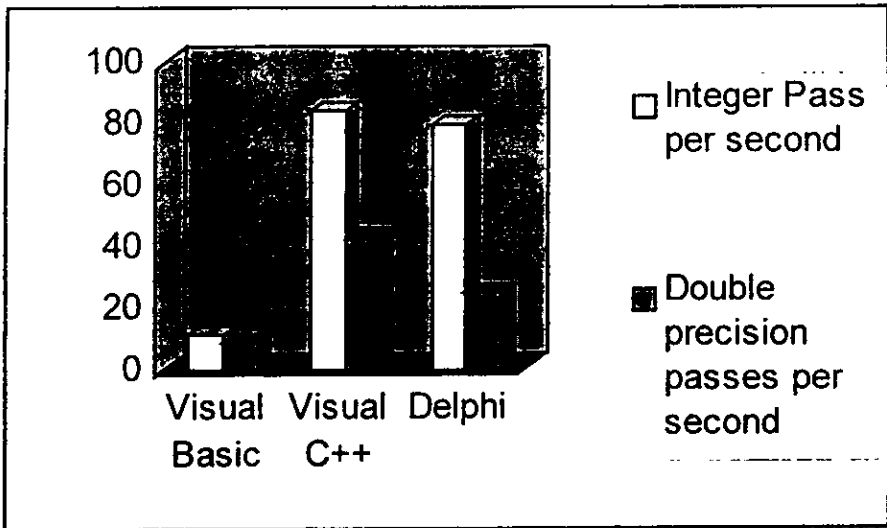
Tabla 2.7.9. Evaluación de Seguridad.

	Test	Visual Basic	Delphi	Power Builder
Display	Empty display	52.509	26.85	86.667
	Labels	0.316	1.295	4.748
	Bitmaps	1.226	3.296	8.925
Database	Overhead	9.75	7.838	6.935
	Small retrieval	0.34	0.612	0.393
	Large retrieval	0.25	0.11	0.161
	In process method	0.000295	0.082	N/A
	Out of process let	1.066	2.572	N/A
	Out of process method	1.039	2.572	N/A
Lenguaje	Sieve of Eratoshenes	360.625	3323.82	16940.12
	String manipulation	900.35	48.13	1941.36

Tabla 2.7.10. Fuente: Carnegie Technology Group, Inc. ( los números pequeños son mejores).

	Integer passes per second	Double precision passes per second
Visual Basic	11.1	8.8
Visual C++	84	42.6
Delphi	74.9	24.9

Tabla 2.7.11. Fuente: CarnegieTechnology Group, Inc. (los números pequeños son mejores).



Gráfica 2.7.1. Interpretación gráfica de la tabla 2.7.11.

Se utilizara Access 97 porque es un programa poderoso y fácil de usar, la empresa telefónica tiene las licencias de Access 97, Visual Basic 5.0 y Windows 98.

De acuerdo al análisis (tablas 2.7.1. – 2.7.11 y el gráfico 2.7.1) no existen razones suficientes para adquirir un nuevo software de bases de datos, ni de interfaz gráfica, puesto que por lo mencionado anteriormente ya se contaba con licencias del software.

El 80% del personal maneja o por lo menos conoce dichas herramientas.

VB es el lenguaje de programación visual con mayor soporte a nivel mundial. Se puede obtener ayuda e información por diversos medios y es relativamente sencillo encontrar desarrolladores especializados.

Hay quienes podrían decir que una de las grandes ventajas de Visual Basic sobre Delphi es su facilidad de aprendizaje y la seguridad que proporciona, lo que se ve reforzado por los comentarios de quienes opinan que aprender Delphi es casi tan complejo como aprender C.

La combinación de Access 97 con Visual Basic 5.0, para el manejo de un volumen de datos considerable (por ejemplo, con una base de datos de un millón de registros) el rendimiento es muy bueno.

Trabajar en Visual Basic 5.0 los programas se procesan con mayor velocidad.





## **CAPITULO 3. DESARROLLO E IMPLEMENTACION DEL SISTEMA**

### **3.1. Aplicación de la metodología elegida (Back End para cada módulo).**

#### **3.1.1. Diagrama de contexto.**

En la figura 3.1.1.1. se muestra el diagrama de contexto del Sistema de Control de Inventarios de nivel cero, en el cual se puede observar el flujo de información dentro del mismo.

Tiene como entradas a los usuarios, que es la entidad que define las necesidades en cuanto a materiales del inventario, y al SIREM que es la entidad en la cual se registran existencias, pedidos y movimientos.

Teniendo como salidas las entidades de:

Almacenes, que es la entidad en la cual se almacenan los materiales y que en un momento dado recibe instrucciones de traspasar materiales con otros almacenes.

Compras, que es la entidad encargada de realizar los pedidos a los proveedores.

Logística, que es la entidad encargada de planear los embarques de los materiales.

### DIAGRAMA DE CONTEXTO DE NIVEL 0

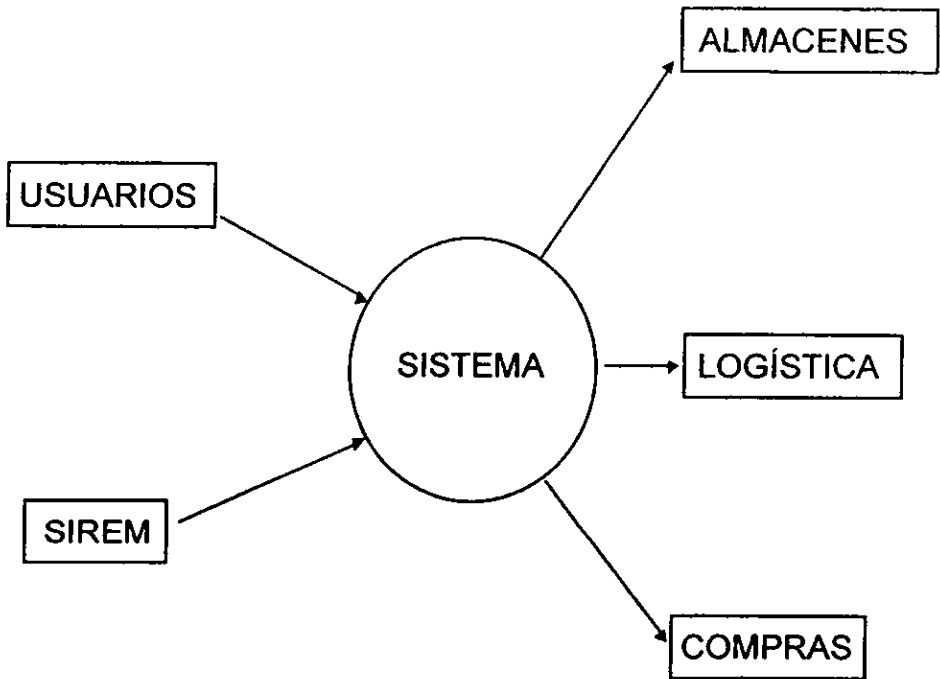
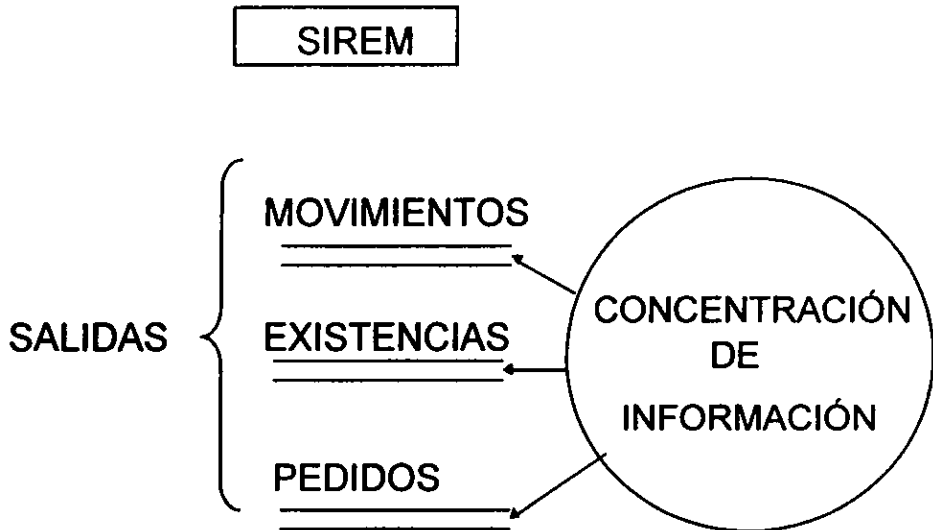


Figura 3.1.1.1. Diagrama de Contexto de Nivel 0.

#### 3.1.2. Diagramas de Flujo de datos

En la figura 3.1.2.1. se puede observar el diagrama de concentración de la información de nivel uno; se almacena en archivos del SIREM, diferenciados por movimientos, existencias y pedidos.

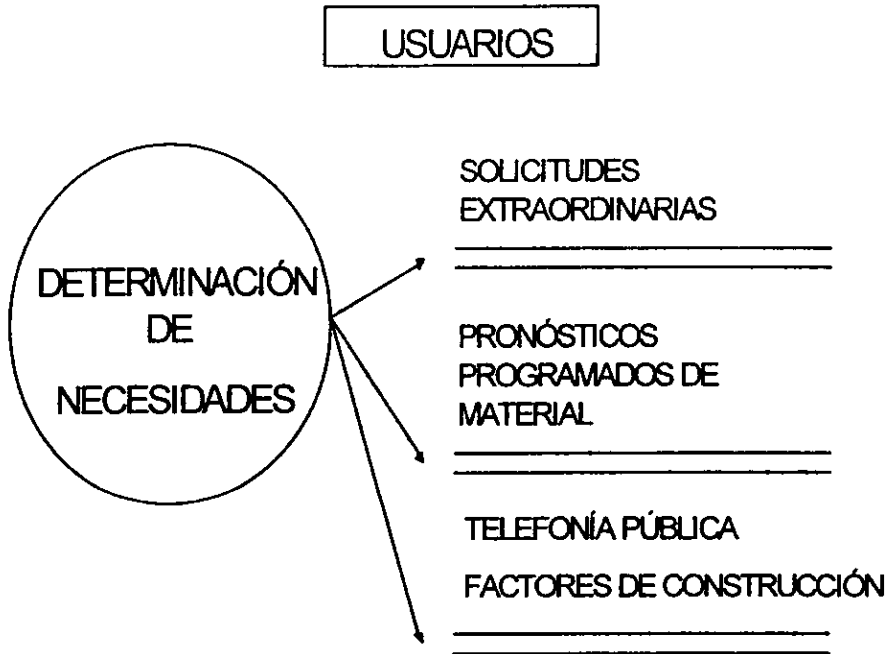
## NIVEL 1 CONCENTRACIÓN DE LA INFORMACIÓN



**Figura 3.1.2.1. Concentración de la información. Nivel 1.**

La figura 3.2.1.2. Muestra el diagrama de determinación de necesidades de nivel uno en el cual el proceso por medio del cual el usuario determina sus necesidades de solicitudes extraordinarias, pronósticos programados de material y factores de construcción.

## NIVEL 1 DETERMINACIÓN DE NECESIDADES



**Figura 3.1.2.2. Determinación de necesidades Nivel 1.**

La figura 3.1.2.3. Se observa el diagrama de movimiento de material de nivel uno en el cual el proceso de movimiento de material, que lleva a cabo el área de almacenes, hace una solicitud de traspaso por medio de la cual se efectúa el movimiento de los materiales

### NIVEL 1 MOVIMIENTO DE MATERIAL



Figura 3.1.2.3. Movimiento de material. Nivel 1.

En la figura 3.1.2.4. Se observa el diagrama de envío de material de nivel cero. El proceso de envíos de material, coordinada por el área de logística, se lleva a cabo por medio de solicitudes de traspaso, las cuales están ordenadas por destino.

### NIVEL 1 ENVIOS DE MATERIAL

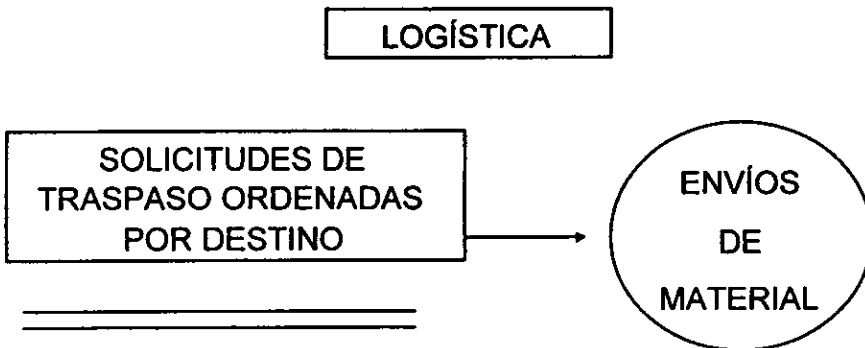
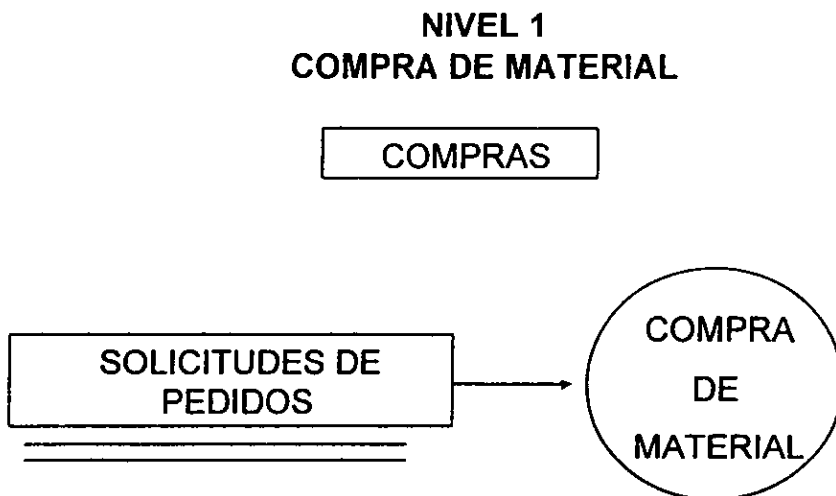


Figura 3.1.2.4. Envíos de Material. Nivel 1.

En la figura 3.1.2.5. se observa el diagrama de compra de material de nivel uno, el área de compras elabora solicitudes de pedidos, para llevar a cabo la compra de materiales.



**Figura 3.1.2.5. Compra de material. Nivel 1.**

La figura 3.1.2.6. representa el flujo de datos de la solución propuesta (Diagrama de nivel uno). En donde el SIREM es la misma entidad definida en el Diagrama de contexto.

Los almacenes proveen pronósticos de materiales (catálogos) a ser surtidos en las localidades que se requieran.

El área de Compras, realiza las compras de material.

De este diagrama de nivel cero se derivan los niveles del uno al once, los cuales se describirán y desglosarán en los siguientes diagramas.

Los diagramas que se mostrarán a continuación, únicamente se pueden desglosar hasta ese nivel, ya que no le siguen otro tipo de procesos. La secuencia es la siguiente:

- 1.- Preparación de pedidos y existencias.
- 2.- Preparación consumos mantenimiento (SIREM).
- 3.- Preparación consumos construcción (PE).
- 4.- Preparación consumos construcción (TP).
- 5.- Preparación pronósticos.
- 6.- Definición de reportes.
- 7.- Generación de reportes
- 8.- Analiza, revisa y evalúa reporte.
- 9.- Carga Catálogo.
- 10.- Mantenimiento almacén.
- 11.- Configuración conexión de archivos.

### DIAGRAMA DE FLUJO DE DATOS NIVEL 1

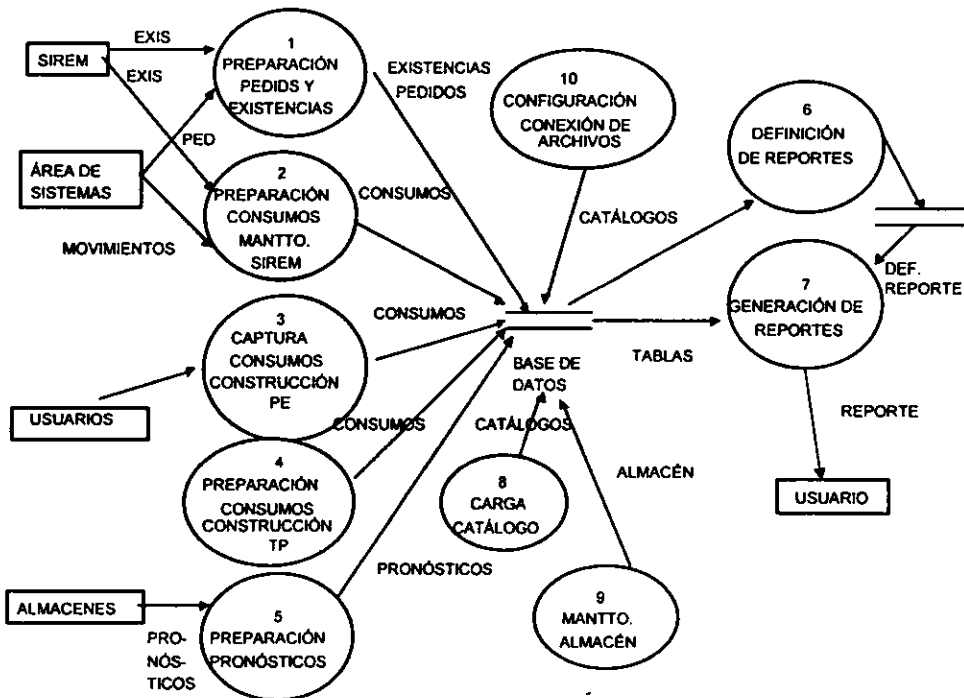


Figura 3.1.2.6. Diagrama de flujo de datos de la solución propuesta Nivel 1.



La figura 3.1.2.7. corresponde al diagrama de nivel dos de la burbuja 1: Preparación de pedidos y existencias de nivel cero; se puede observar el flujo de datos del primer proceso derivado del esquema de la figura 3.1.2.6, para llevar a cabo la generación de pedidos y existencias, el cual se compone de 4 procesos.

El primero es la generación de archivos de existencias; el segundo carga archivo de existencias, valida y completa información; el tercero carga archivo de pedidos, valida y completa información y el cuarto, recupera el archivo de existencias.

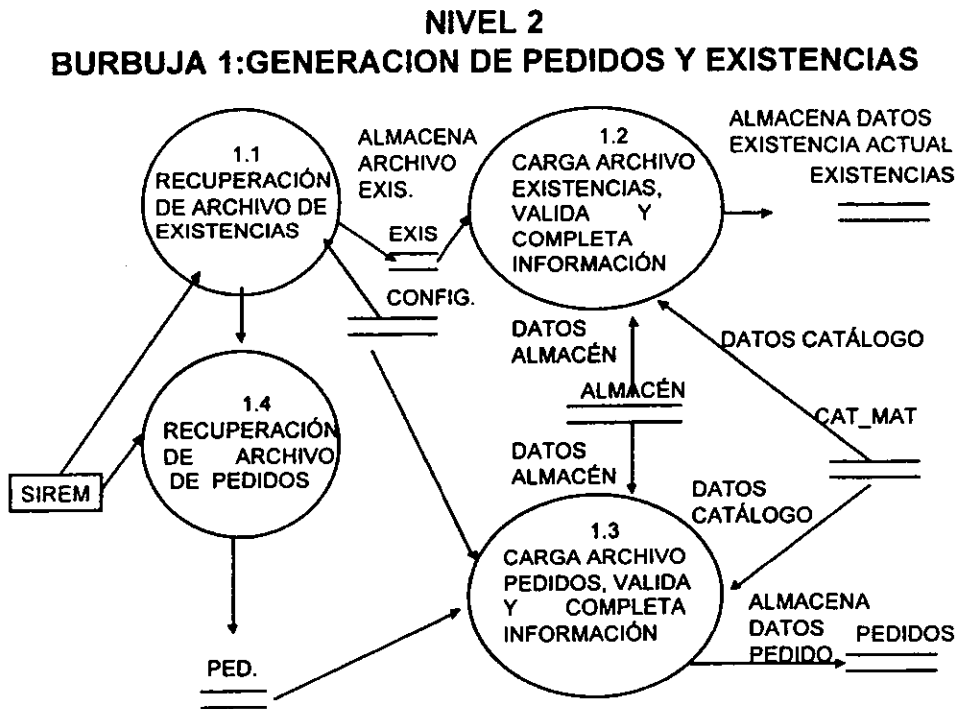


Figura 3.1.2.7. Burbuja 1: Generación de pedidos y existencias. Nivel 2.

La figura 3.1.2.8. Muestra el Diagrama de nivel dos de la burbuja 2: generación de consumos SIREM, el cual realiza el cálculo de éstos. Se consideran los procesos

actuales y los movimientos registrados. Y al haber obtenido los cálculos de consumo, se suman estos por periodo y se obtienen los promedios. Cuenta con los siguientes procesos : recuperación de archivo de movimientos vía FTP, carga a tablas de movimientos, calcula consumo.

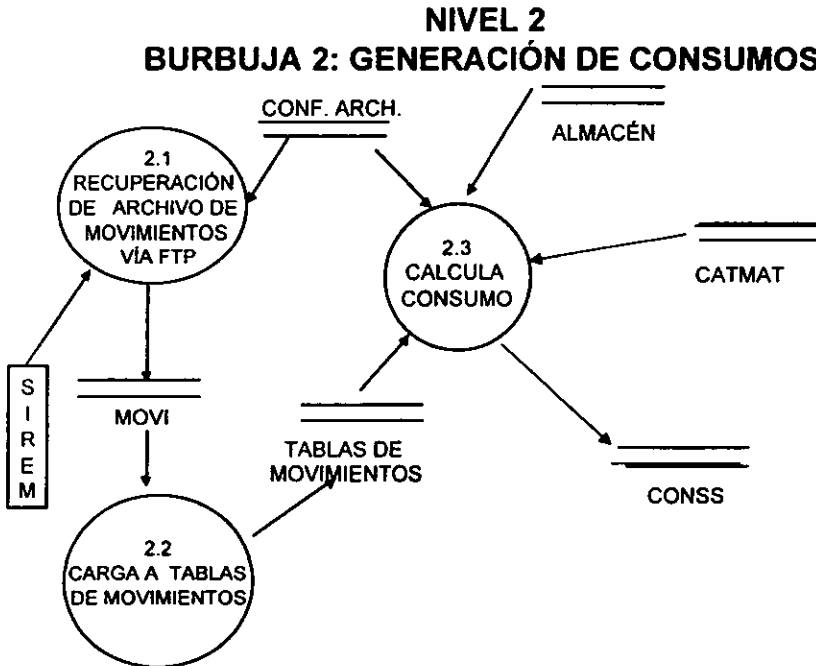


Figura 3.1.2.8. Burbuja2: Generación de consumos. Nivel 2.

La figura 3.1.2.9. muestra el diagrama de nivel 3 de la burbuja 2.3: calculo consumos en la que se muestran los proceso suma consumo por periodo, calcula promedio.

### NIVEL 3 BURBUJA 2.3: CALCULO CONSUMOS

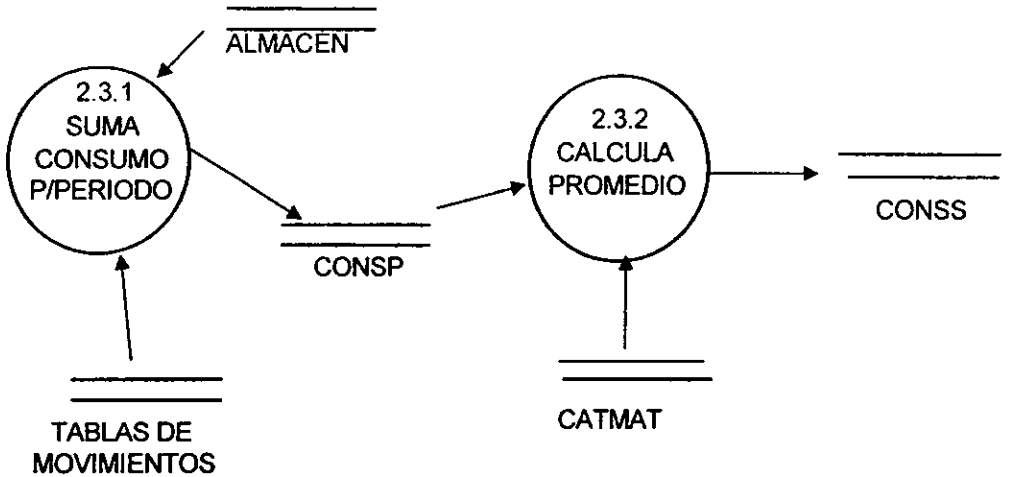
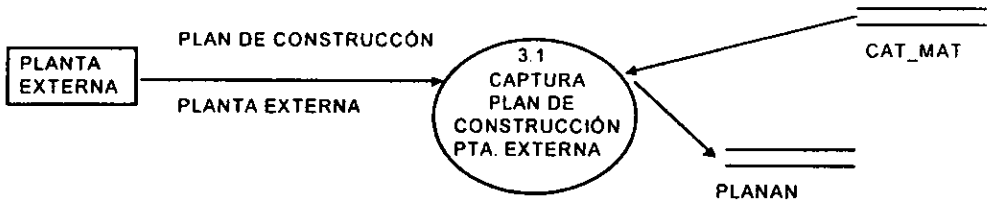


Figura 3.1.2.9. Burbuja 2.3: Cálculo consumos. Nivel 3.

La figura 3.1.2.10. muestra el diagrama de nivel dos de la burbuja 3: preparación de consumos construcción PE (Planta Externa) así como los datos y procesos involucrados para obtener los consumos en el área de construcción; aquí se consulta la información de catálogos y unidades de construcción. Dicha información es de tipo histórica y sirve para obtener estimados de consumo de material. En este se llevan a cabo el proceso de captura de plan de construcción planta externa.

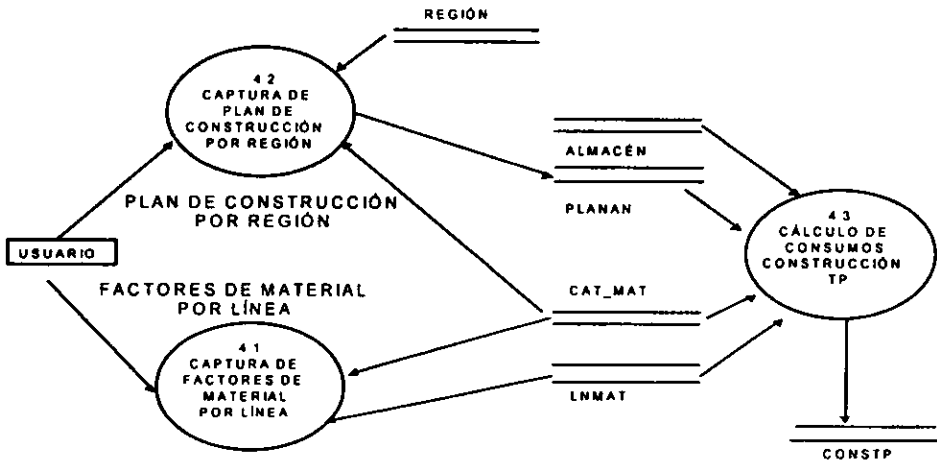
**NIVEL 2**  
**BURBUJA 3: PREPARACIÓN CONSUMOS CONSTRUCCIÓN**



**Figura 3.1.2.10. Burbuja 3: Preparación de consumos construcción PE (Planta Externa). Nivel 2**

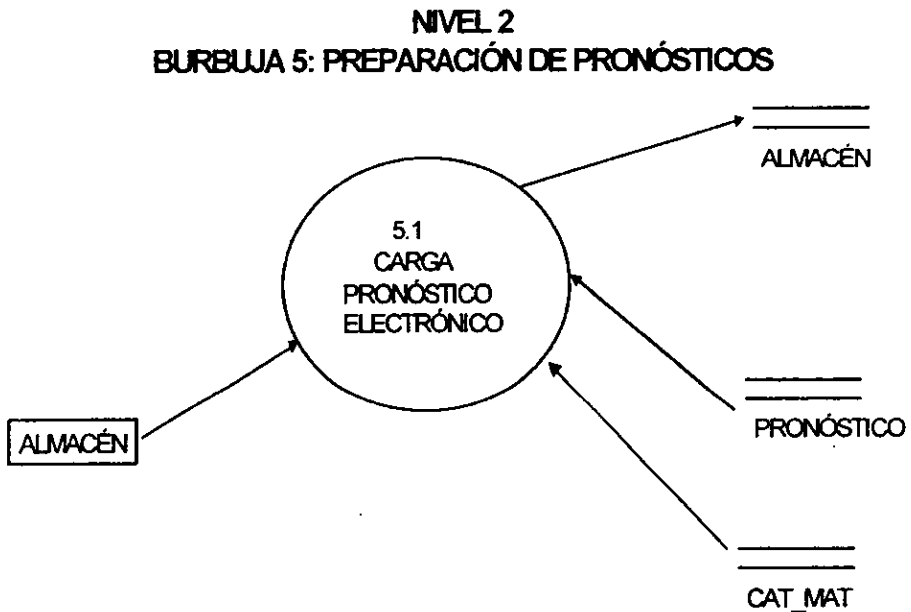
La figura 3.1.2.11. muestra el diagrama de nivel dos de la burbuja 4: generación de consumos, los cuales son conocidos mediante el plan de uso del área de telefonía pública. Cuenta con los procesos: captura del plan de construcción por región, la captura de factores de material y por último el cálculo de consumos construcción.

**NIVEL 2**  
**BURBUJA 4: PREPARACIÓN CONSUMOS CONSTRUCCIÓN**



**Figura 3.1.2.11. Burbuja 4: Generación de consumos. Nivel 2.**

La figura 3.1.2.12. muestra el diagrama de nivel 2 de la burbuja 5: preparación de pronósticos; en el cual los pronósticos son información proporcionada por los almacenes.



**Figura 3.1.2.12. Preparación de pronósticos.**

La figura 3.1.2.13. se muestra el diagrama de nivel 2 de la burbuja 6: definición de reportes; el operador captura las características del reporte y prepara el conjunto de catalogo.

## NIVEL 2 BURBUJA 6: DEFINICIÓN DE REPORTE

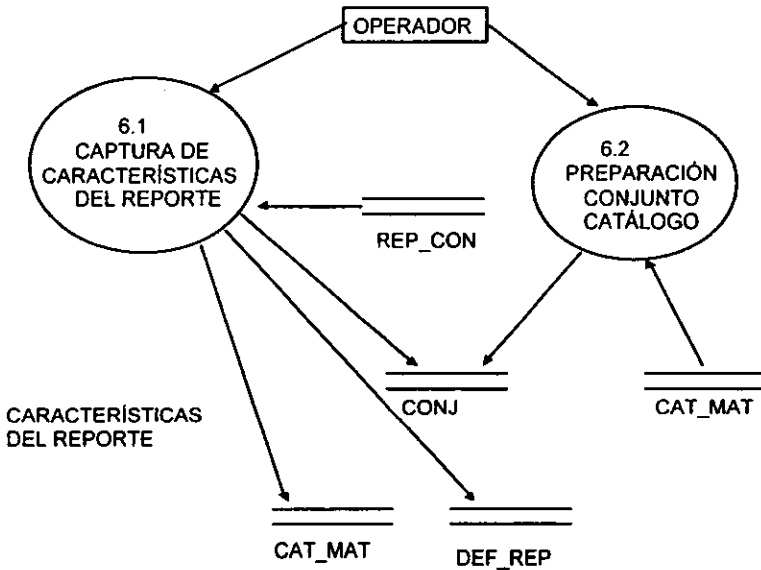
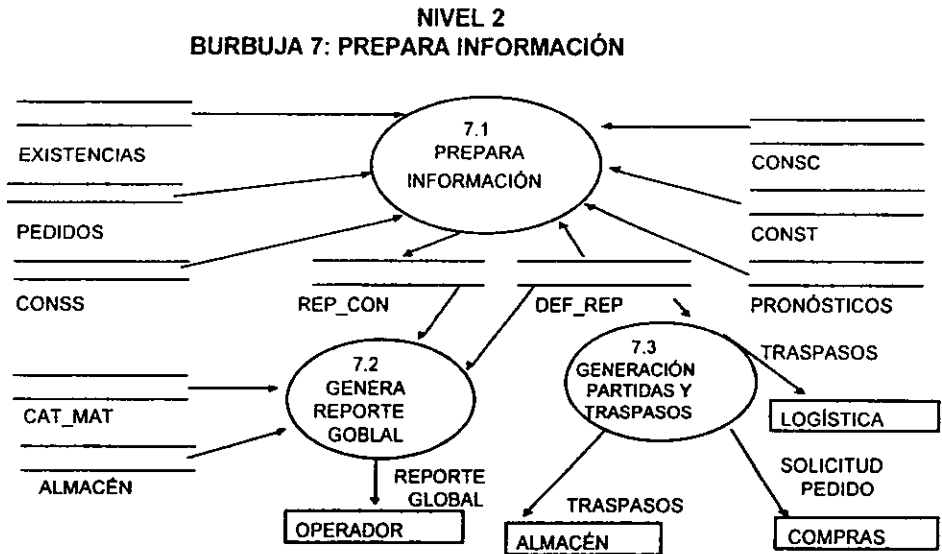


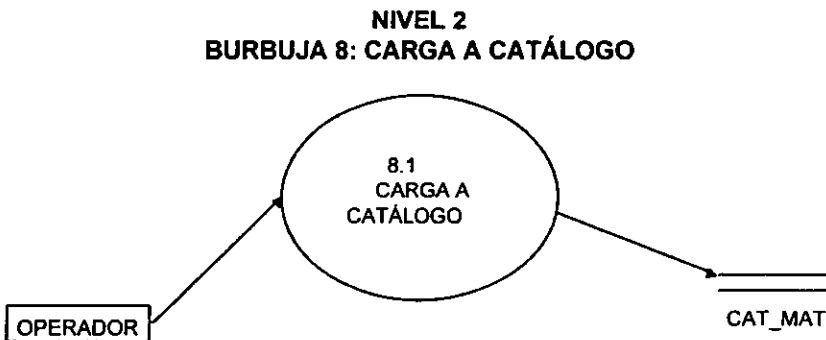
Figura 3.1.2.13 Burbuja 6: Definición de reportes. Nivel2.

En la figura 3.1.2.14. se muestra el diagrama de nivel 2 de la burbuja 7: preparación de reportes; se muestra el proceso de entidades de entrada y salida para la preparación de la información, para la obtención, del reporte general y la generación de partidas y traspasos. Contando con los procesos de preparación de información, generación de reporte global y generación de partidas y traspasos.



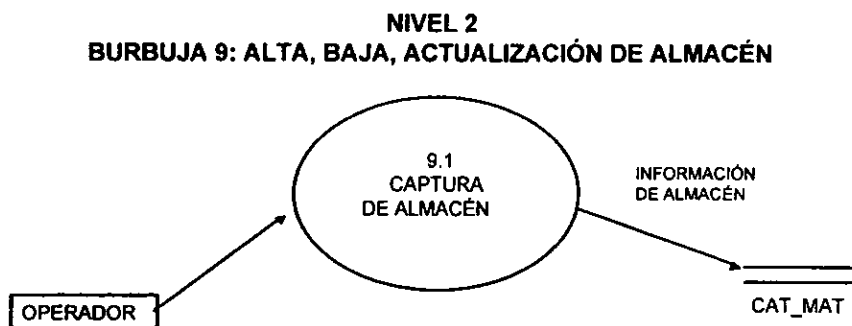
**Figura 3.1.2.14. Burbuja 7: Preparación de la información. Nivel 2.**

En la figura 3.1.2.15. se muestra el diagrama de nivel 2 de la burbuja 8: carga de catálogos. Se muestra la carga de catálogos, en donde la información proviene de un archivo de dBase.



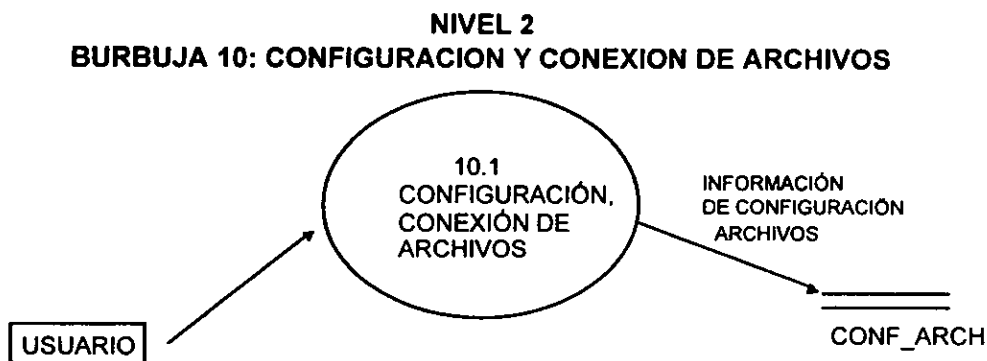
**Figura 3.1.2.15. Burbuja 8: Carga de catálogos. Nivel 2.**

La figura 3.1.2.16. muestra el diagrama de nivel 2 de la burbuja 9: mantenimiento de almacenes: muestra como se lleva a cabo el mantenimiento de almacenes, lo que básicamente es una captura de información.



**Figura 3.1.2.16. Burbuja 9: Mantenimiento de almacenes. Nivel 2.**

La figura 3.1.2.17. muestra el diagrama de nivel 2 de la burbuja 10: configuración, conexión y archivos de entrada. Muestra como se lleva a cabo la configuración, conexión y archivos de entrada.



**Figura 3.1.2.17. Burbuja 10: Configuración, conexión y archivos de entrada. Nivel 2.**



### 3.1.3. Diccionario de Datos

Diccionario de datos: sirve para describir los datos manejados por el sistema, sus características y posibles valores.

#### Tabla: adms

Propiedades

Def. actualizable: TRUE

RecordCount: 0

Columnas

Nombre	Tipo	Tamaño	Descripción
adm	Texto	12	Clave de proyecto

AllowZeroLength: FALSE

Atributos: Longitud variable

Campo de origen: adm

Orden de combinación: General

Posición ordinal: 0

Required: FALSE

Tabla de origen: adms

Índices de tabla

Nombre: Número de campos

Primarykey: 1

Agrupado: FALSE

Cuenta distinta: 0

Extranjero: FALSE

Ignorar Nulos: FALSE

Nombre: Primarykey

Principal: FALSE

Required: FALSE

Única: TRUE

Campos: adm, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins

Users

#### Tabla: almacen

Propiedades

Def. actualizable: TRUE

RecordCount: 0

Columnas

Nombre	Tipo	Tamaño
--------	------	--------

cod_sirem	Texto	3	ESTA ES LA LLAVE*
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cod_sirem		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	almacen		
cod_saci	Texto	3	Almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cod_saci		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	almacen		
nombre	Texto	30	Nombre de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	nombre		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	almacen		
grupodist	Texto	2	Grupo distante de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	grupodist		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	almacen		
region	Texto	15	Región de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		

En el dato región, encontramos sus propiedades en su primera columna, y posteriormente tenemos sus valores, tamaño y descripción.

<b>Tabla: almacen</b>			
Tabla de origen:	almacen		
subgrupo	Texto	15	Subgrupo de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	subgrupo		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	almacen		
categoria	Texto	2	Categoría de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	categoria		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	almacen		
division	Texto	15	División de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	division		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	almacen		
nuevadisvis	Texto	15	Nueva división de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	nuevadisvis		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	almacen		
alm_dist	Número (entero)	2	Almacen distante
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		

Campo de origen: alm\_dist  
 Orden de combinación: General  
 Posición ordinal: 9  
 Required: FALSE  
 Tabla de origen: almacen

**Índices de tabla**

Nombre Número de campos  
 Primarykey 1  
 Agrupado: FALSE

Cuenta distinta: 0

**Tabla: almacen**

Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: Primarykey  
 Principal: FALSE  
 Required: FALSE  
 Única: TRUE  
 Campos: cod\_sirem, Ascen.

**Permisos de usuario**

admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: cat\_tem**

**Propiedades**

Def. actualizable: TRUE :  
 RecordCount: 0  
 Columnas  
 Nombre Tipo Tamaño  
 llave\_extra Texto 7 \* Es la llave extra del sist.  
 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: llave\_extra  
 Orden de combinación: General  
 Posición ordinal: 0  
 Required: FALSE  
 Tabla de origen: cat\_tem  
 catalogo Texto 7 Catalogo temporal  
 AllowZeroLength: FALSE

Atributos: Longitud variable  
 Campo de origen: catalogo  
 Orden de combinación: General  
 Posición ordinal: 1  
 Required: FALSE  
 Tabla de origen: cat\_tem  
 tipo\_cat Texto 1 \*1=Catalogo;2=Conjunto

AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: tipo\_cat  
 Orden de combinación: General

Posición ordinal: 2  
 Required: FALSE  
 Tabla de origen: cat\_tem  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: catalogo**

Propiedades

Def. actualizable: TRUE  
 RecordCount: 0

Columnas

Nombre	Tipo	Tamaño	
catalogo	Texto	7	Catalogo de materiales

AllowZeroLength: FALSE  
 Atributos: Longitud variable

Campo de origen: catalogo  
 Orden de combinación: General

Posición ordinal: 0  
 Required: FALSE

Nombre	Tipo	Tamaño	
unidad	Texto	3	Unidad de materiales

AllowZeroLength: FALSE  
 Atributos: Longitud variable

Campo de origen: unidad  
 Orden de combinación: General

Posición ordinal: 1  
 Required: FALSE

Tabla de origen: catalogo

descripcio	Texto	30	Descripción de materiales
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	descripcio		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	catalogo		
condicion	Texto	3	Condición de materiales
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	condicion		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	catalogo		
Índices de tabla			
Nombre	Número de campos		
Primarykey	1		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	Tabla: catalogo		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	catalogo, Ascend.		
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
	Tabla: con_cat		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			

Nombre conjunto	Tipo Texto	Tamaño 7	Conjunto de categorías
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	conjunto		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	con_cat		
catalogo	Texto	7	Categoría de conjunto
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	con_cat		
Índices de tabla			
Nombre	Número de campos		
Primarykey	2		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	conjunto, Ascend.		
catalogo, Ascendente			
Permisos de usuario			
admin			
	<b>Tabla: con_cat</b>		
Permisos de grupo			
Admins			
Users			
	<b>Tabla: conf_arch</b>		
Propiedades			
Def. actualizable:	TRUE		

RecordCount:	3		
Columnas			
Nombre	Tipo	Tamaño	
nombre	Texto	12	Nombre de configuración
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	nombre		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	conf_arch		
dir_ip	Texto	30	Dirección lógica
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	dir_ip		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	conf_arch		
puerto	Número (entero)	2	Puerto del sistema
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	puerto		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	conf_arch		
clave_acceso	Texto	20	PW de entrada al sistema
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	clave_acceso		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	conf_arch		
pasaporte	Texto	20	Clave de entrada al sistema
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	pasaporte		
Orden de combinación:	General		
Posición ordinal:	4		



Required:	FALSE		
	<b>Tabla: conf_arch</b>		
Tabla de origen:	conf_arch		
ruta_remota	Texto	100	Ruta directa
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ruta_remota		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	conf_arch		
ruta_local	Texto	100	Ruta del lugar donde se encuentra el sistema
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ruta_local		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	conf_arch		
formato_dbase	Sí/No	1	Configuración en formato de dbase
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	formato_dbase		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	conf_arch		
Índices de tabla			
Nombre	Número de campos		
Primarykey	1		
Agrupado:	FALSE		
Cuenta distinta:	3		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	nombre, Ascend.		
Permisos de usuario			

admin

**Tabla: conf\_arch**

Permisos de grupo

Admins

Users

**Tabla: conjunto**

Propiedades

Def. actualizable: TRUE

RecordCount: 0

Columnas

Nombre	Tipo	Tamaño	
conj_id	Texto	7	Este campo es la llave

AllowZeroLength: FALSE

Atributos: Longitud variable

Campo de origen: conj\_id

Orden de combinación: General

Posición ordinal: 0

Required: FALSE

Tabla de origen: conjunto

descripcion	Texto	50	Descripción de conjunto
-------------	-------	----	-------------------------

AllowZeroLength: FALSE

Atributos: Longitud variable

Campo de origen: descripcion

Orden de combinación: General

Posición ordinal: 1

Required: FALSE

Tabla de origen: conjunto

Índices de tabla

Nombre Número de campos

conj\_id 1

Agrupado: FALSE

Cuenta distinta: 0

Extranjero: FALSE

Ignorar Nulos: FALSE

Nombre: conj\_id

Principal: FALSE

Required: FALSE

Única: FALSE

Campos: conj\_id, Ascend.

Primarykey 1  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: Primarykey  
 Principal: FALSE  
 Required: FALSE  
 Única: TRUE  
 Campos: conj\_id, Ascend.  
 Campos: conj\_id, Ascend.

**Tabla: conjunto**

Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

Propiedades **Tabla: conmtp**  
 Def. actualizable: TRUE  
 RecordCount: 0

Columnas	Tipo	Tamaño	
Nombre catalogo	Texto	7	*Código de catalogo
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	conmtp		
region	Texto	15	*Código de región
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	conmtp		
mes	Número (largo)	4	Mes de consumo en

AllowZeroLength:	FALSE		construcción
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	conmtp		
anio	Número (largo)	4	Año de consumo en construcción.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	anio		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	conmtp		
consumoprom	Número (doble)	8	Consumo promedio a 6 meses
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consumoprom		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
	Tabla: conmtp		
Tabla de origen:	conmtp		
Índices de tabla			
Nombre	Número de campos		
Primarykey	4		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	catalogo, Ascend.		
region, Ascendente			
mes, Ascendente			
anio, Ascendente			
Permisos de usuario			
admin			

Permisos de grupo  
Admins  
Users

**Tabla: consc**

Propiedades				
Def. actualizable:	TRUE			
RecordCount:		0		
Columnas				
Nombre		Tipo	Tamaño	
fechaintro	Fecha/Hora		8	Dia,mes,año de calculo e introducción a sistema
AllowZeroLength:	FALSE			
Atributos:	Tamaño fijo			
Campo de origen:	fechaintro			
Orden de combinación:	General			
Posición ordinal:		0		
Required:	FALSE			
Tabla de origen:	consc			
catalogo	Texto		7	*Código de catalogo
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	catalogo			
Orden de combinación:	General			
Posición ordinal:		1		
Required:	FALSE			
Tabla de origen:	consc			
cod_sirem	Texto		3	*Código de almacén
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	cod_sirem			
Orden de combinación:	General			
Posición ordinal:		2		
Required:	FALSE			
Tabla de origen:	consc			
mes_anio	Texto		6	* Mes,Año(MM,AAAA)
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	mes_anio			
Orden de combinación:	General			
Posición ordinal:		3		
Required:	FALSE			
Tabla de origen:	consc			

consumo	Número (doble)	8	Consumo en construcción de planta externa
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consumo		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		

**Tabla: consc**

Tabla de origen:	consc		
Índices de tabla			
Nombre	Número de campos		
Primarykey	4		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	fechaintro, Ascend.		
catalogo, Ascendente			
cod_sirem, Ascendente			
mes_anio, Ascendente			

Permisos de usuario  
admin

Permisos de grupo  
Admins  
Users

**Tabla: consp2**

Propiedades				
Def. actualizable:	TRUE			
RecordCount:	0			
Columnas				
Nombre	Tipo	Tamaño		
catalogo	Texto	7		*Código de catalogo
AllowZeroLength:	FALSE			

Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	consp2		
region	Texto	15	*Código de región
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	consp2		
sumcon	Número (doble)	8	Suma de consumo
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	sumcon		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	consp2		
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
<b>Tabla: conss</b>			
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	Dia,mes,año de cálculo e introducción a sistema.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	conss		

catalogo	Texto	7	*Código de catalogo
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	conss		
almacen	Texto	3	Almacen de consumo
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	almacen		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	conss		
consumprom	Número (doble)	8	Consumo promedio a 6 meses.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consumprom		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	conss		
Índices de tabla			
Nombre	Número de campos		
Primarykey	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
	<b>Tabla: conss</b>		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	fechaintro, Ascend.		
catalogo, Ascendente			
almacen, Ascendente			
Permisos de usuario			
admin			
Permisos de grupo			



Admins  
Users

**Tabla: consspron**

Propiedades				
Def. actualizable:	TRUE			
RecordCount:		0		
Columnas				
Nombre	Tipo	Tamaño		
catalogo	Texto	7		*Código de catalogo
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	catalogo			
Orden de combinación:	General			
Posición ordinal:		0		
Required:	FALSE			
Tabla de origen:	consspron			
almacen	Texto	3		Almacen de consumo promedio.
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	almacen			
Orden de combinación:	General			
Posición ordinal:		1		
Required:	FALSE			
Tabla de origen:	consspron			
consumprom	Número (doble)	8		Consumo promedio a 6 meses
AllowZeroLength:	FALSE			
Atributos:	Tamaño fijo			
Campo de origen:	consumprom			
Orden de combinación:	General			
Posición ordinal:		2		
Required:	FALSE			
Tabla de origen:	consspron			
region	Texto	15		*Código de región
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	region			
Orden de combinación:	General			
Posición ordinal:		3		
Required:	FALSE			
Tabla de origen:	consspron			
Índices de tabla				

Nombre	Número de campos
Primarykey	2
Agrupado:	FALSE
Cuenta distinta:	0

**Tabla: conspron**

Extranjero:	FALSE
Ignorar Nulos:	FALSE
Nombre:	Primarykey
Principal:	FALSE
Required:	FALSE
Única:	TRUE
Campos:	catalogo, Ascend.
almacen, Ascendente	
Permisos de usuario	
admin	
Permisos de grupo	
Admins	
Users	

**Tabla: constp**

Propiedades			
Def. actualizable:	TRUE		
RecordCount:		0	
Columnas			
Nombre		Tipo	Tamaño
catalogo		Texto	7
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:			0
Required:	FALSE		
Tabla de origen:	constp		
region	Texto		15
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:			1
Required:	FALSE		
Tabla de origen:	constp		

\*Código de catalogo

\*Código de región

mes	Número (largo)	4	MES
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	constp		
año	Número (largo)	4	AÑO
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	año		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	constp		
consumo	Número (doble)	8	Consumo de telefonía pública en mes.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consumo		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
	<b>Tabla: constp</b>		
Tabla de origen:	constp		
Índices de tabla			
Nombre	Número de campos		
Primarykey	4		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	catalogo, Ascend.		
region, Ascendente			
mes, Ascendente			
año, Ascendente			
Permisos de usuario			

admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: consumo**

Propiedades				
Def. actualizable:	TRUE			
RecordCount:		0		
Columnas				
Nombre		Tipo	Tamaño	
cat		Texto	7	Catalogo de consumo
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	cat			
Orden de combinación:	General			
Posición ordinal:		0		
Required:	FALSE			
Tabla de origen:	consumo			
alm	Texto		3	Clave de almacén
AllowZeroLength:	FALSE			
Atributos:	Longitud variable			
Campo de origen:	alm			
Orden de combinación:	General			
Posición ordinal:		1		
Required:	FALSE			
Tabla de origen:	consumo			
anio	Número (largo)		4	AÑO
AllowZeroLength:	FALSE			
Atributos:	Tamaño fijo			
Campo de origen:	anio			
Orden de combinación:	General			
Posición ordinal:		2		
Required:	FALSE			
Tabla de origen:	consumo			
mes	Número (largo)		4	MES
AllowZeroLength:	FALSE			
Atributos:	Tamaño fijo			
Campo de origen:	mes			
Orden de combinación:	General			
Posición ordinal:		3		
Required:	FALSE			

Tabla de origen:	consumo		
consumo	Número (doble)	8	Consumo de telefonía pública en mes
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consumo		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		

**Tabla: consumo**

Tabla de origen:	consumo
Índices de tabla	
Nombre	Número de campos
CON_IDX	4
Agrupado:	FALSE
Cuenta distinta:	0
Extranjero:	FALSE
Ignorar Nulos:	FALSE
Nombre:	CON_IDX
Principal:	FALSE
Required:	FALSE
Única:	FALSE
Campos:	cat, Ascendente
alm, Ascendente	
anio, Ascendente	
mes, Ascendente	
Permisos de usuario	
admin	
Permisos de grupo	
Admins	
Users	
*Forman llave	

NOTA: Las siguientes tablas del diccionario de datos se encuentran en el ANEXO B.

## 3.1.4. Diagrama de entidad relación

Relación	Descripción
R1	Qué materiales del catálogo de mantenimiento de proyectos entraron almacén por devolución
R2	Qué materiales del catálogo de mantenimiento de proyectos entraron al almacén por orden de compra
R3	Qué materiales del catálogo de mantenimiento de proyectos salieron del almacén por orden de salida
R4	Cuánto material hay en almacén
R5	Detalle de pedidos por almacén
R6	Detalles de materiales que entraron a los almacenes por devolución
R7	Qué materiales entraron a los almacenes por orden de compra
R8	Qué consumos se tienen en cada almacén
R9	Qué consumos promedios hay por almacén
R10	Qué materia se tienen por movimientos de ajuste
R11	Qué movimientos son de entrada de materiales en el almacén
R12	Qué movimientos son de salida de materiales en el almacén
R13	Qué pedidos tienen el almacén
R14	Qué movimiento realizó el almacén
R15	Qué consumos promedios hay por usuario en los diferentes almacenes
R16	Qué tipo de material se tiene en los almacenes por clave de catálogo
R17	Qué tipo de material se entregó al usuario
R18	Qué material se movió en el almacén
R19	Qué material se pidió en el almacén
R20	Qué material se movió por una orden de salida en el almacén
R21	Qué material entro en el almacén
R22	Qué tipo tuvo movimiento por orden de ajuste
R23	Qué tipo de material se tiene en los consumos promedios
R24	Cantidad y tipo de material que utilizó en los almacenes
R25	Qué tipo de material se recibió en los almacenes por orden de compra
R26	Qué tipo de material se recibió en los almacenes por devolución
R27	Detalles y tipo material que hay en los pedidos de los almacenes
R28	Qué material hay en existencia en el almacén
R29	Factores y tipo de materiales para construcción
R30	Qué subgrupos de material se tienen en los catálogos
R31	Tipo de material se tiene en catálogos para telefonía pública
R32	Qué subgrupos se tienen para los diferentes almacenes regionales
R33	Qué consumos de materiales de telefonía pública hay en las regiones
R34	Qué pedidos detallados hay por región
R35	Qué materiales se tienen en cada una de las diferentes regiones
R36	Qué pronósticos se tienen en la región
R37	Qué número de líneas están planeadas construir por año en cada región
R38	Qué información de materiales se incluyen en los reportes por región
R39	Qué información se definió para los reportes que se generarán

Tabla 3.1.4.1. Relaciones del Diagrama Entidad Relación.

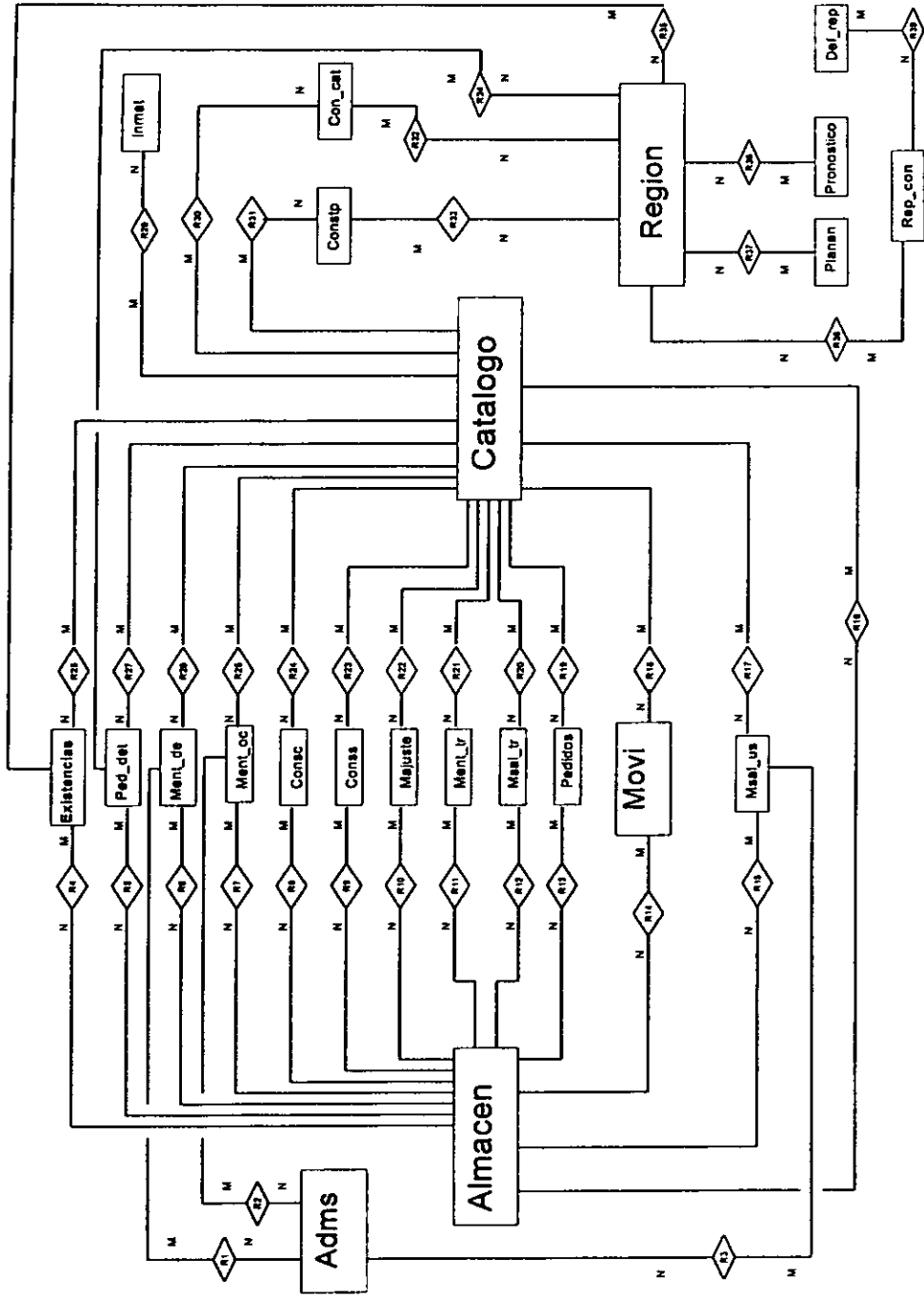


Figura 3.1.4.2. Diagrama Entidad/Relación/Relación del sistema.

### 3.1.5 Normalización

Para normalizar la información de los flujos de datos que intervienen en el sistema, se aplicaron las 3 primeras formas normales (1NF, 2NF y 3NF) de la siguiente forma.

#### FLUJO DE DATOS "MOVI"

un – Unidad organizativa  
ar – Área geográfica  
po – Población  
naf – Nivel de afectación  
cta – Cuenta de mayor  
scta – Subcuenta o auxiliar  
det – Detalle de la subcuenta  
a\_s - Almacén surtidor  
a\_r - Almacén receptor  
alm - Almacén que genera el movimiento  
cat - Catálogo del material  
descr – Descripción del material  
ped - Número de Pedido  
ser - Serie del material  
depto - Departamento  
t\_mov - Tipo de Movimiento  
mot\_s - Motivo de salida  
f\_cap – Fecha de captura del movimiento  
f\_u\_c – Fecha de última compra  
zona - Zona donde se realizó el movimiento  
cdo – Contabilizado  
prov – Proveedor  
uni – Unidad de Medida  
doc – Número del documento  
adm – Código de proyecto  
remi – Remisión  
exped – Expediente del empleado que recibe el material  
movto – Movimiento  
c\_mat – Clase de Material  
catego – Categoría  
cant – Cantidad  
saldo – Saldo de pedido

Figura 3.1.5.1. Flujo de datos MOVI sin normalizar.



En la figura 3.1.5.1 se muestra el flujo de datos MOVI correspondiente al diagrama de flujo de datos "2 Generación de consumos". Este flujo proviene directamente de la entidad SIREM y es necesario normalizarlo junto con todos los demás para poder utilizar su información dentro del sistema en una forma eficiente y poder calcular los consumos de los materiales.

Estos campos tienen su propio dominio y en cada registro solo existe un valor para cada campo, por ejemplo los siguientes campos:

cat: Es la clave que identifica un material; en esta tabla, cada registro trae información de un solo material, por lo que en él solo viene un valor o sea que es atómico.

alm: Es la clave que identifica al almacén en donde se realizó el movimiento del material; de igual forma, solo tiene la información de un solo almacén en cada registro, por lo que también es atómico.

t-mov: Es la clave que identifica el tipo de movimiento que se realizó en ese registro; dado que no es posible registrar diferentes movimientos en un registro, este campo solo tiene un valor y es indivisible.

Viendo estos ejemplos, concluimos que esta tabla ya está en primera forma normal (1NF) por que cumple con la definición de esta forma que dice "La primera forma normal es cuando todas las relaciones de la base de datos cumplen con la propiedad de tener un valor atómico en cada uno de sus atributos".

Tomando en cuenta la definición de 2NF que dice: "Se dice que una relación está en segunda forma normal (2NF) si, y solo si, está en 1NF y todos los atributos que no son

llave, dependen completamente de la llave primaria". Se realiza la división siguiente:

Como ya se llevó a 1NF esta tabla, la primera parte de la definición ya se cumplió; para cumplir la segunda parte podemos ver que la llave principal de la tabla está formada por los tres campos "t\_mov", "alm" y "cat" que se encuentran subrayados en la figura 3.1.5.1. Los campos "a\_s" (almacén surtidor) y "a\_r" (almacén receptor) se reportan cuando el movimiento es un traspaso, en el caso de que el movimiento no lo sea, el campo "a\_s" tiene el mismo valor que "alm"; esto implica que existe redundancia en los datos por no depender funcionalmente de la llave primaria completa, lo que implica que no se encuentra en segunda forma normal (2NF).

Dado que algunos otros campos sólo tienen valor dependiendo el tipo de movimiento, o sea que no dependen funcionalmente de la llave primaria, esta tabla se debe dividir según este campo "t\_mov", quedando las que se muestran en las figuras 3.1.5.2 y 3.1.5.3 anteriores.

Fecha/Hora	Fecha/Hora
un	Texto
ar	Texto
pob	Texto
naf	Texto
cta	Texto
scta	Texto
detscta	Texto
a_r	Texto
alm	Texto
cat	Texto
ped	Texto
f_cap	Fecha/Hora
f_u_c	Fecha/Hora
cdo	Texto
prov	Texto
doc	Texto
adm	Texto
remi	Texto
cant	Número
saldo	Número

Fecha/Hora	Fecha/Hora
alm	Texto
cat	Texto
f_cap	Fecha/Hora
cdo	Texto
cant	Número

Figura 3.1.5.2. Tablas ment\_oc y majuste derivadas de flujo de datos MOVI.

msal_tr : Tabla	
Nombre del campo	Tipo de datos
fechaintr	Fecha/Hora
un	Texto
ar	Texto
pob	Texto
naf	Texto
cta	Texto
scta	Texto
detscta	Texto
a_s	Texto
a_r	Texto
alm	Texto
cat	Texto
f_cap	Fecha/Hora
f_u_c	Fecha/Hora
zona	Texto
cdo	Texto
doc	Texto
cant	Número
saldo	Número

ment_de : Tabla	
Nombre del campo	Tipo de datos
fechaintr	Fecha/Hora
un	Texto
ar	Texto
pob	Texto
naf	Texto
cta	Texto
scta	Texto
detscta	Texto
a_r	Texto
alm	Texto
cat	Texto
depto	Texto
mot_s	Texto
f_cap	Fecha/Hora
f_u_c	Fecha/Hora
zona	Texto
cdo	Texto
doc	Texto
adm	Texto
remi	Texto
exp	Texto
cant	Número
saldo	Número

mesal_us : Tabla	
Nombre del campo	Tipo de datos
fechaintr	Fecha/Hora
un	Texto
ar	Texto
pob	Texto
naf	Texto
cta	Texto
subcta	Texto
detscta	Texto
a_s	Texto
alm	Texto
cat	Texto
depto	Texto
mot_s	Texto
f_cap	Fecha/Hora
f_u_c	Fecha/Hora
zona	Texto
cdo	Texto
doc	Texto
adm	Texto
exped	Texto
catego	Texto
cant	Número
saldo	Número

ment_tr : Tabla	
Nombre del campo	Tipo de datos
fechaintr	Fecha/Hora
un	Texto
ar	Texto
pob	Texto
naf	Texto
cta	Texto
scta	Texto
detscta	Texto
a_s	Texto
a_r	Texto
alm	Texto
cat	Texto
f_cap	Fecha/Hora
f_u_c	Fecha/Hora
cdo	Texto
doc	Texto
cant	Número
saldo	Número

Figura 3.1.5.3. Tablas msal\_tr, ment\_de, mesal\_us y ment\_tr derivadas de Normalizar MOVI.

Se generó una tabla por cada tipo de movimiento que puede venir en el archivo MOVI, las cuales tiene las siguientes descripciones:

ment\_oc – Entradas por orden de compra  
majuste – Ajustes efectuados en el almacén  
msal\_tr – Salidas por traspaso  
ment\_de – Entradas por devolución  
msal\_us – Salidas a usuario o proyecto  
ment\_tr – Entradas por traspaso

Se puede observar que algunos campos que no son relevantes para el tipo de movimiento los cuales fueron eliminados de la tabla correspondiente, se eliminó el campo "a\_s", por ejemplo, en las tablas que no son traspasos, para evitar la redundancia de información que ya se había comentado.

Todos los campos, como dice la definición, ya dependen funcionalmente de la llave primaria que es la misma en cada tabla nueva; esto hacer cumplir la definición, por lo que ya se encuentran en la segunda forma normal (2NF).

Continuando con esta normalización, de estas tablas, es siguiente paso es llevarlas a 3NF lo cual significa, según la definición: "Se dice que una relación esta en tercera forma normal (3NF) sí, y solo si está en 2NF y todos los atributos que no son llave principal, son mutuamente independientes".

Los campos "ser" (serie del material), "uni" (unidad de medida) y "descr" (descripción del material) son características del material (Catálogo), por lo que no dependen del tipo de movimiento ni del almacén, o sea que no dependen funcionalmente de la llave primaria. Para dejar estas tablas en 3NF, se tienen que separar estos campos y todos aquellos que no dependen funcionalmente de la llave primaria, tal como dice la definición. Creando la nueva tabla CATALOGO con la que se puede tener la referencia a estos datos y evitar dependencias funcionales de campos que nos son llave primaria.

De igual forma, el campo "alm" es una llave candidata que indica el almacén que genera el movimiento. En la tabla MOVI no existen campos con características del almacén y que pueden ser necesarios para clasificarlos, pero por ser llave candidata, se puede requerir que se genere un nuevo almacén sin que para ello deba tener movimientos para que exista y este almacén tener diferentes características; por lo que se genera una nueva tabla que contenga estos datos lo que da la oportunidad de tener mas almacenos, sin que para ello se deban tener movimientos en todas las tablas.

El resultado de la creación de estas nuevas tablas de las dos llaves candidatas con sus respectivas características, se presenta en la figura 3.1.5.4.

campo	tipo
catalogo	Texto
unidad	Texto
descripcion	Texto
condicion	Texto
serie	Texto

campo	tipo
cod_sirem	Texto
cod_soci	Texto
nombre	Texto
grupodist	Texto
region	Texto
subgrupo	Texto
categoria	Texto
division	Texto
nuevadivis	Texto
alm_dist	Numérico

Figura 3.1.5.4. Tablas resultantes de analizar las llaves candidatas de MOVI.

Se puede observar en la figura, que se agregaron algunos campos informativos tanto para *catalogo* como para *almacén*, quedando como tablas del sistema independientes de las tablas de movimientos; la llave primaria para la tabla *catalogo* es el campo "catalogo" que apunta a los datos de "cat" de la tabla MOVI, y la llave primaria de la tabla *almacén* es el campo "cod\_sirem" que apunta a los datos de "alm" de la tabla MOVI.

Después de esta descomposición, el resultado son tablas que se encuentran ya en la tercera forma normal 3NF por no tener mas que una sola dependencia funcional entre los campos de la tabla y su llave primaria y carecer de dependencias entre campos que no son esa llave primaria.

En el apéndice C se muestra la normalización del flujo de datos PEDIDOS, de esta forma se normalizan todas las tablas del sistema para que sean funcionales y de fácil acceso.

### 3.2. Generación del código para el procesamiento de la información

Una vez construido el back end del sistema, se procedió a generar todas las rutinas que son necesarias para el procesamiento de la información.

Se partió del módulo principal que es el que va a controlar el flujo y presentación de las pantallas diseñadas para el front end. Adicionalmente se realizaron rutinas de apoyo que son las que se utilizarán para la validación de los datos que se reciban de las pantallas.

Para cada uno de los procesos a realizar, se generaron rutinas que obtienen los resultados, ligados a las formas que las van a llamar.

A continuación se muestra una parte del código que se generó para el módulo principal llamado "Modulo1", las rutinas de validación y el código que se utiliza para validar que el usuario que entre al sistema esté dado de alta y habilitado.

El resto del código se muestra en el Apéndice D.

```
'-----  
-----  
'Archivo: Module1  
'Fecha: Julio-Septiembre de 1999  
'Propósito: Módulo general que implanta  
diversas rutinas de propósito  
'general  
'-----  
-----  
Option Explicit  
Public Type cabeza  
  'cabeza(222) As Byte  
  cabeza(224) As Byte  
  'Nuevos archivos-> cabeza(224) As  
Byte  
End Type  
  
Public Type saldo  
  'cabeza(222) As Byte  
  saldo_fin(14) As Byte  
  'Nuevos archivos-> cabeza(224) As  
Byte  
End Type  
  
Public Type retorno  
  return As String * 1  
End Type  
  
Public Type RecArch 'Definición para  
lectura de archivo de movimientos  
  UN As String * 12  
  AR As String * 12  
  PO As String * 12  
  NAF As String * 12  
  CTA As String * 12
```

```

SCTA As String * 12
xx As String * 12
A_S As String * 12
A_R As String * 12
ALM As String * 12
CAT As String * 12
PED As String * 12
SER As String * 12
DEPTO As String * 12
T_MOV As String * 12
MOT_S As String * 12
F_CAP As String * 12
F_U_C As String * 12
ZONA As String * 12
CDO As String * 12
PROV As String * 12
UNI As String * 12
DOC As String * 12
ADM As String * 12
Remi As String * 12
EXPED As String * 12
MOVTO As String * 12
C_MAT As String * 12
CATEGO As String * 12
CANT As String * 15
End Type

Public fMainForm As fm_Main
Public fmenArchivo As fm_MenArchivo
Global s_gRutaBD As String
Global s_login As String
Global s_gRutaBDAdmin As String
Global s_gPassword As String
Global s_gadmin As Boolean
Global gd_ldb As Database
Global gl_ltb As Recordset

Sub Main()
Dim fLogin As New fm_Login
fm_Splash.Show vbModal
fm_Splash.Refresh
s_gRutaBDAdmin =
"C:\sici\siciadmin"
Set gd_ldb =
OpenDatabase(s_gRutaBDAdmin,
False)

```

```

Set gl_ltb =
gd_ldb.OpenRecordset("bitacora",
dbOpenDynaset)
fLogin.Show vbModal
If Not fLogin.OK Then
'Login Failed so exit app
End
End If
Unload fLogin

Set fMainForm = New fm_Main
Load fMainForm
fMainForm.Show

fMainForm.Visible = False
While s_gRutaBD = ""
Set fmenArchivo = New
fm_MenArchivo
fmenArchivo.Show vbModal
If s_gRutaBD = "" Then
MsgBox "Debe seleccionar una
base de datos antes de continuar",
vbCritical
End If
Wend
Set fMainForm = New fm_Main
fMainForm.Visible = True
End Sub
'Nombre : d_gCalcula_existencia
' Proposito: Por medio de esta función
se realiza el cálculo de existencias
estimadas
'dados la existencia anterior, los pedidos
del mes y el consumo del mes.
'Se consideran varios meses de
pronostico (hasta 6 meses). Para el mes
cero se considera una formula inicial en
algunos casos.
Public Function
d_gCalcula_existencia(i_ICodForm As
Long, i_lmes As Integer, i_IDia As
Integer, d_lExistencia As Double,
d_lPedidos As Double, d_lConsumo As
Double) As Double
Dim d_lres As Double

```



```

Select Case I_IDCodForm
Case 1
    If i_lmes = 0 Then
        d_lres = d_lExistencia +
d_IPedidos - 2 * d_lConsumo -
(d_lConsumo / 30) * (30 - i_lDia)
    Else
        If d_lExistencia > 0 Then
            d_lres = d_lExistencia +
d_IPedidos - d_lConsumo
        Else
            d_lres = d_IPedidos -
d_lConsumo
        End If
    End If
End If
Case 2
    If i_lmes = 0 Then
        d_lres = d_lExistencia +
d_IPedidos - 3 * d_lConsumo
    Else
        If d_lExistencia > 0 Then
            d_lres = d_lExistencia +
d_IPedidos - d_lConsumo
        Else
            d_lres = d_IPedidos -
d_lConsumo
        End If
    End If
End If
Case 3
    If i_lmes = 0 Then
        d_lres = d_lExistencia +
d_IPedidos - (d_lConsumo / 30) * (30 -
i_lDia)
    Else
        d_lres = d_lExistencia +
d_IPedidos - d_lConsumo
    End If
End If
Case Else
    d_lres = 0
End Select
d_gCalcula_existencia = d_lres
End Function

Public Function s_gCadena_fija(dato As
String, longitud As Integer) As String
Dim i_llong As Integer

```

```

Dim s_lcadena As String
i_llong = Len(dato)
s_lcadena = dato
If i_llong < longitud Then
    s_lcadena = dato + String(longitud
- i_llong, " ") + " | "
Else
    s_lcadena = dato + " | "
End If
s_gCadena_fija = s_lcadena
End Function

```

```

Public Function
s_gCadena_fijaEsp(dato As String,
longitud As Integer) As String
Dim i_llong As Integer
Dim s_lcadena As String
s_lcadena = Trim(dato)
i_llong = Len(dato)
If i_llong < longitud Then
    s_lcadena = dato + String(longitud
- i_llong, " ") + " | "
Else
    If i_llong > longitud Then
        s_lcadena = Mid(s_lcadena, 1,
longitud) + " | "
    Else
        s_lcadena = s_lcadena + " | "
    End If
End If
s_gCadena_fijaEsp = s_lcadena
End Function

```

```

'Nombre: s_gNumero_fijo
'Fecha: 15/Agosto/1999
'Proposito: Esta función genera el
número dado en una cadena de la
'longitud solicitada.
Public Function s_gNumero_fijo(numero
As Double, longitud As Integer) As
String
Dim i_llong As Integer
Dim s_lcadena As String
Dim s_ldato As String
Dim l_lNumlong As Long
l_lNumlong = CLng(numero)

```

```

s_Idato = Str(i_Numlong)
i_llong = Len(s_Idato)
s_Icadena = s_Idato
If i_llong < longitud Then
    s_Icadena = String(longitud -
i_llong, " ") + s_Idato + "|"
Else
    s_Icadena = Mid(s_Idato, 1,
longitud) + "|"
End If
s_gNumero_fijo = s_Icadena
End Function

```

```

'Nombre: s_gNombreMes
'Fecha: 15/Agosto/1999
'Proposito: Esta función regresa el mes
del año que se solicite (1..12)
Public Function
s_gNombreMes(numero_mes As
Integer) As String
Dim s_IRes As String
    numero_mes = numero_mes - 1
    numero_mes = (numero_mes Mod
12) + 1
    s_IRes = ""
    Select Case numero_mes
    Case 1
        s_IRes = "ENERO "
    Case 2
        s_IRes = "FEBRERO"
    Case 3
        s_IRes = "MARZO "
    Case 4
        s_IRes = "ABRIL "
    Case 5
        s_IRes = "MAYO "
    Case 6
        s_IRes = "JUNIO "
    Case 7
        s_IRes = "JULIO "
    Case 8
        s_IRes = "AGOSTO "
    Case 9
        s_IRes = "SEPT. "
    Case 10
        s_IRes = "OCTUBRE"

```

```

Case 11
    s_IRes = " NOV. "
Case 12
    s_IRes = " DIC. "
End Select
s_gNombreMes = s_IRes
End Function

```

```

'Proposito. Toma la ruta completa de
archivo en s_IRutaComp y obtiene
'la ruta de directorio s_IRuta y el nombre
de archivo s_Inombre
Sub gObtenRutas(s_IRutaComp As
String, ByRef s_IRuta As String, ByRef
s_INombre As String)
Dim s_IPath As String
Dim i_IPos As Integer
Dim i_ICon As Integer
If IsNull(s_IRutaComp) Or s_IRutaComp
= "" Then
    s_IRuta = ""
    s_INombre = ""
Else
    s_IPath = s_IRutaComp
    i_IPos = InStr(1, s_IPath, ":")
    If i_IPos > 0 Then
        s_IPath = Mid(s_IPath, i_IPos + 1)
        i_ICon = i_IPos
    Else
        i_ICon = 0
    End If
    i_IPos = InStr(1, s_IPath, "\")
    i_ICon = i_ICon + i_IPos
    While i_IPos > 0
        s_IPath = Mid(s_IPath, i_IPos + 1)
        i_IPos = InStr(1, s_IPath, "\")
        i_ICon = i_ICon + i_IPos
    Wend
    s_INombre = s_IPath
    s_IRuta = Mid(s_IRutaComp, 1,
i_ICon)
    If Mid(s_IRuta, Len(s_IRuta), 1) = "\"
Then
        s_IRuta = Mid(s_IRuta, 1,
Len(s_IRuta) - 1)
    End If

```

```

End If
End Sub

'Proposito: Verifica que el caracter
dado sea un digito del 0 al 9
Function IEs_dig(caracter As String) As
Boolean
Dim s_ICara As String
s_ICara = caracter
If s_ICara = "0" Or s_ICara = "1" Or
s_ICara = "2" Or s_ICara = "3" Then
IEs_dig = True
Exit Function
Elseif s_ICara = "3" Or s_ICara = "4"
Or s_ICara = "5" Or s_ICara = "6" Then
IEs_dig = True
Exit Function
Elseif s_ICara = "7" Or s_ICara = "8"
Or s_ICara = "9" Then
IEs_dig = True
Exit Function
End If
IEs_dig = False
End Function

'Proposito: esta función indica si el
contenido en cadena se
'apega al formato de una fecha en
español: DD/MM/AAAA
Public Function
gVerifica_fecha_valida(cadena As
String) As Boolean
Dim s_ICara As String
If cadena = "" Then
gVerifica_fecha_valida = False
Exit Function
End If
If cadena = vbNullString Then
gVerifica_fecha_valida = False
Exit Function
End If
If Len(cadena) > 10 Then
gVerifica_fecha_valida = False
Exit Function
End If
If Mid(cadena, 3, 1) <> "/" Then
gVerifica_fecha_valida = False
Exit Function
End If
If Mid(cadena, 6, 1) <> "/" Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 1, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 2, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 4, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 5, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 7, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 8, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 9, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If
s_ICara = Mid(cadena, 10, 1)
If Not IEs_dig(s_ICara) Then
gVerifica_fecha_valida = False
Exit Function
End If

```

```

End If
gVerifica_fecha_valida = True 'Si
esta bien
End Function

'Proposito: Verifica que la cadena
contiene un anio valido
Public Function
gVerifica_anio_valido(cadena As String)
As Boolean
If cadena = "" Then
gVerifica_anio_valido = False
Exit Function
End If
If cadena = vbNullString Then
gVerifica_anio_valido = False
Exit Function
End If
If Val(cadena) < 1900 Then
gVerifica_anio_valido = False
Exit Function
End If
gVerifica_anio_valido = True 'Si esta
bien
End Function

```

```

Public Sub
gConstruyeComandoFTP(s_IClave As
String, s_IPassword As String,
s_IOrigen As String, s_IDestino As
String)
Dim i_IArchSalida As Integer
i_IArchSalida = FreeFile()
Open "sicilcomftp.txt" For Output
As #i_IArchSalida
Print #i_IArchSalida, s_IClave
Print #i_IArchSalida, s_IPassword
Print #i_IArchSalida, "binary"
Print #i_IArchSalida, "get " +
s_IOrigen + " " + s_IDestino
'Print #i_IArchSalida, "ascii"
'Print #i_IArchSalida, "get " +
s_IOrigen + " " + s_IDestino
Print #i_IArchSalida, "bye"
Close #i_IArchSalida
End Sub

```

```

Sub glmprime_archivo(nombre_archivo
As String)
Dim i_gArchSalida As Integer
Dim s_IBuffer As String
Dim i_lres As Integer
Dim primero As Boolean
If Dir(nombre_archivo) <> "" Then
i_lres = MsgBox("¿ Confirma
impresion de contenido archivo salida
(reporte/traspaso) ?", vbQuestion +
vbYesNo)
If i_lres = vbYes Then
i_gArchSalida = FreeFile()
Open nombre_archivo For Input As
#i_gArchSalida
Printer.Orientation =
vbPRORPortrait
Printer.FontName = "LinePrinter"
Printer.FontSize = 3
primero = True
While Not EOF(i_gArchSalida)
Line Input #i_gArchSalida,
s_IBuffer
If Mid(s_IBuffer, 1, 2) = ">>"
Then
If primero Then
primero = False
Else
Printer.NewPage
End If
'Printer.NewPage
s_IBuffer = Mid(s_IBuffer, 3)
Printer.Print s_IBuffer
Else
Printer.Print s_IBuffer
End If
Wend
Printer.EndDoc
Close #i_gArchSalida
MsgBox "Reporte terminado",
vbExclamation
End If
Else
MsgBox "No existe archivo a
imprimir", vbExclamation

```

```
End If
End Sub
```

```
'-----
'
'Archivo: fm_splash
'Fecha: Julio-Septiembre de 1999
'Proposito: Pantalla inicial de
presentación. Hace uso de un timer
'para mantenerse por un tiempo.
'-----
```

```
Private Sub Form_Load()
    lblVersion.Caption = "Version " &
App.Major & "." & App.Minor & "." &
App.Revision
    lblProductName.Caption = App.Title
End Sub
```

```
Private Sub
fraMainFrame_DragDrop(Source As
Control, X As Single, Y As Single)
End Sub
```

```
Private Sub Timer1_Timer()
    Unload Me
End Sub
```

```
'-----
'
'Archivo: fm_login
'Fecha: Julio-Septiembre de 1999
'Proposito: Módulo de acceso al
sistema.
'-----
```

```
Private Declare Function GetUserName
Lib "advapi32.dll" Alias
"GetUserNameA" (ByVal lpbuffer As
String, nSize As Long) As Long
Public OK As Boolean
Private Sub Form_Load()
    Dim sBuffer As String
    Dim ISize As Long
    sBuffer = Space$(255)
```

```
ISize = Len(sBuffer)
Call GetUserName(sBuffer, ISize)
If ISize > 0 Then
    txtUserName.Text = Left$(sBuffer,
ISize)
Else
    txtUserName.Text = vbNullString
End If
End Sub
```

```
Private Sub cmdCancel_Click()
    OK = False
    Me.Hide
End Sub
```

```
Private Sub cmdOK_Click()
    'To Do - create test for correct
password
'check for correct password
Dim d_ldb As Database
Dim l_ltb As Recordset
Set d_ldb =
OpenDatabase(s_gRutaBDAdmin, True)
l_criterio = "select password, nombre,
activo, admin, nuevo from usuarios
where usuario = " + """" +
txtUserName.Text + """"
Set l_ltb =
d_ldb.OpenRecordset(l_criterio,
dbOpenDynaset)
If l_ltb.EOF Then
    MsgBox "El usuario no está dado
de alta, intente de nuevo!", , "Login"
    txtUserName.SetFocus
    txtUserName.SelStart = 0
    txtUserName.SelLength =
Len(txtUserName.Text)
    txtPassword.Text = ""
Else
    s_gadmin = l_ltb("admin")
    If l_ltb("nuevo") Then
        If txtPassword.Text = "init" Then
            fm_password.Show vbModal
            If fm_password.OK Then
                OK = True
                s_login = txtUserName.Text
```

```

I_Itb.Edit
I_Itb("Password") =
s_gPassword
I_Itb("activo") = True
I_Itb("nuevo") = False
I_Itb.Update
gl_Itb.AddNew
gl_Itb.Fields("usuario") =
s_login
gl_Itb.Fields("basedatos") =
s_gRutaBDAdmin
gl_Itb.Fields("tabla") =
"usuarios"
gl_Itb.Fields("fecha_hora")
= Date + Time
gl_Itb.Fields("Numtrans") =
3
gl_Itb.Update
Me.Hide
Else
MsgBox "Clave inválida,
intente de nuevo!", , "Login"
txtUserName.SetFocus
txtUserName.SetStart = 0
txtUserName.SetLength =
Len(txtUserName.Text)
txtPassword.Text = ""
End If
Else
MsgBox "El Usuario es Nuevo
y debe teclear el indicadir
correspondiente. Consulta a su
Administrador del Sistema",
vbExclamation
End If
Else
If I_Itb("activo") Then
If txtPassword.Text =
I_Itb("Password") Then
OK = True
s_login = txtUserName.Text
Me.Hide
Else
MsgBox "Clave inválida,
intente de nuevo!", , "Login"
txtPassword.SetFocus
txtPassword.SetStart = 0
txtPassword.SetLength =
Len(txtPassword.Text)
End If
Else
MsgBox "El Usuario se
encuentra desabilitado, consulte a su
administrador para habilitarlo", , "Login"
txtUserName.SetFocus
txtUserName.SetStart = 0
txtUserName.SetLength =
Len(txtUserName.Text)
txtPassword.Text = ""
End If
End If
End If
I_Itb.Close
End Sub
'-----
'Archivo: fm_main
'Fecha: Julio-Septiembre de 1999
'Proposito: Módulo principal para
invocar a las demás aplicaciones
'Hace uso de menus y de botones de
uso continuo
'-----
Private Declare Function OSWinHelp%
Lib "user32" Alias "WinHelpA" (ByVal
hwnd&, ByVal HelpFile$, ByVal
wCommand%, dwData As Any)

Private Sub Command1_Click(Index As
Integer)
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_MenArchivo.Show vbModal

```

```
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub
```

```
Private Sub
Command1_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
sbStatusBar.Panels(1).Text = "Accesar
o Crear una Base de Datos"
End Sub
```

```
Private Sub Command2_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_mencarga.Show vbModal
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub
```

```
Private Sub
Command2_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y
As Single)
sbStatusBar.Panels(1).Text = "Presione
el boton para realizar la carga de
información"
End Sub
```

```
Private Sub Command3_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_mendat_calcons.Show vbModal
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub
```

```
Private Sub
Command3_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y
As Single)
sbStatusBar.Panels(1).Text = "Para
capturar datos para el cálculo de
consomos, presiona este boton"
End Sub
```

```
Private Sub Command4_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_mencalcons.Show vbModal
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
```

```

Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub
Command4_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y
As Single)
sbStatusBar.Panels(1).Text =
"Obtención de los consumo promedio
sobre los datos cargados y capturados"
End Sub

```

```

Private Sub Command5_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_menReportes.Show vbModal
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub
Command5_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y
As Single)
sbStatusBar.Panels(1).Text =
"Generación y configuración de
reportes."
End Sub

```

```

Private Sub Command6_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False

```

```

Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_mentablas.Show vbModal
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub
Command6_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y
As Single)
sbStatusBar.Panels(1).Text =
"Mantenimiento a las tablas del sistema"
End Sub

```

```

Private Sub Command7_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
mnuHelp_Click
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True

```

```
End Sub
```

```

Private Sub
Command7_MouseMove(Button As

```



```

Integer, Shift As Integer, X As Single, Y
As Single)
sbStatusBar.Panels(1).Text =
"Información acerca de esta Versión de
su programa"
End Sub

Private Sub Command8_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    If vbYes = MsgBox("¿ Confirma salida
de sistema ?", vbQuestion + vbYesNo)
Then
        End
    End If
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub

Private Sub
Command8_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y
As Single)
    sbStatusBar.Panels(1).Text =
"Abandonar la sección y salir del
sistema"
End Sub

Private Sub Form_Load()
    If s_gadmin Then
        Command6.Enabled = True
    Else
        Command6.Enabled = False
    End If

    Text1.Text = s_gRutaBD
    Me.Left = GetSetting(App.Title,
"Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title,
"Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title,
"Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title,
"Settings", "MainHeight", 6500)
End Sub

Private Sub Form_MouseMove(Button
As Integer, Shift As Integer, X As Single,
Y As Single)
    sbStatusBar.Panels(1).Text =
"Posicione el cursor en algún icono para
ver ayuda"
End Sub

Private Sub Form_Unload(Cancel As
Integer)
    Dim i As Integer
    'close all sub forms
    For i = Forms.Count - 1 To 1 Step -1
        Unload Forms(i)
    Next
    If Me.WindowState <> vbMinimized
Then
        SaveSetting App.Title, "Settings",
"MainLeft", Me.Left
        SaveSetting App.Title, "Settings",
"MainTop", Me.Top
        SaveSetting App.Title, "Settings",
"MainWidth", Me.Width
        SaveSetting App.Title, "Settings",
"MainHeight", Me.Height
    End If
End Sub

Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub

Private Sub mnuViewOptions_Click()
    frmOptions.Show vbModal, Me
End Sub

```

```

Private Sub mnuViewStatusBar_Click()
  If mnuViewStatusBar.Checked Then
    sbStatusBar.Visible = False
    mnuViewStatusBar.Checked =
False
  Else
    sbStatusBar.Visible = True
    mnuViewStatusBar.Checked =
True
  End If
End Sub

```

```

Private Sub mnuViewToolBar_Click()
  If mnuViewToolBar.Checked Then
    tbToolBar.Visible = False
    mnuViewToolBar.Checked = False
  Else
    tbToolBar.Visible = True
    mnuViewToolBar.Checked = True
  End If
End Sub

```

```

Private Sub Inet1_StateChanged(ByVal
State As Integer)
End Sub

```

```

Private Sub
Frame1_MouseMove(Button As Integer,
Shift As Integer, X As Single, Y As
Single)
  sbStatusBar.Panels(1).Text =
"Posicione el cursor en algún icono para
ver ayuda"
End Sub

```

```

Private Sub mn_repap_Click()
  Command1(1).Visible = False
  Command2.Visible = False
  Command3.Visible = False
  Command4.Visible = False
  Command5.Visible = False
  Command6.Visible = False
  Command7.Visible = False
  Command8.Visible = False
  fm_avan.Show

```

```

  Command1(1).Visible = True
  Command2.Visible = True
  Command3.Visible = True
  Command4.Visible = True
  Command5.Visible = True
  Command6.Visible = True
  Command7.Visible = True
  Command8.Visible = True
End Sub

```

```

Private Sub mn_repeu_Click()
  Command1(1).Visible = False
  Command2.Visible = False
  Command3.Visible = False
  Command4.Visible = False
  Command5.Visible = False
  Command6.Visible = False
  Command7.Visible = False
  Command8.Visible = False
  fm_entusu.Show
  Command1(1).Visible = True
  Command2.Visible = True
  Command3.Visible = True
  Command4.Visible = True
  Command5.Visible = True
  Command6.Visible = True
  Command7.Visible = True
  Command8.Visible = True
End Sub

```

```

Private Sub mnu_probor_Click()
  Command1(1).Visible = False
  Command2.Visible = False
  Command3.Visible = False
  Command4.Visible = False
  Command5.Visible = False
  Command6.Visible = False
  Command7.Visible = False
  Command8.Visible = False
  fm_probor.Show
  Command1(1).Visible = True
  Command2.Visible = True
  Command3.Visible = True
  Command4.Visible = True
  Command5.Visible = True
  Command6.Visible = True

```

```
Command7.Visible = True
Command8.Visible = True
End Sub
```

```
Private Sub mnu_procap_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    fm_procap.Show
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub
```

```
Private Sub mnu_ral_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    fm_regalm.Show
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub
```

```
Private Sub mnuadms_Click()
    Command1(1).Visible = False
```

```
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_adms.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub
```

```
Private Sub mnuborra_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    fm_borra.Show
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub
```

```
Private Sub mnuCarAlm_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
```

```

Command8.Visible = False
fm_alma.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

Private Sub mnuCarCat_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_cata.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

Private Sub mnuccon_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_cconf.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True

Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

Private Sub mnuCEP_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_cep.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

Private Sub mnuCM_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_cconm.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```
Private Sub mnuconpe_Click()  
    MsgBox "Opción no disponible",  
    vbInformation  
End Sub
```

```
Private Sub mnucontp_Click()  
    Command1(1).Visible = False  
    Command2.Visible = False  
    Command3.Visible = False  
    Command4.Visible = False  
    Command5.Visible = False  
    Command6.Visible = False  
    Command7.Visible = False  
    Command8.Visible = False  
    fm_tp.Show  
    Command1(1).Visible = True  
    Command2.Visible = True  
    Command3.Visible = True  
    Command4.Visible = True  
    Command5.Visible = True  
    Command6.Visible = True  
    Command7.Visible = True  
    Command8.Visible = True  
End Sub
```

```
Private Sub mnuCP_Click()  
    Command1(1).Visible = False  
    Command2.Visible = False  
    Command3.Visible = False  
    Command4.Visible = False  
    Command5.Visible = False  
    Command6.Visible = False  
    Command7.Visible = False  
    Command8.Visible = False  
    fm_prono.Show  
    Command1(1).Visible = True  
    Command2.Visible = True  
    Command3.Visible = True  
    Command4.Visible = True  
    Command5.Visible = True  
    Command6.Visible = True  
    Command7.Visible = True  
    Command8.Visible = True  
End Sub
```

```
Private Sub mnugen_Click()
```

```
    Command1(1).Visible = False  
    Command2.Visible = False  
    Command3.Visible = False  
    Command4.Visible = False  
    Command5.Visible = False  
    Command6.Visible = False  
    Command7.Visible = False  
    Command8.Visible = False  
    fm_gen.Show  
    Command1(1).Visible = True  
    Command2.Visible = True  
    Command3.Visible = True  
    Command4.Visible = True  
    Command5.Visible = True  
    Command6.Visible = True  
    Command7.Visible = True  
    Command8.Visible = True
```

```
End Sub
```

```
Private Sub mnuHelp_Click()  
    Command1(1).Visible = False  
    Command2.Visible = False  
    Command3.Visible = False  
    Command4.Visible = False  
    Command5.Visible = False  
    Command6.Visible = False  
    Command7.Visible = False  
    Command8.Visible = False  
    fm_About.Show vbModal  
    Command1(1).Visible = True  
    Command2.Visible = True  
    Command3.Visible = True  
    Command4.Visible = True  
    Command5.Visible = True  
    Command6.Visible = True  
    Command7.Visible = True  
    Command8.Visible = True
```

```
End Sub
```

```
Private Sub mnuMant_Click()  
    Command1(1).Visible = False  
    Command2.Visible = False  
    Command3.Visible = False  
    Command4.Visible = False  
    Command5.Visible = False  
    Command6.Visible = False
```

```

Command7.Visible = False
Command8.Visible = False
fm_consirem.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub mnumattm_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_Inmat.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub mnuOpt_Click()
fm_Options.Show
End Sub

```

```

Private Sub mnumttotp_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False

```

```

fm_conmtp.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub mnuplan_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_planc.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub mnuregfac_Click()
Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
fm_regfact.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True

```

```

Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub mnurepcjt_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    fm_conjcat.Show
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub

```

```

Private Sub mnurepcon_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    fm_cnrp.Show
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub

```

```

Private Sub mnurepgen_Click()

```

```

Command1(1).Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Command5.Visible = False
Command6.Visible = False
Command7.Visible = False
Command8.Visible = False
Fm_genrep.Show
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

```

```

Private Sub
tbToolBar_ButtonClick(ByVal Button As
ComctlLib.Button)
    Select Case Button.Key
        Case "NuevaBD"
            mnuFileNew_Click
        Case "AbreBD"
            mnuFileOpen_Click
        Case "ConfCarga"
            fm_cconf.Show
        Case "Conjunto"
            fm_conjcat.Show
        Case "ConfRep"
            fm_cnrp.Show
        Case "GenRep"
            Fm_genrep.Show
        Case "Salir"
            If vbYes = MsgBox("¿ Confirma
salida de sistema ?", vbQuestion +
vbYesNo) Then
                End
            End If
        End Select
    End Sub

```

```

Private Sub mnuHelpContents_Click()
    Dim nRet As Integer

```

```

'if there is no helpfile for this project
display a message to the user
'you can set the HelpFile for your
application in the
'Project Properties dialog
If Len(App.HelpFile) = 0 Then
    MsgBox "Unable to display Help
Contents. There is no Help associated
with this project.", vbInformation,
Me.Caption
Else
    On Error Resume Next
    nRet = OSWinHelp(Me.hwnd,
App.HelpFile, 3, 0)
    If Err Then
        MsgBox Err.Description
    End If
End If
End Sub

```

```

Private Sub mnuHelpSearch_Click()
    Dim nRet As Integer
    'if there is no helpfile for this project
display a message to the user
'you can set the HelpFile for your
application in the
'Project Properties dialog
If Len(App.HelpFile) = 0 Then
    MsgBox "Unable to display Help
Contents. There is no Help associated
with this project.", vbInformation,
Me.Caption
Else
    On Error Resume Next
    nRet = OSWinHelp(Me.hwnd,
App.HelpFile, 261, 0)
    If Err Then
        MsgBox Err.Description
    End If
End If
End Sub

```

```

Private Sub mnuViewRefresh_Click()
'To Do
    MsgBox "Refresh Code goes here!"
End Sub

```

```

Private Sub mnuFileOpen_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    With dlgCommonDialog
        'To Do
        'set the flags and attributes of the
'common dialog control
        .DialogTitle = "Apertura de base de
datos "
        .Filter = "Access DBs
(*.mdb)|*.mdb"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        s_gRutaBD = .FileName
        Text1.Text = .FileName
    End With
'To Do
'process the opened file
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub

```

```

Private Sub mnuFileClose_Click()
'To Do
    s_gRutaBD = "" 'se cierra ruta de
acceso a bd
    Text1.Text = ""
End Sub

```

```

Private Sub mnuFileSave_Click()
'To Do

```



```

    MsgBox "Save Code goes here!"
End Sub

```

```

Private Sub mnuFileSaveAs_Click()
    'To Do
    'Setup the common dialog control
    'prior to calling ShowSave
    dlgCommonDialog.ShowSave
End Sub

```

```

Private Sub mnuFileSaveAll_Click()
    'To Do
    MsgBox "Save All Code goes here!"
End Sub

```

```

Private Sub mnuFileProperties_Click()
    'To Do
    MsgBox "Properties Code goes here!"
End Sub

```

```

Private Sub mnuFilePageSetup_Click()
    dlgCommonDialog.ShowPrinter
End Sub

```

```

Private Sub
mnuFilePrintPreview_Click()
    'To Do
    MsgBox "Print Preview Code goes
here!"
End Sub

```

```

Private Sub mnuFilePrint_Click()
    'To Do
    MsgBox "Print Code goes here!"
End Sub

```

```

Private Sub mnuFileSend_Click()
    'To Do
    MsgBox "Send Code goes here!"
End Sub

```

```

Private Sub mnuFileMRU_Click(Index
As Integer)
    'To Do
    MsgBox "MRU Code goes here!"
End Sub

```

```

Private Sub mnuFileExit_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    If vbYes = MsgBox("¿ Confirma salida
de sistema ?", vbQuestion + vbYesNo)
Then
        End
    End If
    Command1(1).Visible = True
    Command2.Visible = True
    Command3.Visible = True
    Command4.Visible = True
    Command5.Visible = True
    Command6.Visible = True
    Command7.Visible = True
    Command8.Visible = True
End Sub

```

```

Private Sub mnuFileNew_Click()
    Command1(1).Visible = False
    Command2.Visible = False
    Command3.Visible = False
    Command4.Visible = False
    Command5.Visible = False
    Command6.Visible = False
    Command7.Visible = False
    Command8.Visible = False
    With dlgCommonDialog
        'To Do
        'set the flags and attributes of the
        'common dialog control
        .DialogTitle = "Creación de nueva
base de datos "
        .Filter = "Access DBs
(*.mdb)*.mdb"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
    End With

```

```
If Dir(.FileName) <> "" Then
    MsgBox "Atención: Ese nombre
ya esta siendo ocupado por otra base.
Creación cancelada", vbExclamation
    Exit Sub
End If
s_gRutaBD = .FileName
Call gCrea_bd(s_gRutaBD)
Text1.Text = .FileName
End With
'To Do
'process the opened file
Command1(1).Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Command5.Visible = True
Command6.Visible = True
Command7.Visible = True
Command8.Visible = True
End Sub

Private Sub Text1_Change()
    sbStatusBar.Panels(1).Text =
"Trayectoria de la Base de datos Activa"
End Sub

Private Sub Text1_MouseMove(Button
As Integer, Shift As Integer, X As Single,
Y As Single)
    sbStatusBar.Panels(1).Text = "Indica
la trayectoria de la base de datos activa"
End Sub
```

### 3.3. Diseño y construcción del Front-End

Los requerimientos del usuario final marcan la pauta de lo que será el **Front-End** ó vista final que le permitirá la manipulación de su información así como la generación de reportes de manera sencilla.

Intentando darle mayor funcionalidad al sistema y minimizando la posibilidad de error humano al operarlo, se ha diseñado un **Front-End** conformando básicamente por botones de acción ó comando, de fácil operación a través del mouse ó del teclado; incluyendo **combo-boxes** (listas- desplegables) para que el usuario agilice la entrada de información al seleccionar los datos que requiere de una lista previamente definida y obteniendo información relacionada con ésta selección sin necesidad de realizar operaciones complicadas de entrada de datos; barras de navegación estándares con las opciones más frecuentes para la operación de una base de datos (como son las funciones de agregar, eliminar, modificar, avance, retroceso, búsquedas, reportes impresos o en pantalla etc.).

Así mismo, un sistema de ayuda a través "**tooltips**" (mensajes cortos) que aparecen al posicionar el apuntador con el mouse sobre una determinada zona de la pantalla ó hacia algún botón de comando y a través de la barra de estado.

En general, se ha diseñado un sistema totalmente visual, la cual nos permite la generación de formas propias, la reutilización de código fuente, y de los formularios (pantalla de captura).

A continuación se describe cómo fue diseñado nuestro **Front-End** utilizando las herramientas de diseño de Microsoft Visual Basic versión 5.0.

#### **Creación de la interfaz**

Los formularios son la base para crear la interfaz de una aplicación. Se pueden usar

estos para agregar ventanas y cuadros de diálogo a la aplicación. También se pueden usar como contenedores de elementos que no son parte visible de la interfaz de la aplicación. Por ejemplo, puede haber un formulario en la aplicación que sirva como contenedor para gráficos que quiere presentar en otros.

### Construcción de un proyecto

Un proyecto es un contenedor de todos los componentes de un sistema, entre ellos, barras de menús, barras de herramientas, caja de herramientas, formularios, ventana de propiedades, etc.

Para crear un proyecto en Visual Basic 5.0, se selecciona EXE estándar y se presiona el botón abrir de Nuevo proyecto. Visual Basic crea un nuevo proyecto aparecerá un formulario nuevo, se asigna un nombre y un directorio o carpeta donde se almacenará, figura 3.3.1.

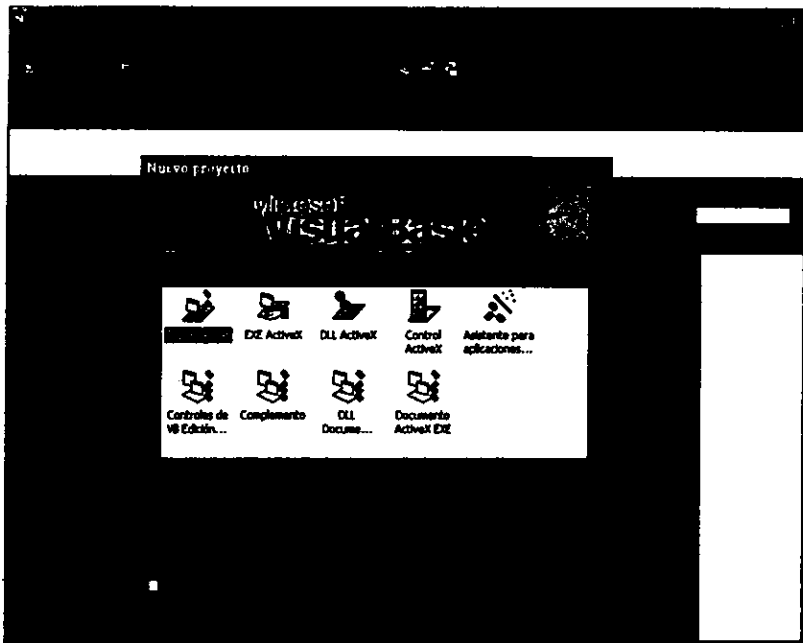


Figura 3.3.1. Nuevo proyecto.

Hay dos tipos de objetos en Visual Basic: formularios y controles. 1.- Los controles son objetos gráficos que nosotros dibujamos sobre un formulario, tales como cajas de texto, botones, etiquetas, marcos, listas, temporizadores. 2.- Un formulario es una ventana sobre la que dibujamos los elementos que el usuario tiene que utilizar para comunicarse con la aplicación, para cambiar el tamaño y la situación del mismo utilizamos el ratón. 3.- Ventana de propiedades en la cual se establecen las propiedades de un objeto. 4.- Explorador de proyecto encontramos los nuevos proyectos. 5.- Barra de herramientas, 6.- Ventana de código, figura 3.3.2.

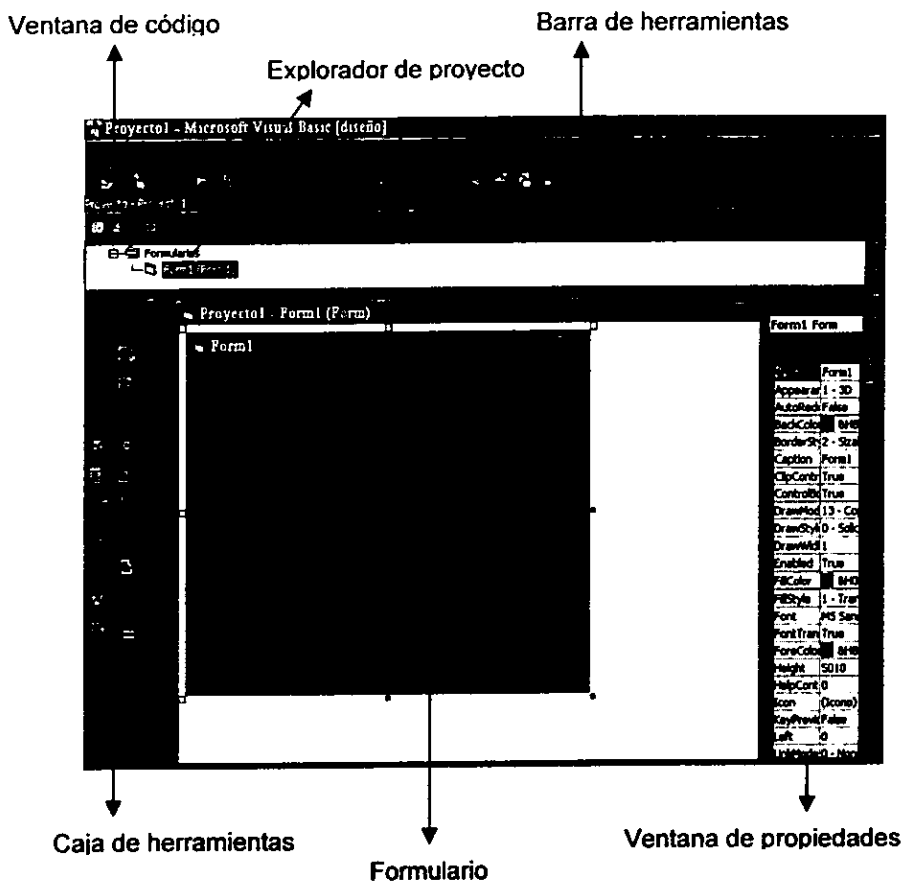


Figura 3.3.2. Visual Basic.

Para dibujar un control mediante el cuadro de herramientas. (Etiquetas y cuadro texto)

1. - Haga clic en la herramienta del control que ha elegido dibujar.
2. - Mueva el puntero del formulario. El puntero adoptará la forma de cruz.
3. - Coloque la cruz donde quiera que aparezca la esquina superior izq. del control.
4. - Arrastre la cruz que el control tenga el tamaño deseado.
5. - Suelte el mouse y escriba el texto en el botón creado, figura 3.3.3.

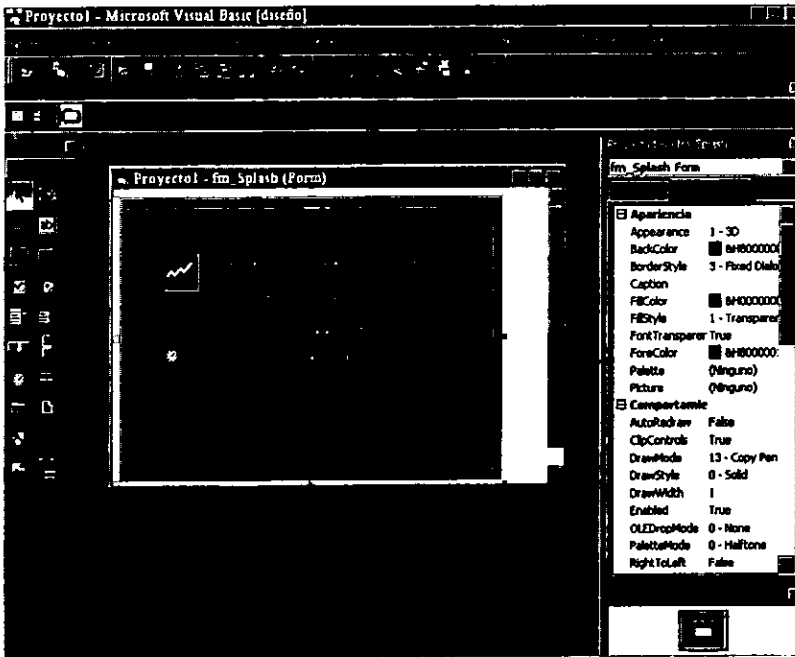


Figura 3.3.3. Etiquetas, temporizador y cuadros de texto.

La herramienta temporizador se diseña. Haciendo doble clic en el control temporizador y este aparece en el formulario, deberá establecer su propiedad Interval, misma que es la más importante de este control. Esta propiedad contiene un valor dentro del intervalo de 1 a 65535, el valor se encuentra en milisegundos. Su finalidad es trabajar en segundo plano y activar un evento en lapsos determinados por ticks del

reloj.

Muchas propiedades de los controles coinciden. La mayoría de los controles tienen las propiedades Left, Height, Top, Width y Visible. Para determinar el carácter que aparece cuando el usuario teclea una contraseña se realiza con la propiedad PasswordChar.

Para establecer o modificar las propiedades desde la ventana Propiedades del formulario.

1. - En el menú Ver, elija Propiedades o haga clic en el botón Propiedades de la barra de herramientas. La ventana propiedades presenta los valores del formulario o control seleccionado.
  2. - En la lista propiedades presentan los valores del formulario o control seleccionado.
  3. - En la columna de la derecha, escriba o seleccione el nuevo valor de la propiedad.
- La ventana Propiedades consta de los siguientes elementos, figura 3.3.4.

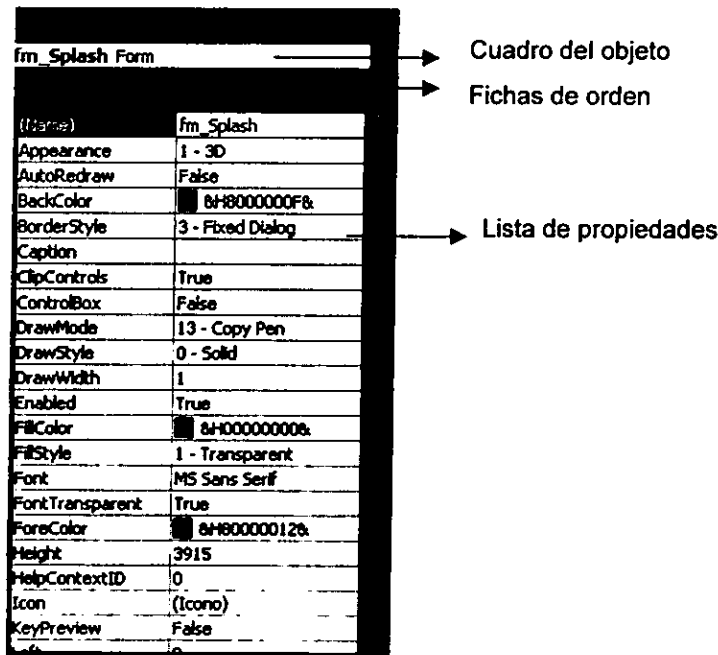


Figura 3.3.4. Ventana Propiedades.

Para asignarle el título al formulario (Acceso al sistema), se selecciona el formulario y a continuación la propiedad **caption** (Título) en la lista de propiedades. Después se sobrescribe en el texto "Form1" el título que deseamos que tenga nuestra pantalla.

Botones de comando. Estos botones determinan el momento en que el usuario desea hacer algo, como salir de la aplicación o empezar a imprimir. Los pasos para agregar un botón son:

Ubicar y ajustar el tamaño del botón en el formulario.

Cambiar las propiedades **Name** y **Caption** del botón. (La propiedad **Caption** contiene el mensaje que mostrará el botón), figura 3.3.5.

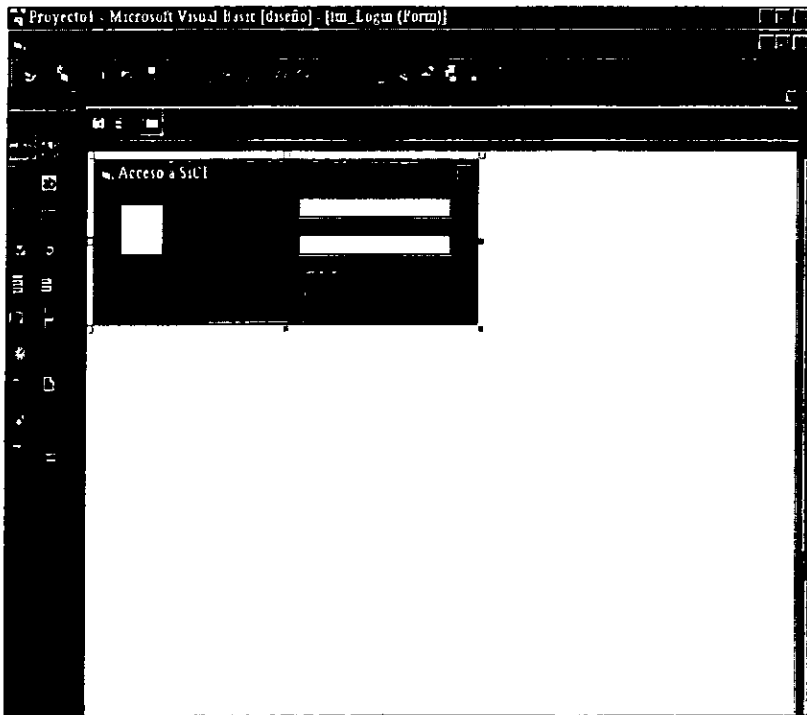


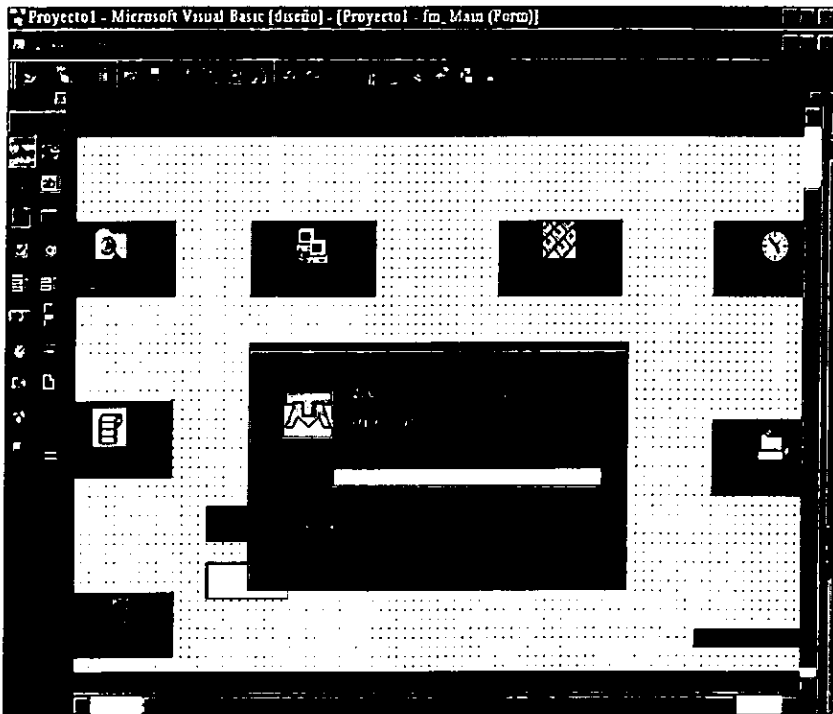
Figura 3.3.5. Nombre de formulario y botones.



El control Image despliega imágenes en su formulario. Las imágenes se encuentran en archivos, y el control determina cómo aparecerá la imagen en la pantalla.

Un control Image no mostrará ninguna imagen hasta que establezca las propiedades adecuadas. La más importante es Picture, pues es con la que se determina la imagen que aparecerá dentro de los límites de este control en el formulario.

Cuando haga clic en esta propiedad, Visual Basic desplegará unos puntos suspensivos que, al oprimirlos, mostrarán el cuadro de diálogo Cargar imagen. Visual Basic se acompaña de varios archivos gráficos, figura 3.3.6.



**Figura 3.3.6. Control Imagen.**

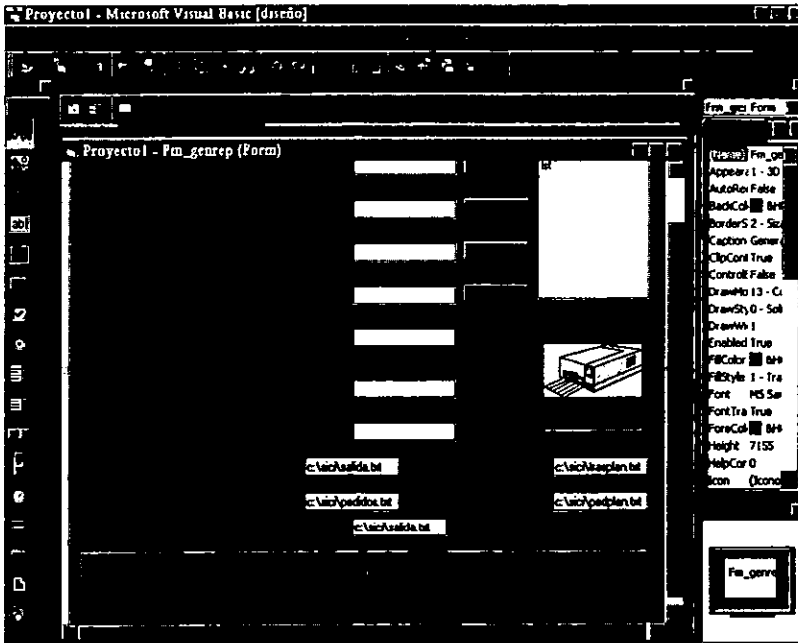


Figura 3.3.7. Barras de desplazamiento y lista desplegable.

Las barras de desplazamiento son a menudo utilizadas en cajas de texto o ventanas para desplazar la información ya sea: hacia abajo, arriba, a la izquierda o a la derecha de la ventana o también como controles se selecciona de la caja de herramientas y en la ventana de propiedades se le especifica la propiedad Min, y el valor máximo mediante la propiedad Max.

Cuando el usuario hace clic encima o debajo del cuadro de desplazamiento éste se desplaza una cantidad negativa o positiva dependiendo del valor de la propiedad **Large Change**, el largo y corto del desplazamiento depende de la propiedad. El usuario puede arrastrar el propio cuadro de desplazamiento para ubicarlo en cualquier posición de la barra de desplazamiento en lugar de cambiar el valor en forma gradual.

La lista desplegable combina una caja de texto y una lista. Permite al usuario escribir lo que desea seleccionar o elegir un elemento de la lista. Una lista desplegable estándar se obtiene poniendo la propiedad `Style` de dicho control a valor 0 (Dropdown Combo).

Cuando el usuario haga clic en la flecha se visualiza la lista de elementos y podrá optar por elegir un elemento de la lista o escribir directamente el elemento deseado en la caja de texto, figura 3.3.7.

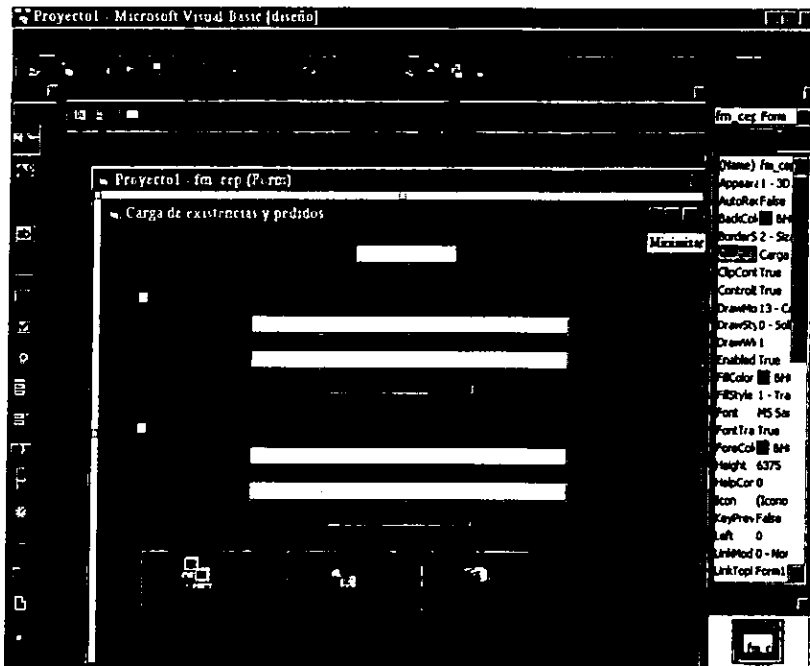


Figura 3.3.8 Barra de avance.

Barra de avance de carga, es una barra que muestra el avance al cargarse un proceso y depende de este mismo.

Inhibir una parte de la pantalla en este caso existencias o pedidos se realiza por medio de la propiedad de `enabled` que devuelve o establece un valor que determina

si un objeto puede responder a eventos generados para valorar de verdadero o falso, figura 3.3.8.

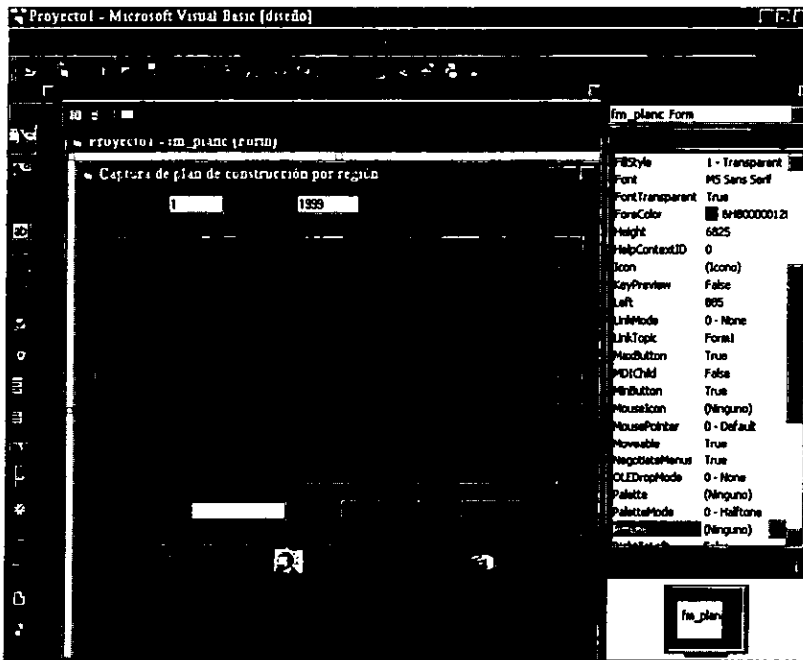


Figura 3.3.9. Control rejilla.

El control rejilla visualiza la información en celdas. Una celda es la región formada por la intersección de una fila y una columna. El usuario puede situarse en una celda cualquiera haciendo clic sobre ella utilizando las teclas de movimiento del cursor. Durante la ejecución, el usuario puede seleccionar una o más celdas, pero puede modificar su contenido escribiendo directamente sobre ellas. La forma de modificar una celda es ejecutando el código apropiado, figura 3.3.9.

Los menús son controles avanzados con propiedades que se establece mediante el editor. En la mayoría de los elementos de menú se requiere especificar las propiedades

Caption y Name, así como la sangría para indicar su independencia. De manera opcional, se les pueden asignar teclas de método abreviado y marcas de verificación.

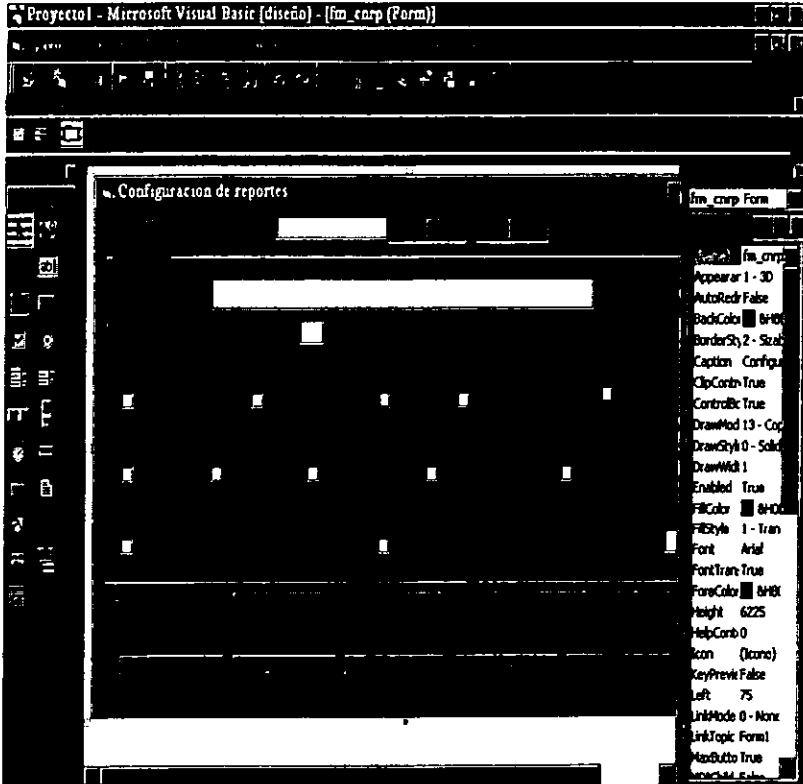


Figura 3.3.10. Casillas de verificación.

Las casillas de verificación son controles CheckBox tienen las mismas propiedades de un OptionButton, excepto que la propiedad Value no sólo indica si el control está seleccionado (si vale 1) o no (si vale 0), sino también indica si está atenuado (si vale 2). Los usuarios utilizan una casilla de verificación atenuado para determinar si una parte de una opción seleccionada es verdadera.

### 3.4. Pruebas e Integración del Sistema

Las pruebas constituyen una parte integral y vital del ciclo de vida del desarrollo de sistemas. Se realizan con el propósito de descubrir defectos y se establecen para mejorar la calidad del sistema.

En la Tabla 3.4.1. Se enlistan los elementos para llevar a cabo las pruebas de integración del sistema, las cuales se describirán más adelante.

Categoría de Pruebas	Consideraciones para la ejecución de pruebas	Infraestructura de pruebas	Recursos Humanos
Unitarias	Riesgos y suposiciones	Ambiente de pruebas	Preparación de las pruebas
Integración	Condiciones y restricciones	Organización de las pruebas	Casos de pruebas
Regresión	Cobertura funcional de las pruebas	Metodología de las pruebas	Procedimientos de las pruebas
Volumen	Descripción de la arquitectura del sistema.	Herramientas de Pruebas	
Aceptación Del Usuario		Puntos de control y aprobaciones	
Caja Blanca		Criterios de suspensión y conclusión de las pruebas.	
Caja Negra			
Estáticas			
Funcionales			
Estructurales			

Tabla 3.4.1. Elementos para llevar a cabo las Pruebas de Integración del Sistema.

### **Categoría de pruebas**

Las categorías de pruebas se deducen directamente de la lista de requerimientos funcionales y estructurales del plan de pruebas. La especificación de las categorías de pruebas a usar es importante ya que basados en ellas podemos determinar la infraestructura de pruebas requerida.

Las categorías de las mismas pueden ser: Unitarias, de Integración, de Regresión, de Concurrencia, de Volumen, de Aceptación, del Sistema, Manuales o Automáticas.

### **Fundamentos para la ejecución de pruebas**

Los fundamentos de las pruebas definen los objetivos esenciales para las pruebas del software.

Es en esta fase donde el ingeniero intentará demoler todo el software que ha construido en las etapas de análisis y diseño.

Las pruebas requieren que se descarten las ideas acerca de lo correcto que es el software desarrollado y que al descubrir los errores, se logre superar cualquier conflicto en el sistema.

### **Las pruebas nos permiten:**

- Sentar las bases para determinar los objetivos y un plan específico de pruebas.
- Asegurar la obtención y formalización de los requerimientos del usuario y verificar que son adquiridos de una manera completa, correcta y consistente.
- Verificar los requerimientos funcionales así como los estructurales y establecerlos como fundamento para realizar las pruebas del sistema.
- Detectar y registrar defectos asociados a los requerimientos establecidos.
- Establecer la documentación de reportes para pruebas realizadas.

### **Pruebas Unitarias**

Categoría de pruebas realizadas sobre un programa o módulo con la intención de encontrar problemas funcionales en la lógica, así como problemas técnicos en el código. La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software – el módulo. Usando la descripción del diseño detallado como guía, se prueban los caminos de control importantes, con el fin de descubrir errores dentro del ámbito del módulo. La complejidad relativa de las pruebas y de los errores descubiertos esta limitada por el alcance estricto establecido por la prueba de unidad. La prueba de unidad siempre esta orientada a la caja blanca y este paso se puede llevar a cabo en paralelo para múltiples módulos.

### **Pruebas de Integración**

Categoría de pruebas realizadas a un grupo de programas para asegurar que los datos y controles sean pasados adecuadamente entre controles. La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción, el objetivo es tomar los módulos probados en unidad y construir una estructura de programas que este de acuerdo con lo que dicta el diseño.

### **Pruebas de Regresión**

Categorías de pruebas selectivas para detectar fallas que se hayan introducido durante las modificaciones a un sistema o componente, que permiten verificar que estas modificaciones no impacten en forma negativa y que se siga cumpliendo con los requerimientos planteados.

### **Pruebas de Volumen**

Categoría de pruebas realizadas para verificar el comportamiento adecuado y eficiente



de una aplicación bajo condiciones de volumen (número de operaciones), competencia de recursos (conurrencia) y carga máxima (velocidad de petición de ejecución de una operación) así como el comportamiento eficiente bajo las condiciones de volumen máximo (cantidad de datos) en las aplicaciones.

### **Pruebas de Caja Negra**

Pruebas funcionales basadas en los requerimientos sin conocimiento sobre cómo fue construido el sistema y usualmente dirigidas a los datos.

### **Pruebas de Caja Blanca**

Pruebas basadas en el conocimiento sobre la lógica y estructura internas. Usualmente dirigidas a la lógica.

### **Pruebas de Aceptación del Usuario**

Categoría de pruebas finales, ejecutadas por el usuario, para asegurar que el sistema satisfaga las necesidades de la organización o usuario final (validan que el sistema construido es el correcto).

### **Pruebas Estáticas**

Consiste en la revisión y validación de los documentos generados en las distintas fases de la vida de un proyecto. Verificación realizada sin ejecutar el código del sistema.

### **Pruebas Funcionales**

Validan los requerimientos de la organización (lo que se supone que el sistema debe hacer), pretenden descubrir errores cometidos en la implantación de dichos requerimientos.

## **Pruebas Estructurales**

Validan la arquitectura del sistema confirmando que todas sus partes funcionen sincronizadamente y que la tecnología esta siendo usada apropiadamente. Se refieren a las características técnicas, como su comportamiento con grandes volúmenes de información, tiempos de respuesta, etc.

### **Consideraciones importantes para la ejecución de las pruebas**

#### **Condiciones y restricciones**

Generalmente son limitaciones o problemas de naturaleza técnica y están relacionadas con el desarrollo del proyecto en sí, la tecnología de pruebas, el estado de los ambientes de pruebas, etc.

#### **Riesgos y suposiciones para las pruebas**

Los riesgos son aquellos factores que pueden afectar negativamente la ejecución de las pruebas. Las suposiciones son las premisas que pueden afectar positiva o negativamente la ejecución de las pruebas complicando o facilitando las actividades de las pruebas.

#### **Cobertura funcional de las pruebas**

Dentro de la cobertura funcional de las pruebas se deben describir y listar de manera clara y concisa las funciones a probar, así como aquellas funciones a no ser probadas aún siendo parte del proyecto, ya que son necesarias especialmente cuando se requiere explicar él porque de su exclusión definiendo el alcance de las pruebas y delimitando responsabilidades.

Además se debe documentar el ciclo del sistema a ser simulado con el objeto de

ejecutar cada una de las funciones objeto de las pruebas. Esta simulación suele ser realizada con muestras de datos fuera de especificaciones.

### **Descripción de la arquitectura del sistema**

Para ello se consideran las especificaciones del software de base sobre el cual esta construido el sistema tal como la plataforma, el software de base de datos, el sistema operativo, el lenguaje de programación, etc.

### **Infraestructura de pruebas**

**La infraestructura de pruebas contempla lo siguiente:**

#### **Organización de las pruebas**

Definir la organización (puestos y responsabilidades) que es requerida para la construcción y ejecución de las pruebas.

#### **Ambiente de pruebas**

Identificación de los ambientes donde se ejecutarán las pruebas, así como mencionar las características generales de los datos de prueba (que datos se necesitan y como se obtendrán). Tomando como base el modelo de datos del proyecto. Esta mención de los datos de prueba, es importante para saber cuántos y cuáles datos serán seleccionados, y para la estimación de la carga de trabajo necesaria para generarlos.

#### **Herramientas de prueba**

En este punto es necesario identificar los productos a utilizar y el uso específico que se hace con ellos.

Básicamente se debe determinar si es preciso vigilar todos los componentes o solamente algunos, el interés de vigilar a determinados componentes, es justificado por la necesidad de verificar cual es el comportamiento interno de dicho componente, esto es, como se realiza el procesamiento de la información. En el caso de estar solamente interesados, en las entradas y salidas de los procesos, es suficiente muchas veces el verificar estas entradas y salidas sin tener en cuenta exactamente cómo se leen y como se generan.

La importancia de este punto radica en la decisión de adoptar el enfoque de "caja blanca" (vigilando el comportamiento interno de los componentes) lo cual implicará una carga adicional de trabajo debido al número adicional de casos de prueba.

### **Metodología de las pruebas**

En este punto es importante determinar si existe un procedimiento de pruebas dentro de la organización o si es necesario elaborarlo y en que medida este procedimiento está integrado con el resto de los sistemas de desarrollo y mantenimiento.

Gracias a esta información, es posible estimar el esfuerzo adicional requerido para la construcción de estos procedimientos, con el objeto de realizar las pruebas.

Se deberá intentar localizar los procedimientos de pruebas de uso general que puedan ser de utilidad.

### **Puntos de control y aprobaciones**

Se deben especificar los puntos de control en el transcurso de la construcción y ejecución de las pruebas, tal como el determinar los puestos de las personas que tendrán que autorizar la continuación de las pruebas acorde con el plan original o asumiendo las variaciones incorporadas al mismo.

### **Criterios de suspensión y conclusión de las pruebas**

Estos criterios se refieren a la suspensión o terminación de la ejecución de los casos de prueba cuando son necesarios otros componentes que no tienen listos ó cuando el número de los defectos encontrados sobrepasan el límite de los esperados para lo cual es necesario regresar a la etapa de desarrollo y verificar las especificaciones.

### **Recursos Humanos**

Identificar las personas específicas a ocupar los puestos en la organización de pruebas. Esta información es de utilidad ya que al asociar personas a los puestos se descubren los factores que les hace falta para el efectivo cumplimiento de sus responsabilidades.

### **Preparación de las pruebas**

Es necesario realizar todas las actividades que aseguren la disponibilidad de la infraestructura de pruebas acorde con las categorías de pruebas a ejecutar.

Los criterios a tener en cuenta para decidir que tipo de plan necesitamos son los siguientes:

1. Tamaño y complejidad del Proyecto.
2. Lista de funciones, requerimientos, condiciones generales tanto funcionales como estructurales.
3. Arquitectura de la aplicación.
4. Condiciones y restricciones.
5. Ambiente de pruebas.

**Criterios para iniciar la ejecución de las pruebas.-** Se deben especificar los eventos de carácter general a acontecer con el objeto de iniciar la ejecución del plan de pruebas.

**Preparación y verificación del ambiente de pruebas.-** Especificar las actividades más

relevantes para la preparación del ambiente de pruebas.

Una vez especificadas las actividades es posible identificar los procedimientos de pruebas necesarios (existentes y los que requieran ser creados).

A partir de los requerimientos funcionales y estructurales se deben generar los casos de pruebas.

### **Casos de pruebas**

Los casos de pruebas son las condiciones de prueba ejecutables, que incluye un conjunto de datos de entrada, resultados esperados y resultados obtenidos.

Los casos de prueba contemplados en la ejecución de estas se toman a partir de los requerimientos funcionales y estructurales con el siguiente criterio:

Datos dentro de especificaciones.

Datos fuera de especificaciones.

### **Procedimientos para las pruebas**

Así como la metodología de pruebas, en caso de no existir un procedimiento dentro de la organización para la realización de las pruebas, se hace necesario elaborar los pasos a seguir para terminar esta fase con óptimos resultados.

### **Ciclo del sistema**

El ciclo del sistema que se va a probar consiste en la transferencia de archivos de movimientos, existencias y pedidos, carga de archivos de necesidades de usuario, captura de necesidades extraordinarias, cálculo de consumos, configuración y generación del reporte principal de análisis, generación de los reportes de pedidos y

traspasos, finalmente análisis de pedidos sin entregar y dotaciones de empleados. Este ciclo se muestra en la figura 3.4.1.

## Flujo de Información

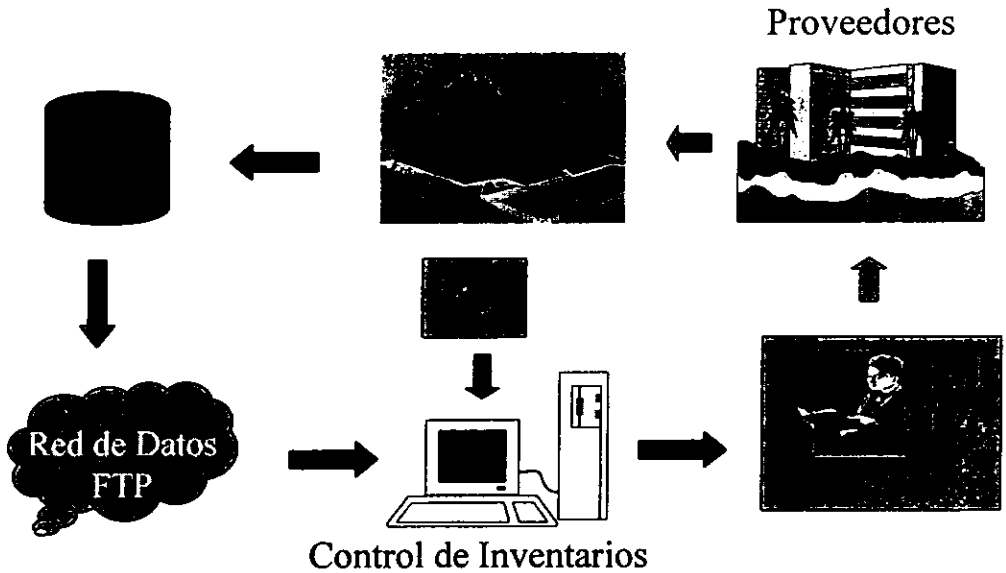


Figura 3.4.1. Ciclo del sistema.

### Reglas de Negocio

Las pruebas que se realizarán al sistema son fundamentadas en la información que maneja la Empresa Telefónica para la que se diseña el sistema. Obedeciendo sus reglas y validaciones para asegurar la integridad de la información dentro del sistema.

### Lista de Reglas de Negocio

### Identificación de almacén

Nombre: Clave de Almacén

Código: almacen

Tipo: definición

Descripción:

Clave de almacén: 3 caracteres alfanuméricos.

Siempre debe existir.

#### **Nombre de almacén**

Nombre: Nombre de Almacén

Código: nombre

Tipo: definición

Descripción:

Nombre de almacén: 30 caracteres alfanuméricos.

Siempre debe existir.

#### **Región de distribución**

Nombre: Región

Código: region

Tipo: definición

Descripción:

Región: 15 caracteres alfanuméricos

Siempre debe existir.

#### **Número de catálogo de material**

Nombre: Catálogo del Material

Código: catalogo

Tipo: definición

Descripción:

Catálogo del material: 7 caracteres alfanuméricos.

Siempre debe existir.



### **Descripción del material**

Nombre: Descripción del Material

Código: descripcio

Tipo: definición

Descripción:

Descripción del material: 30 caracteres alfanuméricos.

Siempre debe existir.

### **Unidad de medida del material**

Nombre: Unidad de Medida

Código: unidad

Tipo: definición

Descripción:

Unidad de medida: 3 caracteres alfanuméricos.

Siempre debe existir.

### **Cuenta contable**

Nombre: Cuenta Contable

Código: cta

Tipo: definición

Descripción:

Cuenta contable: 12 caracteres alfanuméricos.

Siempre debe existir.

### **Fecha de ingreso de datos**

Nombre: Fecha de ingreso

Código: fechaintro

Tipo: definición

Descripción:

Fecha de ingreso: formato fecha dd/mm/aaaa

Siempre debe existir.

### **Fechas de movimiento y proceso**

Nombre: Fecha  
Código: fecha  
Tipo: definición

Descripción:

Fecha: formato fecha dd/mm/aaaa

Se llenarán dependiendo de la información requerida.

### **Cantidad de material**

Nombre: Cantidad de Material  
Código: cant  
Tipo: definición

Descripción:

Clave de Almacén: 9,999,999,999 Caracteres numéricos

Siempre debe existir.

### **Código de proyecto ADM (Autorización de Desembolso Mayor)**

Nombre: Autorización de Desembolso Mayor  
Código: adm  
Tipo: definición

Descripción:

Autorización de desembolso mayor: 12 caracteres alfanuméricos.

Indica si el movimiento afecta a un proyecto.

### **Nombre del Servidor**

Nombre: Nombre del Servidor  
Código: nom\_serv  
Tipo: definición

Descripción:

Nombre del servidor: 12 caracteres alfanuméricos.

Siempre debe existir.

### **Dirección IP**

Nombre: Dirección IP

Código: dir\_ip

Tipo: definición

Descripción:

Clave de Almacén: 13.999.999.999 caracteres numéricos

Siempre debe existir.

### **Puerto de acceso**

Nombre: Puerto de Acceso

Código: puerto

Tipo: definición

Descripción:

Puerto de acceso: 999 caracteres numéricos

Siempre debe existir.

### **Clave de acceso**

Nombre: Clave de Acceso

Código: clave\_acceso

Tipo: definición

Descripción:

Clave de acceso: 8 caracteres alfanuméricos.

Siempre debe existir.

### **Contraseña de acceso**

Nombre: Contraseña de Acceso

Código: pasaporte

Tipo: definición

Descripción:

Contraseña de acceso: 8 caracteres alfanuméricos.

Siempre debe existir.

### **Rutas de archivos**

Nombre: Ruta  
Código: ruta\_  
Tipo: definición

Descripción:

Ruta: 100 caracteres alfanuméricos.

Siempre deben existir.

### **Área de trabajo**

Nombre: Área de Trabajo  
Código: areatm  
Tipo: definición

Descripción:

Área de trabajo: 1 caracteres alfanuméricos.

Siempre debe existir.

## **Puntos de control para pruebas unitarias, de caja blanca e integración**

### **Módulo de Transferencia de archivos**

Especificaciones:

- Captura de fecha de ingreso válida.
- Selección única de archivo a transferir.
- Especificación de nombres de archivo en rutas sugeridas.

Fuera de especificaciones:

- Rechazo en fecha errónea.
- Desactivación de un archivo en caso de seleccionar el otro.
- Rechazo de nombres nulos.

En la figura 3.4.2. se muestra un ejemplo de la forma como se rechaza la fecha errónea en este módulo.

#### Módulo de Carga de Información de usuarios

##### Especificaciones:

- Captura de fecha de ingreso válida.
- Selección única de archivo a cargar.
- Selección de campos que contienen información.

##### Fuera de especificaciones:

- Rechazo de fecha no válida.
- Rechazo de archivo inexistente.

#### Módulo de Captura de Necesidades Extraordinarias

##### Especificaciones:

- Captura del año y área a procesar.
- Incorporación de datos numéricos en matriz de regiones y meses.

##### Fuera de especificaciones:

- Rechazo de año inválido.
- Rechazo de cantidades no numéricas.

En la figura 3.4.3. se muestra la forma como rechaza el sistema un dato en el campo de "año", que no es válido.

#### Módulo de Cálculo de Consumos

##### Especificaciones:

- Captura de fechas de referencia para el cálculo.
- Captura de fecha de registro del resultado del cálculo.
- Cálculo automático de consumos promedio al proporcionar el número de meses a promediar.

##### Fuera de especificación:

- Rechazo de fechas no válidas.
- Rechazo del cálculo automático si no se proporciona el número de meses a promediar.

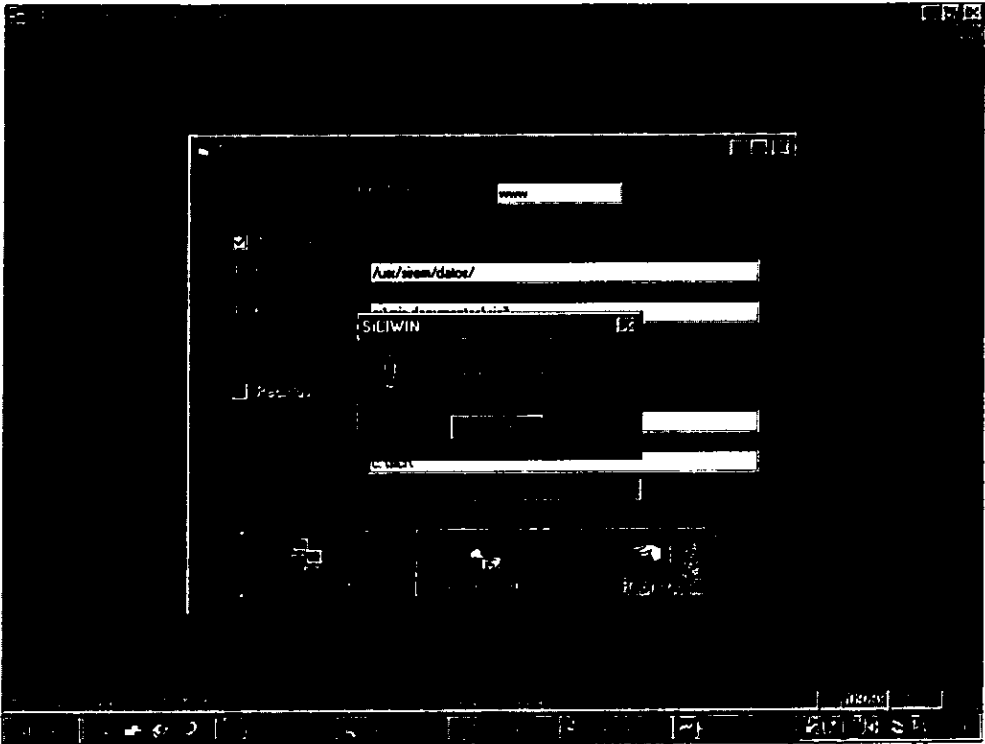


Figura 3.4.2. Muestra de un rechazo de fecha en el módulo de carga de información.

### Módulo de Configuración del Reporte Principal

#### Especificaciones:

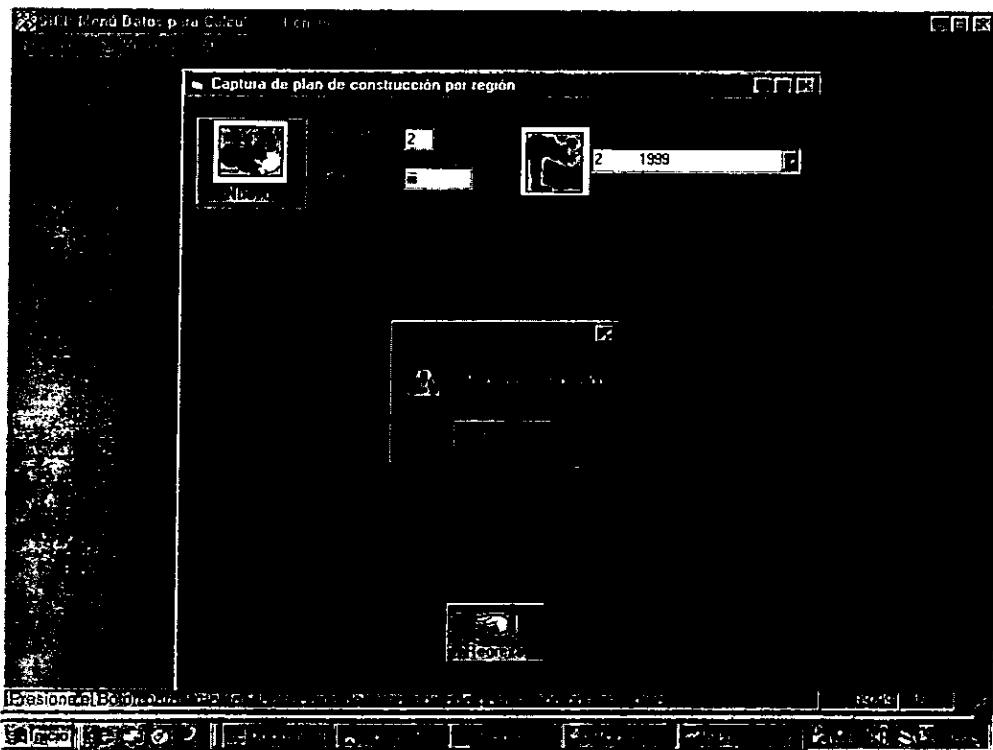
- Selección de las regiones con almacenes auxiliares controlados.
- Creación de conjuntos de catálogos seleccionando los catálogos que los conforman y capturando su descripción.
- Definición del formato de los reportes con su respectiva descripción, proporcionando los datos que lo conforman y los catálogos y/o conjuntos que debe incluir.

#### Fuera de especificaciones:

- Limitación para no seleccionar regiones que no existan.
- Limitación a no poder seleccionar catálogos de material inexistentes en la

creación de conjuntos y rechazo en caso de no proporcionar la descripción correspondiente.

- Limitación a no seleccionar catálogos o conjuntos inexistentes al definir un formato de reporte.
- Rechazo en caso de no proporcionar la descripción del formato y los meses a



procesar.

**Figura 3.4.3. Muestra de Rechazo de año no válido en el módulo de Captura de Necesidades Extraordinarias.**

En la figura 3.4.4. Se muestra la forma como rechaza el sistema cuando se pretende actualizar un formato sin proporcionar la descripción del mismo.

Módulo de Generación del Reporte Principal, de Pedidos y de Traspasos

**Especificaciones:**

- Selección del formato de reporte que se desea generar.
- Selección a través de la fecha de ingreso de los datos de existencias, pedidos y consumos que se considerarán en el reporte.
- Captura de una fecha válida como referencia del reporte.
- Captura de los nombres de archivo de salida de los reportes.
- Generación automática de los reportes tras la pulsación del botón correspondiente.

**Fuera de especificaciones:**

- Limitación a solo seleccionar un formato de reporte existente.
- Limitación a solo seleccionar una fecha de ingreso para los archivos de existencias, pedidos y consumos.
- Rechazo en caso de que la fecha de referencia no sea en el formato correcto.
- Rechazo en caso de no proporcionar el nombre de los archivos de salida.

En la figura 3.4.5. se muestra la forma como se limita la selección del formato de reporte a procesar en el módulo de generación del Reporte Principal, de traspasos y de pedidos.

**Módulo de reportes de Pedidos sin Entregar y Dotaciones de Usuario**

**Especificaciones:**

- Selección a través de la fecha de ingreso de los datos de pedidos y existencias que debe considerar el reporte de pedidos sin entregar.
- Captura de las fechas del periodo a considerar y de referencia.
- Selección de los catálogos a considerar en el reporte de dotaciones de usuario.
- Captura del nombre de los archivos de salida de los reportes.
- Generación automática de los reportes tras la pulsación del botón correspondiente.

**Fuera de especificaciones:**

- Limitación a solo seleccionar las fechas de ingreso registradas en el sistema



tanto de pedidos como de existencias

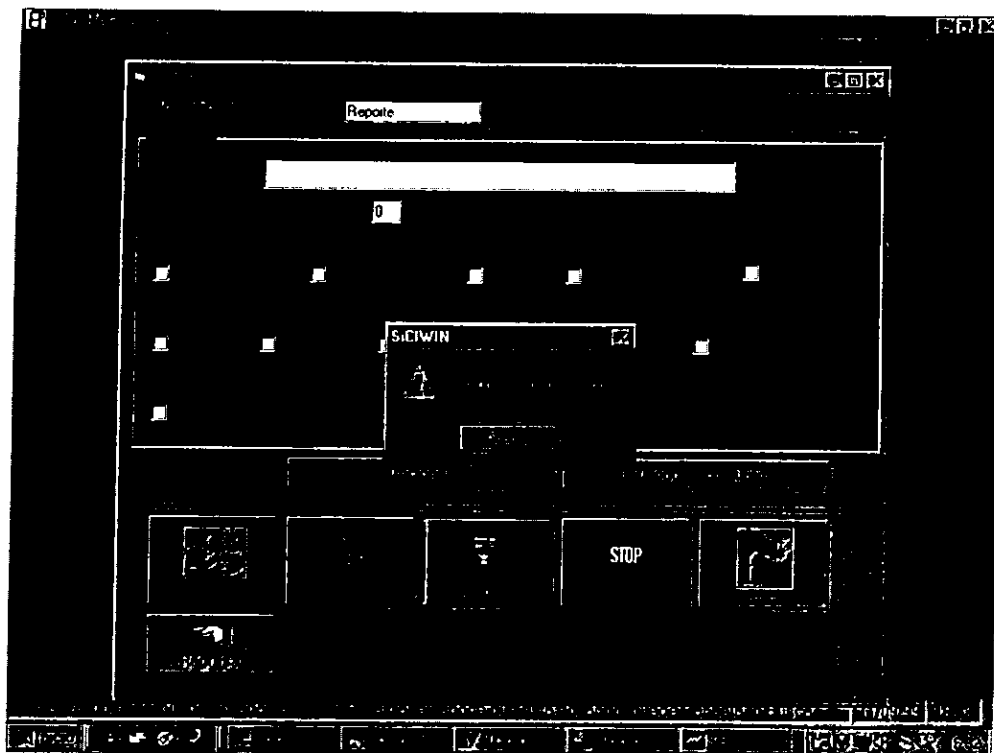


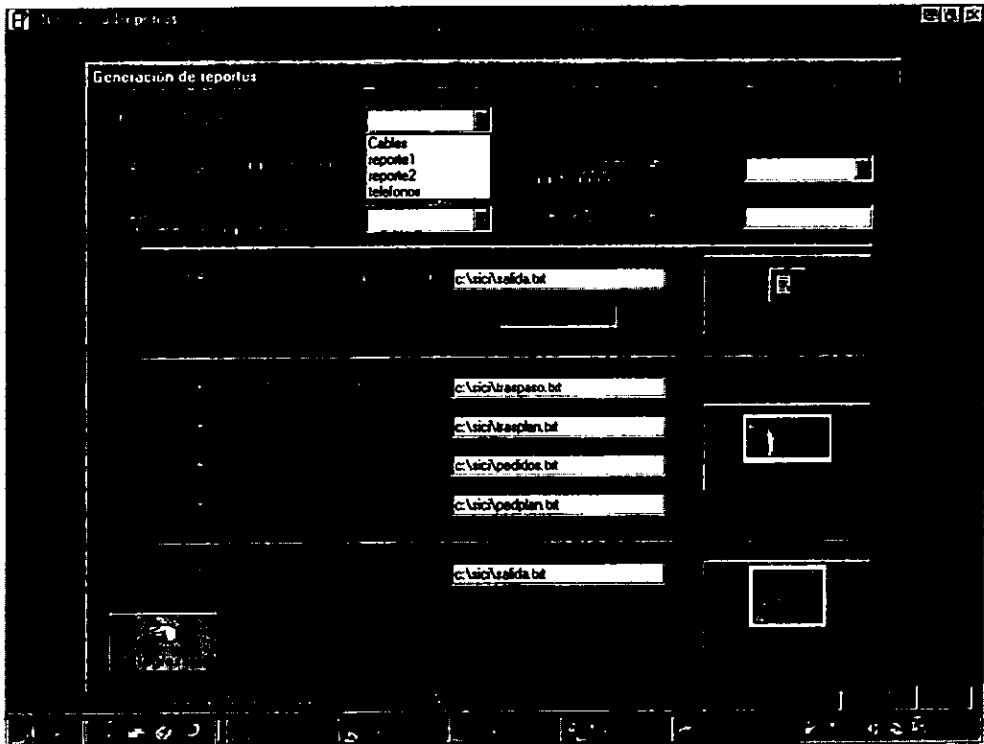
Figura 3.4.4. Muestra de rechazo de un formato de reporte sin descripción.

- Rechazo de fechas de referencia y de rango fuera de formato
- Limitación a solo seleccionar catálogos existentes en el sistema
- Rechazo en caso de no proporcionar un nombre de archivo de salida de los reportes

**Puntos de control para pruebas funcionales, de caja negra y de aceptación del usuario final.**

Para realizar esta etapa de pruebas fue necesario instalar el sistema en un equipo que cumpliera con las características previamente definidas en el manual del usuario, para

que este último lo utilizara siguiendo los procedimientos del mismo con el fin de valorar su funcionalidad, apegándose al siguiente plan:



**Figura 3.4.5. Muestra de la lista desplegable que limita la selección del formato de reporte a procesar.**

- Configuración de las conexiones ftp para la transferencia de archivos por parte del administrador.
- Carga de catálogos del sistema por parte del administrador.
- Creación de usuarios con el perfil correspondiente.
- Transferencia de archivos de pedidos.

- Carga de archivos de pedidos con diferentes fechas.
- Transferencia de archivos de existencias.
- Carga de archivos de existencia con diferentes fechas.
- Transferencia de archivos de movimientos.
- Carga de archivos de movimientos con diferentes fechas.
- Carga de archivos de necesidades y pronósticos de usuarios.
- Captura de necesidades extraordinarias de usuarios para diferentes áreas.
- Ejecución del cálculo de consumo de Telefonía Pública.
- Ejecución del cálculo de consumos promedio de Planta externa.
- Creación de conjuntos de materiales.
- Selección de regiones con existencia en auxiliares controlada.
- Configuración de formatos de reporte.
- Generación del reporte principal con diferentes formatos.
- Generación de los reportes de pedidos y traspasos para cada reporte principal.
- Generación de reportes de Entregas a usuario.
- Generación de reportes de avance de pedidos.
- Generación de los respaldos de información.

Al término de un periodo de un mes de pruebas con este plan, el usuario pudo comprobar que el sistema cubre con todos los requerimientos planteados en la etapa de análisis y quedando satisfecho con los resultados obtenidos, se concluyeron las pruebas y se dio por liberado el sistema para su implantación.

### **Integración e implantación del sistema**

Acuerdos para la liberación del sistema:

En la figura 3.4.6. se muestra la carta de acuerdos de liberación y aceptación del sistema con la que se ampara la completa satisfacción de la empresa y los desarrolladores del sistema denominado "Control de Inventarios para los almacenes de una Empresa Telefónica".

**CARTA DE LIBERACIÓN DE PRODUCTOS**

Proyecto: **91 37 Sistema de Control de Inventarios**  
 Cliente: **Gerencia de Control de Inventarios de Compañía Telefónica, S.A. de C.V.**  
 Atención: **Gerente de Control de Inventarios**  
 Lugar y fecha: **México, D.F. a 17 de Septiembre de 1999**

Por medio de la presente, me es grato informarle que de acuerdo con los términos establecidos para el proyecto anteriormente citado, y gracias al esfuerzo conjunto de su distinguida empresa y el equipo de trabajo de Desarrollo, se concluyeron satisfactoriamente las actividades planteadas para tal fin, por lo que hago entrega oficial del producto que a continuación se describe:

- Carga de existencias, pedidos y movimientos de SIREM vía FTP.
- Carga de pronósticos y configuración de conexión y archivos.
- Captura de plan anual por región.
- Captura de materiales para construcción de líneas en áreas de la Empresa
- Captura de ADM's de mantenimiento.
- Captura de consumos por mantenimiento en telefonía pública.
- Captura de consumo y consumo promedio por datos de SIREM.
- Cálculo de consumo por construcción en telefonía pública.
- Creación de conjuntos de catálogos para generación de reportes dinámicos.
- Configuración de reportes dinámicos, Generación de reportes dinámicos.
- Reporte de entregas a usuario, Reporte de avance de pedidos.
- Carga de catálogos, captura de almacenes.
- Captura de regiones factibles para uso en generación de reportes de plan de pedidos y traspasos.
- Reportes de plan de pedidos y traspasos.
- Manual de usuario, documentación técnica y documentación del programa.
- 1 Disco con código del sistema (93 archivos 22/08/99), 1 disco con documentación (22/08/99) y 6 discos de instalación.

Quedando todas las partes de acuerdo con la terminación del proyecto, y en espera de vernos nuevamente favorecidos con su preferencia, firman de conformidad.

Vo. Bo.

Nombre y firma del responsable  
Líder de proyecto, Desarrollo

Nombre y firma del cliente  
Gerente General de Almacenes  
Compañía Telefónica S.A. de  
C.V.

Nombre y firma del usuario  
Gerente Control de  
Inventarios

**3.4.6. Carta de aceptación del sistema.**

### 3.5. Generación de reportes para la toma de decisiones

El sistema genera varios reportes que son utilizados para la toma de decisiones, estos reportes tienen la característica de poder ser impresos directamente en una impresora local, o importados a una hoja de cálculo para su análisis, darles formato o transportar en forma sencilla. Estos reportes son los siguientes:

#### Reporte Principal

Este reporte es configurable, su objetivo es mostrar un panorama global de los materiales seleccionados para observar los posibles problemas que puedan existir, como el faltante en determinado tiempo y región, sobrantes factibles de traspasar, pedidos que están programados y/o vencidos, pronósticos que puedan estar afectando el consumo, etc.

Las columnas que se pueden observar en este reporte están divididas en dos partes, la primera son datos base que se utilizan para ir haciendo cálculos o son informativos, la segunda son los resultados del cálculo por cada mes y los datos que afectan a cada cálculo.

Dentro de los datos base del reporte, se tienen los siguientes:

- *Catálogo*: Es la clave del material, solo se muestran los materiales que se seleccionaron para el reporte o los conjuntos de ellos, los cuales agrupan los datos de varios materiales. Esta columna siempre aparece.
- *Descripción*: Es la descripción del material listado o del conjunto de materiales; esta columna siempre aparece.

- **Región:** La región, es la que se surte por un almacén distribuidor, por lo que cada región representa un almacén de este tipo. Aparecen todas las regiones que contiene el catálogo de regiones; es una columna que siempre aparece.
- **Exis\_Ac:** "Existencias Actuales"; esta columna es opcional y solo aparece cuando en la configuración del reporte se marca la casilla con el mismo nombre. Se presenta la existencia que se tiene en el archivo de existencias seleccionado, ya sea de todos los almacenes de la región (sumatoria) o solo las del distribuidor, esto dependiendo de que la casilla "Total existencias actuales solo sobre almacenes distribuidores" este marcada o no.
- **Cons\_Mt:** "Consumos Mantenimiento"; es el consumo promedio calculado para mantenimiento y solo aparece si se marca la casilla "Consumo mnto" en la configuración del reporte. Este valor se va disminuyendo de la existencia en cada mes presentado, para obtener la nueva existencia del mes.
- **Stock:** Este dato es la multiplicación del consumo promedio por dos, y representa la cantidad base del material que debe tener la región para seguridad, se resta de inicio de la existencia para comenzar a calcular la posible existencia de los siguientes meses. También es un campo opcional que se va a mostrar solo si se pide en la configuración del reporte.
- **Pend\_Ant:** Este campo presenta la cantidad que se tiene en pedidos con fecha menor a la fecha de referencia con la que se genera el reporte, se imprime al final de las columnas siempre y cuando se seleccione la casilla correspondiente en la configuración del reporte.

Los datos que se pueden presentar por cada mes, también se configuran, si alguno de ellos está marcado para que aparezca, se va a observar y va a sumarse o restarse según su naturaleza. El número de meses que se va a proyectar y observar en el

reporte, también se indica en la configuración del mismo. A continuación se describen los campos que pueden aparecer por mes:

- *Cons\_pr* y *Exi\_Pro*: Aparece la información que esté capturada de pronósticos de construcción para el mes que se presenta; esta cantidad se resta de la existencia inicial o del mes anterior, para presentar la columna de existencia después del pronóstico.
- *Con\_Mtp* y *Exi\_Mtp*: Aparece la información que está capturada como consumo de mantenimiento de telefonía pública para el mes que se presenta; se resta también de la existencia inicial o del mes anterior para presentar una posible existencia después de ese consumo.
- *Cons\_TP* y *Exi\_Ctp*: Muestra la información que se tiene en el sistema como pronósticos de construcción de telefonía pública para el mes que se está procesando; de igual forma se resta a la existencia inicial o del mes anterior, para presentar la nueva resultante para el mes que se presenta.
- *Cons\_PE* y *Exi\_Pe*: En caso de que se haya calculado una proyección de construcción por factor para planta externa, esta aparece en esta columna, según el mes que se está presentando; por ser un consumo de material, se resta de la existencia inicial o del mes anterior presentado, para obtener la columna nueva de existencia para el mes presentado.
- *Pedidos*: Muestra la cantidad que se tiene en pedidos para ese material en esa región, que existan con fecha de entrega en el mes que se presenta, esta cantidad es sumada al resultado que se lleva de existencias.
- *Exi\_Mtp*: Cuando se selecciona en los datos base presentar los consumos promedio de mantenimiento y stock, por cada mes se presenta este campo a

fin de mostrar el resultado de restar el stock y consumo mensual a la existencia inicial o la del mes anterior.

En la figura 3.5.1. se tiene una muestra de un reporte generado; en este se puede observar que se presentan solo algunos catálogos y un conjunto de catálogos. En este ejemplo solo se seleccionó que se mostraran los datos de existencia actual, consumo por mantenimiento, stock y pedidos.

El resultado muestra que conforme se aplica la fórmula de restar el stock y los consumos por mes, se va restando la existencia inicial, cuando esta existencia no alcanza a cubrir la necesidad, se presenta el resultado negativo, indicando que se necesitará un pedido o un traspaso a esa región en ese mes; si esto sucede, se considera que se cubrirá y la existencia inicial para el siguiente mes será cero.

Como se comentó, es un panorama global de los materiales seleccionados a un cierto tiempo en el futuro. Para tomar las decisiones de traspasar o comprar, es necesario hacer un análisis de cada material, para esto el sistema genera otros reportes.

### **Reportes de Traspasos y Pedidos**

Estos reportes resultan del anterior, cuando en el anterior se determina que es correcto lo que se desea, se generan estos. Son el equivalente a hacer el análisis del reporte anterior, determinando por mes si sobra material en una región y falta en otra.

Cuando se detecta un faltante, se buscan sobrantes en las regiones factibles, de existir sobrante se propone el traspaso y se genera una línea en el reporte de traspasos.

Cuando en una región se tiene un faltante en un mes, y no se alcanza a cubrir con los traspasos posibles, se genera una línea en el reporte de pedidos.



Compañía Telefónica S.A de C.V.  
Gerencia de Control de Inventarios

01/02/1999  
Archivo: c:\sis\data.txt

Reporte de Cables para 3 meses

CATALOG	DESCRIPCION	REGION	EXS_AC	CONS_MT	STOCK	FECHAS FEBRERO	EX.MITO FEBRERO	FECHAS MARZO	EX.MITO MARZO	FECHAS ABRIL	EX.MITO ABRIL	FECHAS MAYO	EX.MITO MAYO
2302866	CAB SUBT 20PS 0.4MM SCREB	CELAYA	7935	897	1794	0	5273	0	4376	0	3479	0	2682
2302866	CAB SUBT 20PS 0.4MM SCREB	CHIHUAH	3419	399	738	0	2324	0	1955	0	1566	0	1217
2302866	CAB SUBT 20PS 0.4MM SCREB	GUADAJA	17675	1439	2679	0	13405	0	11995	0	10526	0	9067
2302866	CAB SUBT 20PS 0.4MM SCREB	HERMOSI	8402	80	161	0	8163	0	8083	0	8002	0	7922
2302866	CAB SUBT 20PS 0.4MM SCREB	MAZATLA	3538	194	387	0	2661	0	2768	0	2574	0	2380
2302866	CAB SUBT 20PS 0.4MM SCREB	MERIDA	0	0	0	0	0	0	0	0	0	0	0
2302866	CAB SUBT 20PS 0.4MM SCREB	METRO	27209	9908	19817	0	-2186	0	-9908	0	-9908	0	-9908
2302866	CAB SUBT 20PS 0.4MM SCREB	MONTER	12149	2399	4777	0	5063	0	2674	0	266	0	-2103
2302866	CAB SUBT 20PS 0.4MM SCREB	PUEBLA	8666	1723	3445	0	3755	0	2033	0	310	0	-1413
2302866	CAB SUBT 20PS 0.4MM SCREB	VILLAHE	9195	1222	2443	0	5571	0	4350	0	3128	0	1907
Cables1	Cable de 20 pares SCREB 0.4	CELAYA	0	0	0	0	0	0	0	0	0	0	0
Cables1	Cable de 20 pares SCREB 0.4	CHIHUAH	0	0	0	0	0	0	0	0	0	0	0
Cables1	Cable de 20 pares SCREB 0.4	GUADAJA	0	0	0	0	0	0	0	0	0	0	0
Cables1	Cable de 20 pares SCREB 0.4	HERMOSI	75	0	0	0	75	0	75	0	75	0	75
Cables1	Cable de 20 pares SCREB 0.4	MAZATLA	1220	0	0	0	1220	0	1220	0	1220	0	1220
Cables1	Cable de 20 pares SCREB 0.4	MERIDA	0	0	0	0	0	0	0	0	0	0	0
Cables1	Cable de 20 pares SCREB 0.4	METRO	20569	107	214	0	20242	0	20135	0	20028	0	19921
Cables1	Cable de 20 pares SCREB 0.4	MONTER	0	0	0	0	0	0	0	0	0	0	0
Cables1	Cable de 20 pares SCREB 0.4	PUEBLA	3744	163	326	0	3261	0	3088	0	2936	0	2772
Cables1	Cable de 20 pares SCREB 0.4	VILLAHE	3445	108	215	0	3126	0	3018	0	2910	0	2803

Página 1

Figura 3.5.1. Ejemplo del Reporte Principal.

Al final lo que se obtiene son los reportes de traspasos que se deben de realizar y de pedidos a fincar; éstos son turnados a las áreas responsables de realizar los movimientos (almacenes y compras).

En la figura 3.5.2. se tiene un ejemplo del reporte de traspasos que resulta de analizar el reporte mostrado en la figura 3.5.1.

Compañía Telefónica, S.A. de C.V.  
Gerencia de Control de Inventarios  
Reporte de Traspasos

Catalogo : 2302856

DESTINO	ORIGEN	MES	CANTIDAD
Tomando Consumos	SIREM		
METRO	VILLAHERMOSA	FEBRERO	2186
METRO	VILLAHERMOSA	MARZO	2164
METRO	HERMOSILLO	MARZO	7744
METRO	HERMOSILLO	ABRIL	258
METRO	CHIHUAHUA	ABRIL	1586
METRO	MAZATLAN	ABRIL	2574
METRO	MONTERREY	ABRIL	285
METRO	GUADALAJARA	ABRIL	5205
VILLAHERMOSA	GUADALAJARA	ABRIL	1222
CHIHUAHUA	CELAYA	MAYO	369
HERMOSILLO	CELAYA	MAYO	80
MAZATLAN	GUADALAJARA	MAYO	194
METRO	GUADALAJARA	MAYO	2467
METRO	CELAYA	MAYO	2132

**Figura 3.5.2. Ejemplo del Reporte de Traspasos.**

En el se observan por material los movimientos que deben realizarse para cubrir faltantes en algunas regiones con sobrantes en otras, como por ejemplo:

En el reporte de la figura 3.5.1. se tiene el material 2302856 para la región Metro; en el mes de febrero se observa un valor negativo por -2186 (faltante); observando la primera línea del material en el reporte de la figura 3.5.2., se observa que se solicita el traspaso correspondiente para cubrir esta necesidad, ya que en la región de Villahermosa estaban sobrando 5571 metros.

Compañía Telefónica, S.A. de C.V.  
Gerencia de Control de Inventarios  
Reporte de Pedidos

Catalogo : 2302856

REGION	FEBRERO	MARZO	ABRIL	MAYO
Tomando Consumos SIREM				
CELAYA	0	0	0	0
CHIHUAHUA	0	0	0	0
GUADALAJARA	0	0	0	0
HERMOSILLO	0	0	0	0
MAZATLAN	0	0	0	0
METRO	0	0	0	5309
MONTERREY	0	0	0	2389
PUEBLA	0	0	0	1413
VILLAHERMOSA	0	0	0	1222

**Figura 3.5.3. Reporte de Pedidos.**

En la figura 3.5.3., se tiene un ejemplo del reporte de pedidos resultante del análisis del reporte en la figura 3.5.1. En el se observa que para el mismo material, una vez que se realicen todos los traspasos posibles, existirá un faltante en el mes de mayo que no podrá ser cubierto con ningún traspaso, por lo que se necesitará solicitar compra, por lo que se listan las cantidades, que se requieren y para cuando se requieren.

**Reportes planos de Traspasos y Pedidos.**

Con el objeto de poder proporcionar la información anterior de acciones a tomar, en una forma que facilite su manejo y distribución, se proporcionan paralelamente los archivos planos con esta información, con un formato que puede ser importado como texto a cualquier hoja de cálculo o base de datos.

En la figura 3.5.4. se muestra el ejemplo del reporte plano de traspasos correspondiente al análisis de la figura 3.5.1.

CATALOGO	DESTINO	ORIGEN	MES	CANTIDAD	FECHA_REF
2302856	METRO	VILLAHERMOSA	FEBRERO	2186	01/02/1999
2302856	METRO	VILLAHERMOSA	MARZO	2164	01/02/1999
2302856	METRO	HERMOSILLO	MARZO	7744	01/02/1999
2302856	METRO	HERMOSILLO	ABRIL	258	01/02/1999
2302856	METRO	CHIHUAHUA	ABRIL	1586	01/02/1999
2302856	METRO	MAZATLAN	ABRIL	2574	01/02/1999
2302856	METRO	MONTERREY	ABRIL	285	01/02/1999
2302856	METRO	GUADALAJARA	ABRIL	5205	01/02/1999
2302856	VILLAHERMOSA	GUADALAJARA	ABRIL	1222	01/02/1999
2302856	CHIHUAHUA	CELAYA	MAYO	369	01/02/1999
2302856	HERMOSILLO	CELAYA	MAYO	80	01/02/1999
2302856	MAZATLAN	GUADALAJARA	MAYO	194	01/02/1999
2302856	METRO	GUADALAJARA	MAYO	2467	01/02/1999
2302856	METRO	CELAYA	MAYO	2132	01/02/1999

**Figura 3.5.4. Ejemplo del Reporte Plano de Traspasos.**

De la misma forma, en la figura 3.5.5. se muestra un ejemplo del reporte plano de pedidos correspondiente al mismo análisis.

CATALOGO	DESTINO	MES	CANTIDAD	FECHA_REF
2302856	METRO	MAYO	5309	01/02/1999
2302856	MONTERREY	MAYO	2389	01/02/1999
2302856	PUEBLA	MAYO	1413	01/02/1999
2302856	VILLAHERMOSA	MAYO	1222	01/02/1999

**Figura 3.5.5. Ejemplo del Reporte Plano de Pedidos.**

## Reporte de Avance de Pedidos

El reporte de avance de pedidos, es un reporte que refleja los pedidos, que a la fecha de referencia no han sido saldados en su totalidad; al generarse se puede indicar un porcentaje de tolerancia, cuando un pedido tiene un saldo menor a ese porcentaje no se refleja en este reporte.

Las columnas que se incorporan son las siguientes:

- **NUM ORD:** Número de orden de compra o pedido; es la clave del pedido que está atrasado o tiene un saldo por entregar menor al indicado.
- **ALM:** Clave del almacén distribuidor donde debe ser entregado el material que se encuentra con retraso en sus entregas.
- **CATALOG:** Es el número de catálogo (clave de material) que está pendiente de entregarse en el almacén indicado.
- **DESCRIPCIÓN:** Es la descripción del material que se está reportando, tomada del catálogo de materiales.
- **PROVEEDOR:** Es la clave del proveedor que tiene asignado el pedido, se presenta como referencia para poder ubicar el pedido y poder hacer gestiones para que se haga la entrega lo antes posible.
- **TOT PRO:** Es la cantidad total del pedido que debió ser entregada al almacén indicado y que se encuentra con saldo.
- **% A ENT:** Refleja el porcentaje que debió entregarse, este puede variar del 100% cuando en el pedido se tiene una cantidad por entregar diferente a la original del pedido. Generalmente será el 100%.

- **%REAL:** Es el porcentaje del pedido que ya ha sido entregado; es el campo de referencia para determinar si se presenta o no el renglón, esta columna se compara contra el porcentaje que se desea ver, si la diferencia al 100% es mayor a lo indicado en el reporte, se presenta el renglón.
- **EXI DIS:** Es la existencia actual que se tiene en el almacén distribuidor. Esta columna se incorpora para poder determinar el efecto que tiene el retraso en la entrega, en la región correspondiente, ya que si la cantidad por entregar es muy pequeña con respecto a este campo de existencia, el retraso carece de importancia real; en caso contrario, se sabe que la mayor parte del material con el que se puede disponer, está retrasado en sus entregas.

En el encabezado del reporte se indican los datos generales de la empresa y el departamento que lo está generando, además de que se indican los parámetros con los que se generó:

- **Periodo de análisis:** que es el rango de fechas en las que se desea saber si hay entregas pendientes
- **Fecha de referencia:** Es la fecha que se indica al generar el reporte, y contra la que se compara la fecha de entrega y se determina si el pedido está atrasado.
- **Porcentaje:** Que es el porcentaje de entrega que se seleccionó para generar el reporte.

En la figura 3.5.6. se muestra un ejemplo de los pedidos que tiene saldo vencido por entregar del material que se analizó en la figura 3.5.1.

Compañía Telefónica, S.A. de C.V.  
Gerencia de Control de Inventarios

Fecha referencia: 01/02/1999

Del 01/08/1998 al 01/02/1999 Con diferencia de porcentajes > a 20 %

Reporte de Avance de Pedidos

NUM ORD	ALMI	CATALOG	DESCRIPCION	PROVEEDOR	TOT PRO	% A ENT	% REAL	EXI DIS
5000780	08	2302856	CAB SUBT 20PS 0.4MM SCREB	18215	145485	100	48	12588
5000794	13	2302856	CAB SUBT 20PS 0.4MM SCREB	18215	1830	100	0	31
5003572	11	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	6443	100	0	927
5003572	32	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	23821	100	42	2763
5003572	43	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	16477	100	78	607
5003572	44	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	3669	100	0	5101
5003572	Y0	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	6405	100	2	7161
5007160	Y0	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	3660	100	76	7161
5010816	08	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	60390	100	8	12588
5010816	13	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	10065	100	0	31
5010816	23	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	4575	100	0	883
5010816	43	2302856	CAB SUBT 20PS 0.4MM SCREB	17950	16470	100	0	607
5010816	43	2302874	CAB SUBT 50PS 0.4MM SCREB	17950	13725	100	62	7515
5010818	Y2	2302874	CAB SUBT 50PS 0.4MM SCREB	17950	915	100	0	1075
5010818	Y3	2302874	CAB SUBT 50PS 0.4MM SCREB	17950	1830	100	0	237
5010818	Y7	2302874	CAB SUBT 50PS 0.4MM SCREB	17950	915	100	0	810

Página: 1

Figura 3.5.6. Ejemplo de Reporte de Avance de Pedidos.

Dado que el reporte de avance de pedidos se genera para todos los materiales que existen en el catálogo, no existe relación entre este y el Reporte Principal, la única relación es que se genere con las mismas fechas de existencias y pedidos, y se revisen solo los materiales que se presentaron en dicho reporte.

### **Reporte de Entregas a Usuario**

El reporte de entregas a usuario, aprovecha la información que se tiene cargada en el sistema, para poder reflejar todo el material que se le a entregado a un empleado en un periodo determinado.

Entre el material que se maneja en los almacenes, se encuentra toda la herramienta que usan los trabajadores para realizar el mantenimiento de la planta telefónica; a cada uno de estos empleados se le entrega una dotación de herramienta la cual es importante controlar y conocer si alguno está consumiendo en exceso.

Existen materiales que son especiales y de un costo significativo, por lo que se requiere saber cuantos se le entregan a un empleado y se puede comparar contra las órdenes de servicio que haya atendido.

Para estos análisis, este reporte permite observar todo el material que se entregó a determinados empleados, a través de la clave del mismo que se tiene en cada uno de los movimientos cargados desde el SIREM.

Las columnas que se presentan son las siguientes:

- **CATÁLOGO:** Es la clave del material que se está reportando y que fue entregado a un empleado.



- **EXPEDIENTE:** Es la clave del empleado que se desea analizar y al cual fue entregado el material indicado en la columna "CATÁLOGO".
- **CANTIDAD:** Es la cantidad de unidades que fueron entregadas al empleado indicado, en el periodo solicitado.

En el encabezado, se muestra la fecha con que se generó, los datos de la empresa y el departamento que genera el reporte. Un ejemplo de este reporte se observa en la figura 3.5.7.

En esta figura 3.5.7., se seleccionaron los expedientes 1 y 1000 y se solicitó que mostrara todo los materiales que les habían sido entregados. Se observa que el ordenamiento es por catálogo y luego por expediente.

Compañía Telefónica, S.A. de C.V.19/08/1999  
Gerencia de Control de Inventarios

Reporte de Entregas a Usuarios

CATALOGO	EXPEDIENTE	CANTIDAD
2302883	1000	11197
2302874	1000	4560
2302892	1	2909
2302847	1000	2341
2302892	1000	1876
2302856	1000	1706
2302883	1	780
2302856	1	745
2302838	1000	590
2302865	1000	485
2302865	1	294
2302874	1	263
2302810	1000	60
2302838	1	55
2302810	1	50
2302801	1	48

Página: 1

Figura 3.5.7. Ejemplo del reporte de Entregas a Usuario.

### **3.6. Factibilidad Técnica y Operativa**

Dentro de los elementos viables que tenemos en este sistema, es sin duda la centralización de la información, es decir la definición y creación de una estructura de datos organizada, y normalizada, esto lo realizamos con la implementación de una base de datos relacional, de donde aplicamos las normas y procedimientos que están definidos para la realización de este tipo de bases de datos. El administrador de bases de datos utilizado es Acces 97, con este tenemos la posibilidad de la migración de la base de datos a otros administradores, tales como, Oracle, Foxpro, Clipper, etc.

La centralización de la información nos permite de alguna manera tener mejor control de la misma, es decir, podemos contar con procedimientos de validación, de acceso, y políticas de respaldos, otra de las ventajas que podemos tener es la importancia de cuidar la integridad, y evitar la duplicidad de la información, y con ello la disminución de recursos de cómputo y sin duda tiempo de proceso, para la obtención de resultados.

Utilizando las herramientas de Access podemos tener un mejor manejo y explotación de la información, tal como: consultas, reportes, estadísticas, etc., contar con esta flexibilidad, es sin duda un factor muy importante, ya que uno de los problemas con que contaba el área es en la generación de reportes de interés a ciertas entidades, y esto implicaba mucho tiempo, por falta de una sistema centralizado.

Con la concentración de la información e integración en un sistema, esto facilita las condiciones para que se integre a la red de datos, y por ende, todas las áreas involucradas en la compra de materiales, para que, en un momento, dado tener la facilidad de accezar a la información vía la red, y de esta manera realizar consultas en línea de la misma, que es una de las necesidades que se tenían en el área.

Con la centralización de los diferentes procesos en un solo equipo, la integración de este a la red de datos y la implementación del sistema, se puede prescindir del uso de varios elementos que se involucraban en el proceso anterior, tales como: Equipo de cómputo, Impresoras, Software, Personal especializado, uso de diferentes formatos, etc., y esto para el área representa una gran ayuda, ya que con esto se pueden tener mejores tiempos de respuesta, información confiable y oportuna para las áreas involucradas (Compras, Administrativas, Usuarios, etc.).

El equipo de cómputo necesario y suficiente para el funcionamiento del sistema es: una PC con procesador Pentium a 300MHz, con 64 MB en RAM, 8 GB en disco duro, Tarjeta de Red a 10 MGB Ethernet, con Windows 9X, Access 97, Visual Basic Ver. 5.0, con monitor SVGA de 14 pulgadas, con 4 MB de memoria de vídeo, una impresora Láser con 8 MB en RAM, para la impresión de reportes, este equipo debe estar configurado para acceso a la red, haciendo uso del protocolo FTP para la transferencia de archivos.

Las características antes mencionadas son las mínimas recomendables para que el sistema funcione y cumpla con su objetivo; dado que el sistema está en un ambiente totalmente gráfico, se tiene la necesidad de contar con un procesador, capaz de ejecutar este ambiente de manera rápida y aceptable para el usuario, es por ello que se recomienda como mínimo un procesador Pentium con suficiente memoria RAM (como mínimo 64 MB), y con ello tener tiempos de respuesta aceptables.

Una de las características con las que debe cumplir este equipo es sin duda la capacidad en disco duro, debido a la gran cantidad de información que se maneja en el sistema (100 MB por mes aproximadamente), es por ello que se requiere de un área de almacenamiento tanto para la base de datos activa, como para los respaldos de la misma y por ende de información ya procesada como son reportes, memos, etc.

Debido al flujo de insumos y la integración del sistema a la red de datos de la empresa, es un requisito indispensable que el equipo cuente con este acceso, para ello se requiere de una tarjeta de red, con la características antes mencionadas, ya que el sistema tiene procesos que obtienen información (archivos de movimientos, existencias y pedidos de los almacenes) del sistema SIREM y esto lo realizan vía la red haciendo uso del protocolo de comunicaciones FTP. Debido a la necesidad de contar con información confiable y oportuna, el sistema cuenta con la flexibilidad de que la información, pueda ser consultada desde cualquier área de la empresa, esto desde luego debe hacerse vía la red.

Dado que para el desarrollo e implementación del sistema se utilizó el software de desarrollo Visual Basic ver. 5.0 y Access 97 para ambiente Windows 9X, esto implica desde luego la utilización de Windows 9X.

En muchas ocasiones a la Gerencia de Control de Inventarios, le son solicitados reportes de ciertas características, los cuales en la mayoría de los casos son presentados a personal de la empresa de niveles como: Directores, Subdirectores y Gerentes de Área, y por ende, la presentación de los mismos debe ser con cierta calidad y con formatos adecuados, de donde se tiene la necesidad de contar con una impresora que nos permita obtener estos reportes, para ello recomendamos el uso de una impresora láser con suficiente capacidad de memoria (8 MB), para una buena impresión.

Las formas de operación del sistema son muy amigables ya que la plataforma en la que está implementado, es totalmente gráfica es decir son ambientes Windows 9X y Visual Basic, con esta facilidad podemos prescindir de personal especializado para la decisión de compra de ciertos materiales.

Al integrar el sistema a la red de datos, la obtención de insumos tales como los archivos de pedidos, movimientos, y existencias que el sistema SIREM provee, el sistema lo realiza de manera automática y no se requiere contar con una persona

especializada para la obtención de esta información.

Al integrar los procesos en un sistema la operación se facilita, ya que no se requiere que lo ejecute una persona especializada, ya que el ambiente del sistema va guiando al usuario y que una persona con conocimientos básicos de manejo de ambiente Windows puede operar el sistema sin más complicaciones; con esta facilidad podemos prescindir del manejo de diferentes procesos y herramientas de software (Excel, Word, DbaseXX).

El sistema contempla un módulo capaz de generar reportes, donde el usuario define la información que en un momento dado sea de su interés, con este módulo podemos solventar algunas de las diferentes peticiones de reportes solicitados por entidades involucradas en la compra y control de materiales.

Dentro de las facilidades que tenemos con el sistema, es sin duda el tiempo de respuesta, ya que los procesos y procedimientos anteriores tenían tiempos de respuesta de tres a cuatro días y con este sistema el tiempo de respuesta es de 6 a 8 horas como máximo, y esto con el uso de menos recursos tanto materiales como humanos.

En este nuevo sistema, como ya existe una base de datos, esto implica que debe haber una persona responsable, con los conocimientos necesarios y suficientes para realizar las funciones de administración de la misma y con ello garantizar la integridad y seguridad de la información, los conocimientos que debe tener esta persona son: saber usar las herramientas de Access 97, conocimientos de Windows 9X, redes de datos, y uso de Visual Basic.

El recurso humano que requieren para la operación del sistema, es una persona que cuente con los conocimientos de que es una PC y conocimientos básicos de sistemas operativos Windows 9X. Dado que el sistema es totalmente gráfico, el tiempo estimado de capacitación para la operación adecuada del sistema, es de 1

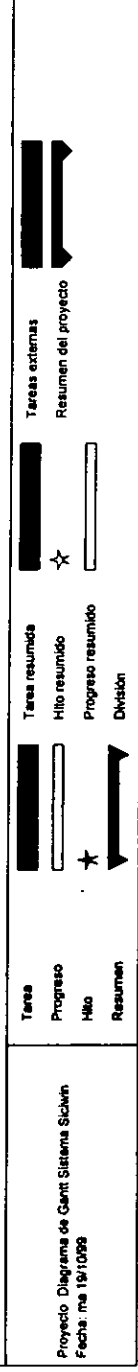
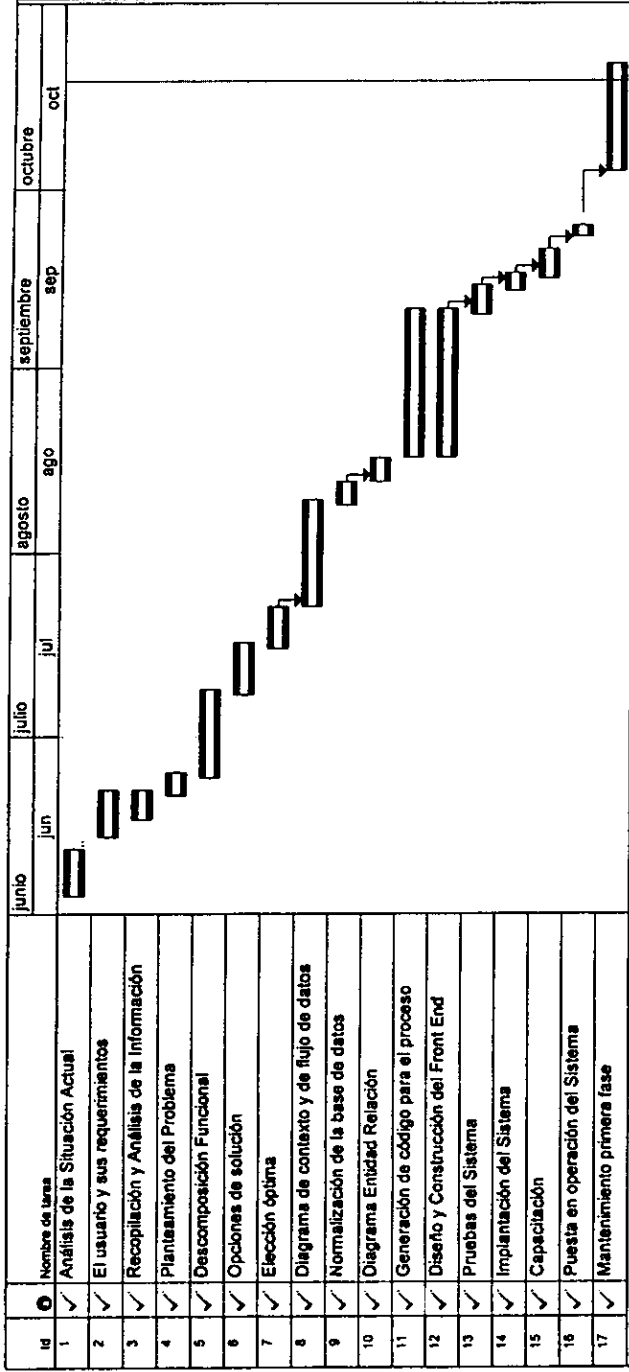
día, ya que el sistema es muy amigable y por ello la facilidad de operación del mismo.

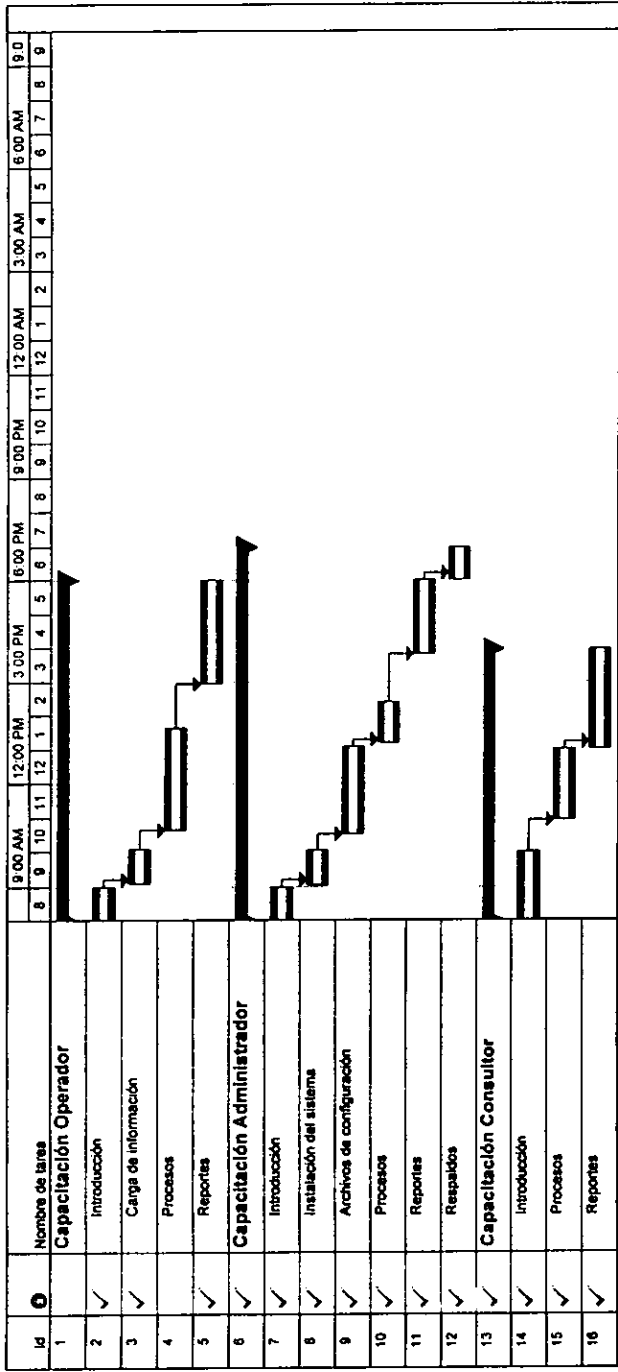
El recurso humano que se requiere para la administración del sistema, debe ser una persona que conozca de redes de computadoras en ambiente Windows 9X protocolos de comunicaciones (TCP/IP), diseño de bases de datos relacionales e implementación con Access 97, conocimientos y uso de las herramientas de Access 97, conocimientos de Visual Basic ver. 5.0. y conocimientos de procedimientos de respaldo de información. El tiempo estimado en la capacitación en la administración del sistema es de 2 días, por la gran cantidad de información que el sistema maneja y sin duda de la importancia de la misma.

Para el desarrollo e implementación del sistema se elaboró un plan de trabajo como se muestra en el siguiente diagrama de Gantt, en este diagrama se encuentran definidas todas las tareas que se realizaron tanto en tiempo y orden. Para el caso de la tarea de capacitación se desarrollo un plan en unidades de tiempo en horas por usuarios, donde se definen los temas y tiempo de exposición para cada uno. En este diagrama se puede observar que el tiempo de capacitación tanto para uso y administración es relativamente corto, ya que estamos hablando de un periodo de 6 a 8 horas por usuario aproximadamente. Y esto sin duda es una flexibilidad en la operación del mismo. Los usuarios que se pueden tener definidos en el sistema son tres como se indica en el diagrama de donde cada uno tiene alguna finalidad con el uso del sistema como se indica a continuación:

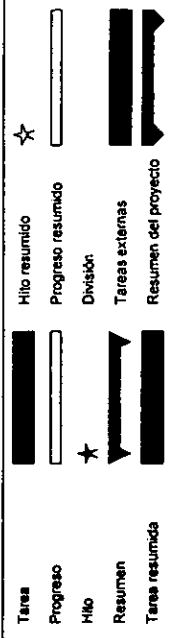
- ◆ **Operador.** Usuario que utiliza el sistema para obtención y generación de reportes de compra y traspaso de materiales.
- ◆ **Consultor.** Usuario de consulta de información del sistema y petición de reportes.
- ◆ **Administrador.** Usuario responsable de creación de usuarios, configuración de módulo de transferencia de archivos, generación de respaldos, mantenimiento a la base de datos, etc.

Capítulo 3: Desarrollo e Implementación del Sistema.





Proyecto: Diagrama de Gantt Capacitación Sistema ScitHn  
 Fecha: ma 19/10/99





En la actualidad el tener definido normas y procedimientos en las empresas, es un factor importante para el control y buen funcionamiento de las mismas y por ende tener un desarrollo ordenado y de alguna manera controlado, esto no queda fuera del alcance de la empresa, ya que, en la empresa existen normas y procedimientos para el desarrollo e implementación de nuevos sistemas.

En todas y cada una de las tareas que se definieron para el desarrollo e implementación del sistema, se cuidó que todas se realizaran en cuanto norma y procedimientos definidos por la empresa y de esta manera poder garantizar que el sistema se desarrollo y se implantó con apego a las normas existentes, de las cuales por mencionar algunas tenemos:

- ◆ El sistema cuenta con una pantalla de inicio en la cual el usuario tiene que teclear la clave y la contraseña que se le asigne, y de esta manera tener control de quien esta usando el sistema.
- ◆ El sistema cuenta en todas sus pantallas con mensajes de descripción de que función realizan todos y cada uno de sus elementos, y de esta manera el usuario cuente con información del funcionamiento del sistema.
- ◆ El sistema cuenta con sus módulos de respaldo como lo define la norma y se apega a las políticas de la misma.
- ◆ Todas y cada una de las pantallas fueron diseñadas con las peticiones del usuario y con apego a los estándares definidos por la empresa.
- ◆ Las herramientas de desarrollo que se utilizaron, son las que la empresa tiene homologadas en sus normas de desarrollo.
- ◆ Todos los nombres de archivos esta creados bajo los estándares definidos por la empresa.
- ◆ En el sistema existe un proceso de transferencia de archivos, esto implica que el proceso fue desarrollado conforme a la norma de acceso a red datos, es decir en el sistema se deben definir datos válidos tales como: usuarios, claves, dirección, etc.

# **Control de Inventarios para los almacenes de una empresa telefónica**

## **Manual de Usuario**

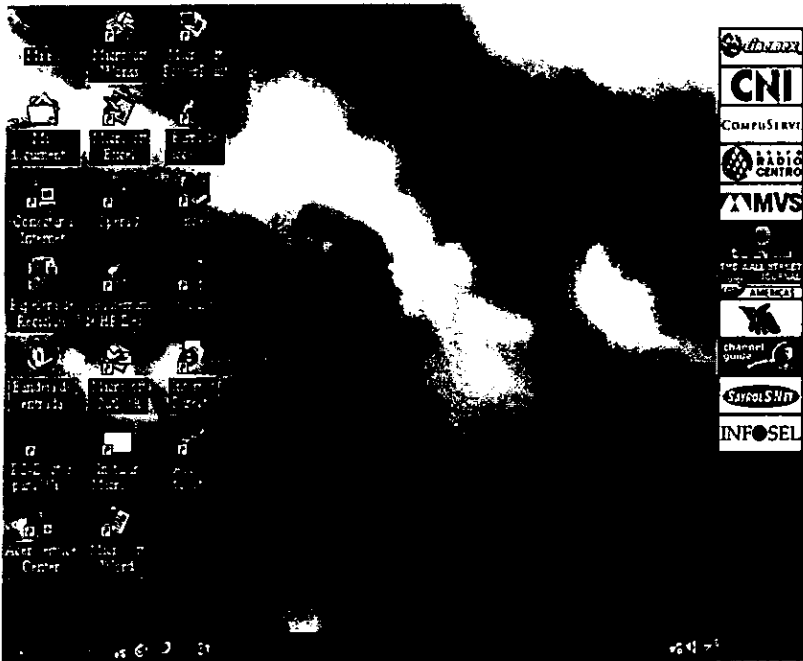
### **Introducción**

Este manual tiene como propósito introducir a los usuarios a la operación y uso del Sistema de Control de Inventarios para una empresa telefónica (SiCI). Los contenidos de este manual explican como iniciar el sistema, la operación de cada uno de los módulos del sistema, los productos generados, las interfaces con otros sistemas y los requerimientos.

El Sistema de Control de Inventarios (SiCI) es una herramienta de software que permite primordialmente obtener reportes de carácter variable acerca de las existencias futuras, así como de las existencias actuales, pedidos y consumos de las diferentes áreas de consumo de la compañía telefónica.

Adicionalmente a los reportes dinámicos que genera el sistema, también se pueden obtener planes de trasposos y pedidos, reporte de entregas a usuarios y reporte de avance de pedidos.

## Inicio del sistema



**Pantalla 1. Ubicación de acceso directo al sistema SiCI (Sistema de Control de Inventarios ).**

La pantalla 1 muestra la pantalla principal del sistema Windows 95. En la parte inferior se encuentra el icono de la aplicación SiCI. Para iniciar la operación posicione el ratón de la computadora sobre ese icono. A continuación presione dos veces el botón derecho del ratón. La aplicación se iniciará.

Al iniciar el sistema presenta la pantalla 2. Esta es la pantalla de presentación del sistema. Dura algunos segundos activada y después se procede a presentar la pantalla de acceso. En esta pantalla (pantalla 3) se debe de proporcionar clave y password de acceso al sistema.



**Pantalla 2. Al arrancar la aplicación se muestra primeramente la pantalla de presentación.**

### **Módulos del sistema**

El sistema esta compuesto de 18 módulos que realizan diversas funciones. Los módulos están organizados en 5 grupos de subsistemas. Estos subsistemas son:

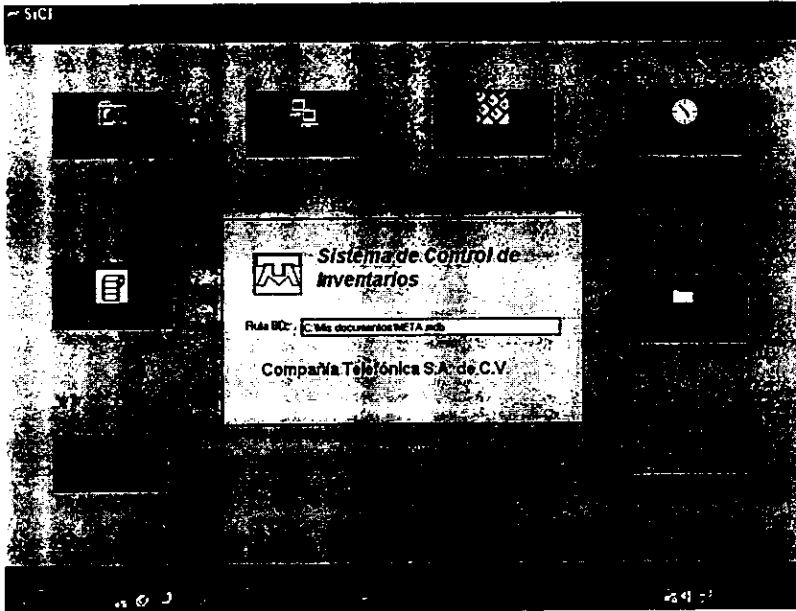
- Carga de información.
- Datos para cálculos de consumos.
- Cálculo de consumos.
- Reportes.
- Tablas.



**Pantalla 3. Se provee una pantalla para acceder al sistema.**

La pantalla 4 muestra estos grupos de módulos en la vista principal del sistema a modo de un menú de opciones en la parte superior de la pantalla. A continuación se presentan a modo de botones las opciones (módulos) más requeridos durante la

operación cotidiana del sistema.



**Pantalla 4.** Aquí se presenta la pantalla principal del sistema que incluye el menú de acceso a los programas.

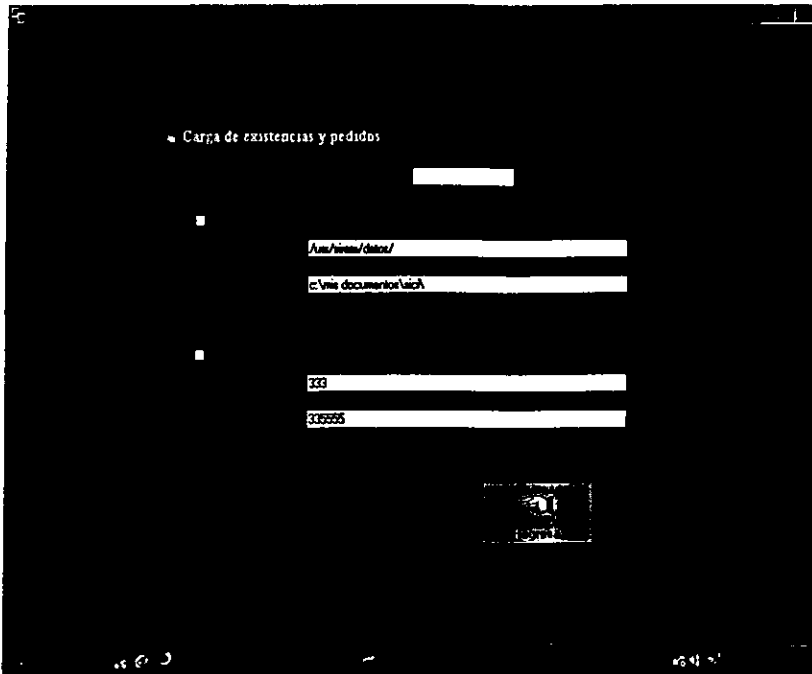
### **Carga de información**

Este conjunto de programas permite cargar información de otros sistemas como pedidos, existencias, movimientos y pronósticos. Estos datos son accedidos cotidianamente de computadoras remotas por lo que se hace uso del protocolo FTP para accederlos.

### **Carga de existencias y pedidos**

Las existencias y pedidos normalmente están contenidos en otras máquinas remotas por lo que primero es necesario traerlas y después cargarlas y procesarlas en el sistema de aquí la necesidad de este módulo. La pantalla muestra la carga de

existencias y pedidos.



**Pantalla 5. Módulo de carga de existencias y pedidos.**

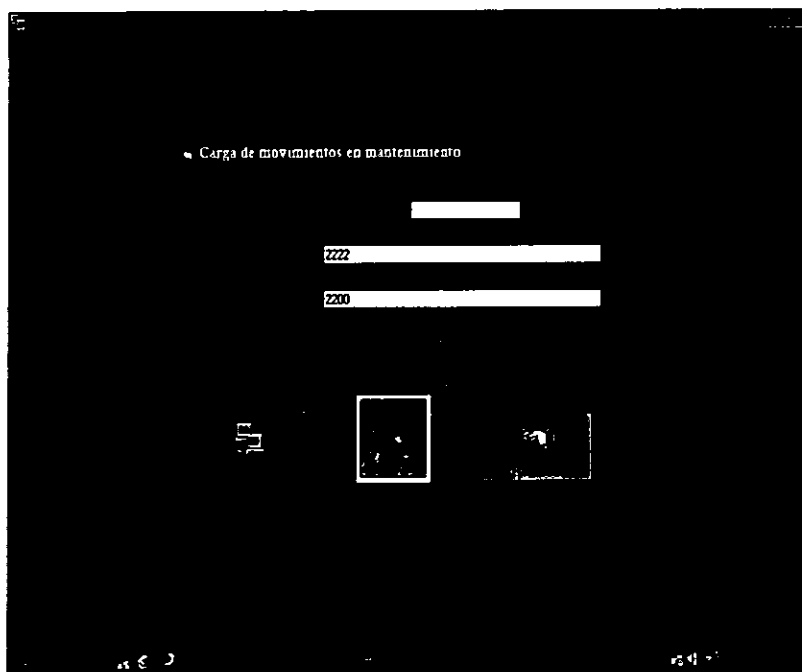
Para traer archivos de computadoras remotas primeramente proporcione la ubicación completa en esa computadora en ruta origen. Después proporcione la ruta y nombre del archivo donde quiere colocar los archivos en la computadora local. Al terminar para indicar que la transferencia es para existencias, pedidos o ambas marque en las cajitas de chequeo para existencias y pedidos.

Al terminar con lo anterior presione el botón **Trae Via FTP**. El sistema inicia la transferencia de archivos. Aparecen 1 o 2 pantallas con la operación de la aplicación

FTP. Al terminar estas pantallas desaparecen.

### Carga de movimientos de SIREM

Este módulo realiza la transferencia de información de máquinas remotas y la carga a la base de datos del sistema.



**Pantalla 6. Módulo de carga de movimientos en mantenimiento. Proviene de SIREM.**

La transferencia de información se realiza vía protocolo FTP. Primero se indica donde esta localizado el archivo que se va a traer con ruta completa en el espacio de archivo origen.



Después debe indicar donde se desea poner el archivo en la máquina local. Esto indicarlo en archivo destino. Al hacer esto iniciar el proceso presionando el botón **Traer Vía FTP**. El sistema genera una aplicación FTP que indica los pasos de la transferencia. Al terminar el proceso esta aplicación desaparece.

Una vez que se tiene el archivo localmente se indica su localización en el espacio de archivo de destino. Especifique la fecha de introducción al sistema. Al terminar esto se arranca el proceso de carga a base de datos interna presionando el botón **Carga a BD**.

Debido a que esto puede ser un proceso largo se incluye una barra gráfica que indica el progreso de la operación. Al terminar el proceso el sistema pregunta por confirmación de inclusión de datos.

### **Carga de pronósticos**

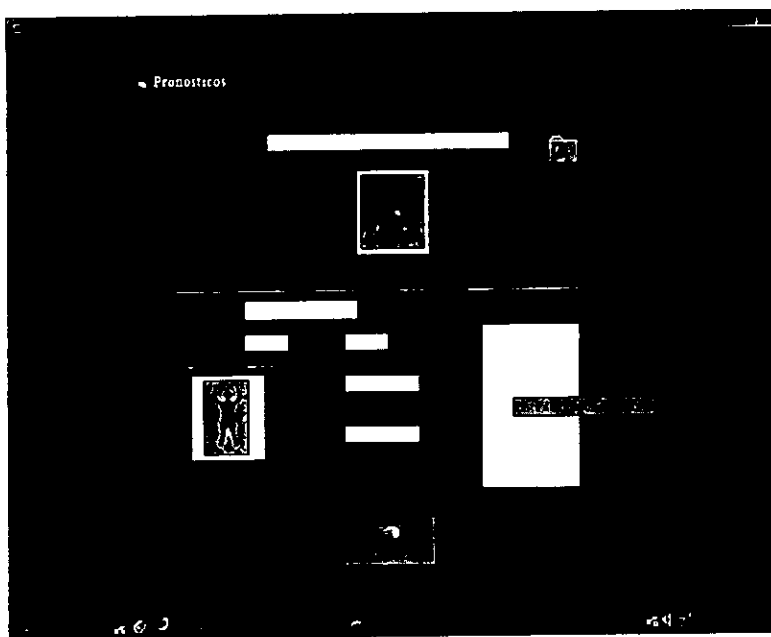
Los pronósticos es información proporcionada por diferentes almacenes con formatos de información variable. Los archivos son generados en dBASE pero contienen campos variables de información. Para resolver esto el presente módulo hace la inserción de los datos de pronósticos insertando la información de los campos únicamente requeridos.

El usuario pone en nombre de archivo la ruta completa del archivo dBASE a cargar. Hecho esto se presiona el botón **carga campos**. El módulo lee la configuración interna de ese archivo y presenta en **Campos dBase** los campos disponibles.

El usuario debe seleccionar los campos pertinentes para catálogo y pronóstico e indicarlos mediante la selección del campo y presionando el botón << para asociarlo a catálogo o a pronóstico.

Al terminar seleccionar en "región" a que región desea se asocien esos pronósticos. En mes y año se indica en que meses del año se deben guardar esos valores.

Al estar listo se presiona el botón **carga archivo** y se procede a cargarlo. Al terminar el sistema pide confirmación de carga.



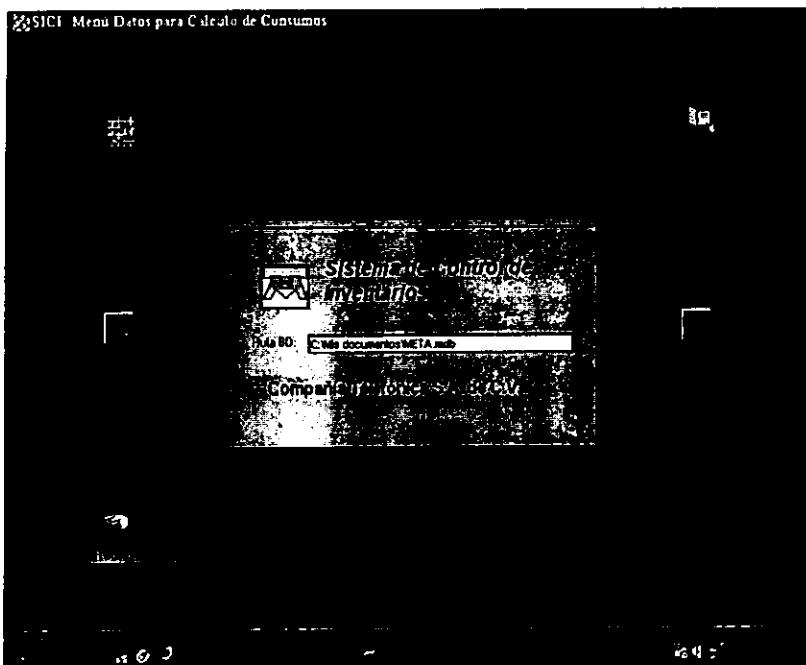
**Pantalla 7. Módulo de carga de pronósticos.**

### **Configuración de conexión a FTP y archivos**

Este módulo es de soporte para indicar la dirección y datos especiales de las computadoras remotas de donde se trae la información vía FTP. Existen tres registros con información para existencias, movimientos y pedidos.

En cada una de ellas se debe indicar la dirección IP, el puerto y las claves de acceso y pasaporte para accesarlas. Para localizar cada uno de los tres registros avance al

próximo registro o anterior con los botones localizados en la parte inferior derecha de la pantalla.

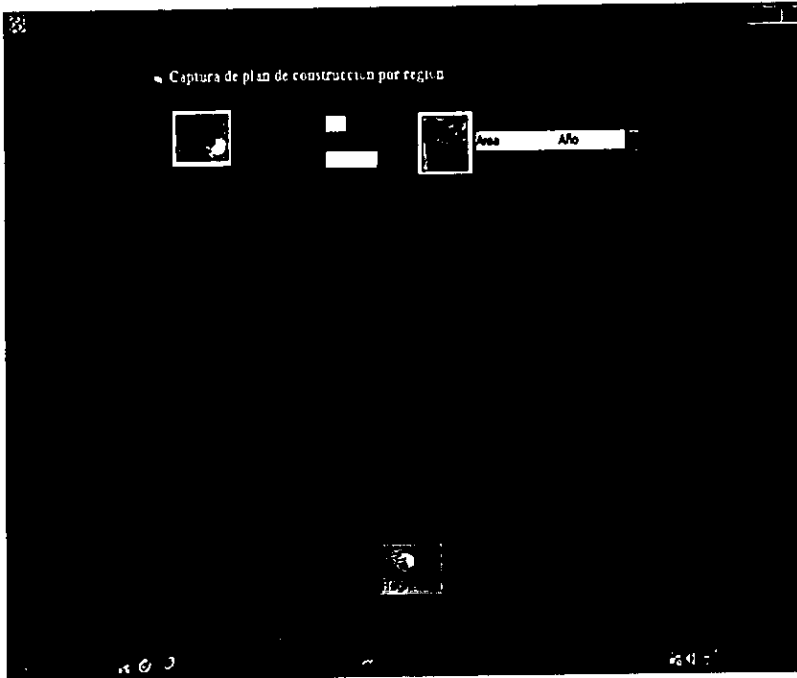


**Pantalla 8. Primera pantalla del módulo de captura del plan de construcción anual por región de la planta telefónica.**

Se muestra en forma calendaria el plan de construcción, el área de construcción de telefonía pública. Para cada mes del año de cada una de las regiones registradas se permite introducir un estimado de líneas telefónicas a instalar.

Para introducir o cambiar un dato solo posicione en la celda deseada con el ratón y presione el botón derecho del ratón. En la parte inferior izquierda debajo de la matriz existe el espacio para cantidad en el que puede cambiar el dato de la celda indicada. Así se puede hacer para cada cambio que se desee.

Al terminar si se considera que los cambios realizados son válidos presione el botón **actualiza**. Si no se desea aceptar los cambios realizados presione el botón **cancela** y la operación no se registrará.



**Pantalla 9. Segunda pantalla del módulo de captura del plan de construcción anual por región. Aquí se presenta toda la información pertinente del área de la compañía telefónica.**

### **Captura de factores de catálogos para construcción de líneas**

Este módulo permite registrar todo el material necesario para la construcción de una línea telefónica. El material es dado en forma de catálogo y se especifica cuanto material es necesario.

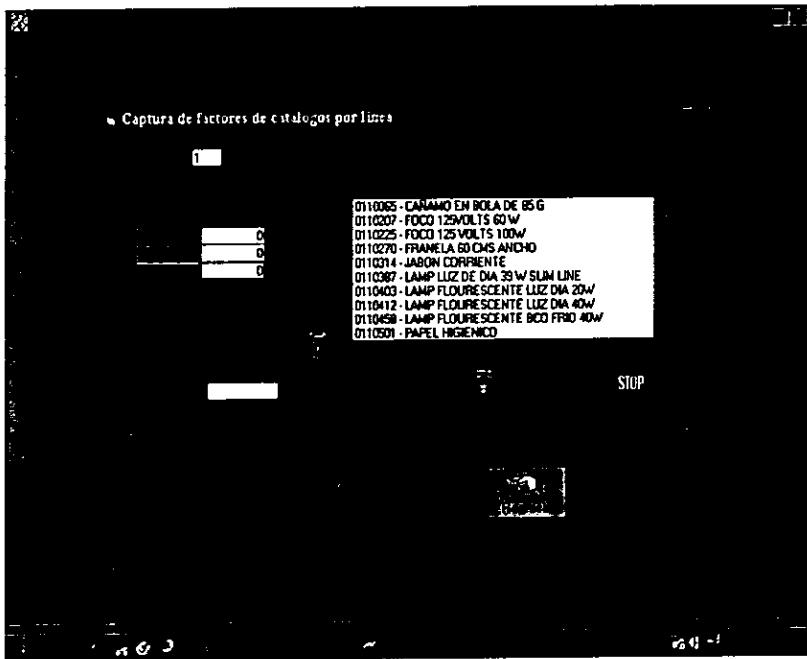
El módulo fue pensado para que se puedan especificar materiales de construcción de

líneas de diferentes áreas de construcción telefónica. El área TM = 1 corresponde al área de construcción en telefonía pública.

Al indicar el área se presiona alta o busca dependiendo a si existe o no el área.

Si el área es aceptada se procede a presentar la otra pantalla que contiene todos los catálogos registrados junto con las cantidades necesarias de cada uno.

Entramos en la pantalla del módulo de captura de factores de catálogos para construcción de líneas.



**Pantalla 10. Primera pantalla del módulo de captura de factores de catálogos para construcción de líneas.**

En la pantalla 10 se indican los catálogos necesarios para la construcción de líneas mediante la selección de catálogos localizados en la lista a la derecha. Una vez que se

encuentra catálogo a incluir se marca con el botón derecho del ratón y se agrega a la lista de materiales necesarios presionando el botón <<. Con esto si no estaba ya incluido se agrega el catálogo.

Una vez hecho lo anterior se marca con el botón derecho del ratón la celda de factor que se quiera cambiar para indicar cuanto de ese catálogo es necesario. En la parte de abajo se hacen cambios que aparecen inmediatamente en la celda de factor elegido.

Este procedimiento se repite con todos los catálogos que se quieran incluir o editar.

Para eliminar un catálogo se presiona la celda a borrar con el botón derecho del ratón y se marca al bote de basura. El sistema pide confirmación para borrar.

Al terminar con las actualizaciones si son validas se presiona el botón **actualiza** con lo que todo se salva.

Si no se desean salvar todos los cambios se presiona el botón **cancela** con lo que no se salvan los cambios realizados.

### **Captura de ADM de mantenimiento**

Los ADM son necesarios para filtrar el cálculo de consumos SIREM de tal forma que solo se consideren consumos de mantenimiento. Por esta razón es que en esta pantalla se registran todos aquellos ADM que estan en el área de mantenimiento. Después al momento de calcular el consumo SIREM solo se consideran los consumos que tengan ADM en esta sección.

La pantalla 11 presenta la captura de ADM de mantenimiento. Sin embargo para ver y editar los ADM es necesario presionar el botón **editar**. Al hacer esto se habilitan los ADM listos a editarse o borrarse.

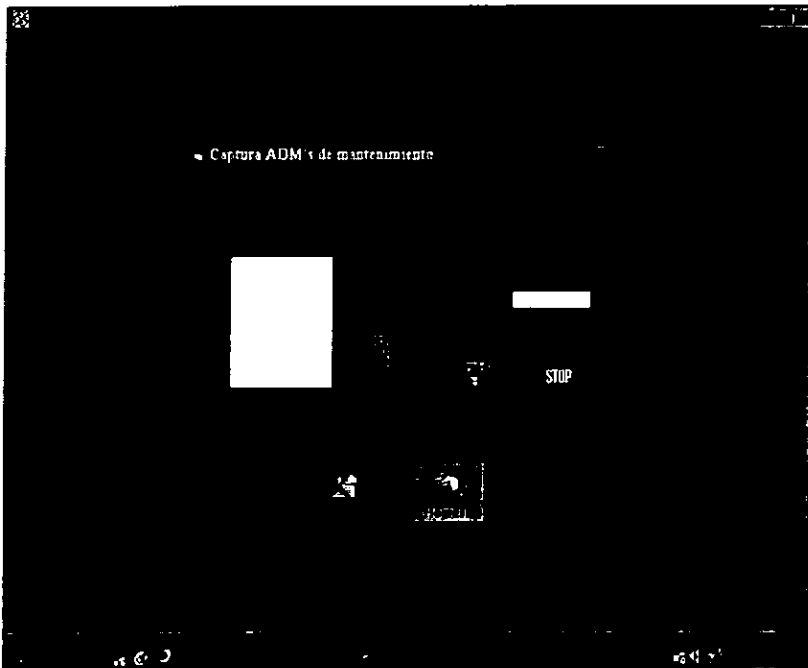
Podemos incluir los ADM que deseemos. Basta con teclear el ADM en el área a la derecha de la pantalla. Una vez hecho esto presionamos el botón << con lo que se agrega a la lista de ADM considerados de mantenimiento.

Para borrar un ADM de la lista basta con seleccionarlo con el botón derecho del ratón en la lista de la izquierda y presionar el dibujo de bote de basura con el botón derecho del ratón. El sistema pide confirmación antes de borrarlo.

Para aceptar los cambios realizados se presiona el botón **actualiza**.

Para terminar de editar sin salvar los cambios realizados presionar el botón **cancela**.

### Cálculo de consumos



**Pantalla 11. Primera pantalla de captura de ADM de mantenimiento.**





Si la información se encuentra disponible se presenta la pantalla 12. Aquí es donde se hace la edición del plan de consumo.

Para cambiar la información de una cierta celda específica posicione el ratón en la celda a cambiar y presione el botón derecho del ratón. La información de esta celda se carga en la parte inferior izquierda abajo de la matriz en el área denominada cantidad.

Allí si cambiamos la información inmediatamente se refleja en la celda seleccionada. Los cambios a las celdas los podemos hacer a tantas celdas como se desee con el procedimiento anterior.

Al terminar para salvar los cambios realizados presione el botón denominada **actualiza**.

Si desea terminar sin salvar los cambios presione el botón denominado **cancela**.

### **Cálculo de consumos provenientes de SIREM**

Este módulo permite hacer el cálculo de consumos y consumos promedios provenientes de SIREM. Al realizar los cálculos los resultados son colocados con fecha de introducción de acuerdo a como se instruya en fecha intro.

Para cálculo de consumo se realizan los siguientes pasos:

- Proporcionar una fecha de referencia para cálculo de consumos. Se toma para seleccionar los movimientos con igual fecha de captura
- Presione el botón **cálculo de consumo**

Con esto se calcula el consumo a partir de la fecha de referencia indicada.

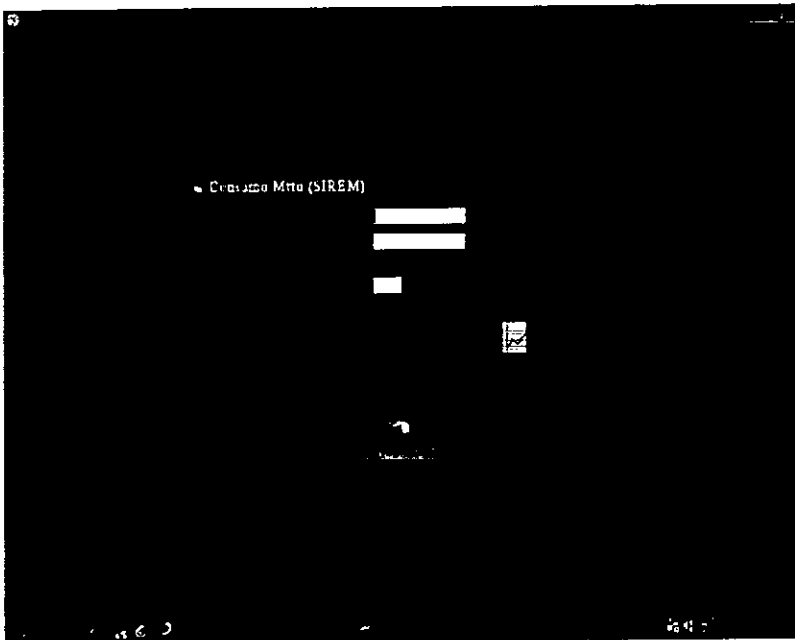
Para el cálculo de consumo promedio se realizan los siguientes pasos

- Proporcionar fecha de introducción al sistema

- Proporcionar una fecha de referencia para cálculo de consumos. Se toma para seleccionar los movimientos con igual fecha de captura.
- Proporcionar el número de meses a promediar.
- Presione el botón **cálculo de consumo promedio**. (Es importante que siempre haga primero cálculo de consumo y después cálculo de consumo promedio).

### **Cálculo de consumos por construcción en telefonía pública**

Para el cálculo de telefonía pública solo basta con proporcionar el año que se desea se calcule. Este cálculo se basa en el plan de construcción anual por región y en los factores de catálogos para la construcción de líneas.



**Pantalla 13. Módulo de cálculo de consumos provenientes de SIREM.**

## Reportes

Este sistema puede generar 2 tipos de reportes: reportes estáticos y dinámicos.

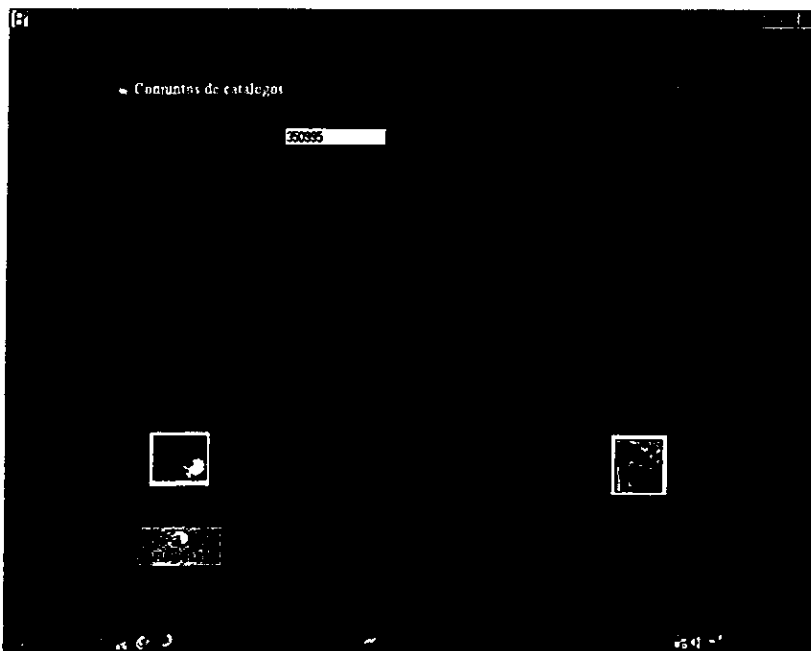
Los reportes estáticos son:

- Reporte de entregas a usuarios.
- Reporte de avance de pedidos.

Los reportes dinámicos del sistema son configurables de tal forma que se puede seleccionar los catálogos de materiales a incluir, la información relevante de catálogos a imprimir, las fuentes de cálculo de existencias futuras.

### Creación de conjuntos de catálogos

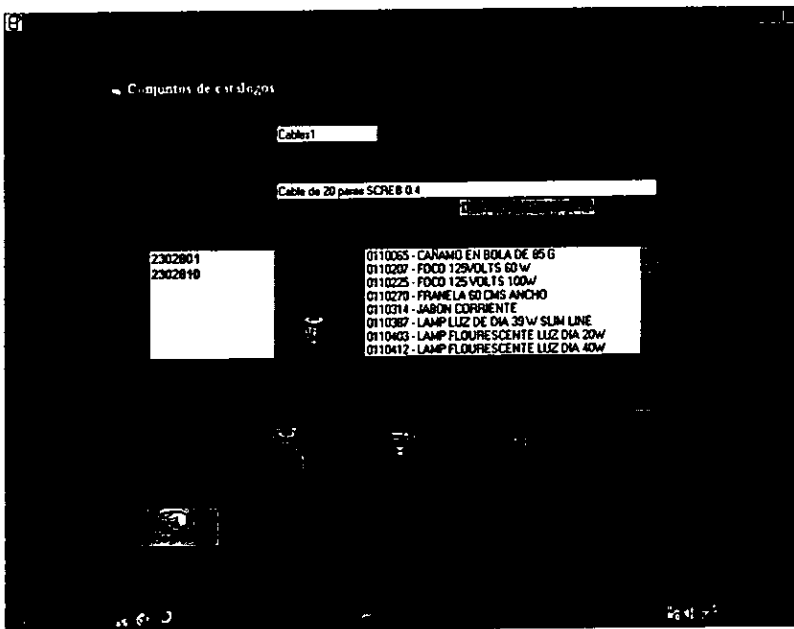
Aquí se introduce la creación de conjuntos que posteriormente se usan en configuración de reportes.



**Pantalla 14. Primera pantalla del módulo de creación de conjuntos de catálogos.**

Los reportes dinámicos permiten organizar la información y seleccionarla de la manera que a cada quien le parezca mejor. Para esto se ha proporcionado esta opción de conjuntos de catálogos cuyo propósito es el de organizar un grupo de catálogos de materiales y asociarlo a un 'conjunto' de tal forma que posteriormente en la configuración de reportes sea posible agregarlo al grupo de materiales a desplegar.

Como muestra la pantalla inicial de esta opción. Como se ve aquí podemos dar de alta o buscar a un conjunto de catálogo específico.



**Pantalla 15. Segunda pantalla del módulo de creación de conjuntos de catálogos.**

Ahora para incluir los catálogos que son parte de este conjunto hacemos lo siguiente: la lista a la derecha incluye todos los catálogos existentes en el sistema, de esta lista podemos buscar los catálogos a incluir y al encontrarlos se seleccionan con el botón derecho del ratón. Al estar marcados se presiona el botón con la leyenda <<. Al hacer

esto el sistema verifica que no estén incluidos ya en la lista de la izquierda. Si no están entonces se agregan como nuevos elementos de este conjunto. Esta operación la podemos realizar varias veces hasta incluir todos los catálogos requeridos. Al final el resultado se ve como en la pantalla 15.

Para salvar los cambios se presiona el botón que dice **Actualiza**. Los cambios son salvados.

Si se desean cancelar los cambios o adiciones se presiona el botón **Cancela**.

Si se desea eliminar todo el conjunto se presiona botón **Borrar**.

Para salir presione la equis en la esquina superior derecha.

### **Configuración de reportes**

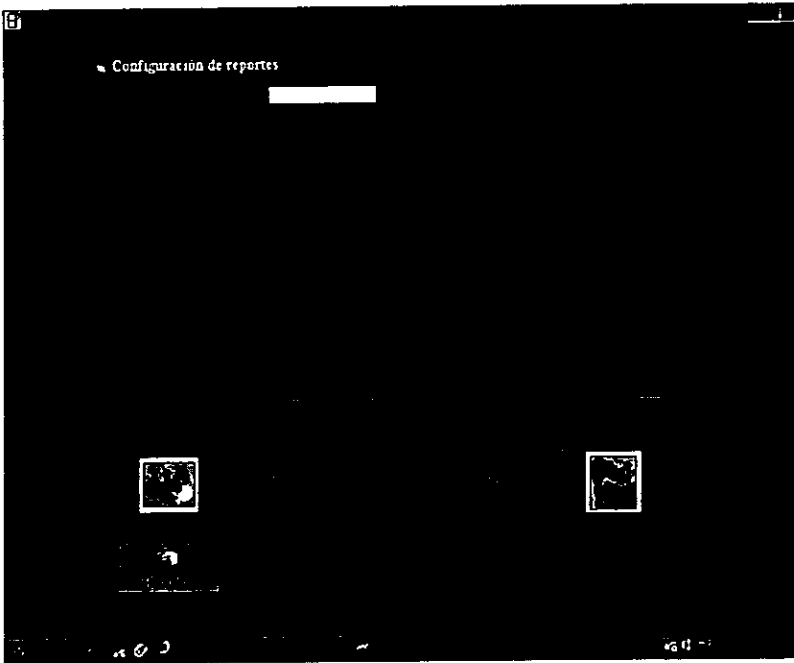
Mediante este módulo podemos generar diferentes tipos de reporte que pueden ser almacenados y posteriormente impresos. La mecánica de operación es como sigue.

Al tener necesidad de un nuevo tipo de reporte que involucre las existencias actuales, pedidos, consumos y existencias estimadas se procede a crear un tipo de reporte. Invocamos a la opción configuración de reportes y aparece la pantalla 16.

Aquí contamos con el ejemplo de un listado de materiales de planta exterior. A este reporte le llamamos 'MATPE' y lo escribimos en el área de identificador de reporte. Después presionamos el botón **alta** y el sistema nos muestran 2 botones que son la información completa con que cuenta el sistema.

El botón **Opciones** invoca a la pantalla 18 que contiene toda la información relativa a como queremos nuestro reporte: ¿Deseamos que se aparezca una determinada columna?, ¿Deseamos que los cálculos se lleven acabo sobre un consumo específico,

sobre otro en especial o sobre ambos?, ¿Deseamos que halla una sustitución por pronósticos en caso de no existir un consumo?, ¿Deseamos utilizar una fórmula dada para cálculo de existencias?



**Pantalla 16. Primera pantalla del módulo de configuración de reportes.**

Si seleccionamos el botón Catálogos y Conjuntos aparece la pantalla 14. En esta pantalla indicamos que catálogos y conjuntos son incluidos en el reporte.

Aquí podemos dar de baja un reporte con el botón Baja.

Para salir presionamos la X en la esquina superior derecha.

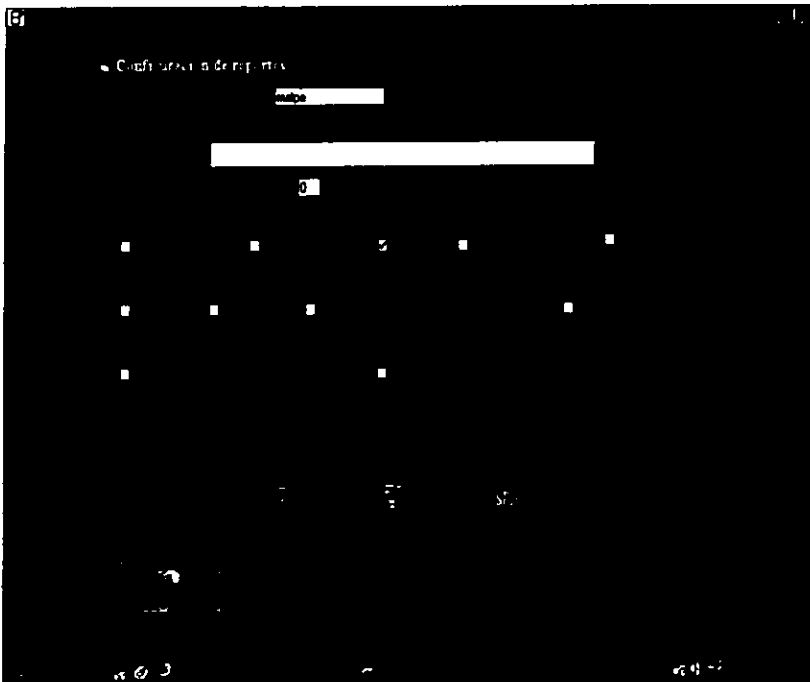
En la pantalla 17 proporcionamos información sobre como queremos que se presente el reporte. En el ejemplo se esta pidiendo que todos los campos relevantes aparezcan en la impresión.

Si no deseamos alguna columna por ejemplo stock entonces apuntamos el ratón a esa parte y marcamos con el botón derecho del ratón. El sistema apaga esa opción.

Debido a que la parte primordial de este reporte es generar estimados de existencias hasta 6 meses, podemos indicar cuantos meses deseamos que sean considerados en una proyección.

Este dato es útil porque en caso que estemos haciendo comparativos de existencias de diferentes consumos entonces en lugar de pedir 6 meses podemos pedir 3 meses para cada uno.

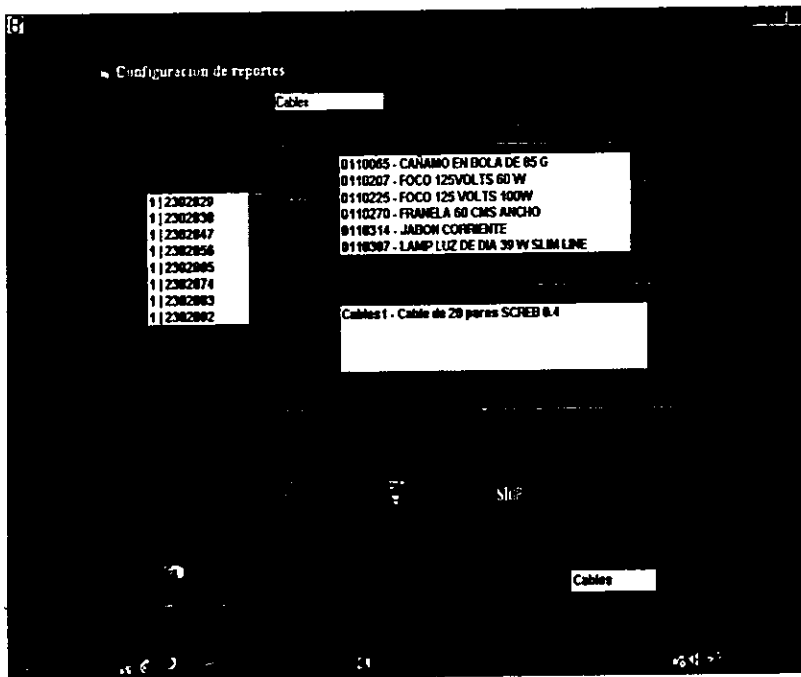
La parte que dice ¿Sobre qué datos desea obtener proyecciones? Sirve para indicar que fuente de datos será usada para la generación de resultados.



**Pantalla 17. Tercera pantalla del módulo de configuración de reportes.**

La pantalla 18 muestra la asignación de catálogos y conjuntos al reporte. A la derecha de la pantalla existen 2 listas con los catálogos y conjuntos existentes. Los usuarios pueden seleccionar de cualquiera de estas 2 listas e incluirlas al reporte.

El conjunto asignado aparece con un 2 para indicar que se trata de un conjunto. La operación descrita se aplica tanto para conjuntos como para catálogos y puede ser repetida tantas veces se requiera.



**Pantalla 18.** Cuarta pantalla del módulo de configuración de reportes.

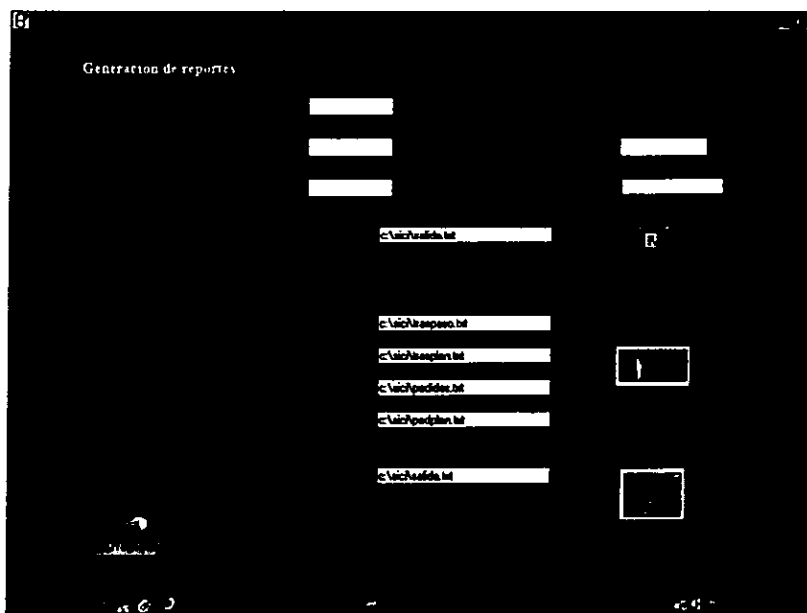
Para borrar un conjunto o catálogo del reporte se selecciona el elemento a borrar de la lista de la izquierda con el botón derecho del ratón. Acto seguido se apunta el ratón al bote de basura asociado y se presiona el botón derecho. El sistema solicita confirmación de la eliminación y borra al elemento.



Una vez que se tienen todos los cambios y adiciones se procede a salvarlos con el botón **Actualiza**. En caso de no desear salvar los cambios se presiona el botón **Cancela**.

### Generación de reportes

La pantalla 19 presenta el generador de reportes dinámicos del sistema. Básicamente este módulo tiene 3 funciones:



**Pantalla 19. Pantalla del módulo de generación de reportes.**

- Generación de reporte dinámico a un archivo texto para su posterior impresión.
- Generación de reportes de plan de pedidos y traspasos a archivos texto para impresión.
- Impresión de archivos texto.

Para la **generación de un reporte dinámico** primero se debe de dar el nombre del identificador que contiene y como está configurado el reporte a generar.

Una vez especificado el archivo de configuración se procede a indicar de que fecha tomar las existencias, pedidos y consumos a utilizar.

**Archivo de salidas (reporte/traspasos)** indica el nombre del archivo donde se almacenará el resultado de la generación. El sistema solo genera el archivo pero no lo imprime automáticamente. Si se desea otro nombre de archivo favor de indicarlo en este espacio.

Para la generación del reporte se presiona el botón que dice **GENERA REPORTE**. El sistema pide confirmación y una vez que inicia se indica en la barra gráfica de avance localizada a la derecha como va el proceso. Dependiendo de la cantidad de catálogos incluidos puede ser que la ejecución tome un tiempo largo por lo que hay que esperar por resultados. Al terminar la generación el sistema lo indica.

La **Generación de reportes de plan de pedidos y traspasos** es un producto que se basa en la misma información que la generación de reportes dinámicos por lo que la información se llena de la misma manera que en la generación de reportes.

Debido a que este proceso genera 4 archivos en lugar de uno entonces se tiene la posibilidad de indicar cuales son los nombres de archivo para cada uno. Mediante los espacios **Archivo de salidas (reporte/traspasos)** y **Archivo de salida (pedidos)** es posible especificar donde se almacenaran los resultados para reporte de traspasos y reporte de pedidos respectivamente.

Para arrancar la generación de reportes de plan de pedidos y traspasos primero cargue la información tal como se indico para generación de reportes y después presione el botón **GEN PEDIDOS Y TR**. Aquí deberá esperar por los resultados que puede que también se tarde un período prolongado (aquí no se hace uso de la barra de avance).

Al terminar los reportes el sistema indica que ha terminado.

La Impresión de resultados se realiza de manera que uno a uno de los archivos se imprimen. Para indicar que archivo se va a imprimir se especifica en el espacio de **Archivo de salidas (reporte/traspasos)**. Al terminar de teclear se presiona el botón **IMPRIMIR** con lo que el archivo indicado es impreso.

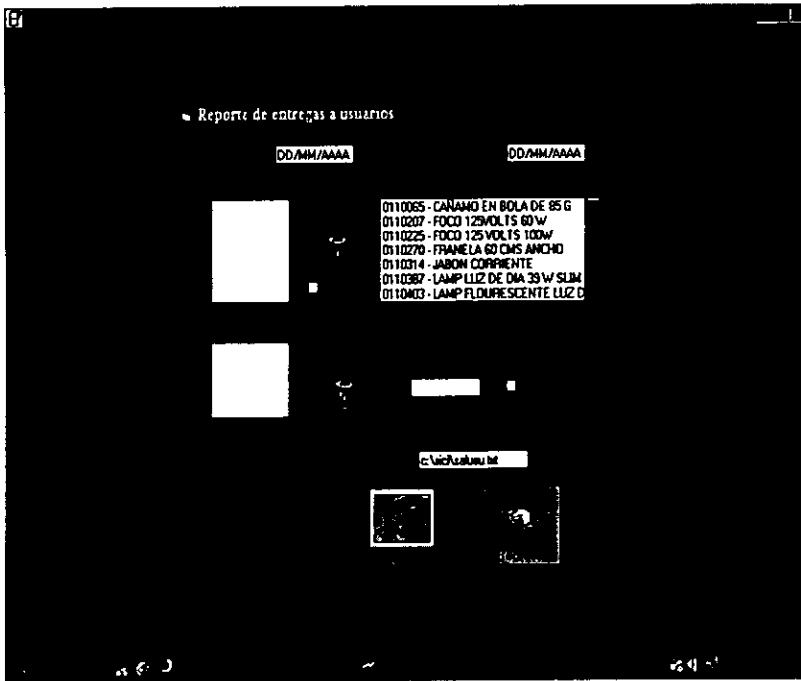
### **Reporte de entregas a usuario**

El reporte de entregas a usuario proporciona un listado del material entregado a los usuarios organizado por expediente. Se debe de proporcionar un rango de fechas de captura y los materiales sobre los que se desea saber información.

Para incluirlo al reporte basta con buscar en la lista de catálogos disponibles localizado a la derecha. Al encontrarlo se marca con el botón derecho del ratón y se presiona el botón <<. El sistema lo incluye en el reporte. Existe otra opción que es la de seleccionar todos los catálogos. Para hacer esto marque a caja **Todos los catálogos** situados a la derecha. Para eliminar un catálogo del reporte marcarlo en la lista de la izquierda con el botón derecho del ratón y marcar con el botón derecho del ratón el bote de basura. El sistema pide confirmación y lo borra.

También se puede especificar el número de expedientes que se desee seguir. Esto se hace en la subpantalla de abajo. Primero se debe de escribir el número de expediente de interés y después presionar el botón << del centro. El sistema incluye el expediente.

También se tiene la opción de seleccionar todos los expedientes marcando la caja indicada **Todos los expedientes** localizado a la derecha. Para eliminar un expediente del reporte marcarlo en la lista de la izquierda con el botón derecho del ratón y marcar con el botón derecho del ratón el bote de basura. El sistema pide confirmación y lo borra.



**Pantalla 20. Pantalla del módulo de reporte de entregas a usuarios.**

El reporte generado residirá en la ruta especificado en el espacio señalado como **nombre de archivo**. Si desea cualquier otro nombre favor de indicarlo aquí.

Una vez que todos los parámetros han sido especificados se realiza la generación del reporte presionando el botón **GENERAR** con el botón derecho del ratón. Al terminar el reporte el sistema lo indica.

De aquí para imprimir el resultado presione el botón **Imprimir** con el botón derecho del ratón. Este seguro que el nombre en **nombre de archivo** sea el correcto.

## **Reporte de avance de pedidos**

El reporte de avance de pedidos presenta el avance de cada pedido en función de las entregas prometidas y las entregas realmente realizadas a lo largo de un plan de entrega establecido.

Para obtener este reporte es necesario que se indiquen las fechas de introducción del archivo de pedidos y existencias.

## **Tablas**

Esta sección cubre tablas que son soporte del sistema que son cambiadas constantemente. Básicamente se tienen 3 tablas que requieren cambios esta son: catálogos, almacenes y regiones factibles.

Catálogos y almacenes son tablas en las que basan su operación muchas de las tablas del sistema.

Regiones factibles es una tabla que permite establecer preferencias de regiones para cada región en el sistema. Estas preferencias son usadas para realizar el traspaso de existencias entre almacenes. Toda esta información es útil en la generación de los reportes de traspasos y pedidos.

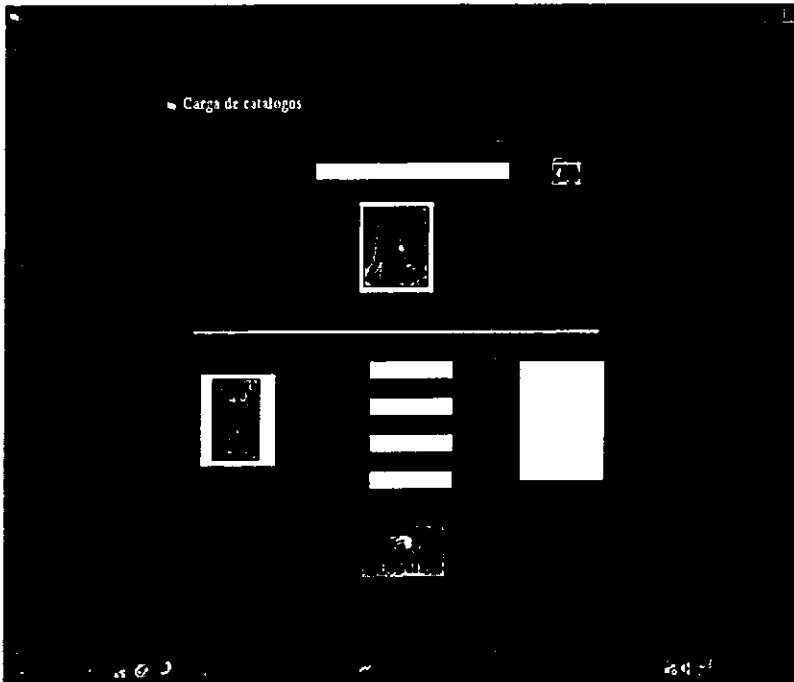
## **Carga de catálogos**

Los catálogos son información proporcionada en archivos dBASE por lo que se debe cargar a la base de datos interna del sistema. Para resolver esto el presente módulo hace la inserción de los datos de catálogos insertando la información de los campos únicamente requeridos.

El usuario pone en nombre de archivo la ruta completa del archivo dBASE a cargar.

Hecho esto se presiona el botón **Carga campos**. El módulo lee la configuración interna de ese archivo y presenta en Campos dBase los campos disponibles. El usuario debe seleccionar los campos pertinentes para los campos catálogo, unidad, descripción y condición e indicarlos mediante la selección del campo y presionando el botón << para asociarlo a cada uno.

Al estar listo se presiona el botón **carga archivo** y se procede a cargarlo. Al terminar el sistema pide confirmación de carga.

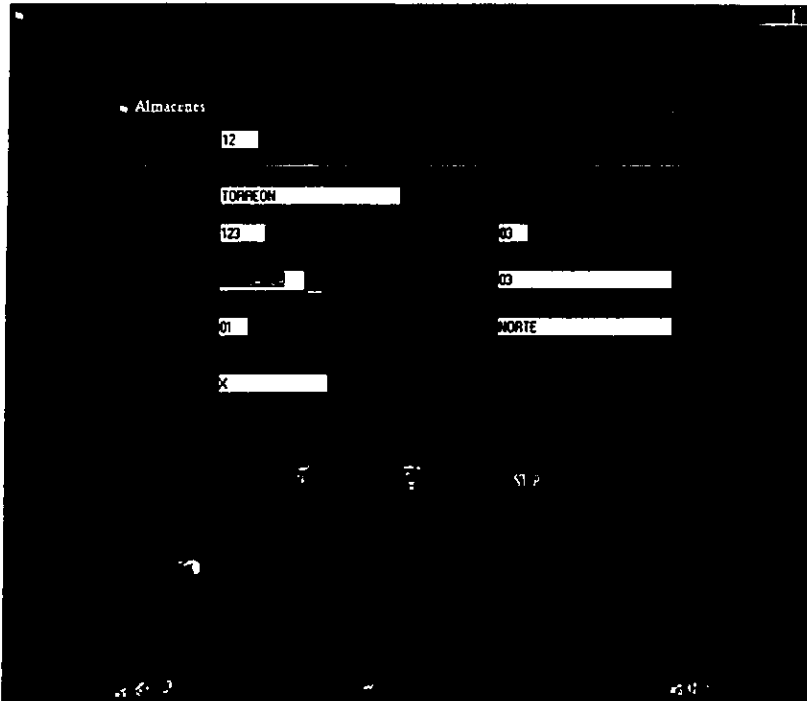


**Pantalla 21. Pantalla del módulo de carga de catálogos.**

### **Captura de almacenes**

La información de los almacenes se cambia y edita en una pantalla de captura. Primero

se muestra la pantalla de selección o alta del almacén. Por ejemplo se trabaja con el almacén 32. Primero se proporciona numero de almacén y se presiona el botón de buscar o alta. El sistema busca y muestra la pantalla 22.



**Pantalla 22. Segunda pantalla del módulo de captura de almacenes.**

Al terminar el usuario debe de presionar el botón **Actualiza** para salvar todos los cambios ocurridos. Si no se desea salvar los cambios entonces presionar **Cancela**.

Para eliminar un almacén se debe de presionar el botón **Baja**.

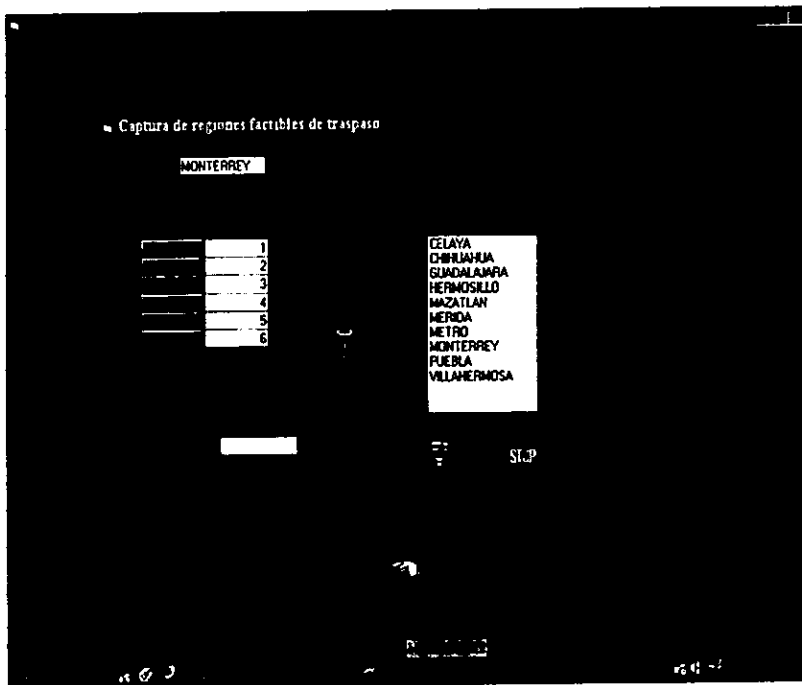
### **Captura de regiones factibles para traspaso**

Las regiones factibles son las regiones más aptas para el envío de trasposos de

material. Cada región posee un conjunto de regiones factibles y es necesario capturarlas para que al momento de que el sistema genere un plan de pedidos y trasposos considere este tipo de relaciones.

Primero muestra la pantalla de búsqueda y alta de regiones. Para escoger alguna región en especial seleccione de la lista de regiones disponibles localizada hacia la derecha en la parte superior de la pantalla

La pantalla 23 muestra a la derecha una lista con todas las regiones registradas. De aquí se deben de localizar aquellas que son aptas para hacer trasposos. Al encontrar en la lista se marca con el botón derecho del ratón y se presiona el botón <<. Sin embargo cabe aquí agregar a otro dato que se llama la preferencia de la región.



**Pantalla 23. Segunda pantalla del módulo de captura de regiones factibles para traspasso.**



Este dato se debe proporcionar cada vez que se incluye una nueva región. Por ejemplo Guadalajara tiene una preferencia por sobre Mazatlán y Metro. Mazatlán tiene una preferencia por sobre Metro y así sucesivamente.

Esta preferencia como se ve se asigna con el número. El número 1 tiene la mayor preferencia. Para cambiar una preferencia posicione el ratón en la celda de la región que desee cambiar y presione el botón derecho del ratón. En la parte inferior izquierda aparece el valor actual de preferencia y allí usted puede poner un nuevo valor.

Si desea borrar alguna región de preferencia, selecciónela con el ratón en la lista de la izquierda y posicione el ratón en el bote de basura. Enseguida presione el botón derecho del ratón. El sistema pide confirmación y se elimina el elemento.

Para salvar los cambios presione el botón **Actualiza**. Si no desea salvar los cambios presione el botón **Cancela**.

## Manual Técnico

El Objetivo de este manual es que el administrador del sistema, cuente con un documento en el cual se describan todos y cada uno de los elementos relevantes para que, pueda realizar sus funciones como tal. Es decir, requerimientos tanto de hardware y software, procedimientos para la instalación y eliminación del sistema, procedimientos de respaldos, configuración del módulo para acceso a la red, y descripción de la base de datos del sistema.

Los requerimientos (mínimos) de hardware y software necesarios para la instalación del sistema son (los que se mencionan en el punto 3.6.):

Un CPU con el siguiente hardware:

- Procesador Pentium II a 300 MHz.
- 64 Mb en memoria RAM.
- 8 GB en disco duro.
- Monitor SVGA de 14 ".
- Mouse.
- Tarjeta de red (Ethernet a 10 Mbits).
- Unidad de disco de 3 ½" de alta densidad.

Los requerimientos de software son los siguientes:

- Sistema operativo Windows 9X.
- Visual Basic versión 5.0.
- Access versión 97.


### Procedimiento para la instalación del sistema

- Insertar el disco uno del kit de instalación del sistema.
- Desde el menú de inicio de Windows 9X, en la opción ejecutar, correr el programa **setup.exe** de A:, aparecerá la siguiente Pantalla 1.



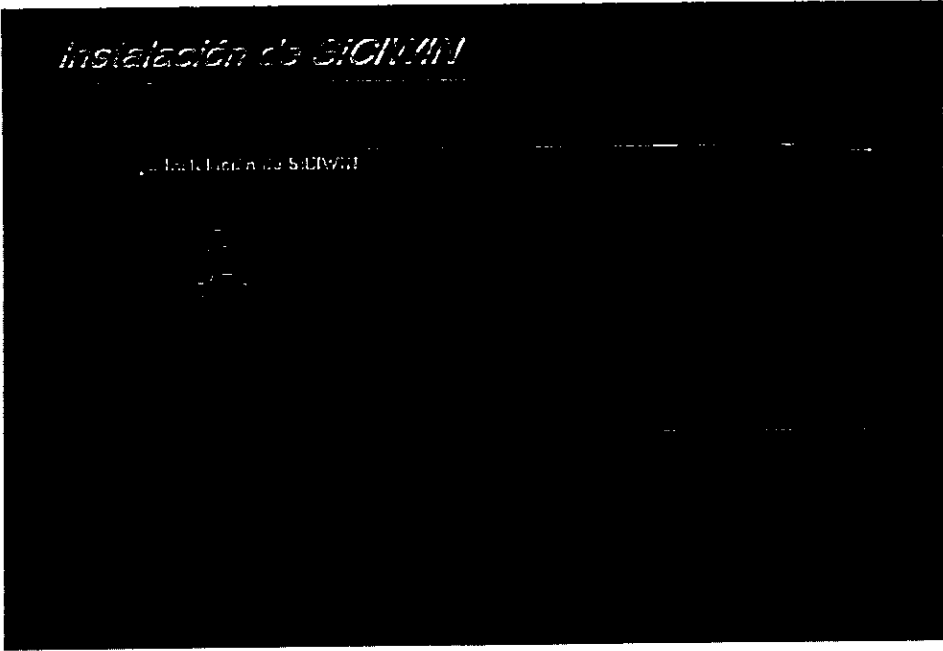
**Pantalla 1. Instalación de SiCIWIN.**

En esta Pantalla 1 se despliega un mensaje de bienvenida, algunas recomendaciones como cerrar todas las aplicaciones, para que el sistema pueda actualizar los archivos que en un momento puedan ser compartidos con otras aplicaciones, y dos opciones una Aceptar opción por omisión; las funciones que estos botones realizan, a continuación se describen:

 Este botón se utiliza para continuar con la instalación del sistema del cual se desprende la Pantalla 2, es decir el subdirectorio donde se instalará el sistema, cabe mencionar que la capacidad necesaria para la instalación del sistema es de 18 MB, además del espacio para la base de datos.

 Este botón se utiliza para cancelar la instalación, es decir se

detiene el proceso de instalación y se despliega la Pantalla 3.



**Pantalla 2. Definición de ruta destino de instalación.**

En esta Pantalla 2 se define la ruta de instalación y se activa el proceso de copia de software en la ruta seleccionada, por omisión el sistema tiene asignado el directorio de archivos de programa y el subdirectorio SiCIWIN, a continuación se describen los elementos que la conforman.



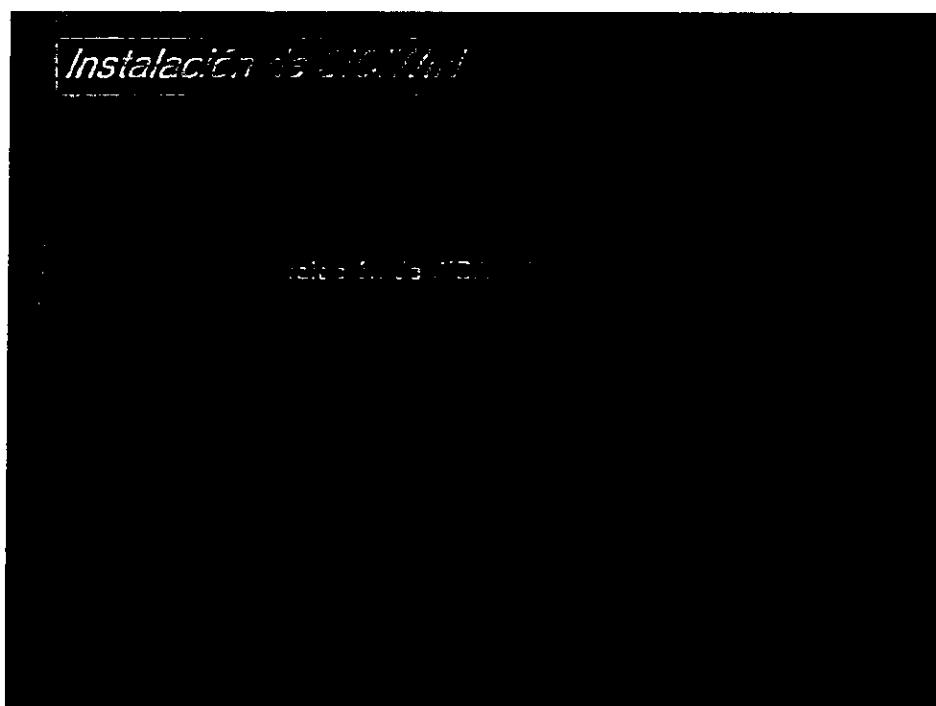
Haciendo clic en este icono se procede a la instalación del software del sistema en la ruta definida y se desplegará la Pantalla 5.



Con este botón se puede definir la ruta donde se instalará el sistema, como se muestra en la Pantalla 2, solo se debe tener en cuenta la capacidad de disco necesaria.



Este botón se utiliza para salir de la instalación y se desplegará la Pantalla 3.



**Pantalla 3. Desea abandonar la instalación del sistema.**

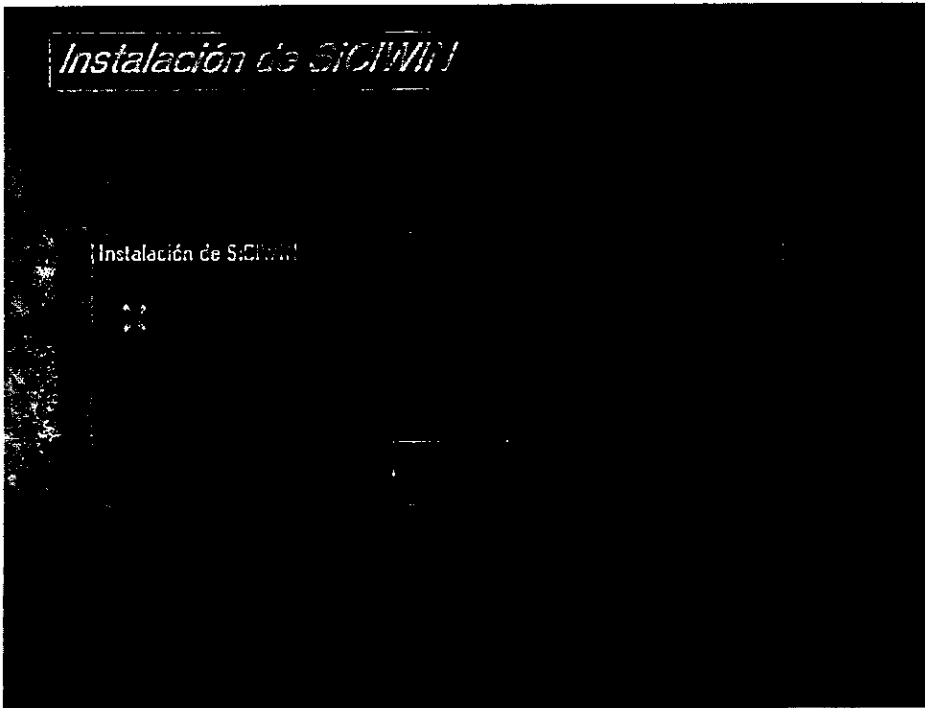
En esta Pantalla 3 se confirma si la instalación del sistema se cancela y se tienen los dos botones que a continuación se describen.



Este botón se utiliza para confirmar la cancelación de la instalación del sistema y desplegará la Pantalla 4.

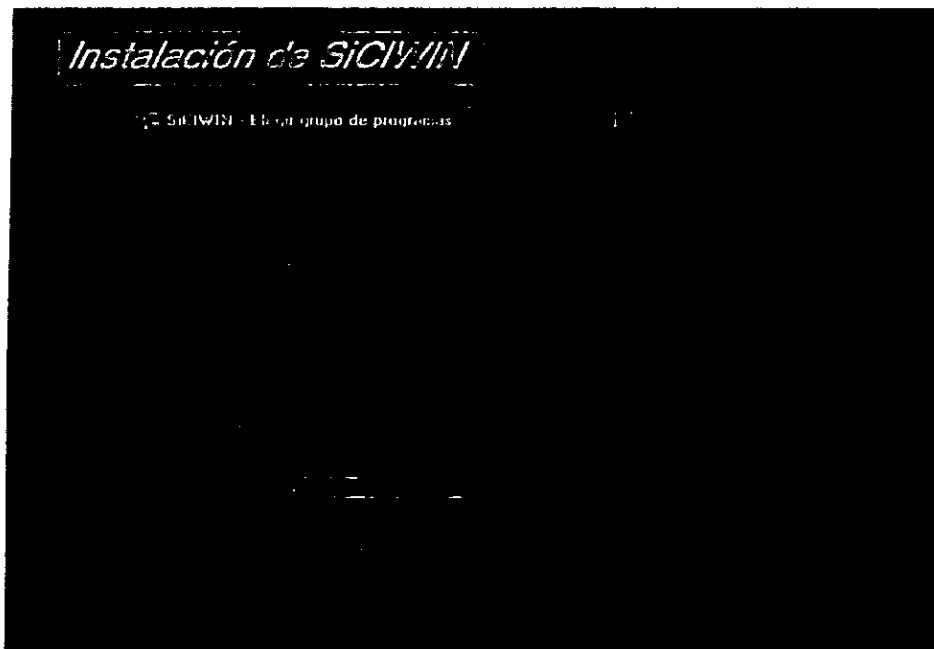


Este botón se utiliza para cancelar la opción desinstalar el sistema y continuar con el proceso de instalación.




**Pantalla 4. Mensaje de cancelación de la instalación del sistema.**


En esta Pantalla 4 se tiene el mensaje de que se interrumpió la instalación del sistema y solo resta hacer clic en el botón de aceptar y se habilitará el proceso de quitar el sistema, como se explica mas adelante en la sección de quitar sistema.

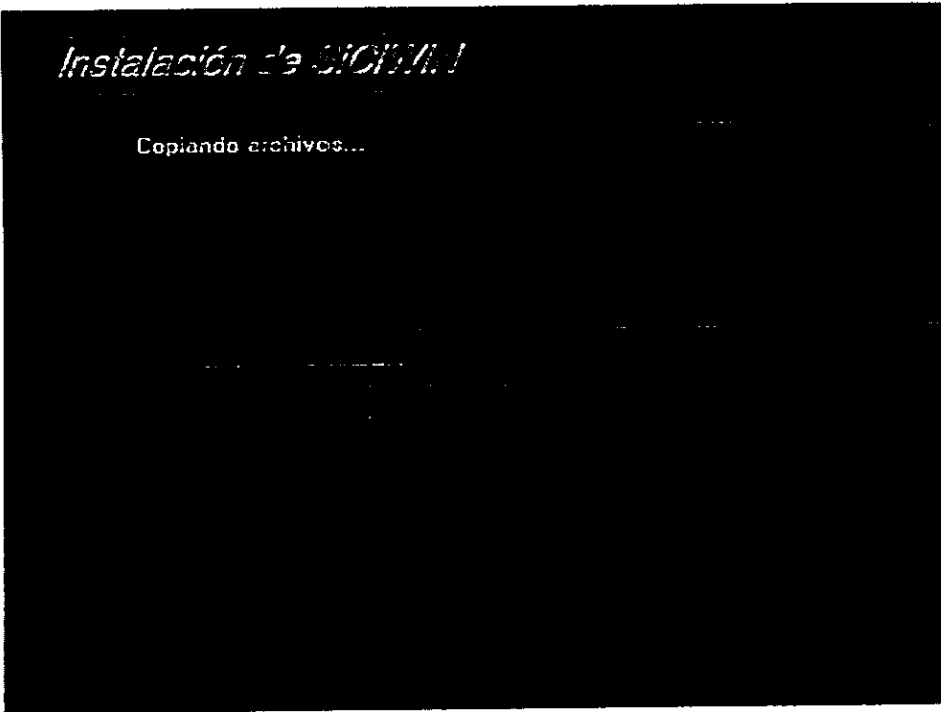


**Pantalla 5. Elección de grupo de programas.**

En esta Pantalla 5 se define el grupo donde se agregará el elemento del sistema, el sistema define un grupo por omisión (SiCIWIN), pero el usuario puede seleccionar algún otro grupo de la lista o en su defecto teclear alguno que crea conveniente; Esta Pantalla tiene dos botones que a continuación de describen sus funciones.

 Una vez seleccionado el grupo de programas en el cual se agregará el sistema, con este botón se habilita el proceso de instalación del sistema al subdirectorío temporal de Windows (C:\WINDOWS\SYSTEM\*.\*) , y después se habilitará la Pantalla 6.

 Este botón se utiliza para cancelación de la instalación del sistema, y desplegará la Pantalla 3.



**Pantalla 6. Copia de archivos al directorio final.**

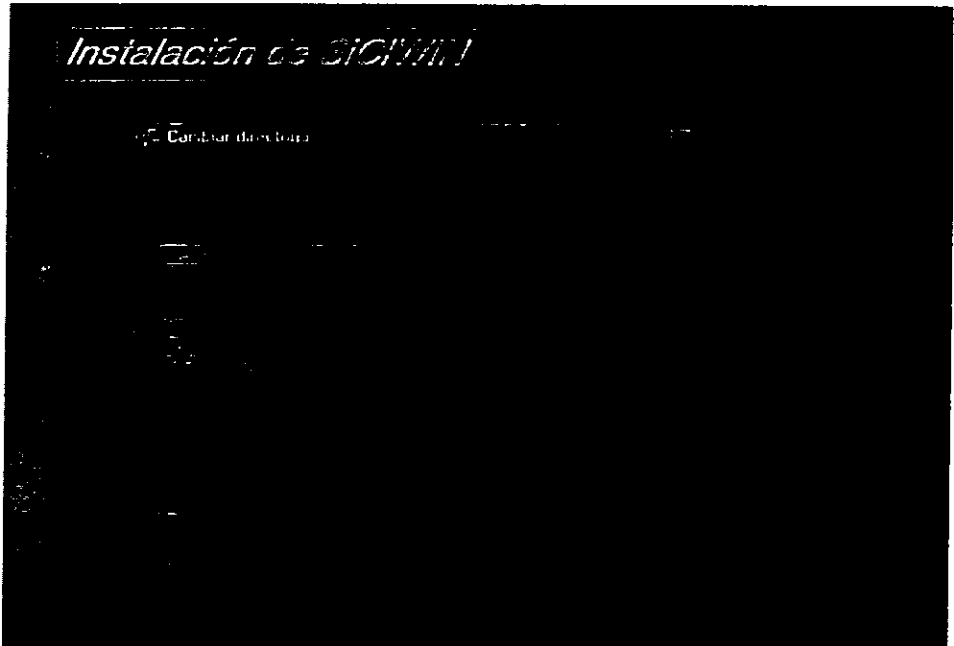
En esta Pantalla 6 se muestra una barra de porcentaje de instalación del sistema del subdirectorio temporal al directorio final, además se tiene un botón que a continuación se describe su función y si no existen errores en el proceso de esta Pantalla se habilitará la Pantalla 8, en cualquier otro caso se habilitara una Pantalla de Error.



Este botón se utiliza para la cancelación de la copia de los archivos del sistema del directorio temporal al directorio

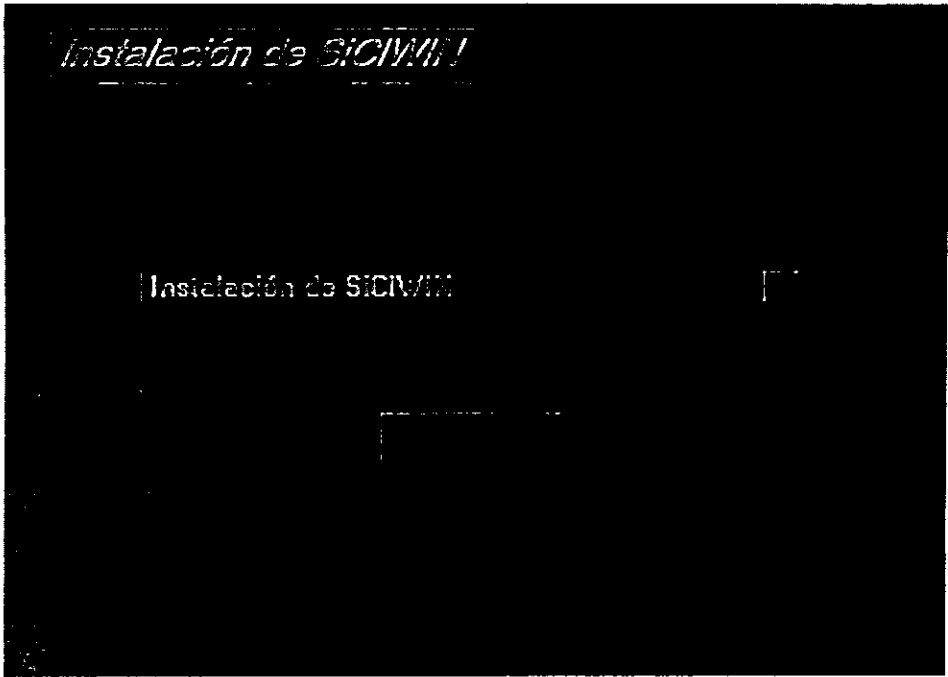


final y por ende la cancelación de la instalación del sistema, y desplegará la Pantalla 4.



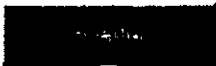
**Pantalla 7. Especifique o seleccione subdirectorio destino.**

En esta Pantalla 7 se tiene la posibilidad de definir o seleccionar otro subdirectorio destino en el cual se instalara el sistema, para realizar esta función la Pantalla 7 esta conformada por los siguientes botones: uno de aceptación y otro de cancelación y su función es igual a la que se tienen en otras Pantallas anteriores donde se tienen estos botones; se tiene un elemento donde se puede teclear directamente el subdirectorio, una lista donde puede seleccionar el subdirectorio y finalmente se tiene una lista donde se puede seleccionar la unidad de disco.



**Pantalla 8. La instalación del sistema ha terminado correctamente.**

Esta Pantalla 8 es la última del proceso de instalación y solo resta dar clic en el botón para terminar el proceso .



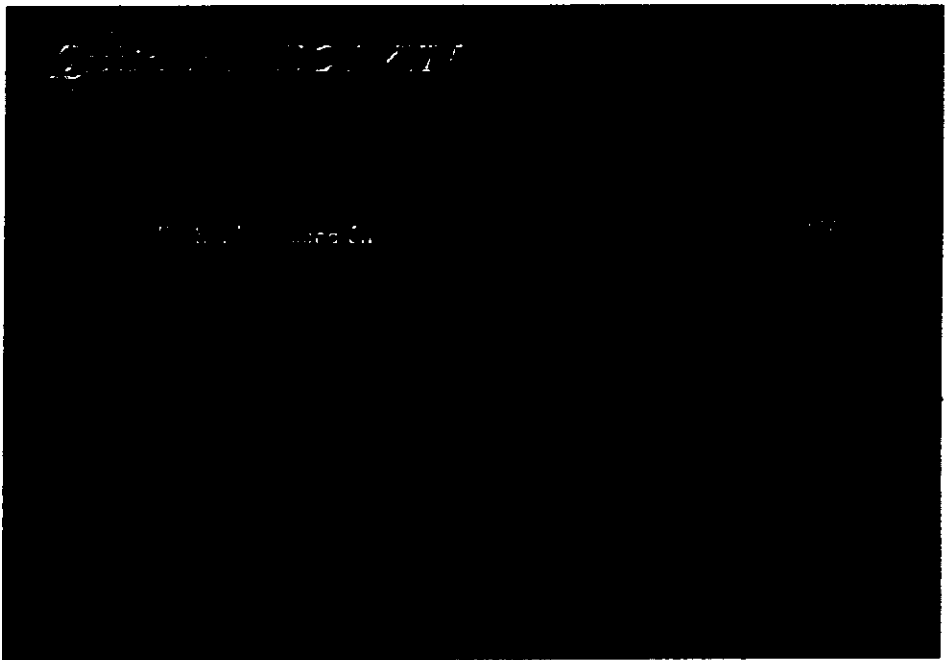
Este botón se utiliza para dar por terminado el proceso de instalación del sistema.

### **Procedimiento para quitar el sistema**

En este procedimiento se definen los pasos para quitar el sistema; esto es una ventaja, ya que en ocasiones existe la posibilidad de migración del sistema a otro equipo de cómputo y si en el equipo original ya no es necesario la utilización del

mismo, no es conveniente que el sistema este instalado utilizando recursos de este, para ello como el sistema esta desarrollado para un ambiente de sistemas Windows 9X; el sistema cuenta con esta flexibilidad, a continuación se describen los pasos para realizarlo:

Desde el menú de inicio de Windows 9X, seleccionar la opción de Configuración y entrar al Panel de control; dentro de esta Pantalla ejecutar el icono de Agregar y quitar programas seleccionando el sistema (SiCIWIN) al dar clic se habilitara la Pantalla 9..



**Pantalla 9. Confirmación de quitar elementos del sistema.**

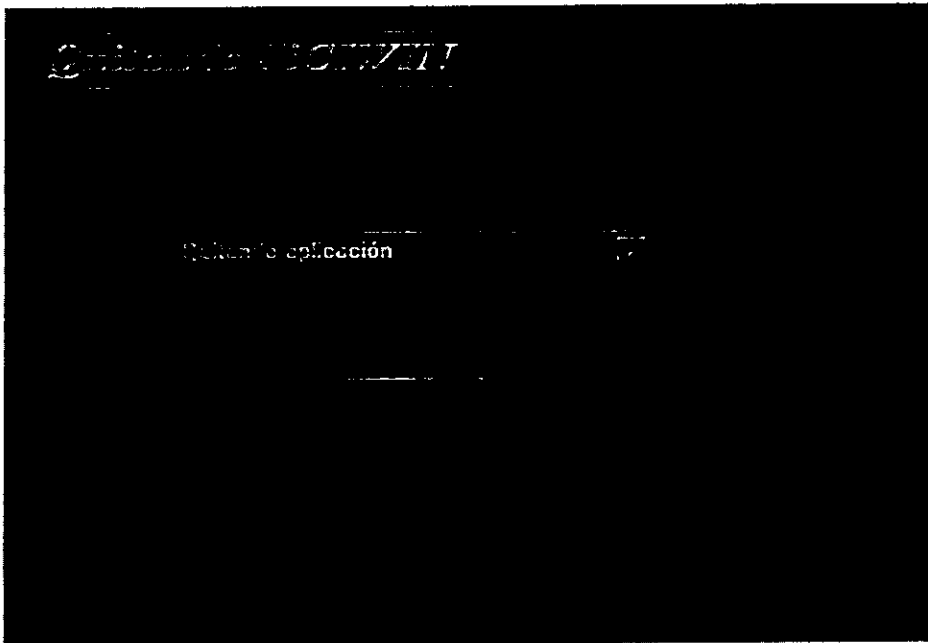
En esta Pantalla 9 se pide que se confirme la operación de quitar todos los componentes del sistema, esta Pantalla tiene dos botones que a continuación de describe:



Este botón se utiliza para confirmar la opción de quitar los elementos del sistema y se desplegará la Pantalla 11.

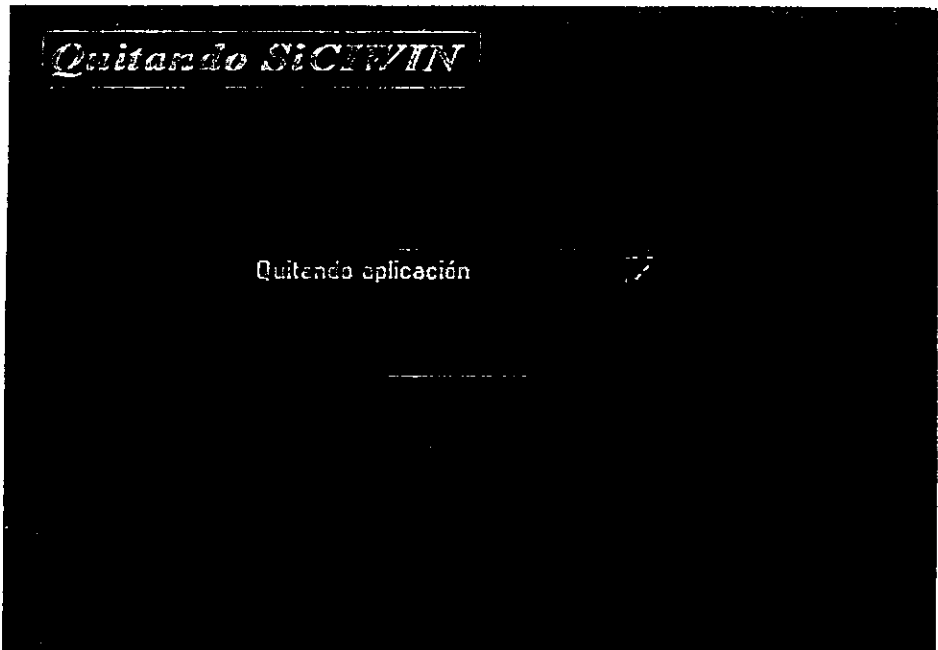


Este botón se utiliza para cancelar la opción de quitar los elementos del sistema y se desplegará la Pantalla 10.



**Pantalla 10. Se anuló la opción de eliminar la aplicación.**

En esta Pantalla 10 solo resta hacer clic en el botón de aceptar para cancelar la opción de eliminación de la aplicación.

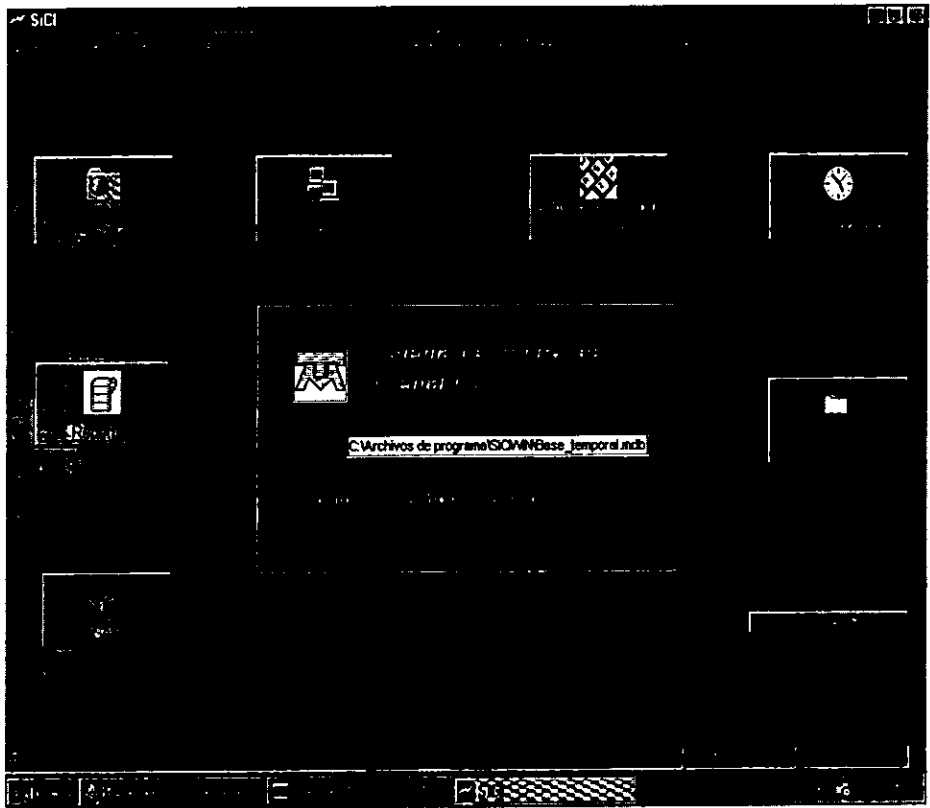


**Pantalla11. Confirmación de la eliminación de la aplicación.**

En esta Pantalla 11 solo resta hacer clic en el botón de aceptar para dar por terminado el proceso de eliminación de la aplicación.

### Procedimiento para crear una nueva base de datos

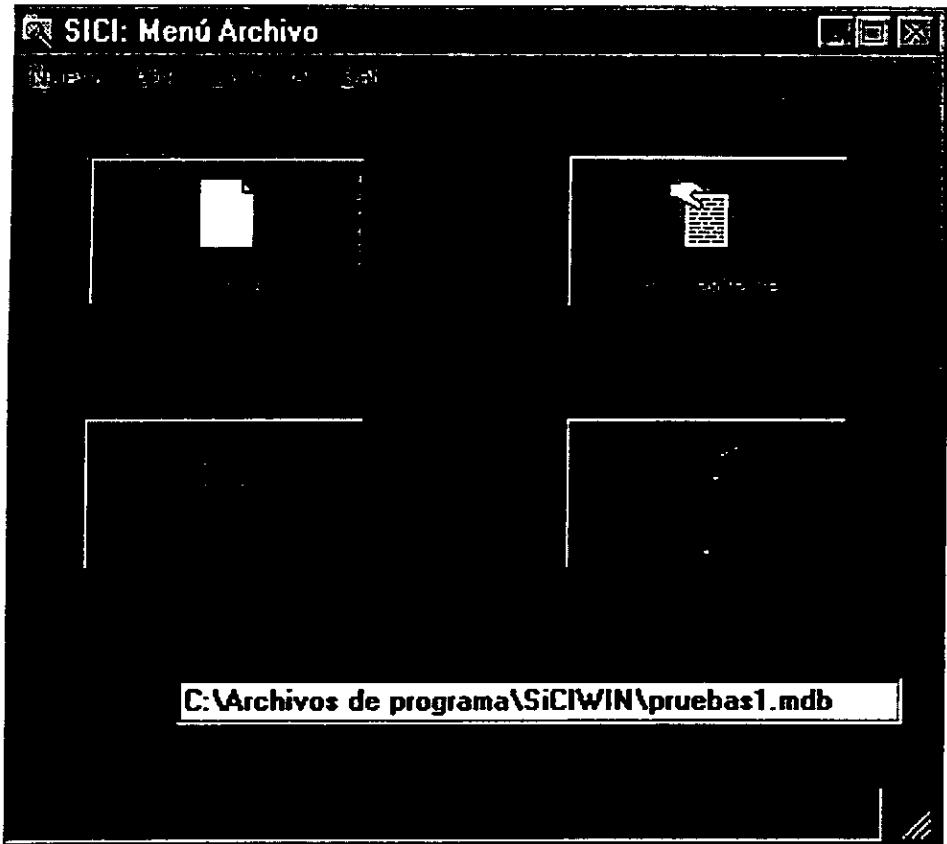
En este procedimiento se indican los pasos para crear una nueva base de datos, para este caso se definió un proceso de creación de la estructura de la base de datos para el sistema.



**Pantalla 12. Principal del Sistema.**



En la Pantalla 12, se tienen definidos los iconos para los diferentes procesos; para la creación de una base de datos utilizaremos este icono y se habilita la Pantalla 13, en la cual se indican dentro de la misma otros procesos.

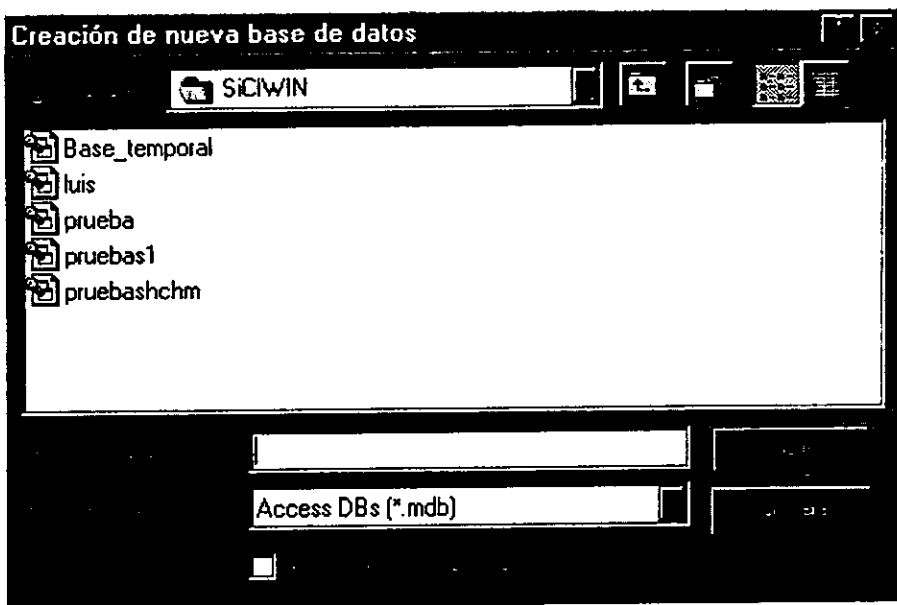


**Pantalla13. Menú Archivo.**

En esta Pantalla 13 se pueden realizar los procesos de: abrir una base de datos que previamente fue creada por el sistema, crear nueva base de datos, continuar con la ejecución del sistema o salir del mismo, para ello en las pantallas se trata de tener iconos que traten de indicar el proceso que realizan.



Para nuestro caso utilizaremos este icono que crea una base de datos, para ello habilita la Pantalla 14. Donde se indicará el nombre y la ruta de la nueva base de datos, para que el sistema tenga definida la estructura de datos necesaria, como se muestra en la misma.



Pantalla14. Nueva base de datos.

En esta Pantalla 14 son proporcionados los datos necesarios: el nombre, la rutas, tipo de apertura de la base de datos; una vez definidos se procede a dar clic en el botón de abrir y de esta manera se concluye la creación y apertura de la nueva base, o en su defecto salir del procesos con el botón de cancelación.

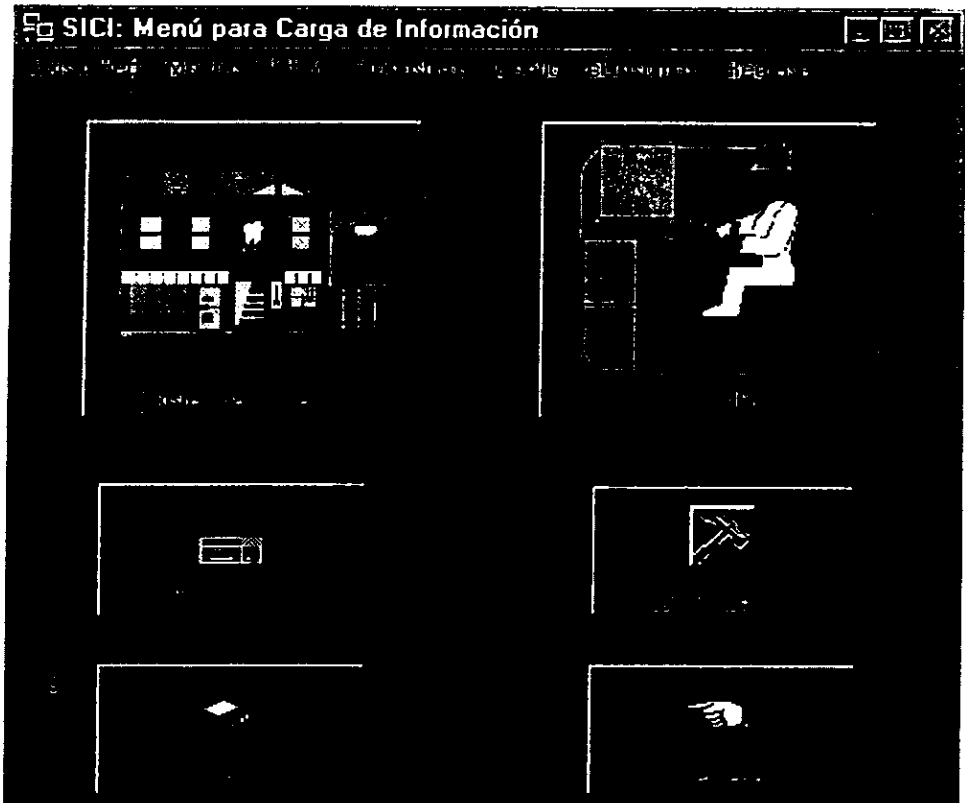
#### Procedimiento para la configuración de transferencia de archivos

En este procedimiento se indican los pasos para la definición de los datos necesarios para que se pueda realizar el proceso de transferencia de los archivos del sistema SIREM (Existencias, Pedidos, Movimientos), que son los insumos para el sistema.



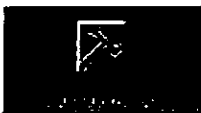
Para realizar esta función se selecciona este icono que esta definido en los procesos de la Pantalla 12 y se habilitara la Pantalla 15.



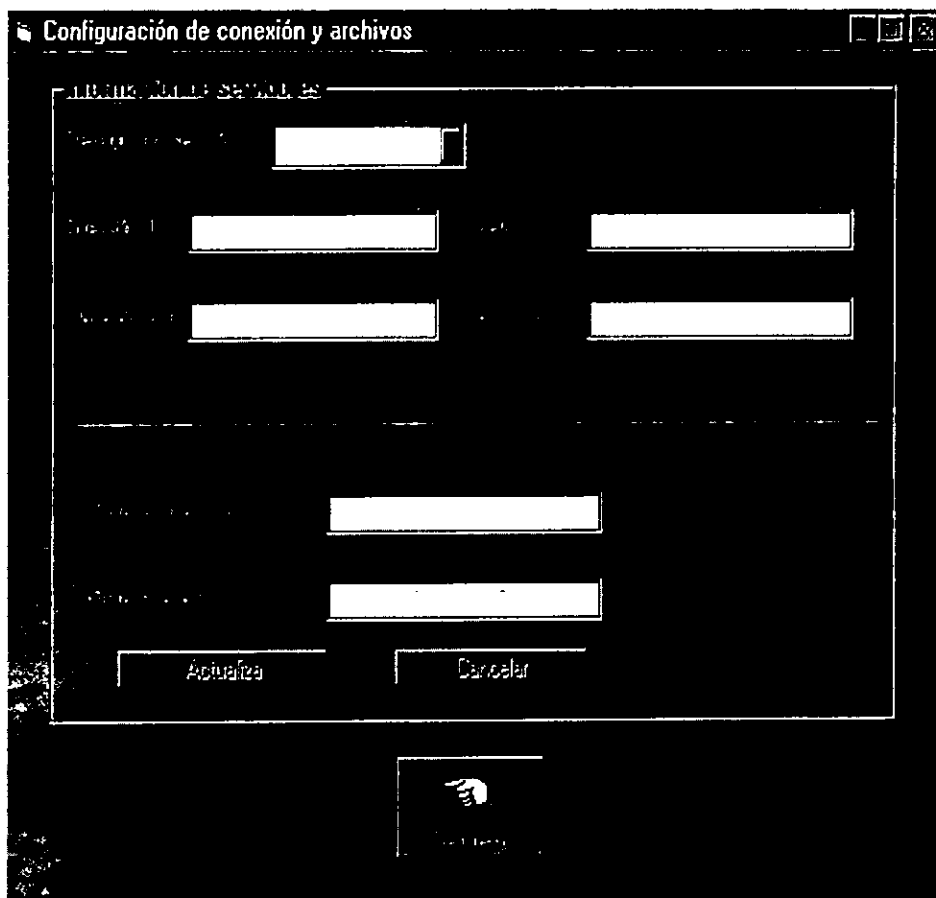


**Pantalla15. Menú para carga de información.**

En esta Pantalla 15, como su nombre lo indica, los procesos que se pueden realizar son: existencias, pedidos, movimientos, pronósticos y configuración para la transferencia de archivos del sistema SIREM, que es donde se tienen los insumos de los diferentes almacenes, y en su defecto se tienen el botón de regresar a la Pantalla 12.



Para realizar las funciones de transferencia de archivos se selecciona este icono, el cual tiene asociada la Pantalla 16.



**Pantalla 16. Configuración de conexión y archivos.**

En esta Pantalla 16 se teclean los datos necesarios para la transferencia de archivos del sistema, los datos solicitados son:

- |                       |   |
|-----------------------|---|
| Descripción servidor: | Tipo de archivo que se transferirá(existencias, movimientos o pedidos). |
| Dirección IP:         | La dirección IP del sistema SIREM (13.44.2.1).                          |
| Puerto:               | Número de puerto (23).  |
| Clave:                | Usuario de acceso al sistema SIREM.                                     |

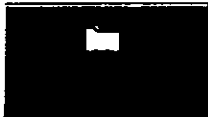
**Password:** Contraseña de acceso al sistema SIREM.  
**Ruta remota del archivo:** Ruta donde están los archivos del sistema SIREM.  
**Ruta local del archivo:** Ruta donde se transferirán los archivos.

Una vez proporcionados los datos, se selecciona el botón actualizar para confirmar y guardar los cambios y posteriormente salir de la Pantalla con el botón de regresar.

### **Procedimientos de captura de catálogos, almacenes, y regiones factibles de traspaso.**

En este procedimiento se describen las herramientas para la captura de información necesaria para el sistema, como son: los catálogos alta nuevos almacenes, y almacenes de los cuales se pueden realizar traspasos de materiales.

Para realizar estos procesos se tienen definidos los siguientes iconos:



Con este icono se habilita la Pantalla 17, en la cual se pueden realizar los procesos que a continuación se describen con sus diferentes iconos asociados a los mismos.



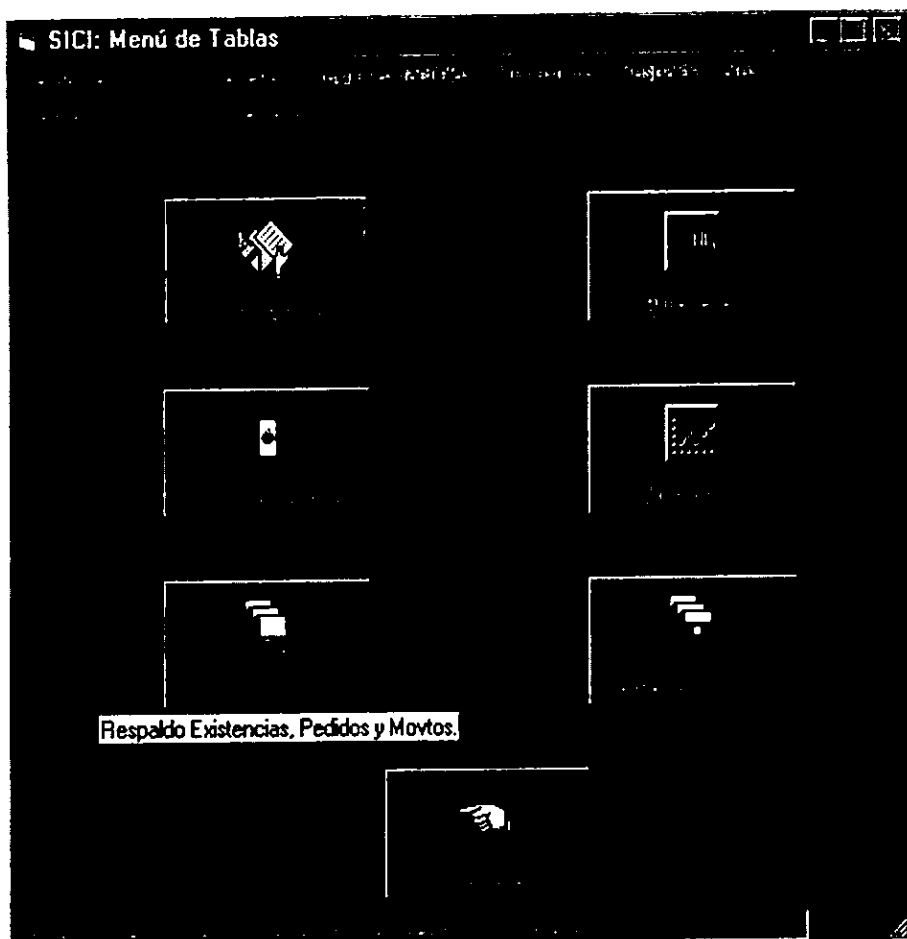
Para la actualización y carga de catálogos nuevos se tiene definido este icono, por el cual se tiene asociada la Pantalla 18.



Para realizar capturas de almacenes se tiene definido con este icono los procesos de: altas, bajas y cambio en los mismos, al seleccionar este icono se habilita la Pantalla 19.

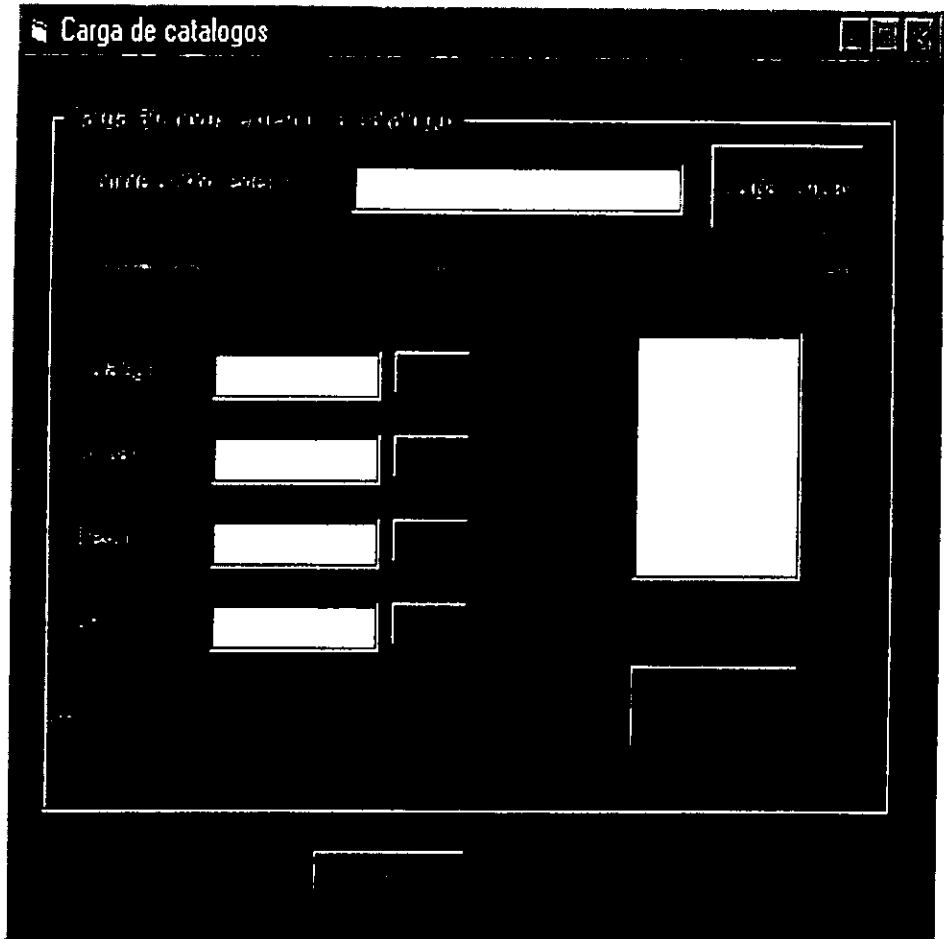


En algunas ocasiones se tiene la posibilidad de poder transferir materiales entre los almacenes, para ello se tiene definido este proceso en este icono, el cual tiene asociada la Pantalla 20.



**Pantalla17. Menú de tablas.**

En esta Pantalla 17 se tiene los procesos de asociados al mantenimiento de información del sistema, tales como: captura de almacenes, carga de catálogos de materiales, almacenes con posibilidades de traspasos y los procesos asociados con respaldo de información, como pedidos, movimientos, existencias, etc.



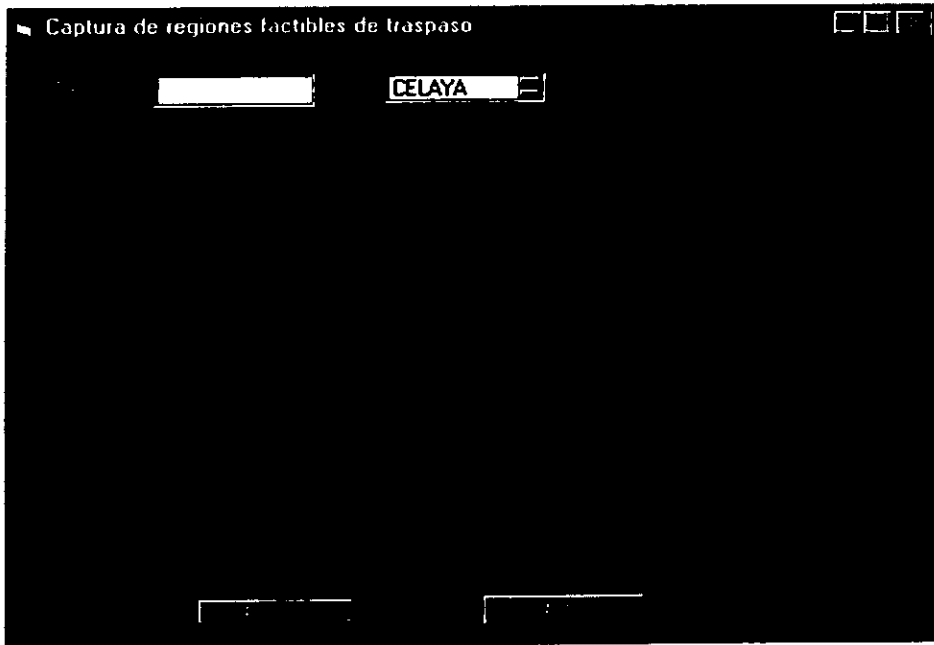
**Pantalla18. Carga de catálogos.**

En esta Pantalla 18 se realiza el proceso de carga de datos en los catálogos de materiales que se utilizan en la empresa, estos son generados por otra área, en la cual se definen los tipos de materiales, agrupación, costo, categoría, etc., y de esta manera se tenga una referencia y control en la compra de los mismos. Esta información es entregada en archivos de formato de dBase XX, y esto implica que debe ser transportada a formato de Access, esto se realiza en esta Pantalla con el proceso carga de dichos archivos a la base de datos del sistema.



Región:	Región asignada.
Categoría:	Categoría asignada (distribuidor, almacén, etc.).
División:	División a la que pertenece.
Grupo de Distribución:	Grupo de distribución asignado.
Subgrupo:	Subgrupo de distribución asignado.

Una vez proporcionados los datos se procede a la selección de la función a realizar, que se tiene definida en los botones de Actualizar, Baja y Cancelar.



**Pantalla20. Captura de regiones factibles de traspaso.**

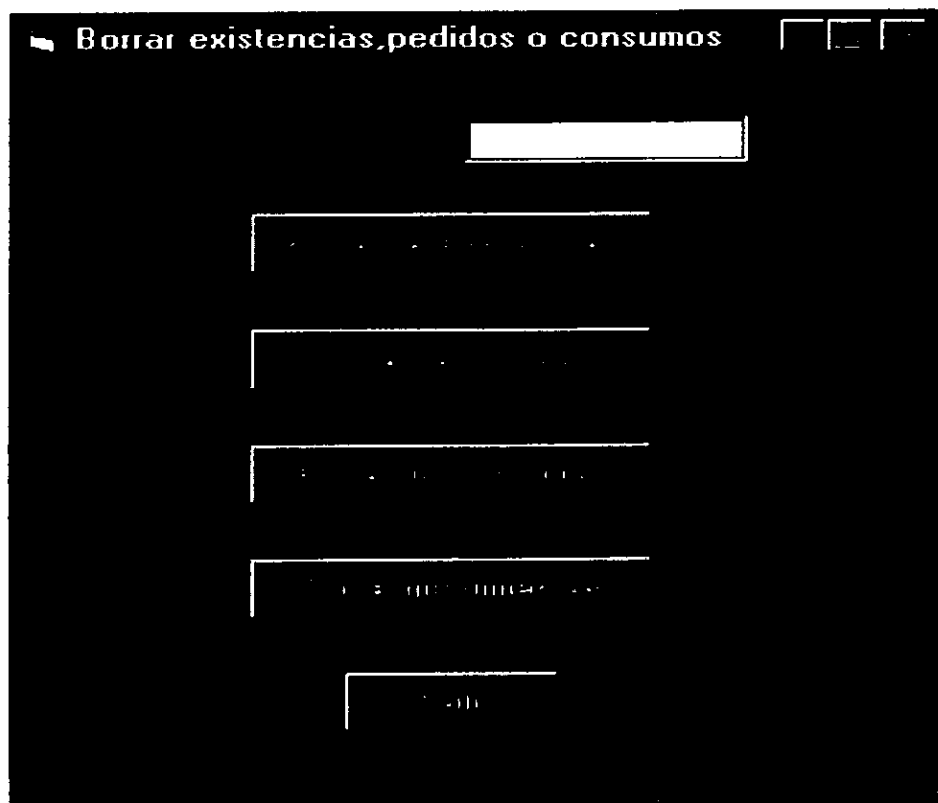
En esta Pantalla 20 se capturan las regiones que un momento dado puedan realizar traspasos de materiales a otras áreas donde se requieran.

## Procedimiento de respaldo de información

Uno de los factores que se deben tener en cuenta en la administración del sistema es conocer los procedimientos de respaldo de información, ya que el volumen de información que maneja es alto y por ende el crecimiento de la base de datos; para ello se tienen definidas las siguientes herramientas en el sistema para realizar esta función, para efectuar estas operaciones se tienen los siguientes procesos que a continuación se da un a breve explicación de los mismos.



En la Pantalla 17 se tiene definido este icono, por medio del cual se gestionarán todos los procesos de respaldo que se tienen definidos como lo indica la Pantalla 21.





Pantalla21. Borra existencias, pedidos o consumos.





---

En esta Pantalla 21 se realizan las funciones de mantenimiento del sistema, tales como respaldar la información de una fecha determinada de los archivos de existencias, pedidos, consumos y movimientos que están en la base de datos del sistema; esto se debe realizar con el fin de liberar espacio en la base, ya que la información que el sistema maneja es mucha y con este proceso se pretende sacar la información que ya no sea relevante para el sistema. Para ello en esta Pantalla se tienen definidos los siguientes botones:

 Con este botón se ejecuta el proceso de respaldo de los registros de la tabla existencias que cumplan con la fecha indicada, una vez respaldados se eliminan de la tabla.

 Con este botón se ejecuta el proceso de respaldo de los registros de la tabla pedidos que cumplan con la fecha indicada, una vez respaldados se eliminan de la tabla.

 Con este botón se ejecuta el proceso de respaldo de los registros de la tabla consumos que cumplan con la fecha indicada, una vez respaldados se eliminan de la tabla.

 Con este botón se ejecuta el proceso de respaldo de los registros de la tabla movimientos que cumplan con la fecha indicada, una vez respaldado se eliminan de la tabla.

## **Conclusiones**

Con el desarrollo de este sistema de control de inventarios, se ha logrado ejercer un alto grado de control administrativo sobre el área de inventarios de la empresa telefónica.

La disponibilidad de la información en línea ha permitido a la alta dirección y a las demás áreas tomar decisiones de compras de materiales que conformaran los inventarios acertada y oportunamente al poder disponer de la información exacta y a tiempo.

El esfuerzo de oficina anteriormente requerido (movilización de varias personas, el manejo manual o en diskettes de la información, etc.) para procesar un gran volumen de transacciones se ha reducido al 90%. Por ejemplo, como resultado los encargados del área de compras pueden dedicar mas tiempo al esfuerzo de la reducción de los costos de adquisiciones.

Es posible analizar un gran volumen de datos que no era factible hacer en ausencia de un sistema de inventarios como el desarrollado.

Con el sistema en línea, el volumen de papeleo desciende considerablemente, lo cual deriva que el costo de la manipulación del inventario y los tramites del mismo, se reduzcan.

Con el desarrollo del sistema, también se logró una interfaz más amigable hombre-máquina, la cual se obtuvo mediante la creación de menús por medio de iconos que proporciona Visual Basic. Esto hace posible que personal no especializado maneje el sistema, dedicando a los especialistas a realizar mayor análisis y desarrollo de propuestas.

El sistema es confiable y seguro, ya que cuenta con niveles de autorización, en el cual

solo se permite la manipulación del operador o el administrador del sistema, dichos niveles establecen que tipo de información podrá manipular cada uno.

Dentro de la seguridad ofrecida por el sistema se cuenta con una bitácora en la cual queda registrado el tipo de movimiento que haya hecho el usuario (bajas o cambios), la hora y el tipo de transacción que este llevo a cabo; ayudando con esto a tener certidumbre de quién realiza qué operaciones.

El desarrollo del sistema de control de inventarios proporciona también procedimientos eficientes de compras, recepción y embarques

Mantiene en existencia el inventario suficiente, para evitar situaciones de falta de productos, los que conducen a pérdidas de productividad en las empresas.

Resuelve el problema de tener en almacén un inventario demasiado grande, evitándose de esta forma el gasto de inmovilizar dinero en artículos que no se necesitan.

Al darse la solución de llevar a cabo la implementación del sistema teniendo como manejador de bases de datos a Access 97 y Visual Basic 5.0, la empresa no tuvo que hacer gastos de compra de licencias de las mismas, dado que ya cuentan con las mismas.

La capacitación, se reducirá a un mínimo, dado que el sistema es bastante amigable y seguro.

El sistema tiene la flexibilidad en la generación de reportes con base a los requerimientos específicos de las diferentes áreas.

La implementación se realizó en una plataforma totalmente gráfica (Windows 9X esto representa una gran ventaja ya que en la actualidad es una de las aplicaciones mas

utilizada en el desarrollo de sistemas.

Esta tesis se desarrolló por medio del programa de Apoyo a la Titulación del Palacio de Minería, el cual nos pareció un excelente medio para elaborar el trabajo, ya que cuenta con personal académico altamente competitivo; así mismo proporciona acceso a la biblioteca del Palacio de Minería dando mayor posibilidad de consulta de los temas desarrollados.

## **Bibliografía**

### **Administración de la Producción**

L. Tawfik, A.M. Chauvel

1ª. Edición

Mc. Graw Hill

### **Administración de Operaciones**

Joseph G. Monks

1ª edición

Mc. Graw Hill

### **Sistemas de Producción Basados en Computadoras**

A.K. Kochhar

3ª. Edición

C.E.C.S.A.

### **Enciclopedia de Visual Basic V.5.**

Francisco Javier Ceballos

Raima

### **Introducción a los Sistemas de Bases de Datos**

C. J. Date

Addison-Wesley

### **Diseño de Bases de Datos**

Wiederhold

Mc. Graw Hill

Manual imprescindible de Access 97

Julian Casas Luengo

Anaya Multimedia

3ª Edición

Energía de software

Roger S. Pressman

Mc. Graw Hill

3ª edición

Fundamentos de programación

Luis Jayanes Aguilar

Mc. Graw Hill

Aprendiendo Unix

Jamey Graner

Prentice Hall

Metodologías de desarrollo

Antonio López F.

Ra- Ma

Fundamentos de Bases de datos

Henry F. Korth, Abraham Silberschatz

Mc. Graw Hill

Windows NT

Sheldon

Mc. Graw Hill

Revista PC-Magazine

Guerra de bases de datos

Contabilidad

Horngren & Harrison

Prentice Hall

# APENDICE A

## Conceptos básicos de Contabilidad Financiera

### Origen y Definición

En el complejo mundo de los negocios en que vivimos, consecuencia de las técnicas modernas, de nuevas ideologías, del incremento de la población, de cambios sociológicos provocados por las últimas guerras y de otras circunstancias, tales como los problemas del petróleo, controles gubernamentales, alianzas estratégicas, el hombre de negocios tiene muchos y variados problemas que le obligan a superarse mediante el estudio y mejor conocimiento de los elementos financieros de las empresas para tratar de mejorar sus rendimientos.

Para el manejo de las empresas, el hombre de negocios necesita en su dinámico y cambiante mundo, de información financiera oportuna y adecuada, entendiéndose por tal la información financiera cuantitativa, confiable y accesible para que le ayude a fortalecer su juicio, en la toma de decisiones sobre algún asunto financiero o de carácter general.

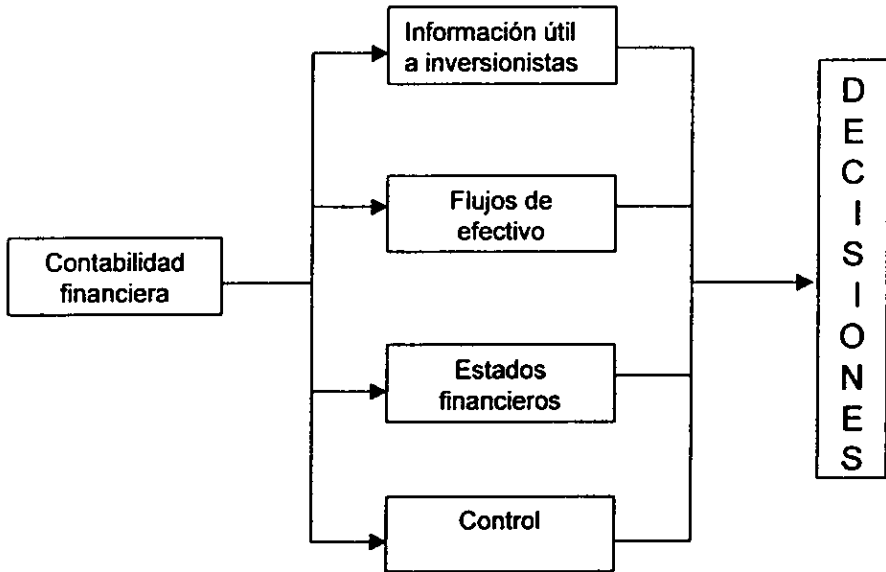
Para obtener dicha información es necesario contar con un sistema de captación de las operaciones con el que mida, clasifique, registre y resuma con claridad en términos de dinero, las transacciones y hechos, para nuestro caso de estudio serán de carácter financiero. De esta manera, el área de administración contará con una fuente de información que le permita:

- a) Coordinar las actividades.
- b) Captar, medir, planear y controlar las operaciones diarias.
- c) Estudiar las fases del negocio y proyectos específicos.
- d) Contar con sistema de información central que pueda servir a los interesados en la empresa, como serían los propietarios, acreedores, gobierno, empleados, posibles inversionistas o público en general.

Definición. Podemos decir que la contabilidad financiera es un medio a través del cual los diversos interesados pueden medir, evaluar y seguir el progreso, estancamiento o retroceso en la situación financiera de las entidades, derivado de la lectura de los datos económicos y financieros, de naturaleza cuantitativa, recolectados, transformados y resumidos en los informes denominados estados financieros, los cuales, en función de su utilidad y confiabilidad, permitirán tomar decisiones relacionadas con las empresas. Desde el punto de vista técnico se dice que. La contabilidad financiera es una técnica que se utiliza para producir sistemáticamente y estructuradamente información cuantitativa de las transacciones



que realiza una entidad económica y de ciertos eventos económicos identificables y cuantificables que la afectan, con objeto de facilitar a los diversos interesados el tomar decisiones en relación con dicha entidad económica. Asimismo, podemos decir que la contabilidad financiera también es utilizada como medio e instrumento de control, para informar sobre el correcto uso de los bienes y recursos de la empresa en la actividades de la misma, figura A1. Finalidades de la contabilidad financiera



**Figura A.1. Finalidades de la contabilidad financiera.**

Es necesario destacar que en la información financiera intervienen singularmente la habilidad y la honestidad de quien la prepara, así como los principios de contabilidad aplicados sobre bases consistentes.

### **Importancia y Utilidad**

La finalidad de la contabilidad financiera y, por ende de la información es proporcionar información que siendo útil, confiable y comprensible, sirva de base para la toma de decisiones. Además del anterior fin o propósito principal, la contabilidad financiera comprende los siguientes tres objetivos generales:

1. Proporcionar información útil para los futuros inversionistas y acreedores, y para otros usuarios que han de tomar decisiones racionales de inversión y de crédito.

2. Preparar información que ayude a los usuarios a determinar los montos, la oportunidad y la incertidumbre de los proyectos de entrada de efectivo asociados con la realización de inversiones dentro de la empresa.
3. Informar acerca de los recursos económicos de una empresa, los derechos sobre éstos y los efectos de las transacciones y acontecimientos que cambien esos recursos y los derechos sobre aquéllos.

Las características fundamentales de la información financiera son las siguientes:

- Utilidad
- Confiabilidad
- Provisionalidad

Cada una de estas características, deben cumplir con una serie de elementos o requisitos para ser válida y cumplir con su función dentro de la información financiera en su conjunto, se puede decir además que estas características no, son excluyentes, por el contrario, deben ser cumplidas todas ellas en todos los casos, en la medida de las posibilidades, dentro de lo que los contadores llaman razonablemente correcto.

La utilidad como característica financiera es definida como la cualidad de adecuarse al propósito del usuario. Esto es evidente si partimos del hecho de que la contabilidad fue creada o elaborada por el hombre para satisfacer sus necesidades de información y sabemos que un satisfactor es útil sólo cuando cumple con las necesidades. La utilidad de la información esta en función de su contenido informativo y oportunidad como se muestra en la figura A.2. Elementos de la utilidad como características de la información financiera.

**Significado de la información.** Representa su cualidad o capacidad de mostrar mediante palabras y cantidades, la entidad y su evolución, su estado presente y en diferentes puntos del tiempo, sus resultados de operación y demás datos pertinentes y necesarios.

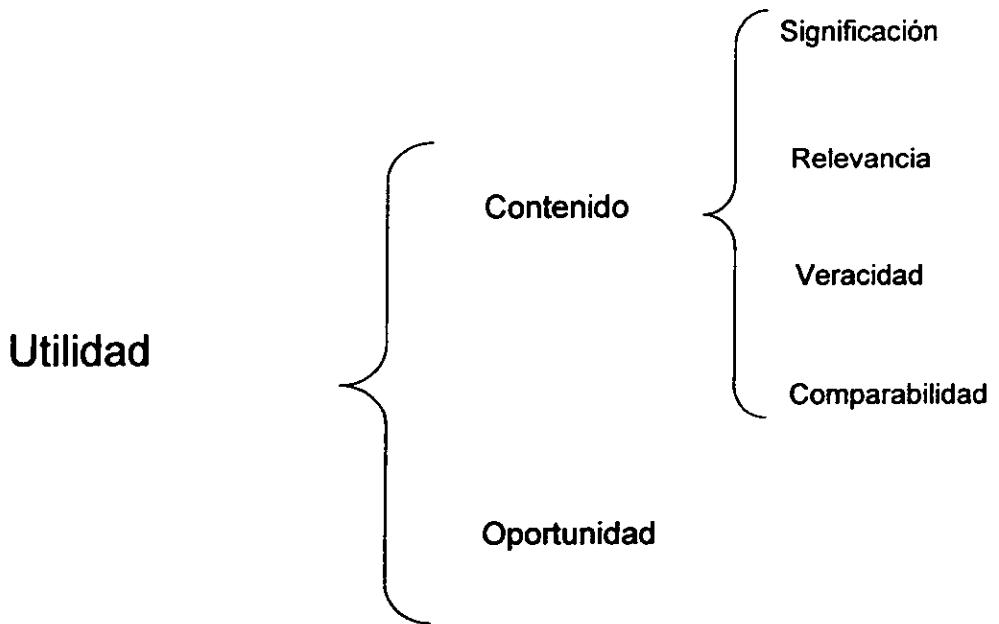
**Relevancia de la información.** Consiste en seleccionar los elementos de la misma que permitan al usuario captar el mensaje y operar con base en ella para lograr sus fines particulares.

**Veracidad de la información.** Es una de las cualidades esenciales de la información financiera, esto implica el hecho de presentar eventos, transacciones y operaciones reales, correctamente valuadas y presentadas a la luz de las herramientas de medición aceptados como válidas por el sistema.

**Comparabilidad de la información.** Es la cualidad de la información de ser válidamente comparable en

los diferentes puntos del tiempo de la entidad y de ser válidamente comparables dos o más entidades entre sí.

**Oportunidad.** Se refiere a que la información llegue a tiempo a manos del usuario, para que puedan usarla y tomar decisiones a tiempo, y de esta manera lograr sus objetivos, fines y metas financieras en la entidad.



**Figura A.2. Elementos de la utilidad como características de la información financiera.**

Asimismo, podemos decir que la contabilidad financiera también es utilizada como medio e instrumento de control interno, para informar sobre el correcto uso de los bienes y recursos de la empresa en las actividades de la misma.

#### **Principios de contabilidad Generalmente aceptados**

Los principios de contabilidad generalmente aceptados pueden ser definidos como las reglas o normas desarrolladas por la técnica contable, que se observan para registrar las operaciones y presentación de estados financieros; es decir, la presentación de la situación financiera y de los resultados de las operaciones hechas a través de los estados financieros no es una presentación arbitraria sino basada en

la experiencia, el desarrollo de la técnica contable y la aceptación general.

Los estados financieros deben explicar bajo qué reglas particulares han sido preparados, para que las personas interesadas en los estados financieros puedan aplicar las técnicas de análisis y comparaciones con otros estados financieros y se formen un juicio respecto a la situación financiera de la empresa; en otras palabras, puede decirse que los llamados principios de contabilidad forman el lenguaje común de la técnica contable.

La técnica contable ha establecido tres clases de conceptos que son: Principios, reglas particulares y criterio prudencial de la aplicación de las reglas particulares.

Los principios se dividen en tres grupos: los que identifican y delimitan el ente económico y sus aspectos financieros, y los que establecen la base para cuantificar las operaciones del ente económico y su presentación, y los que establecen los requisitos cualitativos de la información financiera.

- Los principios de contabilidad que identifican y delimitan el ente económico son:
  - a) **Unidad o entidad.** Se acepta que una empresa tiene vida y operación diferentes de las personas o socios que la forman. Es decir, tiene personalidad jurídica propia y totalmente separada.
  - b) **Realización.** Se presume que los costos y gastos se registran paralelamente a los ingresos que lo originaron.
  - c) **Lapsos o periodos contables.** Se presume que las operaciones de las empresas se dividen en lapsos o ejercicios y al terminar cada uno se efectúa un cierre de las operaciones.
  
- Los principios de contabilidad que establecen la base para cuantificar las operaciones del ente económico y su presentación son:
  - a) **Base de costo o valor histórico.** El poder adquisitivo de la moneda, en su constante fluctuación, tiene importancia secundaria para el registro de las operaciones, siendo de mayor importancia el valor nominal empleado en las operaciones; es decir, las operaciones deben ser registradas al valor de los activos erogados con el fin de unificar diferentes estimaciones de valor y ofrecer una evidencia documental del costo.
  - b) **Continuidad o negocio en marcha.** Bajo esta presunción se establece que una empresa tiene existencia continua y no limitada a un lapso determinado. De no ser así, todos los activos tendrían que presentarse a su valor de realización.
  - c) **Dualidad económica.** La doble dimensión de la estructura contable de la unidad económica es básica para la comprensión y su realización con otras entidades. La dualidad se constituye, por una parte, con los recursos con que cuenta el ente económico para la realización de sus fines y, por la otra, con las fuentes de esos recursos; es decir, los hechos y fenómenos económico-

financieros que afectan el patrimonio de la entidad y al registrarse se aprecia su efecto en la estructura financiera, figura A.3. Dualidad económica

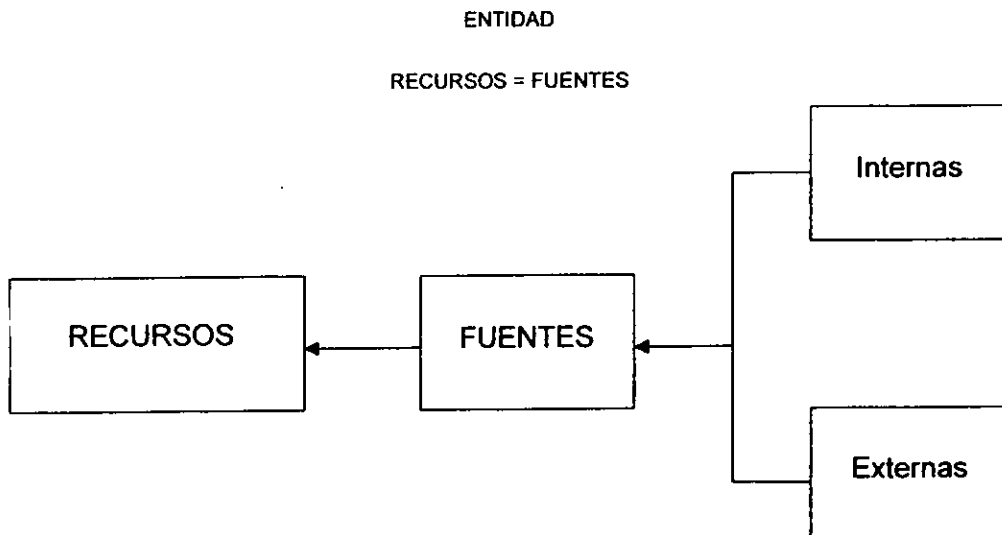


Figura A.3. Dualidad económica.

- Los principios de contabilidad que establecen los requisitos cualitativos de la información financiera son:
  - a) Revelación suficiente. Los estados financieros deben contener en forma clara y comprensible la información necesaria para que el lector pueda juzgar la situación financiera y el resultado de las operaciones.
  - b) Sustancialidad o importancia relativa. La información que contienen los estados financieros debe mostrar aspectos importantes susceptibles de ser cuantificados en unidades monetarias. El no detallar en los estados financieros partidas relativamente insignificantes, hace que la atención del lector se fije en los conceptos fundamentales. Las partidas de reducida importancia relativa con respecto a los totales que se presentan, deben ser agrupadas y presentarse globalmente a menos que la presentación de tales partidas menores sea esencial para alguna finalidad específica. Deben agruparse partidas homogéneas que conduzcan a totales y subtotales cuyo dato sea de utilidad, sin mezclar partidas de naturaleza diferente entre sí.
  - c) Consistencia. La información que se presenta en los estados financieros requiere que se sigan procedimientos de calificación aceptados en estados anteriores o precedentes, con objeto de que en dichos procedimientos no conduzca a conclusiones erróneas, cuando sea necesario adoptar

un cambio que afecte la comparabilidad de los resultados, es necesario indicar el cambio efectuado en los estados financieros y su repercusión en las cifras contables. Igualmente, esto es aplicable a la agrupación y presentación de información.

- Las reglas particulares son las aplicables específicamente a los diferentes conceptos que integran los estados financieros; se dividen en reglas de valuación y reglas de presentación.

Las reglas de valuación son aplicables principalmente a los estados financieros estáticos que muestran la imagen de una situación en una fecha determinada, tal como el estado de situación financiera, y señalan que es conveniente indicar en las partidas de mayor importancia las bases de valuación adoptadas. Se ha venido acostumbrando a hacer tal indicación en los renglones de inversiones en valores, inventarios, activos fijos y en los pasivos de importancia apreciable.

Las reglas de presentación están conformadas por los siguientes elementos:

- a) Terminología. Aun cuando se presume que los lectores de los estados financieros tienen la presentación básica para comprenderlos, el empleo de la terminología técnica debe ser especialmente vigilado con objeto de que su lectura no conduzca a interpretaciones erróneas y muestra con claridad la naturaleza del concepto que se indica.
- b) Extensión. La extensión de un estado financiero debe estar condicionada a un tamaño convencional de lectura, sin condensar conceptos que convengan presentar en forma individual ni presentar análisis que puedan resultar de una extensión tal que la atención del lector se pierda. En el caso de tener la necesidad de presentar análisis o detalles importantes o significativos y que estos ocupen gran parte del estado, se recomienda en algunos casos mostrarlos por separado para resumir en lo posible la extensión de los estados.
- c) Completa información. Los estados financieros deberán contener toda la información necesaria para que sean comprendidos razonablemente por los lectores. Cuando exista algún hecho importante para la debida interpretación de los estados, es conveniente describirlo mediante el uso de indicaciones entre paréntesis o en notas al pie o adjuntas a los estados. Esta información no se limita a la fecha o periodo al que se refieren los estados financieros, sino también a los hechos o sucesos que se conozcan durante el periodo comprendido entre la fecha referente a los estados financieros y la fecha de su emisión.

Las reglas particulares serán tratadas específicamente en cada uno de los conceptos que integran los estados financieros. Sin embargo, conviene anticipar que éstos deben reunir ciertas características generales que son:

- a) Satisfacer necesidades generales de información.

- b) Ser razonables y prácticos en su aplicación.
- c) Originar los resultados equitativos y comprensibles.
- d) Poder aplicarlos en diversas circunstancias.
- e) Producir resultados comparables, de periodo a periodo y también entre la información de diferentes empresas.
- f) Tener adaptabilidad consistente.

### **Registro de operaciones**

Sabemos que la contabilidad financiera es una técnica por medio de la cual las operaciones realizadas por la entidad se convierten en información cuantitativa, misma que se expresa en unidades monetarias y se utiliza en la toma de decisiones, para que esta información pueda ser elaborada, es necesario que las operaciones celebradas por las empresas se registren debidamente, para lograr este objetivo, se han establecido una serie de principios, éstos son:

- Los principios de realización y periodo contable, que nos permiten responder a las preguntas de por qué y cuándo se registran las operaciones, siendo la respuesta que una transacción se debe registrar porque se ha realizado y en el momento en que se celebra.
- El principio del valor histórico original, que reglamenta la base del precio de costo o sus modificaciones para la asignación de unidades monetarias.
- El principio de la dualidad económica, que nos permite tener presente que las transacciones celebradas por las entidades siempre deben mantener el aspecto dual de los recursos y sus fuentes, es decir, del activo y las suma del pasivo y capital contable.

Desde el punto de vista práctico, la contabilidad financiera ha evolucionado también en cuanto a la manera de registrar las operaciones realizadas; esta evolución de los sistemas o procedimientos de registro, ha pasado de los inicialmente llevados de forma manual, a los mecánicos y electromecánicos y finalmente a los electrónicos mediante el empleo de computadoras.

Para el establecimiento de un método de registro, debemos tener en cuenta que estos métodos son parte integrante del sistema de información de la entidad.

Los métodos manuales para el registro de operaciones, como su nombre lo indica, son aquellos que se llevan a cabo de registros efectuados a mano. Existe una variedad de registros manuales, mismo que varían de acuerdo con las necesidades de información de cada entidad particular, si bien es cierto que

algunos de ellos dada la propia evolución han quedado en desuso es pertinente comentarlos. Dentro de los métodos de registro y procesamiento de operaciones manuales por mencionarlos tenemos:

- ◆ Diario Continental
- ◆ Tabular
- ◆ Centralizado
- ◆ Pólizas

Los métodos mecánicos y electromecánicos. Podemos decir son métodos de registro directo y simultáneo que agilizan el proceso de registro de las operaciones y la presentación de la información mediante el empleo de máquinas (mecánicas o electromecánicas), estos métodos fueron superados sorprendentemente por los cambios tecnológicos y sin duda por el creciente número de operaciones realizadas por las empresas, son los factores que hacen que estos métodos no sean tan veloces como se desea, asimismo, el empleo de la electrónica en los métodos de registro y el surgimiento de las generaciones de computadoras hicieron que la tecnología dejara a estos métodos prácticamente en la obsolescencia. En la actualidad los métodos electrónicos que usan computadoras, son los más utilizados.

Existe una serie de disposiciones legales que obligan a las entidades a llevar adecuados sistemas de contabilidad, y estas son:

- a) Código Fiscal de la Federación (CFF) y su reglamento (RCFF).
- b) La Ley del Impuesto Sobre la Renta (ISR).
- c) La Ley del Impuesto al Valor Agregado (IVA).
- d) El Código de Comercio (CC).

### **Ecuación básica de la contabilidad**

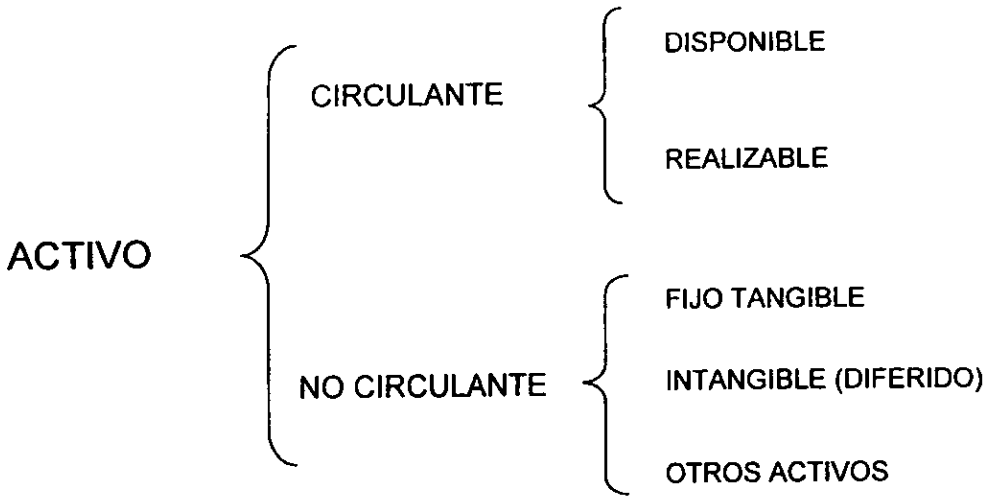
#### **Definición y clasificación del activo**

El activo puede definirse como el conjunto de bienes y derechos reales y personales sobre los que se tiene propiedad, así como cualquier costo o gasto incurrido con anterioridad a la fecha del estado de situación financiera, que debe ser aplicado a ingresos futuros.

Existen principalmente dos formas para clasificar el activo; la primera lo clasifica en tres grupos principales: circulante, fijo y cargos diferidos. La segunda forma reconoce dos grupos únicamente: el activo circulante y el no circulante; el grupo de cargos diferidos quedará incluido en el activo circulante o



activo no circulante, según la intervención directa de las partidas que lo forma en el ciclo financiero a corto plazo o en el ciclo financiero a largo plazo, figura A.4. Clasificación del activo



**Figura A.4. Clasificación del activo.**

El ciclo financiero a corto plazo puede definirse como el tiempo promedio que transcurre entre la adquisición de materiales y servicios, su transformación, su venta y finalmente su recuperación en efectivo. Los activos que intervienen en forma directa en este ciclo financiero son los que se agrupan como activos circulantes.

El ciclo financiero a largo plazo es el que contiene las inversiones de carácter permanente que se efectúan para realizar el objetivo de la empresa. El costo de tales inversiones va interviniendo gradualmente en el ciclo financiero a corto plazo a través de la depreciación, amortización y agotamiento. Estas inversiones son principalmente las que constituyen los activos no circulantes.

Es importante considerar que la base fundamental para hacer la distinción entre circulantes y no circulantes, es principalmente el propósito con que se efectúa la inversión, es decir, si está se hace con una finalidad permanente o no.

Como ejemplo puede citarse la compra de bonos o acciones cotizados en la bolsa con el propósito de inversión temporal y no tener efectivo ocioso. Esta inversión deberá ser considerada como un activo circulante ya que se transformará en efectivo en el transcurso del ciclo financiero a corto plazo. Si por el contrario, la compra se hace con el propósito de inversión permanente, deberá considerarse como un

activo no circulante.

Siguiendo este principio, las diferentes partidas del activo circulante y del no circulante se presentan en el orden de su probable convertibilidad en efectivo. Así como el activo circulante se presenta en el siguiente orden:

- Circulante
  - ◆ Efectivo
  - ◆ Inversiones temporales de inmediata realización
  - ◆ Cuentas de documentos por cobrar
  - ◆ Inventarios
  - ◆ Pagos anticipados

En forma similar, el activo no circulante se acostumbra presentarlo en el orden probable de su liquidez, como sigue:

- No circulante
  - ◆ Cuentas y documentos por cobrar.
  - ◆ Inversiones en valores.
  - ◆ Activo fijo tangible (terrenos, edificios, maquinaria y equipo, etc.).
  - ◆ Activos fijos intangibles (patentes, marcas, crédito mercantil).
  - ◆ Cargos diferidos a largo plazo (gastos de organización, campañas de publicidad, etc.).

Existen ciertos tipos de activos que suscitan la controversia de agruparlos como circulantes o no circulantes. Tal es el caso de los inventarios de las empresas tabacaleras o vinícolas, los cuales generalmente representan una inversión a largo plazo, no obstante que se adquieren con el propósito de venderlos y, por tanto, son inversiones típicamente circulantes.

En vista de la dificultad práctica de determinar el ciclo financiero a corto plazo para los diferentes tipos de empresas, se ha adoptado el término de un año para determinar la clasificación de ciertos activos, por lo cual se acostumbra presentar como activos no circulantes aquellos que aun cuando forman parte del ciclo financiero a corto plazo, representan una inversión cuya convertibilidad en efectivo tomará más de un año. Se hace hincapié en que la base fundamental para hacer la distinción entre circulantes o no circulantes es principalmente el propósito con que se efectúa la inversión, como se señaló con anterioridad.

- Activos Circulantes

Efectivo en caja. Su enunciado en el estado de situación financiera debe indicar la forma en que se está representando, ya sea únicamente en caja, en bancos o en ambos. Por lo general no interesa analizar la forma de inversión sino únicamente enunciarlo en una sola partida. En caso de que existan monedas extranjeras en cantidades importantes, conviene indicar el monto en dicha moneda.

El efectivo mostrado como circulante debe estar sujeto a libre disposición y sin restricciones. En caso de que el efectivo o parte de él esté destinado específicamente a operaciones diferentes a las del ciclo financiero a corto plazo, no debe incluirse como circulante

Los inventarios representan inversiones destinadas a la venta o a la producción para su venta posterior. Así, por ejemplo, materias primas, empaques, manufacturas en proceso, productos terminados, refacciones y materiales indirectos de producción que se consuman dentro del ciclo normal de operaciones.

La regla de valuación de los inventarios es el costo de adquisición o de producción cuando éste sea menor que en el mercado, entendiéndose este último como costo de reposición.

El costo puede determinarse de acuerdo con el sistema y método que cada empresa en particular seleccione tomando en cuenta sus características, y puesto que éstas pueden influir considerablemente en la inversión de los inventarios y en el resultado de las operaciones, debe actuarse con juicio profesional. Los sistemas y métodos deben ser aplicados en forma consistente a menos que se presenten cambios en las condiciones originales, en cuyo caso deberá darse a conocer en los estados financieros.

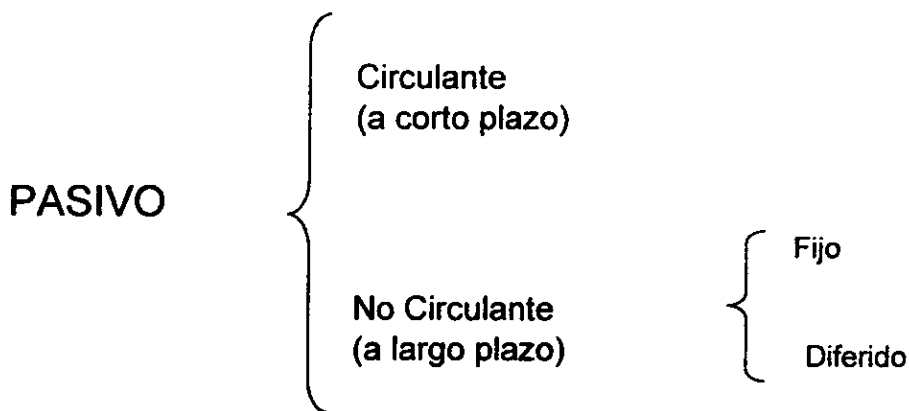
Pueden usarse los sistemas de valuación de costo absorbente o costo directo o marginal, y éstos a su vez puedan manejarse sobre la base de costo histórico y predeterminado, siempre y cuando este último sea semejante al costo histórico bajo condiciones normales de fabricación.

Los inventarios pueden valuarse conforme a los métodos de costo identificado, costo promedio, primeras entradas primeras salidas (PEPS), últimas-entradas primeras salidas (UEPS) y detallistas. Los inventarios que sean obsoletos, dañados o de lento movimiento deben valorizarse a su valor de realización.

### **Definición y clasificación del pasivo**

El pasivo puede definirse como el grupo de obligaciones jurídicas por las cuales el deudor se obliga con el acreedor a pagar con bienes, dinero o servicios. Comprende obligaciones presentes que provengan de operaciones o transacciones pasadas.

Desde el punto de vista de su presentación en el estado de situación financiera, el pasivo debe dividirse en circulante o flotante y no circulante o consolidado. La distinción fundamental entre ellos es el tiempo en el cual deben ser cubiertos. Si la liquidación se produce dentro de un año o en el ciclo normal de operaciones a corto plazo, ha de considerarse como circulante. Si es mayor de dicho lapso, debe clasificarse como no circulante. En caso de clasificar el pasivo circulante con base en el ciclo financiero a corto plazo, deberá revelarse este hecho en los estados financieros mediante una nota. Esta clasificación tiene importancia para apreciar la relación con los activos circulantes o no circulantes y, por consiguiente, la capacidad de pago de la empresa, figura A.5. Clasificación del pasivo



**Figura A.5. Clasificación del pasivo.**

El pasivo debe valorizarse de acuerdo con su monto nominal de restitución o pago, y los pasivos expresados en moneda extranjera, con su conversión al tipo de cambio vigente.

Los pasivos circulantes proceden generalmente de las operaciones de la empresa a corto plazo, tales como compras de mercancías, préstamos para cubrirse a corto plazo, gastos e impuestos acumulados por pagar.

Los conceptos que normalmente forman este grupo son: documentos por pagar a bancos, a proveedores, a otros; cuentas por pagar a proveedores, a otros; deuda a largo plazo con vencimiento a un año; anticipos de clientes; compañías afiliadas; pasivos estimados. Como ejemplos de pasivos estimados o acumulados pueden darse: sueldos, gratificaciones, comisiones, regalías, intereses, participación de utilidades a los empleados, rentas, servicios profesionales, primas de seguros, impuestos.

El pasivo consolidado o a largo plazo se establece por lo general para efectuar inversiones de carácter permanente. Por esta causa el pago se establece en relación con la capacidad de generación de fondos que origina la depreciación o amortización del activo tangible no circulante. Dentro de este concepto encontramos hipotecas, emisión de obligaciones, préstamos refaccionarios, de habilitación o de avío y, excepcionalmente, créditos recibidos de proveedores.

El pasivo contingente lo constituyen las obligaciones que están sujetas a la realización de un hecho, por el cual desaparecerán o se convertirán en pasivos reales, por ejemplo: juicios, reclamaciones de terceros en relación con productos que se les vendieron, garantías, avales, costo de planes de pensiones, jubilaciones, indemnizaciones por separación.

Existen contingencias que no es necesario que se hagan constar en los estados financieros que son inherentes a las operaciones de los negocios y que afecta a muchas empresas, sin no es que a todas, como las posibilidades de una guerra, de una huelga, de una expropiación, de una recesión de los negocios, el cierre de una empresa.

#### **Definición de capital contable**

El término capital contable designa la diferencia que resulta entre el activo y pasivo de una empresa. Refleja la inversión de los propietarios en la entidad y consiste generalmente en sus aportaciones, más sus utilidades retenidas o pérdidas acumuladas, más otros tipos de superávit, como pueden ser las donaciones y la actualización del capital.

El capital contable esta integrado por los siguientes conceptos:

- Capital social
- Otras aportaciones de los socios o accionistas
- Utilidades retenidas
  1. Aplicadas a reservas
  2. Pendientes de aplicar
- Actualización del capital
- Aportaciones no reembolsables hechas por terceros – donaciones
- Utilidad o pérdida del ejercicio

**Capital social.** Es el conjunto de aportaciones de los socios, considerados en la escritura constitutiva o en sus reformas.

El capital social puede estar a su vez integrado como sigue:

1. **Capital autorizado y no emitido:** Lo constituye la diferencia entre el capital de la sociedad autorizada en escrituras y la cantidad que se ha expuesto en suscripción.
2. **Capital emitido no suscrito:** Lo representa aquella parte del capital emitido en las actas de asamblea de accionistas y pendientes de suscribir.
3. **Capital suscrito:** Representa la parte del capital emitido por los socios o accionistas que se comprometen a exhibir.
4. **Capital suscrito y no emitido:** Representa la parte del capital suscrito por los socios o accionistas del cual se encuentra pendiente de recibir la exhibición correspondiente y deberá presentarse en el balance disminuyendo el capital suscrito.
5. **Capital exhibido:** Representa la cantidad que los socios o accionistas han exhibido o aportado efectivamente.

En el capital social pueden existir derechos patrimoniales de socios con diferentes características; por ejemplo: diferencias entre capitales comanditados y comanditarios, en la que los socios comanditarios respondan de las obligaciones sociales sólo hasta una cantidad previamente determinada, o bien, el uso de acciones preferentes de voto limitado.

**Otras aportaciones de los socios o accionistas.** La diferencia entre estas aportaciones y las que hicieron para formar el capital social es únicamente una diferencia jurídica; es decir, las unas se formalizaron a través del acta constitutiva de la sociedad o sus reformas y las otras no necesariamente tienen esta formalidad.

**Utilidades retenidas.** Las utilidades obtenidas por la empresa que no se han capitalizado o distribuido a los accionistas son conservadas en la empresa y, por tanto, forman parte del capital contable, formándose dos grupos:

1. **Utilidades retenidas aplicadas a reservas de manera específica.** Constituidas por disposición de la ley, por disposición de la escritura constitutiva de la empresa o por acuerdos de asambleas de accionistas.
2. **Utilidades retenidas pendientes de aplicar.** Es la parte de las utilidades retenidas que no ha sido aplicada para un fin específico y consiguientemente se encuentra libre. Esta partida debe ser

claramente descrita en los estados financieros.

**Aportaciones no reembolsables hechas por terceros – donaciones.** Cuando se reciben aportaciones no reembolsables de personas diferentes de los socios o accionistas, deberán formar parte del capital contable. En los estados financieros, por la necesidad de una revelación clara y completa, es indispensable que dichas donaciones se presenten como un renglón separado, haciendo los análisis y detalles que las circunstancias dicten.

**Utilidad o pérdida del ejercicio.** La utilidad o pérdida del ejercicio debe considerarse como una parte de las utilidades retenidas pendientes de aplicar. Dado el valor informativo de esta cifra y la conexión con el estado de ingresos y gastos con el estado de la situación financiera, es necesario que la utilidad o pérdida del ejercicio o periodo sea presentada por separado.

## Estados financieros

### Objetivo y utilidad

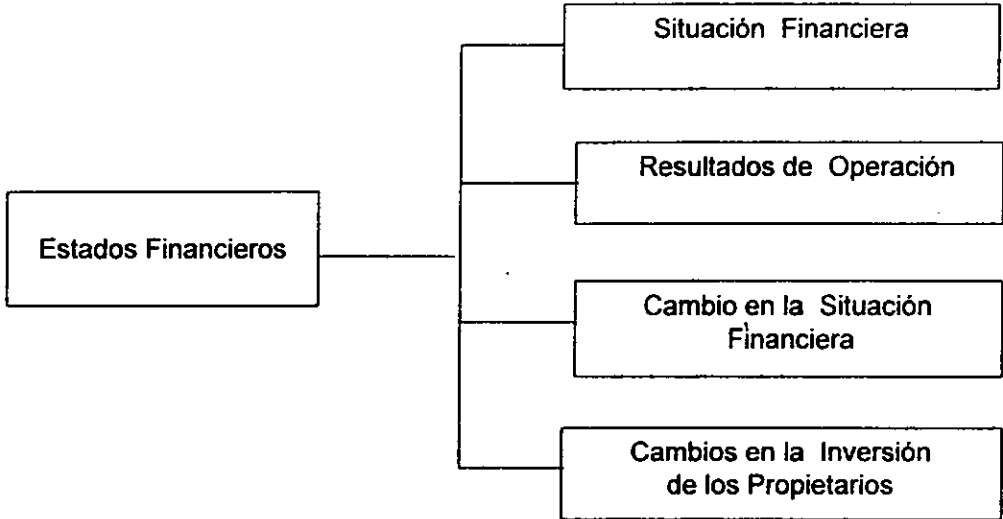
El objetivo que deben cumplir los estados financieros es proporcionar información sobre la situación financiera de la empresa en una fecha determinada y sobre los resultados de sus operaciones y el flujo de fondos por un periodo determinado.

Puesto que los estados financieros son un medio de transmitir información financiera, y no un fin en sí mismos, y que la información contenida en ellos es de interés no tan sólo para la gerencia, los propietarios, los trabajadores, el fisco, los acreedores e inversionistas, y el público en general que se interesa en la entidad, los estados financieros preparados para estos fines, deben procurar o ser capaces de que la información contenida en ellos sea satisfactoria para el usuario general, figura A.6. Objetivos de los estados financieros.

Esta información puede estar constituida de la siguiente manera:

- a) **El nivel de rentabilidad;** es decir: "tomar decisiones de inversión y crédito" y con ello tener un factor de medición respecto a las características de capacidad de crecimiento y estabilidad de la empresa, así como redituabilidad, para asegurar la inversión, la obtención de un rendimiento y la recuperación de esta inversión.
- b) **La posición financiera,** que incluye su solvencia, liquidez, capacidad para generar recursos.
- c) **La capacidad financiera de crecimiento,** es decir la evaluación del origen y las características de los recursos financieros de la empresa, así como el rendimiento de los mismos.

- d) **El flujo de fondos**, es decir formarse un juicio de cómo se ha manejado la empresa y evaluar la gestión de la administración, a través de una evaluación global de la forma en que ésta maneja la rentabilidad, solvencia y capacidad de crecimiento de la empresa.



**Figura A.6. Objetivos de los estados financieros.**

- e) **La habilidad para obtener utilidades**, es decir, la capacidad para obtener ingresos derivados de las operaciones de la entidad, los gastos efectuados para obtenerlos (uso de recursos).
- f) **Los cambios experimentados en las cuentas del capital contable**, es decir, los movimientos que ha experimentado la inversión de los propietarios.

Como resultado de lo anterior se derivan, algunos de los diferentes estados financieros que existen:

1. Estado de posición financiera o balance general.
2. Estado de resultados o de pérdidas y ganancias.
3. Estado de cambios en la situación financiera.
4. Estado de variaciones en el capital contable.
5. Notas a los estados financieros.

**La utilidad de los estados financieros** es sin duda uno de los principales elementos de la contabilidad financiera, que sirven de base para sustentar argumentos y toma de decisiones en la dirección financiera de una empresa, estos argumentos y tomas de decisiones las podemos agrupar en:



- ◆ Toma de decisiones de inversión y crédito, lo que requiere conocer la estructura financiera, la capacidad de crecimiento de la empresa, su estabilidad y redituabilidad.
- ◆ Evaluación de la solvencia y liquidez de la empresa, así como su capacidad para generar fondos.
- ◆ Conocer el origen y las características de sus recursos para estimar la capacidad para generar fondos.
- ◆ Formarse un juicio sobre los resultados financieros de la administración en cuanto a la rentabilidad, solvencia, generación de fondos y capacidad de crecimiento.

**Algunas características de los estados financieros:**

- ◆ Los estados financieros deben adecuarse a los propósitos del usuario y brindar información de tipo general de la entidad, por eso lo que se expresa en ellos debe tener contenido informativo (que debe ser significativo, relevante, veraz y comparable) además de que deben ser presentados con oportunidad.
- ◆ Confiables, al igual que la contabilidad, los estados como productos deben ser estables (consistentes), objetivos y verificables.
- ◆ Provisionalidad, la información contenida en los estados financieros incluye estimaciones necesarias para poder determinar la información pertinente a cada periodo contable, como resultado de haber efectuado cortes convencionales en la vida de la empresa, considerando un negocio en marcha, para la toma de decisiones.

**Balance general**

**Balance general (El estado de situación financiera).** Es un estado financiero que muestra los recursos de que dispone una entidad o empresa para la realización de sus fines (activo) y las fuentes externas e internas de dichos recursos (pasivo y capital contable), a una fecha determinada. De su análisis e interpretación podemos conocer la situación financiera y económica, la liquidez y rentabilidad de una empresa o entidad.

Por prepararse a una fecha determinada, se dice que el balance general es una fotografía de la empresa y que los datos contenidos en él pueden cambiar incluso al día siguiente sin que se haya realizado o celebrado otras transacciones, esto debido a que ciertos eventos económicos o circunstancias hacen que los valores del activo, pasivo o capital sufran cambios o modificaciones (como por ejemplo, los efectos de la devaluación de la moneda, la volatilización o evaporación de las mercancías, los pagos anticipados que se devengan, la depreciación de los activos fijos, las deudas que se pagan).

Este estado financiero puede presentarse de varias maneras, las más comunes son dos:

- ◆ En forma de reporte



COMPañÍA XXX S.A

BALANCE GENERAL AL 30 DE JUNIO DE 19XX

ACTIVO

CIRCULANTE

1)Caja		\$15 000		
Bancos		100 000		
2) Clientes	\$80 000			
Estimación p/ctas. de cobro dudoso	5 000	<u>75 000</u>		
3) IVA acreditable		5 000		
4) Mercancías		120 000	\$ 315 000	

FIJO

5) Terrenos		\$90 000		
6) Edificios	\$140 000			
Depreciación acumulada	<u>70 000</u>	70 000		
7) Mobiliario y equipo	<u>\$50 000</u>			
Depreciación acumulada	<u>25 000</u>	<u>25 000</u>	\$185 000	

DIFERIDO

8) Gastos de constitución		\$5 000		
Amortización acumulada		<u>2 500</u>	2 500	\$502 500

PASIVO

CIRCULANTE

Proveedores		\$95 000		
Documentos por pagar		7 500		
9)Impuestos y derechos retenidos por enterar		3 000		
10)Impuestos y derechos por pagar		5 000	\$ 110 500	

FIJO

11) Documentos por pagar		\$12 000		
Acreedores hipotecarios		<u>20 000</u>	<u>32 000</u>	<u>142 500</u>

**CAPITAL CONTABLE**

**CAPITAL CONTRIBUIDO**

Capital social \$200 000

**CAPITAL GANADO**

12)Utilidad neta del	\$60 000		
Utilidades acumuladas	95 000		
Reserva legal	5 000	160 000	\$360 000

LAS NOTAS QUE SE ACOMPAÑAN SON PARTE INTEGRANTE DE  
ESTE ESTADO FINANCIERO

AUTORIZACION

ELABORADO POR

LIC. ALVARO DAVID PÉREZ  
DIRECTOR GENERAL

C.P. JUAN MANUEL RODRIGUEZ LÓPEZ  
CONTADOR GENERAL

**Figura A.8. Ejemplo de balance general en forma de reporte.**

LA VERTICAL, S.A. DE C.V.			
ESTADO DE SITUACION FINANCIERA AL 28 DE FEBRERO DE 19X_			
		\$ XXXX	
	A	XXXX	
	C	XXXX	
	T	XXXX	
	I	XXXX	
	V	XXXX	
	O	XXXX	
		<hr/>	
	<b>SUMA EL ACTIVO</b>		<b>\$ XXXX</b>
MENOS:			
		<hr/>	
	P	\$ XXXX	
	A	XXXX	
	S	XXXX	
	I	XXXX	
	V	XXXX	
	O	XXXX	
		<hr/>	
	<b>SUMA EL PASIVO</b>		<b>\$ XXXX</b>
IGUAL:			
	<b>CAPITAL CONTABLE</b>		<b>\$ XXXX</b>
		<hr/>	

Figura A.9. Balance general en forma de cuenta.

## APÉNDICE B

### Diccionario de Datos.

Tabla: def_rep			
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
rep_id	Texto	10	*Es la llave
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	rep_id		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	def_rep		
descripcion	Texto	70	Descripción del tipo de reporte
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	descripcion		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	def_rep		
num_meses	Número (entero)	2	Número de meses a calcular (0.....6 meses)
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	num_meses		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	def_rep		
exis_actual	SI/No	1	Si desea que aparezcan EXISTENCIA ACTUAL
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	exis_actual		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	def_rep		
exis_estim	SI/No	1	Si desea que aparezcan y calcu- len EXISTENCIAS ESTIMADAS
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	exis_estim		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	def_rep		
conss	SI/No	1	Si desea existencia estimadas basadas en consumo SIREM
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	conss		
Orden de combinación:	General		

Apéndice B.

Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	def_rep		
constp	Si/No	1	Si desea existencia estimadas basadas en consumo tel. púb.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	constp		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	def_rep		
consc	Si/No	1	Si desea existencia estimadas en consumo Construcción.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consc		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	def_rep		
consx	Si/No	1	Si desea existencia estimadas en consumo nuevo 1(nueva base).
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consx		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	def_rep		
consy	Si/No	1	Si desea existencia estimadas basadas en consumo nuevo 2 (nueva base)
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	consy		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	def_rep		
form_calc	Número (entero)	2	Se especifica formula para cálculo de existencias.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	form_calc		
Orden de combinación:	General		
Posición ordinal:	10		
Required:	FALSE		
Tabla de origen:	def_rep		
stock	Si/No	1	Si desea que aparezca stock (2 veces consumo SIREM)
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	stock		
Orden de combinación:	General		
Posición ordinal:	11		
Required:	FALSE		
Tabla de origen:	def_rep		
ped_ant	Si/No	1	Si desea que aparezca pedidos con fecha anterior a actual.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	ped_ant		
Orden de combinación:	General		

Posición ordinal:	12		
Required:	FALSE		
Tabla de origen:	def_rep		
uso_pron	Si/No	1	Si desea que se utilice pronósticos o no (Si por omisión)
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	uso_pron		
Orden de combinación:	General		
Posición ordinal:	13		
Required:	FALSE		
Tabla de origen:	def_rep		
pedidos	Si/No	1	Pedidos de consumo
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	pedidos		
Orden de combinación:	General		
Posición ordinal:	14		
Required:	FALSE		
Tabla de origen:	def_rep		
solo_alm	Si/No	1	Indicación de cálculo total existencias solo almacenes principales.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	solo_alm		
Orden de combinación:	General		
Posición ordinal:	15		
Required:	FALSE		
Tabla de origen:	def_rep		
pron	Si/No	1	Si desea existencias estimadas basadas en pronósticos únicamente
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	pron		
Orden de combinación:	General		
Posición ordinal:	16		
Required:	FALSE		
	<b>Tabla: def_rep</b>		
Tabla de origen:	def_rep		
conmtp	Si/No		
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	conmtp		
Orden de combinación:	General		
Posición ordinal:	17		
Required:	FALSE		
Tabla de origen:	def_rep		
Índices de tabla			
Nombre	Número de campos		
Primarykey	1		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	rep_id, Ascendente		



Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: existencias**

Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	Fecha introducción a sistema DD MM AAAA*
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	existencias		
catalogo	Texto	7	Toma dato de EXIS. NUM_MAT.*
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	existencias		
cod_sirem	Texto	3	Toma dato de EXIS.WHSE CODE*
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cod_sirem		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	existencias		
on_hand	Número (doble)	8	Toma dato de EXIS.ON_HAND
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	on_hand		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	existencias		
reservados	Número (doble)	8	Toma dato de EXIS.RESERVA- DOS
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	reservados		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	<b>Tabla: existencias</b> existencias		
disponible	Número (doble)	8	Toma dato de EXIS.DISPONIBLE Usado como existencias.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	disponible		

Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	existencias		
region	Texto	15	Región de existencias
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	existencias		
categoria	Texto	2	Se usa para quitar elementos con categoria="02" y región=CELAYA PUEBLA, MONTERREY.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	categoria		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	existencias		
Índices de tabla			
Nombre	Número de campos		
Primarykey	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	Primarykey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	fechaintro, Ascendente		
catalogo, Ascendente			
cod_sirem, Ascendente			
Permisos de usuario			
admin			
	<b>Tabla: existencias</b>		
Permisos de grupo			
Admins			
Users			
	<b>Tabla: expusu</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columns			
Nombre	Tipo	Tamaño	
exped	Texto	12	Expediente del empleado de salidas de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	exped		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	expusu		
Permisos de usuario			

admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: Inmat**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas

Nombre	Tipo	Tamaño	
areatm	Texto	1	*Área de construcción. 1=Telefonía pública;2=Planta exterior.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	areatm		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	Inmat		
catalogo	Texto	7	*Forma llave
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	Inmat		
factor	Número (doble)	8	Cantidad estimada de material utilizada
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	factor		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	Inmat		

Índices de tabla  
 Nombre Número de campos  
 Primarykey 2  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: Primarykey  
 Principal: FALSE  
 Required: FALSE  
 Única: TRUE  
 Campos: areatm, Ascendente  
 catalogo, Ascendente

**Tabla: Inmat**

Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: majuste**

Propiedades  
 Def. actualizable: TRUE

RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	DD MM AAAA: Fecha introducción a sistema*
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	majuste		
alm	Texto	12	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	majuste		
cat	Texto	12	Número ó clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	majuste		
ser	Texto	12	Serie; grupo de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ser		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	majuste		
f_cap	Fecha/Hora	8	Fecha del movimiento
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_cap		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	<b>Tabla: majuste</b> majuste		
cdo	Texto	12	Contabilizado de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cdo		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	majuste		
uni	Texto	12	Unidad para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	uni		

Apéndice B.

---

Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	majuste		
cant	Número (doble)	8	Valor o cantidad
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	cant		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	majuste		
Índices de tabla			
Nombre	Número de campos		
AJU_IDX	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	AJU_IDX		
Principal:	FALSE		
Required:	FALSE		
Única:	FALSE		
Campos:	f_cap, Ascendente		
cat, Ascendente			
alm, Ascendente			
AJUID_IDX	1		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	AJUID_IDX		
Principal:	FALSE		
Required:	FALSE		
Única:	<b>Tabla: majuste</b>		
Campos:	FALSE		
Permisos de usuario	fechaintr, Ascendente		
admin			
Permisos de grupo			
Admins			
Users			
	<b>Tabla: matusu</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
cat	Texto	7	Catalogo; clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	matusu		

Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: ment\_de**

Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	DD MM AAAA: Fecha introducción a sistema *.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	ment_de		
un	Texto	12	Unidad de medida
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	un		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	ment_de		
ar	Texto	12	Area para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ar		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	ment_de		
pob	Texto	12	Población de clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	pob		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	ment_de		
naf	Texto	12	Nivel de afectación(clave aplic.)
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	naf		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	Tabla: ment_de ment_de		
cta	Texto	12	Cuenta para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cta		

Apéndice B.

Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	ment_de		
scta	Texto	12	Subcuenta para clave de aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	scta		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	ment_de		
detscta	Texto	12	Detalle de subcuenta clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	detscta		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	ment_de		
a_r	Texto	12	Almacen receptor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_r		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	ment_de		
alm	Texto	12	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	ment_de		
cat	Texto	12	Clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	10		
Required:	FALSE		
Tabla de origen:	Tabla: ment_de		
serie	ment_de	12	Serie; grupo de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	serie		
Orden de combinación:	General		
Posición ordinal:	11		
Required:	FALSE		
Tabla de origen:	ment_de		
depto	Fecha/Hora	8	Departamento
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		

Campo de origen:	depto		
Orden de combinación:	General		
Posición ordinal:	12		
Required:	FALSE		
Tabla de origen:	ment_de		
mot_s	Texto	12	Motivo de salida de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	mot_s		
Orden de combinación:	General		
Posición ordinal:	13		
Required:	FALSE		
Tabla de origen:	ment_de		
f_cap	Fecha/Hora	8	Fecha del movimiento
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_cap		
Orden de combinación:	General		
Posición ordinal:	14		
Required:	FALSE		
Tabla de origen:	ment_de		
f_u_c	Fecha/Hora	8	Fecha de última compra
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_u_c		
Orden de combinación:	General		
Posición ordinal:	15		
Required:	FALSE		
Tabla de origen:	ment_de		
zona	Texto	12	Zona de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	zona		
Orden de combinación:	General		
Posición ordinal:	16		
Required:	FALSE		
Tabla de origen:	Tabla: ment_de		
cdo	ment_de	12	Código de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cdo		
Orden de combinación:	General		
Posición ordinal:	17		
Required:	FALSE		
Tabla de origen:	ment_de		
uni	Texto	12	Unidad para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	uni		
Orden de combinación:	General		
Posición ordinal:	18		
Required:	FALSE		
Tabla de origen:	ment_de		
doc	Texto	12	Número de documento que registra movimiento en SIREM
AllowZeroLength:	FALSE		



Apéndice B.

---

Atributos:	Longitud variable		
Campo de origen:	doc		
Orden de combinación:	General		
Posición ordinal:	19		
Required:	FALSE		
Tabla de origen:	ment_de		
adm	Texto	12	Clave de proyecto
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	adm		
Orden de combinación:	General		
Posición ordinal:	20		
Required:	FALSE		
Tabla de origen:	ment_de		
remi	Texto	12	Remisión del proveedor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	remi		
Orden de combinación:	General		
Posición ordinal:	21		
Required:	FALSE		
Tabla de origen:	ment_de		
exp	Texto	12	Expediente de usuario
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	exp		
Orden de combinación:	General		
Posición ordinal:	22		
Required:	FALSE		
Tabla de origen:	ment_de		
cant	Número (doble)	8	Valor ó cantidad
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	cant		
Orden de combinación:	General		
Posición ordinal:	23		
Required:	FALSE		
Tabla de origen:	ment_de		
saldo	Número (doble)	8	Saldo del pedido
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	saldo		
Orden de combinación:	General		
Posición ordinal:	24		
Required:	FALSE		
Tabla de origen:	ment_de		
Índices de tabla			
Nombre	Número de campos		
END_IDX	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	END_IDX		
Principal:	FALSE		

Required: FALSE  
 Única: FALSE  
 Campos: f\_cap, Ascendente  
 cat, Ascendente  
 alm, Ascendente  
 ENDID\_IDX 1  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: ENDID\_IDX  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: fechaintro, Ascendente  
 Permisos de usuario  
 admin

**Tabla: ment\_de**

Permisos de grupo  
 Admins  
 Users

**Tabla: ment\_oc**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas

Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	Fecha introducción a sistema DD MM AAAA*
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	ment_oc		
un	Texto	12	Unidad de medida
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	un		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	ment_oc		
ar	Texto	12	Area para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ar		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	ment_oc		
pob	Texto	12	Población para clave de aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	pob		
Orden de combinación:	General		

Apéndice B.

---

Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	ment_oc		
naf	Texto	12	Nivel de afectación de clave aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	naf		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
	<b>Tabla: ment_oc</b>		
Tabla de origen:	ment_oc		
cta	Texto	12	Cuenta para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cta		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	ment_oc		
scta	Texto	12	Subcuenta para clave de aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	scta		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	ment_oc		
detscta	Texto	12	Detalle subcuenta de clave aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	detscta		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	ment_oc		
a_r	Texto	12	Almacen receptor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_r		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	ment_oc		
alm	Texto	12	Clave de almacen
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	ment_oc		
cat	Texto	12	Número ó clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		

Orden de combinación:	General		
Posición ordinal:	10		
Required:	FALSE		
	Tabla: ment_oc		
Tabla de origen:	ment_oc		
ped	Texto	12	Pendientes
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ped		
Orden de combinación:	General		
Posición ordinal:	11		
Required:	FALSE		
Tabla de origen:	ment_oc		
serie	Texto	12	Serie; grupo de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	serie		
Orden de combinación:	General		
Posición ordinal:	12		
Required:	FALSE		
Tabla de origen:	ment_oc		
f_cap	Fecha/Hora	8	Fecha del movimiento
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_cap		
Orden de combinación:	General		
Posición ordinal:	13		
Required:	FALSE		
Tabla de origen:	ment_oc		
f_u_c	Fecha/Hora	8	Fecha de última compra
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_u_c		
Orden de combinación:	General		
Posición ordinal:	14		
Required:	FALSE		
Tabla de origen:	ment_oc		
cdo	Texto	12	Código de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cdo		
Orden de combinación:	General		
Posición ordinal:	15		
Required:	FALSE		
Tabla de origen:	ment_oc		
prov	Texto	12	Clave de proveedor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	prov		
Orden de combinación:	General		
Posición ordinal:	16		
Required:	FALSE		
	Tabla: ment_oc		
Tabla de origen:	ment_oc		
uni	Texto	12	Unidad de clave de aplicación
AllowZeroLength:	FALSE		

Apéndice B.

Atributos:	Longitud variable		
Campo de origen:	uni		
Orden de combinación:	General		
Posición ordinal:	17		
Required:	FALSE		
Tabla de origen:	ment_oc		
doc	Texto	12	Número de documento que registra movimiento en SIREM.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	doc		
Orden de combinación:	General		
Posición ordinal:	18		
Required:	FALSE		
Tabla de origen:	ment_oc		
adm	Texto	12	Clave de proyecto
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	adm		
Orden de combinación:	General		
Posición ordinal:	19		
Required:	FALSE		
Tabla de origen:	ment_oc		
remi	Texto	12	Remisión del proveedor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	remi		
Orden de combinación:	General		
Posición ordinal:	20		
Required:	FALSE		
Tabla de origen:	ment_oc		
cant	Número (doble)	8	Valor ó cantidad
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	cant		
Orden de combinación:	General		
Posición ordinal:	21		
Required:	FALSE		
Tabla de origen:	ment_oc		
saldo	Número (doble)	8	Saldo del pedido
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	saldo		
Orden de combinación:	General		
Posición ordinal:	22		
Required:	FALSE		
Tabla de origen:	Tabla: ment_oc ment_oc		
Índices de tabla			
Nombre	Número de campos		
ENO_IDX	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	ENO_IDX		
Principal:	FALSE		

Required: FALSE  
 Única: FALSE  
 Campos: f\_cap, Ascendente  
 cat, Ascendente  
 alm, Ascendente  
 ENOID\_IDX 1  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: ENOID\_IDX  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: fechaintro, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: ment\_tr**

**Propiedades**  
 Def. actualizable: TRUE  
 RecordCount: 0  
**Columnas**  
 Nombre Tipo Tamaño  
 fechaintro Fecha/Hora 8 Fecha introducción a sistema  
 AllowZeroLength: FALSE DD MM AAAA\*  
 Atributos: Tamaño fijo  
 Campo de origen: fechaintro  
 Orden de combinación: General  
 Posición ordinal: 0  
 Required: FALSE  
 Tabla de origen: ment\_tr  
 un Texto 12 Unidad de medida  
 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: un  
 Orden de combinación: General  
 Posición ordinal: 1  
 Required: FALSE  
 Tabla de origen: ment\_tr  
 ar Texto 12 Area para clave de aplicación  
 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: ar  
 Orden de combinación: General  
 Posición ordinal: 2  
 Required: FALSE  
 Tabla de origen: ment\_tr  
 pob Texto 12 Población para clave de aplic.  
 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: pob  
 Orden de combinación: General  
 Posición ordinal: 3

Apéndice B.

---

Required:	FALSE		
Tabla de origen:	ment_tr		
naf	Texto	12	Nivel de afectación (clave aplic.)
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	naf		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
	<b>Tabla: ment_tr</b>		
Tabla de origen:	ment_tr		
cta	Texto	12	Cuenta para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cta		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	ment_tr		
scta	Texto	12	Subcuenta para clave de aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	scta		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	ment_tr		
detscta	Texto	12	Detalle subcuenta de clave aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	detscta		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	ment_tr		
a_s	Texto	12	Almacén surtidor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_s		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	ment_tr		
a_r	Texto	12	Almacén receptor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_r		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	ment_tr		
alm	Texto	12	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		

Posición ordinal:	10		
Required:	FALSE		
Tabla de origen:	<b>Tabla: ment_tr</b>		
cat	ment_tr		
AllowZeroLength:	Texto	12	Clave de material
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	cat		
Posición ordinal:	General		
Required:	11		
Tabla de origen:	FALSE		
ser	ment_tr		
AllowZeroLength:	Texto	12	Serie; grupo de material
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	ser		
Posición ordinal:	General		
Required:	12		
Tabla de origen:	FALSE		
f_cap	ment_tr		
AllowZeroLength:	Fecha/Hora	8	Fecha del movimiento
Atributos:	FALSE		
Campo de origen:	Tamaño fijo		
Orden de combinación:	f_cap		
Posición ordinal:	General		
Required:	13		
Tabla de origen:	FALSE		
f_u_c	ment_tr		
AllowZeroLength:	Fecha/Hora	8	Fecha de última compra
Atributos:	FALSE		
Campo de origen:	Tamaño fijo		
Orden de combinación:	f_u_c		
Posición ordinal:	General		
Required:	14		
Tabla de origen:	FALSE		
cdo	ment_tr		
AllowZeroLength:	Texto	12	Código de información
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	cdo		
Posición ordinal:	General		
Required:	15		
Tabla de origen:	FALSE		
uni	ment_tr		
AllowZeroLength:	Texto	12	Unidad para clave de aplicación
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	uni		
Posición ordinal:	General		
Required:	16		
Tabla de origen:	FALSE		
doc	<b>Tabla: ment_tr</b>		
AllowZeroLength:	ment_tr		
Atributos:	Texto	12	Número de documento que registra movimiento en SIREM
	FALSE		
	Longitud variable		



Apéndice B.

---

Campo de origen: doc  
 Orden de combinación: General  
 Posición ordinal: 17  
 Required: FALSE  
 Tabla de origen: ment\_tr  
 cant Número (doble) 8 Número ó clave de material  
 AllowZeroLength: FALSE  
 Atributos: Tamaño fijo  
 Campo de origen: cant  
 Orden de combinación: General  
 Posición ordinal: 18  
 Required: FALSE  
 Tabla de origen: ment\_tr  
 saldo Número (doble) 8 Saldo del pedido  
 AllowZeroLength: FALSE  
 Atributos: Tamaño fijo  
 Campo de origen: saldo  
 Orden de combinación: General  
 Posición ordinal: 19  
 Required: FALSE  
 Tabla de origen: ment\_tr  
 Índices de tabla  
 Nombre Número de campos  
 ENT\_IDX 3  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: ENT\_IDX  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: f\_cap, Ascendente  
 cat, Ascendente  
 alm, Ascendente  
 ENTID\_IDX 1  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: ENTID\_IDX  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: fechaintro, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users  
 Tabla: movi  
 Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas

Nombre	Tipo	Tamaño	Unidad de medida
un	Texto	12	Unidad de medida
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	un		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	movi		
ar	Texto	12	Area para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ar		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	movi		
po	Texto	12	Población de clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	po		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	movi		
naf	Texto	12	Nivel de afectación (clave aplic.)
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	naf		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	movi		
cta	Texto	12	Cuenta para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cta		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	<b>Tabla: movi</b> movi		
scta	Texto	12	Subcuenta para clave de aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	scta		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	movi		
xx	Texto	12	
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	xx		
Orden de combinación:	General		
Posición ordinal:	6		

Apéndice B.

---

Required:	FALSE		
Tabla de origen:	movi		
a_s	Texto	12	Almacén surtidor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_s		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	movi		
a_r	Texto	12	Almacén receptor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_r		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	movi		
alm	Texto	12	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	movi		
cat	Texto	12	Número ó clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	10		
Required:	FALSE		
Tabla de origen:	movi		
ped	Texto	12	Pedidos de movimientos
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ped		
Orden de combinación:	General		
Posición ordinal:	11		
Required:	FALSE		
Tabla de origen:	movi		
ser	Texto	12	Serie; grupo de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ser		
Orden de combinación:	General		
Posición ordinal:	12		
Required:	FALSE		
Tabla de origen:	movi		
depto	Texto	12	Departamento
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	depto		
Orden de combinación:	General		

Posición ordinal:	13		
Required:	FALSE		
Tabla de origen:	movi		
t_mov	Texto	12	Tipo de movimiento
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	t_mov		
Orden de combinación:	General		
Posición ordinal:	14		
Required:	FALSE		
Tabla de origen:	movi		
mot_s	Texto	12	Motivo de salida de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	mot_s		
Orden de combinación:	General		
Posición ordinal:	15		
Required:	FALSE		
Tabla de origen:	movi		
f_cap	Fecha/Hora	8	Fecha del movimiento
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_cap		
Orden de combinación:	General		
Posición ordinal:	16		
Required:	FALSE		
Tabla de origen:	<b>Tabla: movi</b> movi		
f_u_c	Fecha/Hora	8	Fecha de última compra
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_u_c		
Orden de combinación:	General		
Posición ordinal:	17		
Required:	FALSE		
Tabla de origen:	movi		
zona	Texto	12	Zona de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	zona		
Orden de combinación:	General		
Posición ordinal:	18		
Required:	FALSE		
Tabla de origen:	movi		
cdo	Texto	12	Código de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cdo		
Orden de combinación:	General		
Posición ordinal:	19		
Required:	FALSE		
Tabla de origen:	movi		
prov	Texto	12	Clave de proveedor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	prov		

Apéndice B.

---

Orden de combinación:	General		
Posición ordinal:	20		
Required:	FALSE		
Tabla de origen:	movi		
uni	Texto	12	Unidad para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	uni		
Orden de combinación:	General		
Posición ordinal:	21		
Required:	FALSE		
Tabla de origen:	movi		
doc	Texto	12	Número de documento que registra movimiento en SIREM
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	doc		
Orden de combinación:	General		
Posición ordinal:	22		
Required:	FALSE		
	<b>Tabla: movi</b>		
Tabla de origen:	movi		
adm	Texto	12	Clave proyecto
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	adm		
Orden de combinación:	General		
Posición ordinal:	23		
Required:	FALSE		
Tabla de origen:	movi		
remi	Texto	12	Remisión del proveedor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	remi		
Orden de combinación:	General		
Posición ordinal:	24		
Required:	FALSE		
Tabla de origen:	movi		
exped	Texto	12	Expediente del empleado salida de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	exped		
Orden de combinación:	General		
Posición ordinal:	25		
Required:	FALSE		
Tabla de origen:	movi		
movto	Texto	12	Movimiento
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	movto		
Orden de combinación:	General		
Posición ordinal:	26		
Required:	FALSE		
Tabla de origen:	movi		
c_mat	Texto	12	Clase de material
AllowZeroLength:	FALSE		

Atributos:	Longitud variable		
Campo de origen:	c_mat		
Orden de combinación:	General		
Posición ordinal:	27		
Required:	FALSE		
Tabla de origen:	movi		
catago	Texto	12	Categoría
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catago		
Orden de combinación:	General		
Posición ordinal:	28		
Required:	FALSE		
Tabla de origen:	Tabla: movi		
cant	Número (doble)	8	Valor ó cantidad
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	cant		
Orden de combinación:	General		
Posición ordinal:	29		
Required:	FALSE		
Tabla de origen:	movi		
saldo	Número (doble)	8	Saldo del pedido
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	saldo		
Orden de combinación:	General		
Posición ordinal:	30		
Required:	FALSE		
Tabla de origen:	movi		
Permisos de usuario	admin		
Permisos de grupo	Admins		
Users	Users		
	<b>Tabla: msal_tr</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	Fecha introducción a sistema
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	msal_tr		
un	Texto	12	Unidad de medida
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	un		
Orden de combinación:	General		
Posición ordinal:	1		

Apéndice B.

---

Required:	FALSE		
Tabla de origen:	msal_tr		
ar	Texto	12	Area para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ar		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	msal_tr		
pob	Texto	12	Poblacion de la clave de aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	pob		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	msal_tr		
naf	Texto	12	Nivel de afectación (clave aplic)
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	naf		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	msal_tr		
cta	Texto	12	Cuenta para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cta		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	msal_tr		
scta	Texto	12	Subcuenta para clave de aplic.
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	scta		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	msal_tr		
detscta	Texto	12	Detalle de subcuenta para clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	detscta		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	msal_tr		
a_s	Texto	12	Almacén surtidor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_s		
Orden de combinación:	General		

Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	msal_tr		
a_r	Texto	12	Almacén receptor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_r		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	msal_tr		
alm	Texto	12	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	10		
Required:	FALSE		
Tabla de origen:	msal_tr		
cat	Texto	12	Número o clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	11		
Required:	FALSE		
Tabla de origen:	msal_tr		
ser	Texto	12	Serie; grupo de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ser		
Orden de combinación:	General		
Posición ordinal:	12		
Required:	FALSE		
Tabla de origen:	msal_tr		
f_cap	Fecha/Hora	8	Fecha de movimiento
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_cap		
Orden de combinación:	General		
Posición ordinal:	13		
Required:	FALSE		
Tabla de origen:	msal_tr		
f_u_c	Fecha/Hora	8	Fecha de última compra
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_u_c		
Orden de combinación:	General		
Posición ordinal:	14		
Required:	FALSE		
Tabla de origen:	msal_tr		
zona	Texto	12	Zona de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	zona		



Apéndice B.

Orden de combinación:	General		
Posición ordinal:	15		
Required:	FALSE		
Tabla de origen:	msal_tr		
cdo	Texto	12	Código de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cdo		
Orden de combinación:	General		
Posición ordinal:	16		
Required:	FALSE		
	<b>Tabla: msal_tr</b>		
Tabla de origen:	msal_tr		
uni	Texto	12	Unidad de clave de aplicación
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	uni		
Orden de combinación:	General		
Posición ordinal:	17		
Required:	FALSE		
Tabla de origen:	msal_tr		
doc	Texto	12	Número de documento que registra movimiento en SIREM
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	doc		
Orden de combinación:	General		
Posición ordinal:	18		
Required:	FALSE		
Tabla de origen:	msal_tr		
cant	Número (doble)	8	Valor ó cantidad
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	cant		
Orden de combinación:	General		
Posición ordinal:	19		
Required:	FALSE		
Tabla de origen:	msal_tr		
saldo	Número (doble)	8	Saldo de pedido
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	saldo		
Orden de combinación:	General		
Posición ordinal:	20		
Required:	FALSE		
Tabla de origen:	msal_tr		
Índices de tabla			
Nombre	Número de campos		
SAT_IDX	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	SAT_IDX		
Principal:	FALSE		
Required:	FALSE		
Única:	FALSE		

**Tabla: msal\_tr**  
 f\_cap, Ascendente

Campos:  
 cat, Ascendente  
 alm, Ascendente  
 SATID\_IDX 1  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: SATID\_IDX  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: fechaintro, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: msal\_us**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas

Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	Fecha introducción a sistema DD MM AAAA*
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	msal_us		
a_s	Texto	12	Almacén surtidor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	a_s		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	msal_us		
alm	Texto	12	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	msal_us		
cat	Texto	12	Número ó clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		

Apéndice B.

---

Tabla de origen:	msal_us		
ser	Texto	12	Serie/grupo de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	ser		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
	<b>Tabla: msal_us</b>		
Tabla de origen:	msal_us		
depto	Texto	12	Departamento
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	depto		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	msal_us		
mot_s	Texto	12	Motivo de salida de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	mot_s		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	msal_us		
f_cap	Fecha/Hora	8	Fecha de movimiento
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_cap		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	msal_us		
f_u_c	Fecha/Hora	8	Fecha de última compra.
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	f_u_c		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	msal_us		
zona	Texto	12	Zona de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	zona		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	msal_us		
cdo	Texto	12	Código de información
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cdo		
Orden de combinación:	General		
Posición ordinal:	10		

Required:	FALSE		
Tabla de origen:	Tabla: msal_us		
uni	msal_us	12	Unidad de clave de aplicación
AllowZeroLength:	Texto		
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	uni		
Posición ordinal:	General		
Required:	11		
Tabla de origen:	FALSE		
doc	msal_us	12	Número de documento que registra movimiento en SIREM
AllowZeroLength:	Texto		
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	doc		
Posición ordinal:	General		
Required:	12		
Tabla de origen:	FALSE		
adm	msal_us	12	Clave proyecto
AllowZeroLength:	Texto		
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	adm		
Posición ordinal:	General		
Required:	13		
Tabla de origen:	FALSE		
exped	msal_us	12	Expediente del empleado de salida de material.
AllowZeroLength:	Texto		
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	exped		
Posición ordinal:	General		
Required:	14		
Tabla de origen:	FALSE		
catego	msal_us	12	Categoría
AllowZeroLength:	Texto		
Atributos:	FALSE		
Campo de origen:	Longitud variable		
Orden de combinación:	catego		
Posición ordinal:	General		
Required:	15		
Tabla de origen:	FALSE		
cant	msal_us	8	Valor ó cantidad
AllowZeroLength:	Número (doble)		
Atributos:	FALSE		
Campo de origen:	Tamaño fijo		
Orden de combinación:	cant		
Posición ordinal:	General		
Required:	16		
Tabla de origen:	FALSE		
saldo	Tabla: msal_us		
AllowZeroLength:	msal_us	8	Saldo del pedido
Atributos:	Número (doble)		
Campo de origen:	FALSE		
	Tamaño fijo		
	saldo		

Apéndice B.

Orden de combinación:	General
Posición ordinal:	17
Required:	FALSE
Tabla de origen:	msal_us
Índices de tabla	
Nombre	Número de campos
SAU_IDX	3
Agrupado:	FALSE
Cuenta distinta:	0
Extranjero:	FALSE
Ignorar Nulos:	FALSE
Nombre:	SAU_IDX
Principal:	FALSE
Required:	FALSE
Única:	FALSE
Campos:	f_cap, Ascendente
cat, Ascendente	
alm, Ascendente	
SAUID_IDX	1
Agrupado:	FALSE
Cuenta distinta:	0
Extranjero:	FALSE
Ignorar Nulos:	FALSE
Nombre:	SAUID_IDX
Principal:	FALSE
Required:	FALSE
Única:	FALSE
Campos:	fechaintro, Ascendente
Permisos de usuario	
admin	
Permisos de grupo	
Admins	
Users	

**Tabla: ped\_det**

Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	Fecha introducción a sistema
AllowZeroLength:	FALSE		DD MM AAAA*
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	ped_det		
po_num	Texto	7	Número de pedido
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	po_num		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	ped_det		
cod_sirem	Texto	3	Toma dato de PED.WHSE_CODE

AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cod_sirem		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	ped_det		
catalogo	Texto	7	Toma dato de PED.WHSE_NUM
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	ped_det		
request_da	Fecha/Hora	8	Fecha pactada de esta entrega
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	request_da		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	ped_det		
qty_ordere	Número (doble)	8	Cantidad programada para esta entrega
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	qty_ordere		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	ped_det		
qty_to_rec	Número (doble)	8	Cantidad total del catalogo del pedido pero no siempre es real
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	qty_to_rec		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	ped_det		
qty_receiv	Número (doble)	8	Cantidad recibida realmente de esta entrega
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	qty_receiv		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	FALSE		
Tabla de origen:	ped_det		
cantidad	Número (doble)	8	Cantidad por recibir
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	cantidad		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	FALSE		
Tabla de origen:	ped_det		

ped_prog	Número (entero)	2	Indicador de entregas parciales
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	ped_prog		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	FALSE		
Tabla de origen:	ped_det		
marca	Texto	2	Marca de pedido
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	marca		
Orden de combinación:	General		
Posición ordinal:	10		
Required:	FALSE		
Tabla de origen:	<b>Tabla: ped_det</b> ped_det		
region	Texto	15	Región de pedido
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	11		
Required:	FALSE		
Tabla de origen:	ped_det		
Índices de tabla			
Nombre	Número de campos		
PED_IDX	5		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	PED_IDX		
Principal:	FALSE		
Required:	FALSE		
Única:	FALSE		
Campos:	fechaintro, Ascendente		
po_num, Ascendente			
cod_sirem, Ascendente			
catalogo, Ascendente			
request_da, Ascendente			
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
Propiedades	<b>Tabla: pedidos</b>		
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
fechaintro	Fecha/Hora	8	DD MM AAAA: Fecha introducción a sistema*
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	fechaintro		

Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	pedidos		
po_num	Texto	7	Número de pedido
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	po_num		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	pedidos		
cod_sirem	Texto	3	Toma dato de PEN.WHSE_CODE
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cod_sirem		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	pedidos		
po_date	Fecha/Hora	8	Fecha de elaboración de pedido
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	po_date		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	pedidos		
pos_status	Texto	10	Status de pedido
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	pos_status		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	<b>Tabla: pedidos</b>		
vendor_num	Texto	10	Número de proveedor
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	vendor_num		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	FALSE		
Tabla de origen:	pedidos		
sched_deli	Texto	1	
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	sched_deli		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	FALSE		
Tabla de origen:	pedidos		
Índices de tabla			
Nombre	Número de campos		
PrimaryKey	3		



Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: PrimaryKey  
 Principal: FALSE  
 Required: FALSE  
 Única: TRUE  
 Campos: fechaintro, Ascendente  
 po\_num, Ascendente  
 cod\_sirem, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: planan**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas

Nombre	Tipo	Tamaño	Area de construcción.1=tel.pub. 2=planta exterior*
areatm	Texto	1	
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	areatm		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	planan		
region	Texto	15	Forman llave*
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	planan		
mes	Número (largo)	4	MES*
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	planan		
anio	Número (largo)	4	AÑO*
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	anio		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	planan		
cantidad	Número (doble)	8	Cantidad planeada
AllowZeroLength:	FALSE		

Atributos:	Tamaño fijo		
Campo de origen:	cantidad		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	<b>Tabla: planan</b>		
Índices de tabla	planan		
Nombre	Número de campos		
PrimaryKey	4		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	PrimaryKey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	areatm, Ascendente		
region, Ascendente			
mes, Ascendente			
año, Ascendente			
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
	<b>Tabla: pronosticos</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
catalogo	Texto	7	*Código de catalogo
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	pronosticos		
region	Texto	15	*Región
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	pronosticos		
mes	Número (largo)	4	*MES
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		

Tabla de origen:	pronosticos		
anio	Número (largo)	4	*AÑO
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	anio		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	pronosticos		
pronos	Número (doble)	8	Pronósticos de consumo del mes
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	pronos		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	<b>Tabla: pronosticos</b>		
Índices de tabla	pronosticos		
Nombre	Número de campos		
PrimaryKey	4		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	PrimaryKey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	catalogo, Ascendente		
region, Ascendente			
mes, Ascendente			
anio, Ascendente			
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
	<b>Tabla: reg_pref</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
region	Texto	15	Región
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	reg_pref		
region_pref	Texto	15	Región preferente
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region_pref		

Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	reg_pref		
prefer	Número (largo)	4	Preferencia
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	prefer		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	reg_pref		
Índices de tabla			
Nombre	Número de campos		
PrimaryKey	2		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	PrimaryKey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	region, Ascendente		
region_pref, Ascendente			
	<b>Tabla: reg_pref</b>		
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
	<b>Tabla: regalm</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
region	Texto	15	Región
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	region		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	regalm		
Índices de tabla			
Nombre	Número de campos		
RAL_IDX	1		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	RAL_IDX		
Principal:	FALSE		
Required:	FALSE		
Única:	FALSE		

Campos: region, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: region**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 10  
 Columnas  

Nombre	Tipo	Tamaño	Región
region	Texto	15	

 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: region  
 Orden de combinación: General  
 Posición ordinal: 0  
 Required: FALSE  
 Tabla de origen: region  
 Índices de tabla  

Nombre	Número de campos
PrimaryKey	1

 Agrupado: FALSE  
 Cuenta distinta: 10  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: PrimaryKey  
 Principal: FALSE  
 Required: FALSE  
 Única: TRUE  
 Campos: region, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: rep\_con**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas  

Nombre	Tipo	Tamaño	
rep_id	Texto	10	*Identificador de reporte

 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: rep\_id  
 Orden de combinación: General  
 Posición ordinal: 0  
 Required: FALSE  
 Tabla de origen: rep\_con  

Nombre	Tipo	Tamaño	
con_cat_id	Texto	7	*Identificar de catalogo o conjunto

 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: con\_cat\_id  
 Orden de combinación: General

Posición ordinal: 1  
 Required: FALSE  
 Tabla de origen: rep\_con  
 tipo\_cat: Texto 1 \*1=CATALOGO;2=CONJUNTO  
 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: tipo\_cat  
 Orden de combinación: General  
 Posición ordinal: 2  
 Required: FALSE  
 Tabla de origen: rep\_con  
 Índices de tabla  
 Nombre: Número de campos  
 con\_cat\_id: 1  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: con\_cat\_id  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: con\_cat\_id, Ascendente  
 PrimaryKey: 2  
**Tabla: rep\_con**  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: PrimaryKey  
 Principal: FALSE  
 Required: FALSE  
 Única: TRUE  
 Campos: rep\_id, Ascendente  
 con\_cat\_id, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users  
**Tabla: sum\_dev**  
 Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas  
 Nombre: Tipo Tamaño  
 cat: Texto 7 Número ó clave de material  
 AllowZeroLength: FALSE  
 Atributos: Longitud variable  
 Campo de origen: cat  
 Orden de combinación: General  
 Posición ordinal: 0  
 Required: FALSE  
 Tabla de origen: sum\_dev  
 alm: Texto 3 Clave de almacén  
 AllowZeroLength: FALSE

Apéndice B.

---

Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	sum_dev		
anio	Número (largo)	4	*AÑO
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	anio		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	sum_dev		
mes	Número (largo)	4	*MES
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	sum_dev		
suma	Número (doble)	8	Suma
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	suma		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	Tabla: sum_dev		
Índices de tabla	sum_dev		
Nombre	Número de campos		
SDEV_IDX	4		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	SDEV_IDX		
Principal:	FALSE		
Required:	FALSE		
Única:	FALSE		
Campos:	cat, Ascendente		
alm, Ascendente			
anio, Ascendente			
mes, Ascendente			
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
	<b>Tabla: sum_dev2</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			

Nombre	Tipo	Tamaño	
cat	Texto	7	Número ó clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	sum_dev2		
alm	Texto	3	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	sum_dev2		
anio	Número (largo)	4	*AÑO
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	anio		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	sum_dev2		
mes	Número (largo)	4	*MES
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	sum_dev2		
suma	Número (doble)	8	Suma
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	suma		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	Tabla: sum_dev2		
Índices de tabla	sum_dev2		
Nombre	Número de campos		
SDEV2_IDX	4		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	SDEV2_IDX		
Principal:	FALSE		
Required:	FALSE		
Única:	FALSE		
Campos:	cat, Ascendente		
alm, Ascendente			
anio, Ascendente			



mes, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: sum\_usu**

Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	Número ó clave de material
cat	Texto	7	
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	sum_usu		
alm	Texto	3	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	sum_usu		
año	Número (largo)	4	*AÑO
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	año		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	sum_usu		
mes	Número (largo)	4	*MES
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	sum_usu		
suma	Número (doble)	8	Suma
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	suma		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
Tabla de origen:	<b>Tabla: sum_usu</b> sum_usu		
Índices de tabla			
Nombre	Número de campos		
SUSU_IDX	4		

Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: SUSU\_IDX  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: cat, Ascendente

alm, Ascendente  
 ano, Ascendente  
 mes, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: sum\_usu2**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas

Nombre	Tipo	Tamaño	
cat	Texto	7	Número o clave de material
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cat		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	sum_usu2		
alm	Texto	3	Clave de almacén
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	alm		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	sum_usu2		
anio	Número (largo)	4	*AÑO
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	anio		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	sum_usu2		
mes	Número (largo)	4	*MES
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	mes		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	sum_usu2		
suma	Número (doble)	8	Suma

AllowZeroLength: FALSE  
 Atributos: Tamaño fijo  
 Campo de origen: suma  
 Orden de combinación: General  
 Posición ordinal: 4  
 Required: FALSE  
**Tabla: sum\_usu2**  
 sum\_usu2  
 Tabla de origen:  
 Índices de tabla  
 Nombre Número de campos  
 SUSU2\_IDX 4  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: SUSU2\_IDX  
 Principal: FALSE  
 Required: FALSE  
 Única: FALSE  
 Campos: cat, Ascendente  
 alm, Ascendente  
 ano, Ascendente  
 mes, Ascendente  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users

**Tabla: TAVAN1**

Propiedades  
 Def. actualizable: TRUE  
 RecordCount: 0  
 Columnas  

Nombre	Tipo	Tamaño	Número de pedido
po_num	Texto	7	
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	po_num		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	TAVAN1		
cod_sirem	Texto	3	Toma dato de PED.WHISE CODE
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	cod_sirem		
Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	TAVAN1		
catalogo	Texto	7	Catalogo
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	2		

Required:	FALSE		
Tabla de origen:	TAVAN1		
sum_prog	Número (doble)	8	Suma programada
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	sum_prog		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	TAVAN1		
sum_rec	Número (doble)	8	Suma de recorte
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	sum_rec		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	FALSE		
	<b>Tabla: TAVAN1</b>		
Tabla de origen:	TAVAN1		
Índices de tabla			
Nombre	Número de campos		
PrimaryKey	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	PrimaryKey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	po_num, Ascendente		
cod_sirem, Ascendente			
catalogo, Ascendente			
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
	<b>Tabla: TAVAN2</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columnas			
Nombre	Tipo	Tamaño	
po_num	Texto	7	Número de pedido
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	po_num		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	TAVAN2		
cod_sirem	Texto	3	Toma de dato de PED.WHISE_
AllowZeroLength:	FALSE		CODE
Atributos:	Longitud variable		
Campo de origen:	cod_sirem		

Apéndice B.

---

Orden de combinación:	General		
Posición ordinal:	1		
Required:	FALSE		
Tabla de origen:	TAVAN2		
catalogo	Texto	7	Catalogo
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	2		
Required:	FALSE		
Tabla de origen:	TAVAN2		
sum_totprog	Número (doble)	8	Suma total programada
AllowZeroLength:	FALSE		
Atributos:	Tamaño fijo		
Campo de origen:	sum_totprog		
Orden de combinación:	General		
Posición ordinal:	3		
Required:	FALSE		
Tabla de origen:	TAVAN2		
Índices de tabla			
Nombre	Número de campos		
PrimaryKey	3		
Agrupado:	FALSE		
Cuenta distinta:	0		
	<b>Tabla: TAVAN2</b>		
Extranjero:	FALSE		
Ignorar Nulos:	FALSE		
Nombre:	PrimaryKey		
Principal:	FALSE		
Required:	FALSE		
Única:	TRUE		
Campos:	po_num, Ascendente		
cod_sirem, Ascendente			
catalogo, Ascendente			
Permisos de usuario			
admin			
Permisos de grupo			
Admins			
Users			
	<b>Tabla: TAVAN3</b>		
Propiedades			
Def. actualizable:	TRUE		
RecordCount:	0		
Columns			
Nombre	Tipo	Tamaño	
catalogo	Texto	7	Catalogo
AllowZeroLength:	FALSE		
Atributos:	Longitud variable		
Campo de origen:	catalogo		
Orden de combinación:	General		
Posición ordinal:	0		
Required:	FALSE		
Tabla de origen:	TAVAN3		
cod_sirem	Texto	3	Toma dato de PED.WHISE_CODE
AllowZeroLength:	FALSE		

---

Atributos: Longitud variable  
 Campo de origen: cod\_sirem  
 Orden de combinación: General  
 Posición ordinal: 1  
 Required: FALSE  
 Tabla de origen: TAVAN3  
 sum\_disp Número (doble) 8 Suma de dispositivos  
 AllowZeroLength: FALSE  
 Atributos: Tamaño fijo  
 Campo de origen: sum\_disp  
 Orden de combinación: General  
 Posición ordinal: 2  
 Required: FALSE  
 Tabla de origen: TAVAN3  
 Índices de tabla  
 Nombre Número de campos  
 PrimaryKey 2  
 Agrupado: FALSE  
 Cuenta distinta: 0  
 Extranjero: FALSE  
 Ignorar Nulos: FALSE  
 Nombre: PrimaryKey  
 Principal: FALSE  
 Required: FALSE  
 Única: TRUE  
 Campos: catalogo, Ascendente  
 cod\_sirem, Ascendente  
  
**Tabla: TAVAN3**  
 Permisos de usuario  
 admin  
 Permisos de grupo  
 Admins  
 Users



## APENDICE C

### NORMALIZACIÓN DEL FLUJO DE DATOS "PEDIDOS"

Continuando con la normalización de las tablas, en la figura C.1, tenemos los datos que contiene el flujo de datos "PEDIDOS", al cual se le aplicarán las primeras tres formas normales para obtener tablas optimizadas y poder administrar la información en el sistema.

#### FLUJO DE DATOS "PEDIDOS"

PO\_STATUS – Estado del pedido  
N\_MATERIAL – Catálogo o clave del material  
DESCRIP – Descripción del material  
ALMACEN – Clave del almacén que recibe el pedido  
N\_PEDIDO – Número de pedido  
F\_PEDIDO – Fecha de elaboración del pedido  
F\_REQ – Fecha requerida del material  
CANTIDAD – Cantidad total solicitada  
C\_XRECT – Cantidad total por recibir (saldo total)  
C\_RECT – Cantidad total recibida  
C\_PROG – Cantidad programada para esta fecha  
C\_RECP – Cantidad recibida de lo programado a esa fecha  
F\_PROG – Fecha programada de recepción  
NUM\_PROV – Clave del proveedor  
LINE\_SEQ – Número de partida del pedido  
SCHED\_SEQ – Número de programa de la partida  
LINE\_STATU – Estado de la línea o partida

Figura C.1. Campos del Flujo de Datos "PEDIDOS".

En esta tabla, dado que son registros que provienen de otro sistema, no existe la posibilidad de que se tengan varios datos en un mismo campo, lo que quiere decir que ya se encuentra en 1NF.

La llave primaria de esta tabla está formada por dos campos "N\_PEDIDO" (Número del pedido) y "ALMACEN" los cuales están subrayados en la figura. Cada pedido tiene datos de encabezado y varias



líneas o partidas que pueden tener diferente catálogo, material y almacén. Los datos que pertenecen al encabezado se agrupan en la tabla "pedidos" que se muestra en la figura C.2 siguiente:

TABLA "pedidos"  
PO\_STATUS  
N\_PEDIDO  
ALMACEN  
F\_PEDIDO  
F\_REQ  
PO\_STATUS  
NUM\_PROV  
CANTIDAD  
C\_XRECT  
C\_RECT

**Figura C.2. Tabla de encabezado de PEDIDOS.**

Estos campos dependen funcionalmente de la llave primaria, solo cambian cuando esta cambia, por lo que se consideran datos de encabezado del pedido.

Los demás datos, no solo dependen de la llave primaria, sino también de la fecha programada, esto porque en un pedido de un almacén se puede agregar varias partidas en las que varíe el catálogo y la fecha de entrega, y para cada fecha y catálogo, se tiene diferente cantidad solicitada.

Al no existir una dependencia funcional de todos los campos con la llave primaria de la tabla, esta tabla necesita ser normalizada. Para ello se conservan en la tabla "pedidos" los campos de encabezado, y se crea una nueva tabla "ped\_det" que contenga los demás datos. Para que exista una relación entre las dos tablas, se repiten los campos llave de la tabla "pedidos" como parte de la llave de la tabla "ped\_det"

Esta nueva tabla "ped\_det" se muestra en la figura C.3 en la que se subrayan los campos N\_PEDIDO, ALMACEN, N\_MATERIAL y F\_PROG que son los que van a formar la llave primaria y de los cuales dependen funcionalmente los demás campos.

Con esta separación de los campos según su dependencia funcional, todos los campos en cada una de las dos tablas, dependen directamente de la llave principal, por lo que decimos que las tablas "ped\_det" y "pedidos" ya se encuentran en 2NF.

Revisando la tabla "ped\_det" en la figura C.3, podemos observar que existe el campo "DESCRIP" que es la descripción del material; es un campo que depende funcionalmente del campo N\_MATERIAL, como

N\_MATERIAL es una llave candidata, esta relación funcional nos hace ver que esta tabla no está en 3NF.

TABLA "ped\_det"  
N\_PEDIDO  
ALMACEN  
N\_MATERIAL  
F\_PROG  
 DESCRIP  
 C\_PROG  
 C\_RECPC  
 LINE\_SEQ  
 SCHED\_SEQ  
 LINE\_STATU  
 CANTIDAD

Figura C.2. Tabla "ped\_det" resultante de "PEDIDOS" para llevar a 2NF.

Para llevar la tabla "ped\_det" a 3NF es necesario sacar el campo "DESCRIP" y relacionarlo en una tabla que contenga "N\_MATERIAL" y "DESCRIP", para que se pueda hacer referencia a ella únicamente con "N\_MATERIAL".

Dentro del sistema ya existe una tabla que tiene esta nueva relación resultante, que es la tabla "catalogo" mostrada en la figura 3.1.5.4; por tal motivo únicamente se elimina el campo "DESCRIP" de la tabla que estamos analizando, quedando como se muestra en la figura C.4.

TABLA "ped\_det"  
N\_PEDIDO  
ALMACEN  
N\_MATERIAL  
F\_PROG  
 C\_PROG  
 C\_RECPC  
 LINE\_SEQ  
 SCHED\_SEQ  
 LINE\_STATU  
 CANTIDAD

Figura C.4. Tabla "ped\_det" resultante de eliminar el campo "DESCRIP".

Con esta eliminación, ya no existe la dependencia funcional de ningún campo con ninguna llave candidata, por lo que podemos decir que ya están estas tablas en 3NF.

Para efectos del sistema, se cambian los nombres de los campos a fin de hacerlos congruentes con las demás tablas y poder hacer relaciones entre ellas. También se eliminan algunos campos que no son útiles al sistema para evitar tener información de más.

En la figura C.5 se muestran las equivalencias de los campos que se van a utilizar en el sistema.

Pedidos	
<u>fechaintro</u>	
<u>po_num</u>	N_PEDIDO
<u>cod_sirem</u>	ALMACEN
po_date	F_PEDIDO
pos_status	PO_STATUS
vendor_num	NUM_PROV

ped_det	
<u>fechaintro</u>	
<u>po_num</u>	N_PEDIDO
<u>cod_sirem</u>	ALMACEN
<u>catalogo</u>	N_MATERIAL
<u>request_da</u>	F_PROG
qty_ordere	C_PROG
qty__to_rec	C_PROG - C_RECPC
qty_receiv	C_RECPC
cantidad	CANTIDAD
ped_prog	LINE_SEQ
marca	LINE_STATU

**Figura C.5. Equivalencia de campos en las tablas dentro del sistema.**

Se puede observar que se añadió el campo "fechaintro" a la llave principal de cada tabla, el cual va a indicar la fecha en que se registró la información, esto para poder manejar diferentes versiones de información dentro del sistema en función de la fecha con la que se registraron los datos.

## APÉNDICE D

### Generación de código

```

-----
'Archivo: fm_about
'Fecha: Julio-Septiembre de 1999
'Proposito: Explicar el proposito de
este sistema
-----

' Reg Key Security Options...
Const KEY_ALL_ACCESS =
&H2003F
' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE =
&H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1
Unicode nul terminated string
Const REG_DWORD = 4
' 32-bit number
Const gREGKEYSYSINFOLOC =
"SOFTWARE\Microsoft\Shared
Tools Location"
Const gREGVALSYSINFOLOC =
"MSINFO"
Const gREGKEYSYSINFO =
"SOFTWARE\Microsoft\Shared
Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"
Private Declare Function
RegOpenKeyEx Lib "advapi32" Alias
"RegOpenKeyExA" (ByVal hKey As
Long, ByVal lpSubKey As String,
ByVal ulOptions As Long, ByVal
samDesired As Long, ByRef
phkResult As Long) As Long
Private Declare Function
RegQueryValueEx Lib "advapi32"
Alias "RegQueryValueExA" (ByVal
hKey As Long, ByVal lpValueName
As String, ByVal lpReserved As
Long, ByRef lpType As Long, ByVal
lpData As String, ByRef lpcbData As
Long) As Long
Private Declare Function
RegCloseKey Lib "advapi32" (ByVal
hKey As Long) As Long

Private Sub Form_Load()
    lblVersion.Caption = "Version " &
App.Major & "." & App.Minor & "." &
App.Revision
    lblTitle.Caption = App.Title
End Sub

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
    Dim rc As Long
    Dim SysInfoPath As String
    ' Try To Get System Info
    Program Path\Name From
    Registry...
    If
    GetKeyValue(HKEY_LOCAL_MACH
    INE, gREGKEYSYSINFO,
    gREGVALSYSINFO, SysInfoPath)
    Then
        ' Try To Get System Info
        Program Path Only From Registry...
        Elseif
        GetKeyValue(HKEY_LOCAL_MACH
        INE, gREGKEYSYSINFOLOC,
        gREGVALSYSINFOLOC,
        SysInfoPath) Then
            ' Validate Existence Of
            Known 32 Bit File Version
            If (Dir(SysInfoPath &
            "MSINFO32.EXE") <> "") Then
                SysInfoPath =
                SysInfoPath &
                "MSINFO32.EXE"
            ' Error - File Can Not Be
            Found...
            Else
                GoTo SysInfoErr
            End If
            ' Error - Registry Entry Can Not
            Be Found...
            Else
                GoTo SysInfoErr
            End If
            Call Shell(SysInfoPath,
            vbNormalFocus)
            Exit Sub
        SysInfoErr:
            MsgBox "System Information Is
            Unavailable At This Time",
            vbOKOnly
            End Sub

    Public Function
    GetKeyValue(KeyRoot As Long,
    KeyName As String, SubKeyRef As
    String, ByRef KeyVal As String) As
    Boolean
        Dim i As Long
        ' Loop Counter
        Dim rc As Long
        ' Return Code
        Dim hKey As Long
        ' Handle To An Open Registry Key
        Dim hDepth As Long
        '
        Dim KeyValType As Long
        ' Data Type Of A Registry Key
        Dim tmpVal As String
        ' Temporary Storage For A Registry
        Key Value
        Dim KeyValSize As Long
        ' Size Of Registry Key Variable
        '
        ' Open RegKey Under KeyRoot
        (HKEY_LOCAL_MACHINE...)
        rc = RegOpenKeyEx(KeyRoot,
        KeyName, 0, KEY_ALL_ACCESS,
        hKey) ' Open Registry Key
        If (rc <> ERROR_SUCCESS)
        Then GoTo GetKeyError
        Handle Error...
        tmpVal = String$(1024, 0)
        ' Allocate Variable Space
        KeyValSize = 1024
        ' Mark Variable Size
        '
        ' Retrieve Registry Key Value...
        rc = RegQueryValueEx(hKey,
        SubKeyRef, 0, KeyValType, tmpVal,
        KeyValSize) ' Get/Create Key
        Value
        If (rc <> ERROR_SUCCESS)
        Then GoTo GetKeyError
        Handle Errors
        If (Asc(Mid(tmpVal, KeyValSize,
        1))) = 0) Then ' Win95 Adds
        Null Terminated String...
            tmpVal = Left(tmpVal,
            KeyValSize - 1) ' Null
            Found, Extract From String
            Else
                ' WinNT Does NOT Null Terminate
                String...
                tmpVal = Left(tmpVal,
                KeyValSize) ' Null Not
                Found, Extract String Only
                End If
        '
        ' Determine Key Value Type For
        Conversion...
        Select Case KeyValType
        ' Search Data Types...
        Case REG_SZ
        ' String Registry Key Data Type
            KeyVal = tmpVal
        ' Copy String Value
        Case REG_DWORD
        ' Double Word Registry Key Data
        Type
            For i = Len(tmpVal) To 1
            Step -1
                ' Convert Each
                Bit
                KeyVal = KeyVal +
                Hex(Asc(Mid(tmpVal, i, 1))) ' Build
                Value Char. By Char.
                Next
        End Sub
    End Sub

```

## Apéndice D.

```

        KeyVal = Format$("&h" +
KeyVal)
        * Convert
Double Word To String
    End Select
    GetKeyValue = True
* Return Success
    rc = RegCloseKey(hKey)
* Close Registry Key
    Exit Function
* Exit
GetKeyError: * Cleanup After An
Error Has Occured...
    KeyVal = ""
* Set Return Val To Empty String
    GetKeyValue = False
* Return Failure
    rc = RegCloseKey(hKey)
* Close Registry Key
End Function

-----
'Archivo: fm_adms
'Fecha: Julio-Septiembre de 1999
'Proposito: Captura de ADMs del
area de mantenimiento
-----

Option Explicit
Dim t_gtb As Recordset
Dim d_gdb As Database
Dim w_gws As Workspace

Private Sub cm_actualiza_Click()
Dim i As Integer
Dim i_lres As Integer
    i_lres = MsgBox(" Confirme la
Modificación ?", vbYesNo)
    If i_lres = vbYes Then
        For i = 0 To It_adms.ListCount - 1
            t_gtb.Seek "=", It_adms.List(i)
            If t_gtb.NoMatch Then
                gl_itb.AddNew
                gl_itb.Fields("usuario") =
$_login
                gl_itb.Fields("basedatos") =
$_gRutaBD
                gl_itb.Fields("tabla") =
"adms"
                gl_itb.Fields("fecha_hora") =
Date + Time
                gl_itb.Fields("Numtrans") = 1
                gl_itb.Update
                t_gtb.AddNew
                t_gtb.Fields("adm") =
It_adms.List(i)
                t_gtb.Update
            End If
            Next i
            Frame1.Enabled = False
            w_gws.CommitTrans
            cm_editar.Enabled = True
        End If
    End Sub

Private Sub
cm_actualiza_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al Presionar
el Botón, se Graban los cambios la
tabla de ADM's que haya efectuado"
End Sub

Private Sub cm_agrega_Click()
Dim i As Integer
    If It_adm.Text <> vbNullString
Then
        For i = 0 To It_adms.ListCount -
1
            If It_adm.Text =
It_adms.List(i) Then
                MsgBox "ADM ya
existente"
            End If
            Next i
            It_adms.AddItem It_adm.Text
        End If
    End Sub

Private Sub
cm_agrega_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Presione el
Botón para agregar a la tabla de
ADM's la que se tiene en el campo
ADM"
End Sub

Private Sub cm_cancela_Click()
    Frame1.Enabled = False
    w_gws.Rollback
    cm_editar.Enabled = True
End Sub

Private Sub
cm_cancela_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al presionar
el botón, cancela todos los cambios
que haya efectuado en la tabla de
ADM's"
End Sub

Private Sub cm_editar_Click()
    If t_gtb.RecordCount > 0 Then
        It_adms.Clear
        t_gtb.MoveLast
        t_gtb.MoveFirst
        While Not t_gtb.EOF
            It_adms.AddItem
t_gtb.Fields(0)
            t_gtb.MoveNext
        Wend
    End If
    Frame1.Enabled = True
    cm_editar.Enabled = False
    w_gws.BeginTrans
End Sub

End Sub

Private Sub
cm_editar_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Presione
este botón para inciar las
modificaciones a la tabla de ADM's"
End Sub

Private Sub cm_salir_Click()
    Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Regresa al
Menú: Datos para Cálculo de
Consumos"
End Sub

Private Sub Form_Load()
    Set w_gws =
DBEngine.CreateWorkspace("ADMS
", "admin", "", dbUseJet)
    Set d_gdb =
w_gws.OpenDatabase(s_gRutaBD,
True)
    Set t_gtb =
d_gdb.OpenRecordset("adms",
dbOpenTable)
    t_gtb.Index = "Primarykey"
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Posicione el
Mouse en algún boton o Campo
para Obtener ayuda de su función"
End Sub

Private Sub
Frame1_DragDrop(Source As
Control, X As Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Posicione el
Mouse en algún boton o Campo
para Obtener ayuda de su función"
End Sub

Private Sub
It_adms_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Muestra la

```

lista de ADM s que existen en el sistema actualmente"  
End Sub

```
Private Sub pc_borra_Click()
Dim i_lres As Integer
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
If It_adms ListIndex <> -1 Then
t_gtb.Seek "=",
It_adms.List(It_adms ListIndex)
If Not t_gtb.NoMatch Then
t_gtb.Delete
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
"adms"
gl_ltb.Fields("fecha_hora")
= Date + Time
gl_ltb.Fields("Numtrans") =
2
gl_ltb.Update
End If
It_adms.RemoveItem
It_adms.ListIndex
End If
End If
End Sub
```

```
Private Sub
pc_borra_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al presionar
el Bote, Elimina del sistema la ADM
que se haya seleccionado en la lista"
End Sub
```

```
Private Sub
tx_adm_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Proporcione
en este espacio la ADM que desea
Agregar al sistema"
End Sub
```

```
-----
'Archivo: fm_alma
'Fecha: Julio-Septiembre de 1999
'Propósito: Captura de Almacenes
-----
```

```
Option Explicit
Dim d_hws As Workspace
Dim d_idb As Database
Dim tb_DR As Recordset
Dim b_lHayTransaccion As Boolean
```

```
Private Sub cb_almacen_Click()
fm_mentablas.sb_mentablas.Panels(
1).Text = "Seleccione el Almacén
que desea Consultar o Modificar"
Dim i_lres As Integer
Dim s_lCata As String
Dim s_lStrId As String
Dim s_lCriterio As String
Dim s_lLinea As String
Dim s_lB Blancos As String
s_lB Blancos = " " 'cadena con 7
blancos
Text1.Text = cb_almacen.Text
s_lStrId = " cod_sirem = " +
Text1.Text + ""
tb_DR.FindFirst s_lStrId
If tb_DR.NoMatch Then
MsgBox "No hay almacenes
con ese identificador",
vbExclamation
Else
d_hws.BeginTrans
b_lHayTransaccion = True
tb_DR.Edit
Text1.Enabled = False
Frame1.Enabled = True
Frame1.Visible = True
cm_control(0).Enabled =
False
cm_control(1).Enabled =
True
cm_control(2).Enabled =
True
cm_control(3).Enabled =
True
cm_control(4).Enabled =
False
cb_almacen.Visible = False
cmd_cancela.Visible = False
'Carga a memoria
tx_codsaci.Text =
tb_DR.Fields("cod_saci")
tx_nombre.Text =
tb_DR.Fields("nombre")
tx_gpodist.Text =
tb_DR.Fields("grupodist")
cb_region.Text =
tb_DR.Fields("region")
tx_subgpo.Text =
tb_DR.Fields("subgrupo")
tx_cat.Text =
tb_DR.Fields("categoria")
tx_div.Text =
tb_DR.Fields("division")
tx_nvaDiv.Text =
tb_DR.Fields("nuevadivis")
End If
End Sub
```

```
Private Sub cm_control_Click(Index
As Integer)
Dim i_lres As Integer
Dim s_lCata As String
Dim s_lStrId As String
Dim s_lCriterio As String
Dim s_lLinea As String
Dim s_lB Blancos As String
Dim i_ltb As Recordset
```

```
s_lB Blancos = " " 'cadena con 7
blancos
Select Case Index
Case 0 'Alta
If Text1.Text = "" Or
IsNull(Text1.Text) Then
MsgBox "Asigne un
identificador", vbExclamation
Exit Sub
End If
Text1.Enabled = False
s_lStrId = " cod_sirem = " +
Text1.Text + ""
tb_DR.FindFirst s_lStrId
If Not tb_DR.NoMatch Then
MsgBox "Atención: El
identificador ya existe use otro",
vbExclamation
Exit Sub
End If
Text1.Enabled = False
d_hws.BeginTrans
b_lHayTransaccion = True
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"almacen"
gl_ltb.Fields("fecha_hora") =
Date + Time
gl_ltb.Fields("Numtrans") = 1
gl_ltb.Update
tb_DR.AddNew
'Carga a memoria
tx_codsaci.Text = ""
tx_nombre.Text = ""
tx_gpodist.Text = ""
cb_region.Text = ""
tx_subgpo.Text = ""
tx_cat.Text = ""
tx_div.Text = ""
tx_nvaDiv.Text = ""
Frame1.Enabled = True
Frame1.Visible = True
cm_control(0).Enabled =
False
cm_control(1).Enabled =
True
cm_control(2).Enabled =
True
cm_control(3).Enabled =
True
cm_control(4).Enabled =
False
Case 1 'Baja
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
tb_DR.Delete
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
gl_ltb.Fields("tabla") =
"almacen"
gl_ltb.Fields("fecha_hora")
= Date + Time
```

```

2      gl_itb.Fields("Numtrans") =
      MsgBox "Almacenes
      invalido"
      Exit Sub
      End If
      Text1.Text = ""
      Text1.Enabled = True
      d_lws.CommitTrans
      b_iHayTransaccion = False
      Text1.Text = ""
      Text1.Enabled = True
      Frame1.Enabled = False
      Frame1.Visible = False
      cm_control(0).Enabled =
      True
      cm_control(1).Enabled =
      False
      cm_control(2).Enabled =
      False
      cm_control(3).Enabled =
      False
      cm_control(4).Enabled =
      True
      Text1.Enabled = True
      End If
      Case 3 "Cancela Cambios
      d_lws.Rollback
      b_iHayTransaccion = False
      Text1.Text = ""
      Text1.Enabled = True
      Frame1.Enabled = False
      Frame1.Visible = False
      cm_control(0).Enabled =
      True
      cm_control(1).Enabled =
      False
      cm_control(2).Enabled =
      False
      cm_control(3).Enabled =
      False
      cm_control(4).Enabled =
      True
      Text1.Enabled = True
      End If
      Case 4 "Busca
      cb_almacen.Visible = True
      cmd_cancela.Visible = True
      cm_control(0).Enabled = False
      cm_control(1).Enabled = False
      cm_control(2).Enabled = False
      cm_control(3).Enabled = False
      cm_control(4).Enabled = False
      Text1.Enabled = False
      Set d_kdb =
      d_lws.OpenDatabase(s_gRutaBD,
      True)
      Set l_itb =
      d_kdb.OpenRecordset("select
      cod_sirem from almacen order by
      cod_sirem", dbOpenDynaset)
      l_itb.MoveLast
      l_itb.MoveFirst
      cb_almacen.Clear
      While Not l_itb.EOF
      cb_almacen.AddItem
      l_itb("cod_sirem")
      l_itb.MoveNext
      Wend
      End Select
      End Sub
      Private Sub
      cm_control_MouseMove(Index As
      Integer, Button As Integer, Shift As
      Integer, X As Single, Y As Single)
      fm_mentablas.sb_mentablas.Panels(
      1).Text = "Con este menú de
      botones puede Crear, Borrar o
      Actualizar el Almacén editado.
      También Cancela o Busca"
      End Sub
      Private Sub cmd_cancela_Click()
      cm_control(0).Enabled = True
      cm_control(1).Enabled = False
      cm_control(2).Enabled = False
      cm_control(3).Enabled = False
      cm_control(4).Enabled = True
      cb_almacen.Visible = False
      cmd_cancela.Visible = False
      Text1.Enabled = True
      End Sub
      Private Sub
      cmd_cancela_MouseMove(Button
      As Integer, Shift As Integer, X As
      Single, Y As Single)
      fm_mentablas.sb_mentablas.Panels(
      1).Text = "Presione este botón para
      Cancelar la Búsqueda de Almacenes
      y Regresar al Principio de la Forma"
      End Sub
      Private Sub Command1_Click()
      Unload Me
      End Sub
      Private Sub
      Command1_MouseMove(Button As
      Integer, Shift As Integer, X As
      Single, Y As Single)
      fm_mentablas.sb_mentablas.Panels(
      1).Text = "Presione este botón para
      Regresar al Menú de Tablas"
      End Sub
      Private Sub Form_Load()
      Inicializa espacio de trabajo
      CONREP para transac
      Dim l_itb As Recordset
      Set d_lws =
      DBEngine.CreateWorkspace("ALMA
      ", "admin", "", dbUseJet)
      Set d_kdb =
      d_lws.OpenDatabase(s_gRutaBD,
      True)
      Set tb_DR =
      d_kdb.OpenRecordset("almacen",
      dbOpenDynaset)
      cb_almacen.Visible = False
      cmd_cancela.Visible = False
      Set d_kdb =
      d_lws.OpenDatabase(s_gRutaBD,
      True)
      Set l_itb =
      d_kdb.OpenRecordset("select region
      from region order by region",
      dbOpenDynaset)
      l_itb.MoveLast
      l_itb.MoveFirst
  
```

```

cb_region.Clear
While Not i_Itb.EOF
    cb_region.AddItem
i_Itb("region")
i_Itb.MoveNext
Wend
b_!HayTransaccion = False
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Posicione el raton en un
botón para saber la acción que
realiza"
End Sub

```

```

Private Sub Form_Unload(Cancel As
Integer)

```

```

If b_!HayTransaccion Then
d_lws.Rollback
Exit Sub
End If
tb_DR.Close
d_ldb.Close
d_nws.Close
End Sub

```

```

Private Sub
Frame1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Posicione el raton en un
botón para saber la acción que
realiza"
End Sub

```

```

Private Sub
Label5_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Seleccione la Región a la
cual pertenece el Almacén"
End Sub

```

```

Private Sub
Text1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Proporcione en este
espacio, la clave de un Almacén
Nuevo a Registrar en el sistema"
End Sub

```

```

Private Sub tx_cat_Change()

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Inicie la categoría del
Almacén ya sea Distribuidor o
Auxiliar"
End Sub

```

```

Private Sub tx_codsaci_Change()

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Proporcione o Corrija el
Código SACI del Almacén Que está
Editando"
End Sub

```

```

Private Sub
tx_div_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Indique a que Dirección
Divisional pertenece el almacén que
está editando"
End Sub

```

```

Private Sub tx_gpodist_Change()

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Proporcione o Corrija el
Grupo de Distribución del Almacén
(Informativo)"
End Sub

```

```

Private Sub tx_nombre_Change()

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Campo para colocar o
Corregir el Nombre del Almacén que
se está Editando"
End Sub

```

```

Private Sub
tx_nvaDiv_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Proporcione el Nombre de
la nueva División a la que pertenece
el Almacén"
End Sub

```

```

Private Sub
tx_subgrp MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Teclee el Subgrupo al que
pertenece el Almacén"
End Sub

```

```

-----
'Archivo: fm_avan
'Fecha: Julio-Septiembre de 1999
'Proposito: Generación de reporte de
avance de pedidos.
-----

```

```

Option Explicit
Dim s_gPedidos As String
'Fecha de introduccion de pedidos

```

```

Dim s_gExistencias As String
'Fecha de introduccion de
existencias
Dim s_gFechaRef As String
'Fecha de Referencia
Dim s_gFechaIni As String
'Fecha de Referencia
Dim s_gFechaFin As String
'Fecha de Referencia
Dim i_gDifer As Integer
Dim d_gdb As Database
Dim q_gqfd As QueryDef
Dim q_gqfd_1 As QueryDef
Dim q_grst_1 As Recordset
'Contiene resultados de reporte
Const Long_num = 10

```

```

Sub gFormatea_reporte()
Dim i_!ArchSalida As Integer
Dim s_!Header0 As String
Dim s_!Header1 As String
Dim s_!Header2 As String
Dim i_!NumLinea As Integer
Dim i_!NumPagina As Integer
Dim s_!po_num As String
Dim s_!cod_sirem As String
Dim s_!Catalogo As String
Dim s_!descripcion As String
Dim s_!vendedor_num As String
Dim d_!sum_totprog As Double
Dim d_!por1 As Double
Dim d_!por2 As Double
Dim d_!sum_disp As Double
Dim s_!Buffer As String
'Se genera reporte principal con
porcentajes a entregar
'y porcentajes realmente
entregados hasta fecha de
referencia
Set q_gqfd_1 =
d_gdb.QueryDefs("RAVAN4")
q_gqfd_1.Parameters(0) =
s_gPedidos
Set q_grst_1 =
q_gqfd_1.OpenRecordset(dbOpenS
napshot)

```

```

'Se organizan datos y se genera
reporte
i_!ArchSalida = FreeFile()
Open tx_salava.Text For Output
As #i_!ArchSalida
i_!NumLinea = 1
i_!NumPagina = 1
s_!Header0 = " Reporte
de Avance de Pedidos"
s_!Header1 = ""
s_!Header2 = "] NUM ORD ["
s_!Header2 = s_!Header2 + " ALM
]"
s_!Header2 = s_!Header2 + "
CATALOG ["
s_!Header2 = s_!Header2 + "
DESCRIPCION ]"
s_!Header2 = s_!Header2 + "
PROVEEDOR ["
s_!Header2 = s_!Header2 + "
TOT PRO ["
s_!Header2 = s_!Header2 + "% A
ENT ]"

```





```

If CDate(tx_fechaRef.Text) >
CDate(tx_fechaFin.Text) Then
MsgBox "Fecha referencia debe
estar dentro de rango fechas".
vbExclamation
Exit Sub
End If
If CDate(tx_fechaRef.Text) <
CDate(tx_fechaIni.Text) Then
MsgBox "Fecha referencia debe
estar dentro de rango fechas".
vbExclamation
Exit Sub
End If
i_res = MsgBox(" Confirme
Generación de Reporte ? ",
vbYesNo)
If i_res = vbYes Then
"Se convierta diferencia
porcentual 21 Oct 97
i_gDifer =
Val(tx_diferencia.Text)
Set d_gdb =
OpenDatabase(s_gRutaBD, True)
gGeneraAvanca
gFormatea_reporte
End If
End Sub

```

```

Private Sub
cm_genera_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione este botón
para iniciar la generación del reporte
de Pedidos"
End Sub

```

```

Private Sub cm_imprime_Click()
Call
gImprime_archivo(tx_salava.Text)
End Sub

```

```

Private Sub
cm_imprime_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione este botón
para mandar a imprimir el archivo
indicado en el cuadro de arriba"
End Sub

```

```

Private Sub cm_salir_Click()
Unload Me
End Sub

```

```

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione este botón
para regresar al Menu de Reportes"
End Sub

```

```

Private Sub Form_Load()
Dim d_idb As Database
Dim l_tfb As Recordset
Dim l_dia As String
Dim i_mes As String
Set d_idb =
OpenDatabase(s_gRutaBD, True)
Set l_tfb =
d_idb.OpenRecordset("select distinct
fechaintro from existencias order by
fechaintro", dbOpenDynaset)
If l_tfb.RecordCount <> 0 Then
l_tfb.MoveLast
l_tfb.MoveFirst
While Not l_tfb.EOF
If Day(l_tfb("fechaintro")) < 10
Then
l_dia = "0" +
CStr(Day(l_tfb("fechaintro")))
Else
l_dia =
CStr(Day(l_tfb("fechaintro")))
End If
If Month(l_tfb("fechaintro")) <
10 Then
l_mes = "0" +
CStr(Month(l_tfb("fechaintro")))
Else
l_mes =
CStr(Month(l_tfb("fechaintro")))
End If
cb_existencias.AddItem l_dia +
"/" + l_mes + "/"
CStr(Year(l_tfb("fechaintro")))
l_tfb.MoveNext
Else
cb_existencias.AddItem ("No
hay Existencias")
End If
l_tfb.Close
Set l_tfb =
d_idb.OpenRecordset("select distinct
fechaintro from pedidos order by
fechaintro", dbOpenDynaset)
If l_tfb.RecordCount <> 0 Then
l_tfb.MoveLast
l_tfb.MoveFirst
While Not l_tfb.EOF
If Day(l_tfb("fechaintro")) < 10
Then
l_dia = "0" +
CStr(Day(l_tfb("fechaintro")))
Else
l_dia =
CStr(Day(l_tfb("fechaintro")))
End If
If Month(l_tfb("fechaintro")) <
10 Then
l_mes = "0" +
CStr(Month(l_tfb("fechaintro")))
Else
l_mes =
CStr(Month(l_tfb("fechaintro")))
End If
cb_pedidos.AddItem l_dia +
"/" + l_mes + "/"
CStr(Year(l_tfb("fechaintro")))
l_tfb.MoveNext
Wend

```

```

Else
cb_pedidos.AddItem ("No hay
Pedidos")
End If
l_tfb.Close
End Sub

```

```

Private Sub tx_existencias_Change()
End Sub

```

```

Private Sub tx_pedidos_Change()
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

```

```

Private Sub
Label1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Seleccione aquí la
fecha del archivo de pedidos que
desea utilizar"
End Sub

```

```

Private Sub
Label2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Seleccione aquí la
fecha del archivo de Existencias que
desea utilizar para ver si es crítico el
retraso en el pedido"
End Sub

```

```

Private Sub
tx_diferencia_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "De el porcentaje de
tolerancia que considerará para no
reportar una entrega incompleta de
un pedido"
End Sub

```

```

Private Sub
tx_fechaFin_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione la fecha
final a considerar en el rango de
pedidos para buscar pedidos
vencidos y generar el reporte"

```

End Sub

```
Private Sub
tx_fechaIni_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione la fecha a
partir de la cual desea realizar el
reporte de Pedidos vencidos"
End Sub
```

```
Private Sub
tx_fechaRef_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione la fecha
referencia a partir de la cual se
realizará la comparación de la fecha
de entrega de los pedidos
reportados"
End Sub
```

```
Private Sub
tx_salava_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_rhenrepor.Pane
ls(1).Text = "Nombre del archivo
donde será depositado el reporte,
también será el que se toma para
mandar a impresión"
End Sub
```

-----  
'Archivo: fm\_borra  
'Fecha: Julio-Septiembre de 1999  
'Propósito: Borra de base de datos  
existencias, pedidos y consumos  
sirem  
'la base de datos interna de Access  
-----

Option Explicit

```
Private Sub cm_borra_Click(Index
As Integer)
Dim i_lres As Integer
Dim d_ldb As Database
Dim s_ICad As String
Dim d_IFecha As Date
Dim w_IVs As Workspace
If Not
gVerifica_fecha_valida(tx_fecha.Text)
Then
MsgBox "Especifique una fecha
de entrada valida", vbExclamation
Exit Sub
End If
d_IFecha = CDate(tx_fecha.Text)
i_lres = MsgBox("Confirma
eliminación día =" +
Str(Day(d_IFecha)) + " mes =" +
Str(Month(d_IFecha)) + " año=" +
```

```
Str(Year(d_IFecha)) + " ?", vbYesNo
+ vbExclamation)
```

```
If i_lres = vbYes Then
Set w_IVs =
DBEngine.CreateWorkspace("BORR
A", "admin", "", dbUseJet)
Set d_ldb =
w_IVs.OpenDatabase(s_gRutaBD,
True)
```

```
Select Case Index
Case 0 'existencias
w_IVs.BeginTrans
s_ICad = "DELETE FROM
existencias WHERE fechaintro = #"
+ Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
existencias registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
Case 1 'pedidos
w_IVs.BeginTrans
s_ICad = "DELETE FROM
ped_det WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
pedidos registros:" +
Str(d_ldb.RecordsAffected)
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
Case 2 'Consumos
w_IVs.BeginTrans
s_ICad = "DELETE FROM
conss WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
consumos sirem registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
Case 3 'Movimientos
w_IVs.BeginTrans
s_ICad = "DELETE FROM
ment_tr WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
ent_traspaso registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
msal_tr WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
sal_traspaso registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
majuste WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
ajuste registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
ment_de WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
ent_devolucion registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
ment_de WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
ent_devolucion registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
ment_de WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
ent_devolucion registros:" +
Str(d_ldb.RecordsAffected)
```

```
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
ment_de WHERE fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
```

```
d_ldb.Execute s_ICad
MsgBox "Se borraron de
ent_devolucion registros:" +
Str(d_ldb.RecordsAffected)
```

```

i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
msal_us WHERE fechaintr = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
d_ldb.Execute s_ICad
MsgBox "Se borraron de
sal_usuario registros:" +
Str(d_ldb.RecordsAffected)
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
w_IVs.BeginTrans
s_ICad = "DELETE FROM
ment_oc WHERE fechaintr = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#"
d_ldb.Execute s_ICad
MsgBox "Se borraron de
ent_ocompra registros:" +
Str(d_ldb.RecordsAffected)
i_lres = MsgBox("Confirma
eliminaciones realizadas ?",
vbYesNo + vbExclamation)
If i_lres = vbYes Then
w_IVs.CommitTrans
Else
w_IVs.Rollback
End If
End Select
d_ldb.Close
w_IVs.Close
End Sub

Private Sub
cm_borra_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)

fm_mentablas.sb_mentablas.Panels(
1).Text = "Al Presionar estos
botones se borran los Archivos
Registrados con la fecha
seleccionada"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
cm_elige(1).Enabled = True
cm_elige(2).Enabled = True
cm_elige(3).Enabled = True
Else
MsgBox "La tabla no tiene
registros"
End If
r_ITabla.Close
d_ldBASE.Close
Else 'Tabla plana ASCII
End If
Exit Sub
termina
MsgBox "Catalogos. El archivo
especificado tiene algun problema
verifique", vbExclamation
'd_ldBASE.Close
Exit Sub
End Sub

Private Sub
cm_campos_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
cargar los campos que contiene el
Archivo dBase seleccionado arriba"
End Sub

Private Sub cm_carga_Click()
Dim d_ldBASE As Database
Dim r_ITabla As Recordset
Dim w_IVs As Workspace
'Para workspace
Dim d_ldb As Database 'Para
base de datos
Dim t_ICat As Recordset 'Para
catalogos
Dim s_IRut As String
Dim s_INom As String
Dim s_lregion As String 'La
region a asignar
Dim s_ICat As String 'Nombre
de campo a asignar a catalogo
Dim s_lUni As String 'Nombre
de campo a asignar a unidad
Dim s_lDes As String 'Nombre
de campo a asignar a descripción
Dim s_lCon As String 'Nombre
de campo a asignar a condición
Dim s_lBusca As String 'Buffer
para búsqueda de datos
Dim i_lPos As Integer 'Variable
auxiliar
Dim t_li As Integer 'Variable a
auxiliar
Dim s_lCat_tab As String
'Nombre de campo a asignar a
catalogo
Dim i_lres As Integer 'Variable
para dialogo de confirmación
If tx_cat.Text = vbNullString Then
MsgBox "Especifique campo de
catalogo", vbExclamation
Exit Sub
Else
s_ICat = tx_cat.Text
End If

Option Explicit

Private Sub cm_campos_Click()
Dim d_ldBASE As Database
Dim r_ITabla As Recordset
Dim s_IRut As String
Dim s_lNom As String
Dim t_li As Integer
If tx_archivo.Text = vbNullString
Or Dir(tx_archivo.Text) = "" Then
MsgBox "Proporcione un
nombre de archivo ", vbExclamation
Exit Sub
End If
It_campos.Clear
tx_cat.Text = ""
tx_uni.Text = ""
tx_des.Text = ""
tx_con.Text = ""

'Abrir archivo dbase
Call gObtenRutas(tx_archivo.Text,
s_IRut, s_lNom)
If True Then 'tabla_dbase then
On Error GoTo termina

' Open the external dBASE
database.
Set d_ldBASE =
OpenDatabase(s_IRut, False, False,
"dBASE III;")

' Abre tabla EXISTEN
Set r_ITabla =
d_ldBASE.OpenRecordset(s_lNom)

If r_ITabla.RecordCount > 0
Then
It_campos.Clear
For i_li = 0 To
r_ITabla.Fields.Count - 1
It_campos.AddItem
r_ITabla.Fields(i_li).Name
Next i_li
cm_elige(0).Enabled = True

```

## Apéndice D.

```

If tx_uni.Text = vbNullString Then
    MsgBox "Especifique campo de
unidad ", vbExclamation
Exit Sub
Else
    s_iUni = tx_uni.Text
End If
If tx_des.Text = vbNullString Then
    MsgBox "Especifique campo de
descripción ", vbExclamation
Exit Sub
Else
    s_iDes = tx_des.Text
End If
If tx_con.Text = vbNullString Then
    MsgBox "Especifique campo de
condición ", vbExclamation
Exit Sub
Else
    s_iCon = tx_con.Text
End If
i_fres = MsgBox(" Confirme carga
del Catálogo ? ", vbYesNo)
If i_fres = vbYes Then
    'Abrir archivo dbase
    Call gObtenRutas(tx_archivo.Text,
s_iRut, s_iNom)
    If True Then 'tabla_dbase then
        On Error GoTo termina
        ' Open the external dBASE
database.
        Set d_ldbBASE =
OpenDatabase(s_iRut, False, False,
"dBASE III;")
        ' Abre tabla EXISTEN
        Set r_ITabla =
d_ldbBASE.OpenRecordset(s_iNom)
        If r_ITabla.RecordCount > 0
Then
            r_ITabla.MoveLast
            r_ITabla.MoveFirst
            Set w_MWs =
DBEngine.CreateWorkspace("CATA"
, "admin", "", dbUseJet)
            Set d_ldb =
w_MWs.OpenDatabase(s_gRutaBD,
True)
            Set t_iCat =
d_ldb.OpenRecordset("catalogo",
dbOpenTable)
            t_iCat.Index = "PrimaryKey"
            If t_iCat.RecordCount > 0
Then
                t_iCat.MoveLast
                End If
                On Error GoTo 0 'termina3
                w_MWs.BeginTrans
                'Proceso de carga a tabla
Catalogos
                While Not r_ITabla.EOF
                    'Falta proteger en caso
que si existan
                    If Not
IsNull(r_ITabla.Fields(s_iCat)) Then
                        s_iCat_tab =
Mid(CStr(r_ITabla.Fields(s_iCat)), 1,
7)
                        t_iCat.Seek "=",
                        s_iCat_tab
                        If Not t_iCat.NoMatch
Then
                            With t_iCat
                                .Edit
                                .Fields("unidad") =
r_ITabla.Fields(s_iUni)
                                .Fields("descripcio") =
r_ITabla.Fields(s_iDes)
                                .Fields("condicion") =
r_ITabla.Fields(s_iCon)
                                .Update
                                gl_itb.AddNew
                                gl_itb.Fields("usuario") = s_login
                                gl_itb.Fields("basedatos") =
s_gRutaBD
                                gl_itb.Fields("tabla") =
"catalogo"
                                gl_itb.Fields("fecha_hora") = Date +
Time
                                gl_itb.Fields("Numtrans") = 3
                                gl_itb.Update
                                End With
                                Else
                                    With t_iCat
                                        gl_itb.AddNew
                                        gl_itb.Fields("usuario") = s_login
                                        gl_itb.Fields("basedatos") =
s_gRutaBD
                                        gl_itb.Fields("tabla") =
"catalogo"
                                        gl_itb.Fields("fecha_hora") = Date +
Time
                                        gl_itb.Fields("Numtrans") = 1
                                        gl_itb.Update
                                        .AddNew
                                        .Fields("catalogo") =
s_iCat_tab
                                        .Fields("unidad") =
r_ITabla.Fields(s_iUni)
                                        .Fields("descripcio") =
r_ITabla.Fields(s_iDes)
                                        .Fields("condicion") =
r_ITabla.Fields(s_iCon)
                                        .Update
                                        End With
                                        End If
                                        End If
                                        r_ITabla.MoveNext
                                        DoEvents
                                    Wend
                                    If vbYes = MsgBox("¿
Confirma actualización de Catalogos
?", vbYesNo) Then
                                        w_MWs.CommitTrans
                                        Else
                                            w_MWs.Rollback
                                        End If
                                    t_iCat.Close
                                    d_ldb.Close
                                    w_MWs.Close
                                    Else
                                        MsgBox "La tabla no tiene
registros"
                                        End If
                                        r_ITabla.Close
                                        d_ldbBASE.Close
                                        Else 'Tabla plana ASCII
                                        End If
                                        End If
                                        Exit Sub
                                    termina:
                                        MsgBox "Catalogos: El archivo
especificado tiene algun problema
verifique", vbExclamation
                                        'd_ldbBASE.Close
                                        Exit Sub
                                    termina2:
                                        MsgBox "Catalogos: No se
puede cargar la base de datos
principal ", vbExclamation
                                        t_iCat.Close
                                        d_ldb.Close
                                        Exit Sub
                                    termina3:
                                        MsgBox "Catalogos: Error en
carga de datos ", vbExclamation
                                        r_ITabla.Close
                                        d_ldbBASE.Close
                                        Exit Sub
                                        End Sub
                                        Private Sub
cm_carga_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
                                        fm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
Iniciar la carga del catálogo de
Materiales, con los campos
establecidos"
                                        End Sub
                                        Private Sub cm_elige_Click(Index As
Integer)
                                        Select Case Index
                                            Case 0 'Catalogo
                                                If ft_camos.ListIndex <> -1
Then
                                                    tx_uni.Text =
ft_camos.List(ft_camos.ListIndex)
                                                    Else
                                                        MsgBox "Seleccione un
campo "
                                                    End If
                                                    Case 1 'Unidad
                                                        If ft_camos.ListIndex <> -1
Then
                                                            tx_uni.Text =
ft_camos.List(ft_camos.ListIndex)
                                                            Else
                                                                MsgBox "Seleccione un
campo "
                                                            End If
                                                            Case 2 'Descripción

```

```

If It_campos.ListIndex <> -1
Then
    tx_des.Text =
It_campos.List(It_campos.ListIndex)
Else
    MsgBox "Seleccione un
campo "
End If
Case 3 "Condición
If It_campos.ListIndex <> -1
Then
    tx_con.Text =
It_campos.List(It_campos.ListIndex)
Else
    MsgBox "Seleccione un
campo "
End If
End Select
End Sub

```

```

Private Sub
cm_elige_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
que escoja el campo seleccionado,
como el que contiene el dato que
está del indicada al lado izquierdo"
End Sub

```

```

Private Sub cm_salir_Click()
Unload Me
End Sub

```

```

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
regresar al Menú de Tablas"
End Sub

```

```

Private Sub cmd_buscaarch_Click()
With dgComm1
.DialogTitle = "Busqueda de
archivo dbf de pronósticos"
.Filter = "dBaseIII (*.dbf)".dbf
.ShowOpen
If Len(FileName) = 0 Then
Exit Sub
End If
tx_archivo.Text = .FileName
End With
End Sub

```

```

Private Sub
cmd_buscaarch_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Presione el botón para
buscar el archivo en su PC"
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Posicione el raton en un
botón para saber la acción que
realiza"
End Sub

```

```

Private Sub
Frame1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Posicione el raton en un
botón para saber la acción que
realiza"
End Sub

```

```

Private Sub
Label2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Muestra el campo
seleccionado que contiene en el
archivo los Números de Catálogo"
End Sub

```

```

Private Sub
Label3_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Muestra el campo
seleccionado que contiene en el
archivo la Unidad de Medida de los
materiales del catálogo"
End Sub

```

```

Private Sub
Label4_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Muestra el campo
seleccionado que contiene en el
archivo la DESCRIPCIÓN de los
materiales del catálogo"
End Sub

```

```

Private Sub
Label5_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Muestra el campo
seleccionado que contiene en el
archivo la CONDICIÓN de los
materiales el catálogo"
End Sub

```

```

Private Sub
It_campos_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Lista de Campos que
contiene el archivo seleccionada.
Seleccione un campo para asignarlo
a algún valor en la Base de Datos"
End Sub

```

```

Private Sub
tx_archivo_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mentablas.sb_mentablas.Panels(
1).Text = "Nombre del Archivo que
contiene el catálogo de materiales"
End Sub

```

```

'-----
'Archivo: fm_cconf
'Fecha: Julio-Septiembre de 1999
'Propósito: Captura de servidores y
rutas de acceso de archivos remotos
'-----

```

```

Dim d_ldb As Database
Dim l_ltb As Recordset
Dim tg_ltb As Recordset
Dim lv_inicio As Integer

```

```

Private Sub Data1_Validate(Action
As Integer, Save As Integer)

```

```
End Sub
```

```

Private Sub cb_Config_Click()
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set tg_ltb =
d_ldb.OpenRecordset("conf_arch",
dbOpenDynaset)
l_condicion = "nombre =" + "" +
cb_Config.Text + ""
tg_ltb.FindFirst l_condicion
If IsNull(tg_ltb.Fields("dir_ip"))
Then

```

```

Text1.Text = ""
Else
Text1.Text =
tg_ltb.Fields("dir_ip")
End If
If
IsNull(tg_ltb.Fields("Clave_acceso"))
Then
Text2.Text = ""
Else
Text2.Text =
tg_ltb.Fields("Clave_acceso")
End If
If IsNull(tg_ltb.Fields("puerto"))
Then
Text3.Text = ""
Else

```

```

        Text3.Text =
    tg_ltb.Fields("puerto")
    End If
    If
    IsNull(tg_ltb.Fields("pasaporte"))
    Then
        Text4.Text = ""
    Else
        Text4.Text =
    tg_ltb.Fields("pasaporte")
    End If
    If
    IsNull(tg_ltb.Fields("ruta_remota"))
    Then
        Text6.Text = ""
    Else
        Text6.Text =
    tg_ltb.Fields("ruta_remota")
    End If
    If
    IsNull(tg_ltb.Fields("ruta_local"))
    Then
        Text7.Text = ""
    Else
        Text7.Text =
    tg_ltb.Fields("ruta_local")
    End If
    Text1.Enabled = True
    Text2.Enabled = True
    Text3.Enabled = True
    Text4.Enabled = True
    Text6.Enabled = True
    Text7.Enabled = True
    cmd_Actualiza.Enabled = True
    cmd_cancela.Enabled = True
    End Sub

    Private Sub cb_Config_Scroll()

    fm_mencarga.SB_mencarga.Panels(
    1).Text = "Seleccione Un Servidor de
    Archivos a Configurar"
    End Sub

    Private Sub cm_salir_Click()
    Unload Me
    End Sub

    Private Sub
    cm_salir_MouseMove(Button As
    Integer, Shift As Integer, X As
    Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
    1).Text = "Al Presionar el Boton,
    Regresa el Menú para Carga de
    Información"
    End Sub

    Private Sub cmd_Actualiza_Click()
    Dim l_res As Integer
    l_res = MsgBox("Confirme
    Modificación ?", vbYesNo)
    If l_res = vbYes Then
        Set d_ldb =
    OpenDatabase(s_gRutaBD, True)
        Set tg_ltb =
    d_ldb.OpenRecordset("conf_arch",
    dbOpenDynaset)

        l_condicion = "nombre = " + "" +
    cb_Config.Text + ""
        tg_ltb.FindFirst l_condicion
        tg_ltb.Edit
        tg_ltb.Fields("dir_ip").Value =
    Text1.Text

        tg_ltb.Fields("clave_acceso").Value
    = Text2.Text
        tg_ltb.Fields("puerto").Value =
    Text3.Text
        tg_ltb.Fields("pasaporte").Value =
    Text4.Text
        tg_ltb.Fields("ruta_remota").Value
    = Text6.Text
        tg_ltb.Fields("ruta_local").Value =
    Text7.Text
        tg_ltb.Update
        gl_ltb.AddNew
        gl_ltb.Fields("usuario") = s_login
        gl_ltb.Fields("basedatos") =
    s_gRutaBD
        gl_ltb.Fields("tabla") = "conf_arch"
        gl_ltb.Fields("fecha_hora") = Date
    + Time
        gl_ltb.Fields("Numtrans") = 3
        gl_ltb.Update
        MsgBox("Registro actualizado")
        Text1.Enabled = False
        Text2.Enabled = False
        Text3.Enabled = False
        Text4.Enabled = False
        Text6.Enabled = False
        Text7.Enabled = False
        cmd_Actualiza.Enabled = False
        cmd_cancela.Enabled = False
        Text1.Text = ""
        Text2.Text = ""
        Text3.Text = ""
        Text4.Text = ""
        Text6.Text = ""
        Text7.Text = ""
    End If
    End Sub

    Private Sub
    cmd_Actualiza_MouseMove(Button
    As Integer, Shift As Integer, X As
    Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
    1).Text = "Al Presionar este boton,
    se Graban los datos que se
    Presentan para el Servidor
    Seleccionado"
    End Sub

    Private Sub cmd_cancela_Click()
    Text1.Enabled = False
    Text2.Enabled = False
    Text3.Enabled = False
    Text4.Enabled = False
    Text6.Enabled = False
    Text7.Enabled = False
    cmd_Actualiza.Enabled = False
    cmd_cancela.Enabled = False
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = ""
    End Sub

    Text6.Text = ""
    Text7.Text = ""
    End Sub

    Private Sub
    fm_mencarga.SB_mencarga.Panels(
    1).Text = "Al Precional este boton,
    Se CANCELAN todos los cambios y
    queda sin afectación el Registro del
    Servidor Seleccionado"
    End Sub

    Private Sub Form_Load()
    Set d_ldb =
    OpenDatabase(s_gRutaBD, True)
    Set t_ltb =
    d_ldb.OpenRecordset("select
    nombre from conf_arch order by
    nombre", dbOpenSnapshot)
    cb_Config.Clear
    t_ltb.MoveLast
    t_ltb.MoveFirst
    While Not t_ltb.EOF
        cb_Config.AddItem
    t_ltb.Fields("nombre")
    t_ltb.MoveNext
    Wend
    t_ltb.Close
    d_ldb.Close
    Text1.Enabled = False
    Text2.Enabled = False
    Text3.Enabled = False
    Text4.Enabled = False
    Text6.Enabled = False
    Text7.Enabled = False
    cmd_Actualiza.Enabled = False
    cmd_cancela.Enabled = False
    Text1.Text = "_____ "
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = ""
    Text6.Text = ""
    Text7.Text = ""
    End Sub

    Private Sub
    Form_MouseMove(Button As
    Integer, Shift As Integer, X As
    Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
    1).Text = "Posicione el Mouse en un
    Campo o Boton para Obtener
    Ayuda"
    End Sub

    Private Sub
    Frame1_MouseMove(Button As
    Integer, Shift As Integer, X As
    Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
    1).Text = "Seleccione un Servidaro
    con la Flacha en el campo de
    Descripción de Servidor, para
    Editarlo"

```

```

End Sub

Private Sub
Text1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(1).Text = "Proporcione la Direccion IP del Servidor que contiene el Archivo Indicado"
End Sub

Private Sub
Text2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(1).Text = "Proporciona la CLAVE DE ACCESO al Servidor de archivos"
End Sub

Private Sub
Text3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(1).Text = "Proporcione el Puerto para Conectarse al Servidor que está Configurando"
End Sub

Private Sub
Text4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(1).Text = "Proporcione la CLAVE SECRETA que se requiere para tener acceso a este Servidor"
End Sub

Private Sub
Text6_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(1).Text = "Proporcione la Ruta por Defecto en el Equipo Servidor, donde se va a ENCONTRAR el Archivo"
End Sub

Private Sub Text7_Change()

fm_mencarga.SB_mencarga.Panels(1).Text = "Proporcione la Ruta por Defecto en el SU Equipo PC, donde se va a DEPOSITAR el Archivo"
End Sub

'-----
'Archivo: fm_cconm

'Fecha Julio-Septiembre de 1999
'Proposito. Carga de archivo de movimientos en formato ASCII hacia 'las diferentes tablas de movimientos.
'-----

Option Explicit
Dim s_gClave As String
Dim s_gPassword As String
Dim s_gDirFTP As String
Dim w_gws As Workspace
'Espacio de trabajo
Dim d_gdb As Database 'Base de datos
Dim t_gEC As Recordset 'ENTRADAS X ORDEN DE COMPRA
Dim t_gSU As Recordset 'SALIDAS A USUARIO
Dim t_gED As Recordset 'ENTRADAS X DEVOLUCION
Dim t_gAJ As Recordset 'AJUSTES
Dim t_gST As Recordset 'SALIDAS X TRASPASO
Dim t_gET As Recordset 'ENTRADAS X TRASPASO

Private Sub lLee_Movimientos()
Dim MyRecord As RecArch
Declare variable.
Dim cabe As cabeza
Dim retorno1 As retorno
Dim saldo1 As saldo
Dim i_gArchivoEntrada As Integer
Dim s_idirarchivo As String
Dim i_cuenta As Integer
Dim s_llident As String

'Para control barras de progreso
Dim l_iNum As Long
Dim l_iCount As Long
Dim d_lres As Double
Dim d_lporc As Double
Dim l_itam As Long
Dim f_lFecha As Date
pb_mov.Value = 0
DoEvents
On Error GoTo Maneja_error
f_lFecha = CDate(tx_fecha.Text)
Set w_gws = DBEngine.CreateWorkspace("MOV" & "admin", "", dbUseJet)
Set d_gdb = w_gws.OpenDatabase(s_gRutaBD, True)
Set t_gEC = d_gdb.OpenRecordset("ment_oc", dbOpenTable)
Set t_gSU = d_gdb.OpenRecordset("msal_us", dbOpenTable)
Set t_gED = d_gdb.OpenRecordset("ment_de", dbOpenTable)
Set t_gAJ = d_gdb.OpenRecordset("majuste", dbOpenTable)

Set t_gST = d_gdb.OpenRecordset("msal_tr", dbOpenTable)
Set t_gET = d_gdb.OpenRecordset("ment_tr", dbOpenTable)
On Error GoTo Maneja_error2
s_idirarchivo = tx_destino.Text
i_gArchivoEntrada = FreeFile()
l_itam = FileLen(s_idirarchivo)
l_iNum = (l_itam - 225) / 379
Open s_idirarchivo For Binary As #i_gArchivoEntrada
l_iCount = 1
d_lporc = 2
Get i_gArchivoEntrada, , cabe
w_gws.BeginTrans
On Error GoTo Maneja_error3
Get i_gArchivoEntrada, , MyRecord
While Not EOF(l_gArchivoEntrada)
s_llident = Mid(MyRecord.T_MOV, 1, 2)
Select Case s_llident
Case "EC"
With t_gEC
gl_ltb.AddNew
gl_ltb.Fields("usuario") = s_login
gl_ltb.Fields("basedatos") = s_gRutaBD
gl_ltb.Fields("tabla") = "ment_oc"
gl_ltb.Fields("fecha_hora") = Date + Time
gl_ltb.Fields("Numtrans") = 1
gl_ltb.Update
AddNew
!fechaintr = f_lFecha
!JUN = MyRecord.UN
!AR = MyRecord.AR
!POB = MyRecord.PO
!NAF = MyRecord.NAF
!CTA = MyRecord.CTA
!SCTA = MyRecord.SCTA
!detscta = MyRecord.xx
!A_R = MyRecord.A_R
!ALM = MyRecord.ALM
!CAT = MyRecord.CAT
!PED = MyRecord.PED
!SERie = MyRecord.SER
!F_CAP = MyRecord.F_CAP
!F_U_C = MyRecord.F_U_C
!CDO = MyRecord.CDO
!PROV = MyRecord.PROV
!UNI = MyRecord.UNI
!DOC = MyRecord.DOC
!Remi = MyRecord.Remi
!CANT = Val(MyRecord.CANT)
.Update
End With
Case "SU"
With t_gSU
gl_ltb.AddNew
gl_ltb.Fields("usuario") = s_login
gl_ltb.Fields("basedatos") = s_gRutaBD

```



```

    gl_itb.Fields("tabla") =
    "msal_us"
    gl_itb.Fields("fecha_hora") =
    Date + Time
    gl_itb.Fields("Numtrans") = 1
    gl_itb.Update
    AddNew
    Ifechaintro = f_IFecha
    !UN = MyRecord.UN
    !AR = MyRecord.AR
    !POb = MyRecord.PO
    !NAF = MyRecord.NAF
    !CTA = MyRecord.CTA
    !SubCTA = MyRecord.SCTA
    !detscta = MyRecord.xx
    !A_S = MyRecord.A_S
    !ALM = MyRecord.ALM
    !CAT = MyRecord.CAT
    !SER = MyRecord.SER
    !DEPTO = MyRecord.DEPTO
    !MOT_S = MyRecord.MOT_S
    !F_CAP = MyRecord.F_CAP
    !F_U_C = MyRecord.F_U_C
    !ZONA = MyRecord.ZONA
    !CDO = MyRecord.CDO
    !UNI = MyRecord.UNI
    !DOC = MyRecord.DOC
    !ADM = MyRecord.ADM
    !EXPED = MyRecord.EXPED
    !CATEG =
    MyRecord.CATEG
    !CANT =
    Val(MyRecord.CANT)
    .Update
    End With
    Case "ED"
    With t_gED
    gl_itb.AddNew
    gl_itb.Fields("usuario") =
    s_login
    gl_itb.Fields("basedatos") =
    s_gRutaBD
    gl_itb.Fields("tabla") =
    "ment_de"
    gl_itb.Fields("fecha_hora") =
    Date + Time
    gl_itb.Fields("Numtrans") = 1
    gl_itb.Update
    AddNew
    Ifechaintro = f_IFecha
    !UN = MyRecord.UN
    !AR = MyRecord.AR
    !POb = MyRecord.PO
    !NAF = MyRecord.NAF
    !CTA = MyRecord.CTA
    !SCTA = MyRecord.SCTA
    !detscta = MyRecord.xx
    !A_R = MyRecord.A_R
    !ALM = MyRecord.ALM
    !CAT = MyRecord.CAT
    !SERie = MyRecord.SER
    !DEPTO = MyRecord.DEPTO
    !MOT_S = MyRecord.MOT_S
    !F_CAP = MyRecord.F_CAP
    !F_U_C = MyRecord.F_U_C
    !ZONA = MyRecord.ZONA
    !CDO = MyRecord.CDO
    !UNI = MyRecord.UNI
    !DOC = MyRecord.DOC
    !ADM = MyRecord.ADM
    !EXPED = MyRecord.EXPED
    !CANT =
    Val(MyRecord.CANT)
    .Update
    End With
    Case "AJ"
    With t_gAJ
    gl_itb.AddNew
    gl_itb.Fields("usuario") =
    s_login
    gl_itb.Fields("basedatos") =
    s_gRutaBD
    gl_itb.Fields("tabla") =
    "majuste"
    gl_itb.Fields("fecha_hora") =
    Date + Time
    gl_itb.Fields("Numtrans") = 1
    gl_itb.Update
    AddNew
    Ifechaintro = f_IFecha
    !ALM = MyRecord.ALM
    !CAT = MyRecord.CAT
    !SER = MyRecord.SER
    !F_CAP = MyRecord.F_CAP
    !CDO = MyRecord.CDO
    !UNI = MyRecord.UNI
    !CANT =
    Val(MyRecord.CANT)
    .Update
    End With
    Case "ST"
    With t_gST
    gl_itb.AddNew
    gl_itb.Fields("usuario") =
    s_login
    gl_itb.Fields("basedatos") =
    s_gRutaBD
    gl_itb.Fields("tabla") =
    "msal_tr"
    gl_itb.Fields("fecha_hora") =
    Date + Time
    gl_itb.Fields("Numtrans") = 1
    gl_itb.Update
    AddNew
    Ifechaintro = f_IFecha
    !UN = MyRecord.UN
    !AR = MyRecord.AR
    !POb = MyRecord.PO
    !NAF = MyRecord.NAF
    !CTA = MyRecord.CTA
    !SubCTA = MyRecord.SCTA
    !detscta = MyRecord.xx
    !A_S = MyRecord.A_S
    !A_R = MyRecord.A_R
    !ALM = MyRecord.ALM
    !CAT = MyRecord.CAT
    !SER = MyRecord.SER
    !F_CAP = MyRecord.F_CAP
    !F_U_C = MyRecord.F_U_C
    !ZONA = MyRecord.ZONA
    !CDO = MyRecord.CDO
    !UNI = MyRecord.UNI
    !DOC = MyRecord.CDO
    !ADM = MyRecord.ADM
    !EXPED = MyRecord.EXPED
    !CANT =
    Val(MyRecord.CANT)
    .Update
    End With
    Case "ET"
    With t_gET
    gl_itb.AddNew
    gl_itb.Fields("usuario") =
    s_login
    gl_itb.Fields("basedatos") =
    s_gRutaBD
    gl_itb.Fields("tabla") =
    "ment_tr"
    gl_itb.Fields("fecha_hora") =
    Date + Time
    gl_itb.Fields("Numtrans") = 1
    gl_itb.Update
    AddNew
    Ifechaintro = f_IFecha
    !UN = MyRecord.UN
    !AR = MyRecord.AR
    !POb = MyRecord.PO
    !NAF = MyRecord.NAF
    !CTA = MyRecord.CTA
    !SCTA = MyRecord.SCTA
    !detscta = MyRecord.xx
    !A_S = MyRecord.A_S
    !A_R = MyRecord.A_R
    !ALM = MyRecord.ALM
    !CAT = MyRecord.CAT
    !SER = MyRecord.SER
    !F_CAP = MyRecord.F_CAP
    !F_U_C = MyRecord.F_U_C
    !CDO = MyRecord.CDO
    !UNI = MyRecord.UNI
    !DOC = MyRecord.DOC
    !CANT =
    Val(MyRecord.CANT)
    .Update
    End With
    Case Else
    MsgBox "Elemento con t_mov
    = " + MyRecord.T_MOV + " no
    reconocido"
    End Select
    l_Count = l_Count + 1
    d_ires = CDBl(l_Count /
    l_Num) * 100#
    If d_ires > d_iporc And d_iporc
    < 98 Then
    pb_mov.Value = d_iporc
    d_iporc = d_iporc + 2
    End If
    DoEvents
    Get t_gArchivoEntrada, .
    retorno1
    If retorno1 retom = "" Then
    Get t_gArchivoEntrada, .
    saldo1

```

```

End If
Get t_gArchivoEntrada, .
MyRecord
Wend
Close #i_gArchivoEntrada
If vbYes = MsgBox("¿ Confirma
actualización de movimientos ?
Cargando " + Str(L_Count) + "
registros", vbYesNo) Then
w_gws.CommitTrans
Else
w_gws.Rollback
End If
MsgBox "tarea terminada"
t_gEC.Close
t_gSU.Close
t_gED.Close
t_gAJ.Close
t_gST.Close
t_gET.Close
d_gdb.Close
w_gws.Close
Exit Sub

```

```

Maneja_error:
MsgBox "Se presento un error en
la base de datos o tabla en lectura,
verifique", vbExclamation
Exit Sub

```

```

Maneja_error2:
MsgBox "Se presento un error (2)
en la base de datos o tabla en
lectura, verifique", vbExclamation
t_gEC.Close
t_gSU.Close
t_gED.Close
t_gAJ.Close
t_gST.Close
t_gET.Close
d_gdb.Close
w_gws.Close
Exit Sub

```

```

Maneja_error3:
MsgBox "Se presento un error (3)
en la base de datos o tabla en
lectura, verifique", vbExclamation
w_gws.Rollback
t_gEC.Close
t_gSU.Close
t_gED.Close
t_gAJ.Close
t_gST.Close
t_gET.Close
d_gdb.Close
w_gws.Close
Exit Sub
End Sub

```

```

Private Sub cm_cargabd_Click()
Dim f_IFecha As Date
Dim d_ldb As Database
Dim r_icount As Recordset
If Not
gVerifica_fecha_valida(tx_fecha.Text
) Then ' = vbNullString Then
MsgBox "Especifique una fecha
de introducción valida".
vbExclamation
Exit Sub

```

```

End If
If tx_destino.Text = "" Then
MsgBox "Debe especificar el
nombre y directorio del archivo
donde se tienen los Movimientos",
vbExclamation
Exit Sub
End If
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
f_IFecha = CDate(tx_fecha.Text)
Set r_icount =
d_ldb.OpenRecordset("select
count(*) from majuste where
fechaintro = #" +
Str(Month(f_IFecha)) + "/" +
Str(Day(f_IFecha)) + "/" +
Str(Year(f_IFecha)) + "#")
If r_icount.Fields(0) > 0 Then
MsgBox "Datos con la fecha
intro ya existen"
r_icount.Close
d_ldb.Close
Exit Sub
End If
r_icount.Close
Set r_icount =
d_ldb.OpenRecordset("select
count(*) from ment_de where
fechaintro = #" +
Str(Month(f_IFecha)) + "/" +
Str(Day(f_IFecha)) + "/" +
Str(Year(f_IFecha)) + "#")
If r_icount.Fields(0) > 0 Then
MsgBox "Datos con la fecha
intro ya existen"
r_icount.Close
d_ldb.Close
Exit Sub
End If
r_icount.Close
Set r_icount =
d_ldb.OpenRecordset("select
count(*) from ment_oc where
fechaintro = #" +
Str(Month(f_IFecha)) + "/" +
Str(Day(f_IFecha)) + "/" +
Str(Year(f_IFecha)) + "#")
If r_icount.Fields(0) > 0 Then
MsgBox "Datos con la fecha
intro ya existen"
r_icount.Close
d_ldb.Close
Exit Sub
End If
r_icount.Close
Set r_icount =
d_ldb.OpenRecordset("select
count(*) from ment_tr where
fechaintro = #" +
Str(Month(f_IFecha)) + "/" +
Str(Day(f_IFecha)) + "/" +
Str(Year(f_IFecha)) + "#")
If r_icount.Fields(0) > 0 Then
MsgBox "Datos con la fecha
intro ya existen"
r_icount.Close
d_ldb.Close
Exit Sub
End If

```

```

r_icount.Close
Set r_icount =
d_ldb.OpenRecordset("select
count(*) from msal_tr where
fechaintro = #" +
Str(Month(f_IFecha)) + "/" +
Str(Day(f_IFecha)) + "/" +
Str(Year(f_IFecha)) + "#")
If r_icount.Fields(0) > 0 Then
MsgBox "Datos con la fecha
intro ya existen"
r_icount.Close
d_ldb.Close
Exit Sub
End If
r_icount.Close
Set r_icount =
d_ldb.OpenRecordset("select
count(*) from msal_us where
fechaintro = #" +
Str(Month(f_IFecha)) + "/" +
Str(Day(f_IFecha)) + "/" +
Str(Year(f_IFecha)) + "#")
If r_icount.Fields(0) > 0 Then
MsgBox "Datos con la fecha
intro ya existen"
r_icount.Close
d_ldb.Close
Exit Sub
End If
r_icount.Close
d_ldb.Close
Call lLee_Movimientos "Lee y
carga a tablas de mov's"
End Sub

```

```

Private Sub
cm_cargabd_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Realiza la carga del
Archivo Copiado, hacia la Base de
Datos, Registrándola con la fecha
indicada"
End Sub

```

```

Private Sub cm_ftp_Click()
Dim s_iComando As String
Dim s_iOrigen As String
Dim s_iDestino As String
Dim v_iReturnValue As Variant
Dim i_li As Integer
If tx_origen.Text = "" Then
MsgBox "Debe especificar el
nombre y la ruta del archivo origen
de Movimientos", vbExclamation
Exit Sub
End If
If tx_destino.Text = "" Then
MsgBox "Debe especificar el
nombre y directorio del archivo
donde se colocarán los
Movimientos", vbExclamation
Exit Sub
Else
s_iOrigen = tx_origen.Text
s_iDestino = tx_destino.Text

```

```

Call
gConstruyeComandoFTP(s_gClave,
s_gPassword, s_lOrigen, s_lDestino)
s_lComando =
"c:\WINDOWS\FTP EXE " + "
s:\sic\comftp.txt" + " " + s_gDirFTP
v_lReturnVal =
Shell(s_lComando, vbNormalFocus)
For i_li = 0 To 1000
DoEvents
Next i_li
End If
End Sub

```

```

Private Sub
cm_ftp_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Ejecuta la Copia via FTP
del Archivo desde el Archivo Remoto
a su Equipo"
End Sub

```

```

Private Sub cm_salir_Click()
Unload Me
End Sub

```

```

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Regresa al Menú para
Carga de Información"
End Sub

```

```

Private Sub Form_Load()
Dim s_lKey As String
Dim d_lDb As Database
Dim t_ltb As Recordset
Set d_lDb =
OpenDatabase(s_gRutaBD, True)
Set t_ltb =
d_lDb.OpenRecordset("conf_arch",
dbOpenSnapshot)
With t_ltb
s_lKey = " nombre =
'movimientos"
.FindFirst s_lKey
If Not .NoMatch Then
If Not
IsNull(.Fields("clave_acceso")) Then
tx_origen.Text =
.Fields("ruta_remota")
End If
If Not
IsNull(.Fields("clave_acceso")) Then
tx_destino.Text =
.Fields("ruta_local")
End If
If Not
IsNull(.Fields("clave_acceso")) Then
s_gClave =
.Fields("clave_acceso")
Else
s_gClave = ""
End If

```

```

If Not
IsNull(.Fields("pasaporte")) Then
s_gPassword =
.Fields("pasaporte")
Else
s_gPassword = ""
End If
If Not IsNull(.Fields("dir_ip"))
Then
s_gDirFTP =
.Fields("dir_ip")
Else
s_gDirFTP = ""
End If
End If
End With
t_ltb.Close
d_lDb.Close
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Posicione el Mouse en Un
Campo o Boton para Saber su
función."
End Sub

```

```

Private Sub
pb_mov_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Marca el Avance en la
Carga de los datos de
MOVIMIENTOS a la Base de Datos"
End Sub

```

```

Private Sub
tx_destino_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Indica la Ruta y Nombre
local en donde se desea colocar el
archivo de MOVIMIENTOS Copiado
del equipo Remoto"
End Sub

```

```

Private Sub
tx_fecha_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Teclee la Fecha con la
que desea Ingresar los datos de
MOVIMIENTOS a la Base de Datos"
End Sub

```

```

Private Sub
tx_origen_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Indica la Ruta u nombre
donde se encuentra el Archivo de
MOVIMIENTOS a Traer"
End Sub

```

```

-----
'Archivo: fm_cep
'Fecha: Julio-Septiembre de 1999
'Propósito: Transferencia via FTP de
archivos de existencias y pedidos.
'Carga de estos archivos en dBase
hacia la base de datos interna en
Access
-----

```

```

Option Explicit
Dim s_gFecha As String
Dim s_gClaveExis As String
Dim s_gPasswordExis As String
Dim s_gDirFTPExis As String
Dim s_gClavePed As String
Dim s_gPasswordPed As String
Dim s_gDirFTPPed As String

```

```

'Proposito: Valida información y sube
a Access. Carga de acuerdo
'a tipo de archivo: dBase o plano.
Sub lCarga_existencias()
Dim d_lDbASE As Database
Dim r_lExis As Recordset
Dim d_lDb As Database
Dim t_lExis As Recordset
Dim w_lWs As Workspace
Dim s_lRut As String
Dim s_lNom As String

```

```

'Para control barras de progreso
Dim l_lNum As Long
Dim l_lCount As Long
Dim d_lRes As Double
Dim d_lPorc As Double
Dim l_lRes As Integer
Dim r_lCount As Recordset
Dim d_lFecha As Date

```

```

'Carga de existencias
PARAMETERS (Fecha_entrada)
DateTime;
'INSERT INTO existencias (
FechaIntro, cod_sirem, catalogo,
on_hand, reservados, disponible )
SELECT (Fecha_entrada),
whse_code, num_mat, on_hand,
reservados, disponible
FROM exis0800;
PB_exi.Value = 0
DoEvents
Call
gObtenRutas(tx_exisdes.Text,
s_lRut, s_lNom)
If True Then tabla_dbase then
On Error GoTo termina

' Open the external dBASE
database

```

```

Set d_IdBASE =
OpenDatabase(s_IrUt, False, False,
"dBASE III.")

' Abre tabla EXISTEN
Set r_IExis =
d_IdBASE.OpenRecordset(s_INom)

If r_IExis.RecordCount > 0 Then
    i_INum =
r_IExis.RecordCount
    r_IExis.MoveLast
    r_IExis.MoveFirst
    On Error GoTo termina2
    Set w_IWs =
DBEngine.CreateWorkspace("EXIS",
"admin", "", dbUseJet)
    Set d_Idb =
w_IWs.OpenDatabase(s_gRutaBD,
True)
    Set t_IExis =
d_Idb.OpenRecordset("existencias",
dbOpenTable)
    t_IExis.Index = "PrimaryKey"
    d_IFecha =
CDate(tx_fecha.Text)
    Set r_Icoun =
d_Idb.OpenRecordset("select
count(*) from existencias where
fechaintro = #" +
Str(Month(d_IFecha)) + "/" +
Str(Day(d_IFecha)) + "/" +
Str(Year(d_IFecha)) + "#")
    If r_Icoun.Fields(0) > 0 Then
        MsgBox "Datos con ta
fecha intro ya existen"
        r_Icoun.Close
        t_IExis.Close
        d_Idb.Close
        w_IWs.Close
        r_IExis.Close
        d_IdBASE.Close
        Exit Sub
    End If
    i_ICount = 0
    d_Iporc = 10
    On Error GoTo termina3
    w_IWs.BeginTrans
    While Not r_IExis.EOF
        t_IExis.Seek "=",
CDate(tx_fecha.Text),
r_IExis.Fields("num_mat"),
r_IExis.Fields("almacen")
        If t_IExis.NoMatch Then
            With t_IExis
                gl_Itb.AddNew
                gl_Itb.Fields("usuario") =
s_login
                gl_Itb.Fields("basedatos") =
s_gRutaBD
                gl_Itb.Fields("tabla") =
"existencias"
                gl_Itb.Fields("fecha_hora") = Date +
Time
                gl_Itb.Fields("Numtrans") = 1
                gl_Itb.Update
            End With
        End If
    End While
    w_IWs.CommitTrans
    w_IWs.Rollback
    t_IExis.Close
    d_Idb.Close
    w_IWs.Close
    r_IExis.Close
    d_IdBASE.Close
    Exit Sub

```

```

AddNew
.Fields("fechaintro") =
CDate(tx_fecha.Text)
.Fields("cod_sirem") =
r_IExis.Fields("almacen")
.Fields("catalogo") =
r_IExis.Fields("num_mat")
.Fields("on_hand") =
r_IExis.Fields("cantidad")
.Fields("reservados") =
r_IExis.Fields("reservados")
.Fields("disponible") =
r_IExis.Fields("disponible")
.Update
End With
Else
    i_Ires = MsgBox("Hay
elementos repetidos, cancela?",
vbYesNo + vbExclamation)
    If i_Ires = vbYes Then
        w_IWs.Rollback
        t_IExis.Close
        d_Idb.Close
        w_IWs.Close
        r_IExis.Close
        d_IdBASE.Close
        Exit Sub
    End If
    End If
    r_IExis.MoveNext
    i_ICount = i_ICount + 1
    d_Ires = CDb(i_ICount /
i_INum) * 100#
    If d_Ires > d_Iporc Then
        PB_exi.Value = d_Iporc
        d_Iporc = d_Iporc + 10
    End If
    DoEvents
    Wend
    If vbYes = MsgBox("¿
Confirma adición de existencias?
Cargando " + Str(i_ICount) + "
registros", vbYesNo) Then
        w_IWs.CommitTrans
    Else
        w_IWs.Rollback
    End If
    t_IExis.Close
    d_Idb.Close
    w_IWs.Close
    r_IExis.Close
    d_IdBASE.Close
    Else 'Tabla plana ASCII
    End If
    PB_exi.Value = 0
    Exit Sub

```

```

Exit Sub
termina3:
MsgBox "Existencias Error en
carga de datos ", vbExclamation
t_IExis.Close
d_Idb.Close
w_IWs.Close
r_IExis.Close
d_IdBASE.Close
Exit Sub
End Sub

'Propósito: Valida información y sube
a Access. Carga de acuerdo
'a tipo de archivo: dBase o plano.
Sub ICarga_pedidos()
Dim d_IdBASE As Database
Dim r_IPed As Recordset
Dim d_Idb As Database
Dim t_IPed As Recordset
Dim w_IWs As Workspace
Dim s_IrUt As String
Dim s_INom As String

'Para control barras de progreso
Dim i_INum As Long
Dim t_ICount As Long
Dim d_Ires As Double
Dim d_Iporc As Double
Dim i_Ires As Integer
Dim r_ICount As Recordset
Dim d_IFecha As Date

' Carga de pedidos
PARAMETERS [Fecha]
DateTime;
INSERT INTO ped_det (
fechaintro, po_num, cod_sirem,
catalogo, request_da, qty_ordere,
qty_to_rec, qty_receiv, cantidad,
ped_prog )
SELECT [Fecha], po_num,
whse_code, item_num, request_da,
qty_ordere, qty_to_rec, qty_receiv,
qty_ordere-qty_receiv, 0
FROM ped0718;
PB_ped.Value = 0
DoEvents
Call gObtenRutas(tx_peddes.Text,
s_IrUt, s_INom)
If True Then 'tabla_dbase then
    On Error GoTo termina

' Open the external dBASE
database.
Set d_IdBASE =
OpenDatabase(s_IrUt, False, False,
"dBASE III.")

' Abre tabla EXISTEN
Set r_IPed =
d_IdBASE.OpenRecordset(s_INom)

If r_IPed.RecordCount > 0 Then
    i_INum =
r_IPed.RecordCount
    r_IPed.MoveLast
    r_IPed.MoveFirst
    On Error GoTo termina2

```



```

d_IdBASE Close
Exit Sub
End Sub

Private Sub ck_exis_Click()
If ck_exis.Value = False Then
ck_ped.Enabled = True
tx_rutaped.Enabled = True
tx_peddes.Enabled = True
Label3.Enabled = True
Label5.Enabled = True
cm_frp.Enabled = False
cm_cargabd.Enabled = False
Else
ck_ped.Enabled = False
tx_rutaped.Enabled = False
tx_peddes.Enabled = False
Label3.Enabled = False
Label5.Enabled = False
cm_frp.Enabled = True
cm_cargabd.Enabled = True
End If
End Sub

Private Sub
ck_exis_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Al Marcar, indica que
desea Extraer el Archivo de
EXISTENCIAS para Cargarlo con la
Fecha Indicada"
End Sub

Private Sub ck_ped_Click()
If ck_ped.Value = False Then
ck_exis.Enabled = True
tx_rutaaxis.Enabled = True
tx_exisdes.Enabled = True
Label2.Enabled = True
Label4.Enabled = True
cm_frp.Enabled = False
cm_cargabd.Enabled = False
Else
ck_exis.Enabled = False
tx_rutaaxis.Enabled = False
tx_exisdes.Enabled = False
Label2.Enabled = False
Label4.Enabled = False
cm_frp.Enabled = True
cm_cargabd.Enabled = True
End If
End Sub

Private Sub
ck_ped_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Al Marcar, indica que
desea Extraer el Archivo de
PEDIDOS para Cargarlo con la
Fecha Indicada"
End Sub

Private Sub cm_cargabd_Click()
Dim s_IComando As String
Dim s_IDestino As String
Dim v_IReturnValue As Variant
Dim i_li As Integer
If Not
gVerifica_fecha_valida(tx_fecha.Text
) Then ' = vbNullString Then
MsgBox "Especifique una fecha
de captura", vbExclamation
Exit Sub
End If
s_gFecha = Format(tx_fecha.Text,
"dd/mm/yyyy")
If ck_exis.Value = 1 Then
If tx_exisdes.Text = "" Then
MsgBox "Debe especificar el
nombre y directorio donde colocar el
archivo de Existencias",
vbExclamation
Exit Sub
Else
'Abre directamente bd
ICarga_existencias
End If
End If
If ck_ped.Value = 1 Then
If tx_peddes.Text = "" Then
MsgBox "Debe especificar el
nombre y directorio donde colocar el
archivo de Pedidos", vbExclamation
Exit Sub
Else
ICarga_pedidos
End If
End If
End Sub

Private Sub
cm_cargabd_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Realiza la carga del
Archivo Copiado, hacia la Base de
Datos. Registrándola con la fecha
indicada"
End Sub

Private Sub cm_frp_Click()
Dim s_IComando As String
Dim s_IOrigen As String
Dim s_IDestino As String
Dim v_IReturnValue As Variant
Dim i_li As Integer
If Not
gVerifica_fecha_valida(tx_fecha.Text
) Then ' = vbNullString Then
MsgBox "Especifique una fecha
de captura", vbExclamation
Exit Sub
End If
s_gFecha = Format(tx_fecha.Text,
"dd/mm/yyyy")
If ck_exis.Value = 1 Then
If tx_rutaaxis.Text = "" Then
MsgBox "Especifique la ruta
donde se encuentra el archivo de
Existencias", vbExclamation
Exit Sub
Else
s_IOrigen = tx_rutaaxis.Text
s_IDestino = tx_exisdes.Text
Call
gConstruyeComandoFTP(s_gClave
Exis, s_gPasswordExis, s_IOrigen,
s_IDestino)
s_IComando =
"c:WINDOWS\FTP.EXE " + " -
s:\sic\comftp.txt" + " " +
s_gDirFTPExis
v_IReturnValue =
Shell(s_IComando, vbNormalFocus)
For i_li = 1 To 7000
DoEvents
Next i_li
End If
End If
If ck_ped.Value = 1 Then
If tx_rutaped.Text = "" Then
MsgBox "Especifique la ruta
donde se encuentra el archivo de
Pedidos", vbExclamation
Exit Sub
Else
s_IOrigen = tx_rutaped.Text
s_IDestino = tx_peddes.Text
Call
gConstruyeComandoFTP(s_gClave
Ped, s_gPasswordPed, s_IOrigen,
s_IDestino)
s_IComando =
"c:WINDOWS\FTP.EXE " + " -
s:\sic\comftp.txt" + " " +
s_gDirFTPPed
v_IReturnValue =
Shell(s_IComando, vbNormalFocus)
For i_li = 1 To 7000
DoEvents
Next i_li
End If
End If
End Sub

Private Sub
cm_frp_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Ejecuta la Copia via FTP
del Archivo desde el Archivo Remoto
a su Equipo"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

'Proposito: Se cargan a tablas de
pedidos y / o existencias dada
la fecha de carga y el nombre de las
tablas
Private Sub Command1_Click()
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Regresa al Menú para
Carga de Información"
End Sub

'Propósito: Se carga forma inicial y
se inicializan rutas de archivos
'a cargar a sistema
Private Sub Form_Load()
Dim s_iKey As String
Dim d_ldb As Database
Dim t_ltb As Recordset
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set t_ltb =
d_ldb.OpenRecordset("conf_arch",
dbOpenSnapshot)
With t_ltb
s_iKey = " nombre =
'existencias'"
.FindFirst s_iKey
If Not .NoMatch Then
If Not
IsNull(.Fields("ruta_remota")) Then
tx_rutaaxis.Text =
.Fields("ruta_remota")
End If
If Not
IsNull(.Fields("ruta_local")) Then
tx_exisdes.Text =
.Fields("ruta_local")
End If
If Not
IsNull(.Fields("clave_acceso")) Then
s_gClaveExis =
.Fields("clave_acceso")
Else
s_gClaveExis = ""
End If
If Not
IsNull(.Fields("pasaporte")) Then
s_gPasswordExis =
.Fields("pasaporte")
Else
s_gPasswordExis = ""
End If
If Not IsNull(.Fields("dir_ip"))
Then
s_gDirFTPExis =
.Fields("dir_ip")
Else
s_gDirFTPExis = ""
End If
End With
s_iKey = " nombre = 'pedidos'"
.FindFirst s_iKey
If Not .NoMatch Then
If Not
IsNull(.Fields("ruta_remota")) Then
tx_rutaped.Text =
.Fields("ruta_remota")
End If
If Not
IsNull(.Fields("ruta_local")) Then
tx_peddes.Text =
.Fields("ruta_local")
End If
If Not
IsNull(.Fields("clave_acceso")) Then

```

```

s_gClavePed =
.Fields("clave_acceso")
Else
s_gClavePed = ""
End If
If Not
IsNull(.Fields("pasaporte")) Then
s_gPasswordPed =
.Fields("pasaporte")
Else
s_gPasswordPed = ""
End If
If Not IsNull(.Fields("dir_ip"))
Then
s_gDirFTPPed =
.Fields("dir_ip")
Else
s_gDirFTPPed = ""
End If
End With
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Posicione el Mouse en Un
Campo o Boton para Saber su
función."
End Sub

Private Sub
PB_exe_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Marca el Avance en la
Carga de los datos de
EXISTENCIAS a la Base de Datos"
End Sub

Private Sub
PB_ped_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Marca el Avance en la
Carga de los datos de PEDIDOS a la
Base de Datos"
End Sub

Private Sub
tx_exisdes_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Indica la Ruta y Nombre
local en donde se desea colocar el
archivo de EXISTENCIAS Copiado
del equipo Remoto"
End Sub

```

```

Private Sub
tx_fecha_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Teclee la Fecha con la
que desea Ingresar los datos de
Existencias y/o Pedidos"
End Sub

Private Sub
tx_peddes_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Indica la Ruta y Nombre
local en donde se desea colocar el
archivo de PEDIDOS Copiado del
equipo Remoto"
End Sub

Private Sub
tx_rutaaxis_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Indica la Ruta u nombre
donde se encuentra el Archivo de
EXISTENCIAS a Traer"
End Sub

Private Sub
tx_rutaped_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Indica la Ruta u nombre
donde se encuentra el Archivo de
PEDIDOS a Traer"
End Sub

```

---

```

'Archivo: fm_cnrp
'Fecha: Julio-Septiembre de 1999
'Propósito: Configurador de reportes
del sistema. Permite especificar
'características del reporte a generar
y asocia catálogos y conjuntos a
'el reporte

```

---

```

Option Explicit
Dim ws As Workspace
Dim db As Database
Dim tb As Recordset
Dim tb_DR As Recordset
Dim tb_RC As Recordset
Dim b_iHayTransaccion As Boolean

Private Sub cb_reportes_Click()
Dim i_lres As Integer
Dim s_iCata As String
Dim s_iStrld As String

```

```

Dim s_IGrupo As String
Dim s_ILinea As String
Dim s_IBlancos As String
Dim d_idb As Database
Dim t_ltb As Recordset
s_IBlancos = " " *cadena con 7
blancos
Text1.Text = cb_reportes.Text
s_IDStrId = "REP_ID = " +
Text1.Text + ""
tb_DR.FindFirst s_IDStrId
If tb_DR.NoMatch Then
MsgBox "No hay reportes
con ese identificador",
vbExclamation
Else
ws.BeginTrans
b_lHayTransaccion = True
tb_DR.Edit
Text1.Enabled = False
Command1.Enabled = True
Command2.Enabled = True
Command3(0).Enabled =
False
Command3(1).Enabled =
True
Command3(2).Enabled =
True
Command3(3).Enabled =
True
Command3(4).Enabled =
False

'Carga a memoria
tx_des.Text =
tb_DR.Fields("descripcion")
tx_forcat.Text =
Str(tb_DR.Fields("form_calc"))
tx_nummes.Text =
Str(tb_DR.Fields("num_meses"))
ck_concpe.Value =
If(tb_DR.Fields("consc"), 1, 0)
ck_concpt.Value =
If(tb_DR.Fields("constp"), 1, 0)
ck_conman.Value =
If(tb_DR.Fields("conss"), 1, 0)
ck_exisact.Value =
If(tb_DR.Fields("exis_actua"), 1, 0)
ck_exisest.Value =
If(tb_DR.Fields("exis_estim"), 1, 0)
ck_stock.Value =
If(tb_DR.Fields("stock"), 1, 0)
ck_pedidos.Value =
If(tb_DR.Fields("pedidos"), 1, 0)
ck_pron.Value =
If(tb_DR.Fields("pron"), 1, 0)
ck_conmtp.Value =
If(tb_DR.Fields("conmtp"), 1, 0)
List2.Clear
s_ICata =
tb_DR.Fields("rep_id")
Set d_idb =
OpenDatabase(s_gRutaBD, True)
s_IGrupo = "SELECT
rep_con.tipo_cat,
rep_con.con_cat_id FROM rep_con
"
s_IGrupo = s_IGrupo + "
ORDER BY 2;"
Set t_ltb =
d_idb.OpenRecordset(s_IGrupo)
If t_ltb.RecordCount > 0 Then
t_ltb.MoveFirst
While Not t_ltb.EOF
s_ILinea = t_ltb.Fields(0)
+ " | " + t_ltb.Fields(1)
List2.AddItem s_ILinea
t_ltb.MoveNext
Wend
End If
t_ltb.Close
d_idb.Close
End If
cb_reportes.Enabled = False
cmd_cancela.Enabled = False
End Sub

Private Sub
ck_concpe_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar este cuadro,
el reporte tomará en cuenta los
pronósticos de Construcción de
Planta Externa"
End Sub

Private Sub ck_concpt_Click()

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar este cuadro,
el reporte tomará en cuenta los
pronósticos de Construcción de
Telefonía Pública"
End Sub

Private Sub
ck_concpt_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar este cuadro,
el reporte tomará en cuenta los
pronósticos de Construcción de
Telefonía Pública"
End Sub

Private Sub
ck_conman_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar aquí, el
reporte va a incluir la cantidad en
pedidos que se encuentre pendiente
de recibir en cada mes"
End Sub

Private Sub
ck_pron_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar este cuadro,
se incluirán en el reporte los
pronósticos de mantenimiento y los
restará mes por mes"
End Sub

Private Sub
ck_soloalm_MouseMove(Button As

```



Integer, Shift As Integer, X As Single, Y As Single)

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar este cuadro,
el reporte solo considerará la
existencia de los almacenes
distribuidores ignorando la de los
auxiliares"
End Sub
```

```
Private Sub
ck_stock_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar este cuadro,
el reporte calculará el stock de
seguridad en función del consumo
promedio"
End Sub
```

```
Private Sub ck_usapro_Click()
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar este cuadro,
el reporte considerará los pronóstico
sumados a los consumos promedio
de mantenimiento"
End Sub
```

```
Private Sub cmd_cancela_Click()
Command3(0).Enabled = True
Command3(1).Enabled = False
Command3(2).Enabled = False
Command3(3).Enabled = False
Command3(4).Enabled = True
Text1.Enabled = True
cb_reportes.Visible = False
cmd_cancela.Visible = False
End Sub
```

```
Private Sub
cmd_cancela_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione para Regresar
al Inicio, y Cancelar la Búsqueda"
End Sub
```

```
Private Sub cmd_regresa_Click()
Unload Me
End Sub
```

```
Private Sub
cmd_regresa_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione para Regresar
al Menú de Reportes"
End Sub
```

```
Private Sub Command1_Click()
'Nombre:
'Fecha : 04/08/99
```

'Propósito: habilitar frame de opciones y deshabilitar frame de conjuntos

```
Frame1.Visible = True
Frame1.Enabled = True
Frame2.Visible = False
Frame2.Enabled = False
End Sub
```

```
Private Sub
Command1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione el boton para
establecer o modificar las opciones
del reporte"
End Sub
```

```
Private Sub Command2_Click()
```

'Nombre:  
'Fecha : 04/08/99  
'Propósito: habilitar frame de conjuntos y deshabilitar frame de opciones

```
Frame1.Visible = False
Frame1.Enabled = False
Frame2.Visible = True
Frame2.Enabled = True
End Sub
```

```
Private Sub
Command2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione este boton,
para indicar los catálogos y
conjuntos de ellos, que se
presentarán en el reporte"
End Sub
```

```
Private Sub Command3_Click(Index
As Integer)
```

```
Dim i_lres As Integer
Dim s_iCata As String
Dim s_iStrId As String
Dim s_iCriterio As String
Dim s_iLinea As String
Dim s_iBlancos As String
Dim d_ldb As Database
Dim t_ltb As Recordset
s_iBlancos = " " * cadena con 7
blancos
```

```
Select Case Index
Case 0 'Alta
If Text1.Text = "" Or
IsNull(Text1.Text) Then
MsgBox "Asigne un
identificador", vbExclamation
Exit Sub
End If
s_iStrId = " REP_ID = " +
Text1.Text + ""
tb_DR.FindFirst s_iStrId
If Not tb_DR.NoMatch Then
```

MsgBox "Atención. El identificador ya existe use otro", vbExclamation

```
Exit Sub
End If
Text1.Enabled = False
ws.BeginTrans
b_iHayTransaccion = True
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"def_rep"
gl_ltb.Fields("fecha_hora") =
Date + Time
gl_ltb.Fields("Numtrans") = 1
gl_ltb.Update
tb_DR.AddNew
```

```
'Carga a memoria
tx_des.Text = ""
tx_forcal.Text = 1
tx_nummes.Text = 0
ck_concpe.Value = 0
ck_concnp.Value = 0
ck_conman.Value = 0
ck_exisact.Value = 0
ck_exisest.Value = 0
ck_pedant.Value = 0
ck_stock.Value = 0
ck_usapro.Value = 0
ck_pedidos.Value = 0
ck_soloalm.Value = 0
ck_pron.Value = 0
ck_conmtp.Value = 0
```

```
List2.Clear
Command1.Enabled = True
Command2.Enabled = True
Command3(0).Enabled =
False
Command3(1).Enabled =
True
Command3(2).Enabled =
True
Command3(3).Enabled =
True
Command3(4).Enabled =
False
cb_reportes.Visible = False
cmd_cancela.Visible = False
Case 1 'Baja
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
tb_DR.Delete
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
gl_ltb.Fields("tabla") =
"def_rep"
gl_ltb.Fields("fecha_hora")
= Date + Time
gl_ltb.Fields("Numtrans") =
2
```

```

gl_Itb.Update
s_!StrId = "DELETE FROM
rep_con WHERE rep_id = "" +
Text11.Text + ""
db.Execute s_!StrId
MsgBox "Definición
Reporte eliminada"
End If
ws.CommitTrans
b_!HayTransaccion = False
Text11.Text = ""
Text11.Enabled = True
List2.Clear

Frame1.Enabled = False
Frame1.Visible = False
Frame2.Enabled = False
Frame2.Visible = False
Command1.Enabled = False
Command2.Enabled = False
Command3(0).Enabled =
True
Command3(1).Enabled =
False
Command3(2).Enabled =
False
Command3(3).Enabled =
False
Command3(4).Enabled =
True
cmd_cancela.Visible = False
cb_reportes.Visible = False
Case 2 'Actualiza
If tx_des.Text = "" Or
IsNull(tx_des.Text) Then
MsgBox "Asigne un
nombre a reporte", vbExclamation
Exit Sub
End If
If tx_des.Text = "" Or
IsNull(tx_des.Text) Then
MsgBox "Asigne un
nombre a reporte", vbExclamation
Exit Sub
End If
tb_DR.Fields("rep_id") =
Text11.Text

'Carga a TABLA
tb_DR.Fields("descripcion") =
tx_des.Text
tb_DR.Fields("form_calc") =
Val(tx_forcal.Text)
tb_DR.Fields("num_meses")
= Val(tx_nummes.Text)
tb_DR.Fields("consc") =
If(ck_concpe.Value = 1, True, False)
tb_DR.Fields("constp") =
If(ck_concpt.Value = 1, True, False)
tb_DR.Fields("conss") =
If(ck_comman.Value = 1, True,
False)
tb_DR.Fields("exis_actua") =
If(ck_exisact.Value = 1, True, False)
tb_DR.Fields("exis_estim") =
If(ck_exisest.Value = 1, True, False)
tb_DR.Fields("ped_ant") =
If(ck_pedant.Value = 1, True, False)
tb_DR.Fields("stock") =
If(ck_stock.Value = 1, True, False)

tb_DR.Fields("uso_pron") =
If(ck_usapro.Value, True, False)
tb_DR.Fields("pedidos") =
If(ck_pedidos.Value, True, False)
tb_DR.Fields("solo_alm") =
If(ck_soloalm.Value, True, False)
tb_DR.Fields("pron") =
If(ck_pron.Value, True, False)
tb_DR.Fields("comntp") =
If(ck_comntp.Value, True, False)
If tb_DR.Fields("rep_id") <>
"" Then
gl_Itb.AddNew
gl_Itb.Fields("usuario") =
s_login
gl_Itb.Fields("basedatos")
= s_gRutaBD
gl_Itb.Fields("tabla") =
"def_rep"
gl_Itb.Fields("fecha_hora")
= Date + Time
gl_Itb.Fields("Numtrans") =
3
gl_Itb.Update
tb_DR.Update
Else
MsgBox "Identificador de
Reporte es invalido"
Exit Sub
End If
Text11.Text = ""
Text1.Enabled = True
i_!tres = MsgBox("Confirma
Actualización del Registro ?",
vbYesNo)
If i_!tres = vbYes Then
ws.CommitTrans
MsgBox "El registro ha
sido Actualizado, Teclea Aceptar
para continuar"
Else
Exit Sub
End If
b_!HayTransaccion = False
Frame1.Enabled = False
Frame1.Visible = False
Frame2.Enabled = False
Frame2.Visible = False
Command1.Enabled = False
Command2.Enabled = False
Command3(0).Enabled =
True
Command3(1).Enabled =
False
Command3(2).Enabled =
False
Command3(3).Enabled =
False
Command3(4).Enabled =
True
cb_reportes.Visible = False
cmd_cancela.Visible = False
Case 3 'Cancela Cambios
ws.Rollback
b_!HayTransaccion = False
Text11.Text = ""
Text1.Enabled = True
List2.Clear
Frame1.Enabled = False

Frame1.Visible = False
Frame2.Enabled = False
Frame2.Visible = False
Command1.Enabled = False
Command2.Enabled = False
Command3(0).Enabled =
True
Command3(1).Enabled =
False
Command3(2).Enabled =
False
Command3(3).Enabled =
False
Command3(4).Enabled =
True
cmd_cancela.Visible = False
cb_reportes.Visible = False
Case 4 'Busca
Command3(0).Enabled = False
Command3(1).Enabled = False
Command3(2).Enabled = False
Command3(3).Enabled = False
Command3(4).Enabled = False
Text1.Enabled = False
cb_reportes.Visible = True
cmd_cancela.Visible = True
cb_reportes.Enabled = True
cmd_cancela.Enabled = True
cb_reportes.Clear
Set d_!ldb =
OpenDatabase(s_gRutaBD, True)
Set t_!tbb =
d_!ldb.OpenRecordset("select distinct
rep_id from def_rep order by rep_id",
dbOpenSnapshot)
t_!tbb.MoveLast
t_!tbb.MoveFirst
While Not t_!tbb.EOF
cb_reportes.AddItem
t_!tbb.Fields("rep_id")
t_!tbb.MoveNext
Wend
t_!tbb.Close
d_!ldb.Close
End Select
End Sub

Private Sub
Command3_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Con estos botones,
puede Agregar, Borrar o Actualizar
el reporte que está editando, puede
Cancelar o Buscar"
End Sub

Private Sub Command5_Click()
Dim i_!i As Integer
Dim s_!linea As String
Dim s_!iRepid As String
Dim s_!iCata As String
Dim s_!iTipo As String
Dim s_!iStrId As String
If Text11.Text <> "" Then
If IsNull(tb_DR.Fields("rep_id"))
Or tb_DR.Fields("rep_id") = "" Then

```

```

    tb_DR.Fields("rep_id") =
Text1.Text
    tb_DR.Update
    gl_tlb.AddNew
    gl_tlb.Fields("usuario") =
s_login
    gl_tlb.Fields("basedatos") =
s_gRutaBD
    gl_tlb.Fields("tabla") = "def_rep"
    gl_tlb.Fields("fecha_hora") =
Date + Time
    gl_tlb.Fields("Numtrans") = 3
    gl_tlb.Update
    s_iStrid = " REP_ID = " +
Text1.Text + ""
    tb_DR.FindFirst s_iStrid
    If Not tb_DR.NoMatch Then
        tb_DR.Edit
        End If
    End If
    For i_li = 0 To
    It_catalogo.ListCount - 1
        If It_catalogo.Selected(i_li)
        Then
            s_iRepld =
            tb_DR.Fields("rep_id")
            s_iCata =
            Left(It_catalogo.List(i_li), 7)
            s_iTipo = "1"
            s_iStrid = " rep_id = " +
            s_iRepld + " and con_cat_id = " +
            s_iCata + ""
            s_iStrid = s_iStrid + " and
            tipo_cat = " + s_iTipo + ""
            tb_RC.FindFirst s_iStrid
            If tb_RC.NoMatch Then
                gl_tlb.AddNew
                gl_tlb.Fields("usuario") =
s_login
                gl_tlb.Fields("basedatos")
            = s_gRutaBD
                gl_tlb.Fields("tabla") =
"rep_con"
                gl_tlb.Fields("fecha_hora")
            = Date + Time
                gl_tlb.Fields("Numtrans") =
1
                gl_tlb.Update
                tb_RC.AddNew
                tb_RC.Fields("rep_id") =
            tb_DR.Fields("rep_id")
                tb_RC.Fields("con_cat_id")
            = Left(It_catalogo.List(i_li), 7)
                tb_RC.Fields("tipo_cat") =
            "1"
                tb_RC.Update
                s_iLinea = "1" + " | " +
            Left(It_catalogo.List(i_li), 7)
                List2.AddItem s_iLinea
            Else
                MsgBox "Ya existe en
            reporte", vbExclamation
            End If
        End If
        Next i_li
    Else

```

```

        MsgBox "Asigne un
        identificador a este reporte",
        vbExclamation
        End If
    End Sub

Private Sub
Command5_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    fm_menReportes.sb_menrepor.Pane
    ls(1).Text = "Presione este botón
    para agregar el catálogo
    seleccionado, a la lista de catálogos
    para mostrar en el reporte"
    End Sub

Private Sub Command7_Click()
    Dim i_li As Integer
    Dim s_iLinea As String
    Dim s_iRepld As String
    Dim s_iCata As String
    Dim s_iTipo As String
    Dim s_iStrid As String
    If Text1.Text <> "" Then
        If IsNull(tb_DR.Fields("rep_id"))
        Or tb_DR.Fields("rep_id") = "" Then
            Text1.Text
            tb_DR.Update
            gl_tlb.AddNew
            gl_tlb.Fields("usuario") =
s_login
            gl_tlb.Fields("basedatos") =
s_gRutaBD
            gl_tlb.Fields("tabla") = "def_rep"
            gl_tlb.Fields("fecha_hora") =
            Date + Time
            gl_tlb.Fields("Numtrans") = 3
            gl_tlb.Update
            s_iStrid = " REP_ID = " +
            Text1.Text + ""
            tb_DR.FindFirst s_iStrid
            If Not tb_DR.NoMatch Then
                tb_DR.Edit
                End If
            For i_li = 0 To
            It_conjunto.ListCount - 1
                If It_conjunto.Selected(i_li)
                Then
                    s_iRepld =
                    tb_DR.Fields("rep_id")
                    s_iCata =
                    Left(It_conjunto.List(i_li), 7)
                    s_iTipo = "2"
                    s_iStrid = " rep_id = " +
                    s_iRepld + " and con_cat_id = " +
                    s_iCata + ""
                    s_iStrid = s_iStrid + " and
                    tipo_cat = " + s_iTipo + ""
                    tb_RC.FindFirst s_iStrid
                    If tb_RC.NoMatch Then
                        gl_tlb.AddNew
                        gl_tlb.Fields("usuario") =
                    s_login
                        gl_tlb.Fields("basedatos")
                    = s_gRutaBD

```

```

            gl_tlb.Fields("tabla") =
            "rep_con"
            gl_tlb.Fields("fecha_hora")
            = Date + Time
            gl_tlb.Fields("Numtrans") =
            1
            gl_tlb.Update
            tb_RC.AddNew
            tb_RC.Fields("rep_id") =
            tb_DR.Fields("rep_id")
            tb_RC.Fields("con_cat_id")
            = Left(It_conjunto.List(i_li), 7)
            tb_RC.Fields("tipo_cat") =
            "2"
            tb_RC.Update
            s_iLinea = "2" + " | " +
            Left(It_conjunto.List(i_li), 7)
            List2.AddItem s_iLinea
        Else
            MsgBox "Ya existe en
            reporte"
            End If
        End If
        Next i_li
    Else
        MsgBox "Solo puede asignar a
        reportes ya existentes ",
        vbExclamation
        End If
    End Sub

Private Sub
Command7_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    fm_menReportes.sb_menrepor.Pane
    ls(1).Text = "Presione este botón
    para agregar el conjunto
    seleccionado, a la lista que
    presentará el Reporte"
    End Sub

Private Sub Form_Load()
    Dim d_idb As Database
    Dim t_tlb As Recordset
    Dim s_iCriterio As String

    'Reposiciona el segundo frame
    para en caso que lo usen
    b_HayTransaccion = False
    Frame1.Left = 120
    Frame1.Top = 480
    Frame2.Left = 120
    Frame2.Top = 480

    'Inicializa espacio de trabajo
    CONREP para transac
    Set ws =
    DBEngine.CreateWorkspace("CONR
    EP", "admin", "", dbUseJet)
    Set db =
    ws.OpenDatabase(s_gRutaBD,
    True)
    Set tb_DR =
    db.OpenRecordset("def_rep",
    dbOpenDynaset)

```

```

Set tb_RC =
db.OpenRecordset("rep_con",
db.OpenDynaset)

' Carga de conjuntos y catalogos
para mostrar
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
s_ICriterio = "SELECT conj_id,
descripcion FROM conjunto ORDER
BY 1"
Set t_ltb =
d_ldb.OpenRecordset(s_ICriterio)
If t_ltb.RecordCount > 0 Then
t_ltb.Clear
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
t_ltb.AddItem
t_ltb.Fields(0) + " - " + t_ltb.Fields(1)
t_ltb.MoveNext
Wend
End If
t_ltb.Close
s_ICriterio = "SELECT catalogo,
descripcion FROM catalogo ORDER
BY 1"
Set t_ltb =
d_ldb.OpenRecordset(s_ICriterio)
If t_ltb.RecordCount > 0 Then
t_ltb.Clear
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
t_ltb.AddItem
t_ltb.Fields(0) + " - " + t_ltb.Fields(1)
t_ltb.MoveNext
Wend
End If
t_ltb.Close
d_ldb.Close
cb_reportes.Visible = False
cmd_cancela.Visible = False
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

```

```

Private Sub Form_Unload(Cancel As
Integer)
If b_lHayTransaccion Then
ws.Rollback
MsgBox "Termine la
transaccion en proceso"
Exit Sub
Else
MsgBox "Terminando modulo"
End If
tb_DR.Close
tb_RC.Close
db.Close
ws.Close
End Sub

```

```

Private Sub Image1_Click()

```

```

End Sub

```

```

Private Sub
Frame1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

```

```

Private Sub
Frame2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

```

```

Private Sub
Frame3_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

```

```

Private Sub
Frame4_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

```

```

Private Sub
List2_MouseMove(Button As Integer,
Shift As Integer, X As Single, Y As
Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "En esta ventana se
muestran los Catálogos que están
incluidos en el reporte. Seleccione
uno si desea eliminarlo"
End Sub

```

```

Private Sub
t_catalogo_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Lista total de materiale
existentes. Escoja uno para agregar
al Reporte"
End Sub

```

```

Private Sub
t_conjunto_MouseMove(Button As

```

```

Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Lista de Conjuntos de
Catálogos existentes en el sistema.
Seleccione uno para agregar al
reporte"
End Sub

```

```

Private Sub Picture1_Click()
Dim s_ICata As String
Dim s_IRepld As String
Dim s_IStrId As String
Dim i_lres As Integer
Dim s_ITipo As String
Dim i_li As Integer
Dim s_ICriterio As String
Dim s_ILinea As String
Dim d_ldb As Database
Dim t_ltb As Recordset

```

```

If Not IsNull(tb_DR.Fields("rep_id"))
And tb_DR.Fields("rep_id") <> ""
Then
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
For i_li = 0 To List2.ListCount - 1
If List2.Selected(i_li) Then
s_IRepld =
tb_DR.Fields("rep_id")
s_ICata =
Mid(List2.List(i_li), 5)
s_ITipo =
Mid(List2.List(i_li), 1, 1)
s_IStrId = " rep_id = " +
s_IRepld + " and con_cat_id = " +
s_ICata + ""
s_IStrId = s_IStrId + " and
tipo_cat = " + s_ITipo + ""
tb_RC.FindFirst s_IStrId
If tb_RC.NoMatch Then
MsgBox "No hay
rep_con asociado", vbExclamation
Else
tb_RC.Delete
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login

```

```

gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"rep_con"
gl_ltb.Fields("fecha_hora") = Date +
Time
gl_ltb.Fields("Numtrans") = 2
gl_ltb.Update
End If
End If
Next i_li
'Se recargan conjuntos y
catalogos asignados a reporte
List2.Clear
s_ICata = tb_DR.Fields("rep_id")

```

```

Set d_ldb =
ws.OpenDatabase(s_gRutaBD,
True)
s_ICriterio = "SELECT
rep_con.tipo_cat,
rep_con.con_cat_id FROM rep_con
"
s_ICriterio = s_ICriterio + " Where
rep_con.rep_id = " + Chr(34) +
s_ICata + Chr(34)
s_ICriterio = s_ICriterio + "
ORDER BY 2;"
Set t_1tb =
d_ldb.OpenRecordset(s_ICriterio)
If t_1tb.RecordCount > 0 Then
t_1tb.MoveFirst
While Not t_1tb.EOF
s_iLinea = t_1tb.Fields(0) + " }
" + t_1tb.Fields(1)
List2.AddItem s_iLinea
t_1tb.MoveNext
Wend
End If
t_1tb.Close
d_ldb.Close
End If
Else
MsgBox "Primero asigne un
identificador de reporte",
vbExclamation
End If
End Sub

Private Sub
Picture1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione en este bote
para eliminar el catálogo que haya
seleccionado"
End Sub

Private Sub
Text1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione el nombre
del reporte que desea Crear"
End Sub

Private Sub
tx_des_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione la
descripción del Reporte y que desea
que aparezca en el Encabezado del
mismo"
End Sub

Private Sub
tx_forca1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Sin efecto por ahora, en
versiones posteriores, se podrá
escoger la fórmula a utilizar en los
cálculos"
End Sub

Private Sub
tx_nummes_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Indique el número de
meses que desea se pronostiquen
en el reporte (Max. 6)"
End Sub

'-----
'Archivo: fm_conjcat
'Fecha: Julio-Septiembre de 1999
'Proposito: Definición de conjuntos
de catalogos via tablas relacionales
'-----

Option Explicit
Dim ws As Workspace
Dim db As Database
Dim tb As Recordset
Dim TB_CON As Recordset
Dim TB_CC As Recordset
Dim b_iHayTransaccion As Boolean

Private Sub cb_conj_cata_Click()
Dim i_lres As Integer
Dim s_iCata As String
Dim s_iStrId As String
Dim s_iCriterio As String
Dim d_ldb As Database
Dim t_1tb As Recordset
Text1.Text = cb_conj_cata.Text
s_iStrId = " CONJ_ID = " +
Text1.Text + ""
TB_CON.FindFirst s_iStrId
If TB_CON.NoMatch Then
MsgBox "No hay conjuntos
con ese identificador",
vbExclamation
Else
ws.BeginTrans
b_iHayTransaccion = True
TB_CON.Edit
Text1.Enabled = False
Text2.Text =
TB_CON.Fields("descripcion")
Text2.Enabled = True
It_concat.Enabled = True
Frame1.Enabled = True
Frame1.Visible = True
Picture1.Enabled = True
Command1(0).Enabled =
False
Command1(1).Enabled =
True
Command1(2).Enabled =
True
Command1(3).Enabled =
True
Command1(4).Enabled =
False
cb_conj_cata.Visible = False
cmd_cancela.Visible = False
End Sub

Private Sub cmd_cancela_Click()
Command1(0).Enabled = True
Command1(1).Enabled = False
Command1(2).Enabled = False
Command1(3).Enabled = False
Command1(4).Enabled = True
Text1.Enabled = True
cb_conj_cata.Visible = False
cmd_cancela.Visible = False
End Sub

Private Sub
cmd_cancela_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Cancela la búsqueda de
conjuntos y Regresa para escoger
otro Conjunto"
End Sub

Private Sub cmd_regresa_Click()
Unload Me
End Sub

Private Sub
cmd_regresa_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Regresa al Menú de
Reportes y abandona la Edición de
Conjuntos"
End Sub

```

```

Private Sub Command1_Click(Index
As Integer)
Dim i_res As Integer
Dim s_cata As String
Dim s_sIstrd As String
Dim s_criterio As String
Dim d_kdb As Database
Dim t_ltb As Recordset
Select Case Index
Case 0 'Alta
If Text1.Text = "" Or
IsNull(Text1.Text) Then
MsgBox "Asigne un
identificador", vbExclamation
Exit Sub
End If
s_Istrd = " CONJ_ID = " +
Text1.Text + ""
TB_CON.FindFirst s_Istrd
If Not TB_CON.NoMatch
Then
MsgBox "Atención: El
identificador ya existe use otro",
vbExclamation
Exit Sub
End If
ws.BeginTrans
b_HayTransaccion = False
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"conjunto"
gl_ltb.Fields("fecha_hora") =
Date + Time
gl_ltb.Fields("Numtrans") = 1
gl_ltb.Update

TB_CON.AddNew

Text1.Enabled = False
Text2.Text = ""
Text2.Enabled = True
It_concat.Clear
It_concat.Enabled = True
Picture1.Enabled = True

Frame1.Enabled = True
Frame1.Visible = True
Command1(0).Enabled =
False
Command1(1).Enabled =
True
Command1(2).Enabled =
True
Command1(3).Enabled =
True
Command1(4).Enabled =
False
Case 1 'Baja
i_res = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_res = vbYes Then
TB_CON.Delete
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"conjunto"
gl_ltb.Fields("fecha_hora") = Date +
Time
gl_ltb.Fields("Numtrans") = 2
gl_ltb.Update

s_Istrd = "DELETE FROM
con_cat WHERE conjunto = " +
Text1.Text + ""
db.Execute s_Istrd
MsgBox "Conjunto
eliminado"
End If
ws.CommitTrans
b_HayTransaccion = False
Text1.Text = ""
Text1.Enabled = True
Text2.Text = ""
Text2.Enabled = False
It_concat.Clear
It_concat.Enabled = False
Picture1.Enabled = False
Frame1.Enabled = False
Frame1.Visible = False
Command1(0).Enabled =
True
Command1(1).Enabled =
False
Command1(2).Enabled =
False
Command1(3).Enabled =
False
Command1(4).Enabled =
True
Case 2 'Actualiza
Carga a TABLA
TB_CON.Fields("conj_id") =
Text1.Text

TB_CON.Fields("descripcion") =
Text2.Text
If TB_CON.Fields("conj_id")
<> "" Then
If
TB_CON.Fields("descripcion") <> ""
Then
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"conjunto"
gl_ltb.Fields("fecha_hora") = Date +
Time
gl_ltb.Fields("Numtrans") = 3
gl_ltb.Update

TB_CON.Update
MsgBox "Debe teclear
la descripción del grupo"
Exit Sub
End If
ws.CommitTrans
b_HayTransaccion = False
Text1.Text = ""
Text1.Enabled = True
Text2.Text = ""
Text2.Enabled = False
It_concat.Clear
It_concat.Enabled = False
Picture1.Enabled = False
Frame1.Enabled = False
Frame1.Visible = False
Command1(0).Enabled =
True
Command1(1).Enabled =
False
Command1(2).Enabled =
False
Command1(3).Enabled =
False
Command1(4).Enabled =
True
Case 3 'Cancela Cambios
ws.Rollback
b_HayTransaccion = False
Text1.Text = ""
Text1.Enabled = True
Text2.Text = ""
Text2.Enabled = False
It_concat.Clear
It_concat.Enabled = False
Frame1.Enabled = False
Frame1.Visible = False
Picture1.Enabled = False
Command1(0).Enabled =
True
Command1(1).Enabled =
False
Command1(2).Enabled =
False
Command1(3).Enabled =
False
Command1(4).Enabled =
True
Case 4 'Busca
Command1(0).Enabled = False
Command1(1).Enabled = False
Command1(2).Enabled = False
Command1(3).Enabled = False
Command1(4).Enabled = False
Text1.Enabled = False
cb_conj_cata.Visible = True
cmd_cancela.Visible = True
cb_conj_cata.Clear
Set d_kdb =
OpenDatabase(s_gRutaBD, True)
Set t_ltb =
d_kdb.OpenRecordset("select distinct

```

```

conjunto from con_Cat order by
conjunto", dbOpenSnapshot)
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
    cb_conj_cata.AddItem
t_ltb.Fields("conjunto")
t_ltb.MoveNext
Wend
t_ltb.Close
d_ldb.Close
End Select
End Sub

'Nombre:
'Fecha: 5 agosto 99
'Proposito: Búsqueda de elemento
de tabla catalogos
'

Private Sub Command2_Click()
End Sub

Private Sub
Command1_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al presionar estos
boton, se crea, Graba, Borra,
Actualiza, o Busca un Conjunto de
Catálogos con la identificación
Teclada arriba"
End Sub

'Descripción: Al activarse este boton
se agregan todos los catalogos
en el catalogo. Se agregan solo que
no existan.

Private Sub Command3_Click()
Dim i_li As Integer
Dim s_iConId As String
Dim s_iCata As String
Dim s_iStrId As String
Dim v_iMarca As Variant
If Text1.Text <> "" Then
    If
IsNull(TB_CON.Fields("conj_id")) Or
TB_CON.Fields("conj_id") = "" Then
        TB_CON.Fields("conj_id") =
Text1.Text
        TB_CON.Update
        gl_ltb.AddNew
        gl_ltb.Fields("usuario") =
s_login
        gl_ltb.Fields("basedatos") =
s_gRutaBD
        gl_ltb.Fields("tabla") =
"conjunto"
        gl_ltb.Fields("fecha_hora") =
Date + Time
        gl_ltb.Fields("Numtrans") = 3
        gl_ltb.Update
        s_iStrId = " CONJ_ID = " +
Text1.Text + ""
        TB_CON.FindFirst s_iStrId
        If Not TB_CON.NoMatch
            Then
                TB_CON.Edit
            End If
            End If
            For i_li = 0 To
            It_catalogo.ListCount - 1
                If It_catalogo.Selected(i_li)
                Then
                    s_iConId =
                    TB_CON.Fields("conj_id")
                    s_iCata =
                    Left(It_catalogo.List(i_li), 7)
                    s_iStrId = " CONJUNTO = "
                    + s_iConId + " and CATALOGO = "
                    + s_iCata + " "
                    TB_CC.FindFirst s_iStrId
                    If TB_CC.NoMatch Then
                        gl_ltb.AddNew
                        gl_ltb.Fields("usuario") =
                        s_login
                        gl_ltb.Fields("basedatos")
                        = s_gRutaBD
                        gl_ltb.Fields("tabla") =
                        "con_cat"
                        gl_ltb.Fields("fecha_hora")
                        = Date + Time
                        gl_ltb.Fields("Numtrans") =
                        1
                        gl_ltb.Update
                        TB_CC.AddNew
                        TB_CC.Fields("conjunto")
                        = TB_CON.Fields("conj_id")
                        TB_CC.Fields("catalogo")
                        = s_iCata
                        TB_CC.Update
                        It_concat.AddItem s_iCata
                    Else
                        MsgBox "Ya existe en
                        conjunto", vbExclamation
                    End If
                    End If
                    Next i_li
                Else
                    MsgBox "Asigne un
                    identificador a este conjunto",
                    vbExclamation
                End If
            End Sub

            Private Sub
            Command3_MouseMove(Button As
            Integer, Shift As Integer, X As
            Single, Y As Single)

            fm_menReportes.sb_menrepor.Pane
            ls(1).Text = "Al presionar este boton,
            agrega el catálogo seleccionado, al
            Conjunto de Catálogos"
            End Sub

            Private Sub Form_Load()
            Dim d_ldb As Database
            Dim t_ltb As Recordset
            Dim s_iCriterio As String
            Text1.Text = ""
            Text2.Text = ""
            b_lHayTransaccion = False

            'Inicializa espacio de trabajo
            CONCAT para transac
            Set ws =
            DBEngine.CreateWorkSpace("CONC
            AT", "admin", "", dbUseJet)
            Set db =
            ws.OpenDatabase(s_gRutaBD,
            True)
            Set TB_CON =
            db.OpenRecordset("conjunto",
            dbOpenDynaset)
            Set TB_CC =
            db.OpenRecordset("con_cat",
            dbOpenDynaset)

            ' Carga de conjuntos y catalogos
            para mostrar
            Set d_ldb =
            OpenDatabase(s_gRutaBD, True)
            s_iCriterio = "SELECT catalogo,
            descripcion FROM catalogo ORDER
            BY 1 "
            Set t_ltb =
            d_ldb.OpenRecordset(s_iCriterio)
            If t_ltb.RecordCount > 0 Then
                It_catalogo.Clear
                t_ltb.MoveLast
                t_ltb.MoveFirst
                While Not t_ltb.EOF
                    It_catalogo.AddItem
                    t_ltb.Fields(0) + " - " + t_ltb.Fields(1)
                t_ltb.MoveNext
                Wend
            End If
            t_ltb.Close
            d_ldb.Close
            cb_conj_cata.Visible = False
            cmd_cancela.Visible = False
            End Sub

            Private Sub
            Form_MouseMove(Button As
            Integer, Shift As Integer, X As
            Single, Y As Single)

            fm_menReportes.sb_menrepor.Pane
            ls(1).Text = "Posicione el mouse en
            un boton para ver su función"
            End Sub

            Private Sub Form_Unload(Cancel As
            Integer)
            If b_lHayTransaccion Then
                ws.Rollback
                TB_CON.Close
                TB_CC.Close
                db.Close
                ws.Close
            Else
                MsgBox "Terminando modulo"
            End If
            End Sub

            Private Sub
            Frame1_MouseMove(Button As
            Integer, Shift As Integer, X As
            Single, Y As Single)

            fm_menReportes.sb_menrepor.Pane

```

```

Is(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

Private Sub
Frame2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Seleccione un Conjunto
de Catálogos para Consultar o
Modificar"
End Sub

Private Sub
lt_catalogo_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Catálogo Completo de
Materiales, para que escoja el que
desee agregar al Conjunto"
End Sub

Private Sub
lt_concat_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Lista de Catálogos que
se van agrupar en el Conjunto que
se está editando"
End Sub

Private Sub Picture1_Click()
Dim s_iCata As String
Dim s_iConj As String
Dim s_iStrId As String
Dim i_lres As Integer
Dim s_iTipo As String
Dim i_li As Integer
Dim s_iCriterio As String
Dim s_iLinea As String
Dim d_ldb As Database
Dim t_ltb As Recordset

If Not
IsNull(TB_CON.Fields("conj_id"))
And TB_CON.Fields("conj_id") <> ""
Then
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
For i_li = 0 To lt_concat.ListCount
- 1
If lt_concat.Selected(i_li) Then
s_iCata = lt_concat.List(i_li)
s_iConj =
TB_CON.Fields("conj_id")
s_iStrId = " catalogo = " +
s_iCata + " and conjunto = " +
s_iConj + ""
TB_CC.FindFirst s_iStrId
If TB_CC.NoMatch Then
MsgBox "No hay con_cat
asociado", vbExclamation
Else
TB_CC.Delete
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
gl_ltb.Fields("tabla") =
"con_cat"
gl_ltb.Fields("fecha_hora")
= Date + Time
gl_ltb.Fields("Numtrans") =
2
gl_ltb.Update
End If
End If
Next i_li

'Se recargan catalogos asignados
a CONJUNTO
lt_concat.Clear
s_iCata =
TB_CON.Fields("conj_id")
Set d_ldb =
ws.OpenDatabase(s_gRutaBD,
True)
s_iCriterio = "SELECT
con_cat.catalogo FROM con_cat "
s_iCriterio = s_iCriterio + " Where
con_cat.conjunto = " + Chr(34) +
s_iCata + Chr(34)
s_iCriterio = s_iCriterio + "
ORDER BY 1;"
Set t_ltb =
d_ldb.OpenRecordset(s_iCriterio)
If t_ltb.RecordCount > 0 Then
t_ltb.MoveFirst
While Not t_ltb.EOF
s_iLinea = t_ltb.Fields(0)
lt_concat.AddItem s_iLinea
t_ltb.MoveNext
Wend
End If
t_ltb.Close
d_ldb.Close
End If
Else
MsgBox "primero asigne un
identificador de conjunto",
vbExclamation
End If
End Sub

Private Sub Picture3_Click()

End Sub

Private Sub
Picture1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Elimina del Conjunto de
Catálogos, el que está seleccionado
en el cuadro de la izquierda"
End Sub

Private Sub
Text1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione la
Descripción del Conjunto de
Catálogos para saber a que
materiales se refiere"
End Sub

'-----
'Archivo: fm_conmtp
'Fecha: Julio-Septiembre de 1999
'Propósito: Captura de consumos por
mantenimiento en telefonía pública.
'-----

Option Explicit
Dim regiones(1 To 32) As String
Dim i_gNumRegion As Integer
Dim w_gws As Workspace
Dim d_gdb As Database
Dim t_gPA As Recordset
Dim t_gCATA As Recordset

Private Sub cb_catalogo_Click()
Dim i_ltb As Recordset
Dim i_criterio As String
Set d_gdb =
w_gws.OpenDatabase(s_gRutaBD,
True)
i_criterio = "select descripcio from
catalogo WHERE catalogo = " +
cb_catalogo.Text + ""
Set i_ltb =
d_gdb.OpenRecordset(i_criterio)
i_ltb.MoveLast
i_ltb.MoveFirst
tx_descripcion = i_ltb(0)
i_ltb.Close
End Sub

Private Sub cm_actualiza_Click()
Dim s_iCatalogo As String
Dim i_lAnio As Integer
Dim i_lregion As Integer
Dim i_lmes As Integer
Dim i_lres As Integer
i_lres = MsgBox("Confirme
Actualización ?", vbYesNo)
If i_lres = vbYes Then
s_iCatalogo =
CStr(cb_catalogo.Text)
i_lAnio = Val(cb_año.Text)
For i_lregion = 1 To
i_gNumRegion

```



```

For i_lmes = 1 To 12
  msg_1.Col = i_lmes
  msg_1.Row = i_lregion
  t_gPA.Seek "=", s_lCatalogo,
regiones(i_lregion), i_lmes, i_lAnio
If Not t_gPA.NoMatch Then
  t_gPA.Edit

t_gPA.Fields("consumoprom") =
Val(msg_1.Text)
t_gPA.Update
gl_tlb.AddNew
gl_tlb.Fields("usuario") =
s_login
gl_tlb.Fields("basedatos")
= s_gRutaBD
gl_tlb.Fields("tabla") =
"conmtp"
gl_tlb.Fields("fecha_hora")
= Date + Time
gl_tlb.Fields("Numtrans") =
3
gl_tlb.Update
End If
Next i_lmes
Next i_lregion
w_gws.CommitTrans
Frame1.Enabled = False
Frame1.Visible = False
cb_catalogo.Enabled = True
cb_año.Enabled = True
cm_busca(0).Enabled = True
cm_busca(1).Enabled = True
End If
End Sub

Private Sub
cm_actualiza_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Al presionar este
Botón, se graban los cambios y
modificaciones realizadas"
End Sub

Private Sub cm_busca_Click(Index
As Integer)
Dim s_lCatalogo As String
Dim i_lAnio As String
Dim i_lregion As Integer
Dim i_lmes As Integer
If cb_catalogo.Text = vbNullString
Or cb_catalogo.Text = "" Then
  MsgBox "Proporcionar catalogo
valido", vbExclamation
Exit Sub
End If
t_gCATA.Seek "=",
cb_catalogo.Text
If t_gCATA.NoMatch Then
  MsgBox "Proporcionar catalogo
valido", vbExclamation
Exit Sub
End If
If Not
gVerifica_año_valido(cb_año.Text)
Then * = vbNullString Or cb_año.Text
= "" Then

```

```

MsgBox "Proporcionar Año
valido", vbExclamation
Exit Sub
End If
Select Case Index
Case 0 'Alta
  w_gws.BeginTrans
  s_lCatalogo =
CStr(cb_catalogo.Text)
i_lAnio = Val(cb_año.Text)
For i_lregion = 1 To
t_gNumRegion
  For i_lmes = 1 To 12
    msg_1.Col = i_lmes
    msg_1.Row = i_lregion
    t_gPA.Seek "=",
s_lCatalogo, regiones(i_lregion),
i_lmes, i_lAnio
    If t_gPA.NoMatch Then
      gl_tlb.AddNew
      gl_tlb.Fields("usuario") =
s_login
      gl_tlb.Fields("basedatos") =
s_gRutaBD
      gl_tlb.Fields("tabla") =
"conmtp"
      gl_tlb.Fields("fecha_hora") = Date +
Time
      gl_tlb.Fields("Numtrans") = 1
      gl_tlb.Update
      t_gPA.AddNew
      t_gPA.Fields("catalogo")
= s_lCatalogo
      t_gPA.Fields("region") =
regiones(i_lregion)
      t_gPA.Fields("mes") =
i_lmes
      t_gPA.Fields("año") =
i_lAnio
      msg_1.Text = 0
      t_gPA.Update
    Else
      MsgBox "Este año ya
esta registrado para este catalogo",
vbExclamation
      w_gws.Rollback
      Exit Sub
    End If
  Next i_lmes
Next i_lregion
Case 1 'Busca
  s_lCatalogo =
CStr(cb_catalogo.Text)
i_lAnio = Val(cb_año.Text)
For i_lregion = 1 To
t_gNumRegion
  For i_lmes = 1 To 12
    msg_1.Col = i_lmes
    msg_1.Row = i_lregion
    t_gPA.Seek "=",
s_lCatalogo, regiones(i_lregion),
i_lmes, i_lAnio
    If t_gPA.NoMatch Then

```

```

MsgBox "No existe año
registrado a esta area ",
vbExclamation
Exit Sub
Else
  msg_1.Text =
t_gPA.Fields("consumoprom")
End If
Next i_lmes
Next i_lregion
w_gws.BeginTrans
End Select
Frame1.Enabled = True
Frame1.Visible = True
cb_catalogo.Enabled = False
cb_año.Enabled = False
cm_busca(0).Enabled = False
cm_busca(1).Enabled = False
msg_1.Row = 1
msg_1.Col = 1
t_row.Caption = msg_1.Row
t_col.Caption = msg_1.Col
End Sub

Private Sub
cm_busca_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Al presionar estos
botones, permiten Crear o Consultar
una nueva matriz de consumos para
el Catálogo y Año seleccionados"
End Sub

Private Sub cm_cancela_Click()
w_gws.Rollback
Frame1.Enabled = False
Frame1.Visible = False
cb_catalogo.Enabled = True
cb_año.Enabled = True
cm_busca(0).Enabled = True
cm_busca(1).Enabled = True
End Sub

Private Sub
cm_cancela_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Al presionar este
botón, CANCELA todos los cambios
que haya realizado"
End Sub

Private Sub cm_salir_Click()
t_gCATA.Close
t_gPA.Close
d_gdb.Close
w_gws.Close
Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Regresa al Menú de
Cálculo de Consumos"
End Sub

```

```

Private Sub Form_Load()
Dim i_li As Integer
Dim d_ldb As Database
Dim t_ltb As Recordset
msg_1.Cols = 13
msg_1.Row = 0
msg_1.Col = 1
msg_1.Col = 1
msg_1.Text = "Enero"
msg_1.Col = 2
msg_1.Text = "Febrero"
msg_1.Col = 3
msg_1.Text = "Marzo"
msg_1.Col = 4
msg_1.Text = "Abril"
msg_1.Col = 5
msg_1.Text = "Mayo"
msg_1.Col = 6
msg_1.Text = "Junio"
msg_1.Col = 7
msg_1.Text = "Julio"
msg_1.Col = 8
msg_1.Text = "Agosto"
msg_1.Col = 9
msg_1.Text = "Septiembre"
msg_1.Col = 10
msg_1.Text = "Octubre"
msg_1.Col = 11
msg_1.Text = "Noviembre"
msg_1.Col = 12
msg_1.Text = "Diciembre"
msg_1.Col = 0

```

```

' Carga de Regiones
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set t_ltb =
d_ldb.OpenRecordset("select region
from region order by region",
dbOpenSnapshot)
t_ltb.MoveLast
t_ltb.MoveFirst
If t_ltb.RecordCount > 0 Then
msg_1.Rows =
t_ltb.RecordCount + 1
i_li = 1
While Not t_ltb.EOF
msg_1.Row = i_li
msg_1.Text =
t_ltb.Fields("region")
regions(i_li) =
t_ltb.Fields("region")
t_ltb.MoveNext
i_li = i_li + 1
Wend
i_gNumRegion = i_li - 1
Else
i_gNumRegion = 0
End If
t_ltb.Close
msg_1.Row = 1
msg_1.Col = 1
Set t_ltb =
d_ldb.OpenRecordset("select

```

```

catalogo from catalogo order by
catalogo", dbOpenSnapshot)
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
cb_catalogo.AddItem
t_ltb("catalogo")
t_ltb.MoveNext
Wend
t_ltb.Close
d_ldb.Close
Dim l_año As Integer
For l_año = 1990 To 2100
cb_año.AddItem l_año
Next

```

```

' Inicializa espacio de trabajo
CONREP para transac
Set w_gws =
DBEngine.CreateWorkspace("conrmt
p", "admin", "", dbUseJet)
Set d_gdb =
w_gws.OpenDatabase(s_gRutaBD,
True)
Set t_gPA =
d_gdb.OpenRecordset("conrmt",
dbOpenTable)
t_gPA.Index = "PrimaryKey"
Set t_gCATA =
d_gdb.OpenRecordset("catalogo",
dbOpenTable)
t_gCATA.Index = "PrimaryKey"
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Posicione el cursor en
algún botón o campo para ver acción
que realiza"
End Sub

```

```

Private Sub
Frame1_DragDrop(Source As
Control, X As Single, Y As Single)

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Posicione el cursor en
algún botón o campo para ver acción
que realiza"
End Sub

```

```

Private Sub
Label1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Seleccione el Catálogo
al que corresponden los consumos a
buscar o crear"
End Sub

```

```

Private Sub Label2_Click()

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Seleccione el Año al

```

que corresponden los consumos a  
buscar o crear"  
End Sub

```

Private Sub msg_1_Click()
lb_row.Caption = msg_1.Row
lb_col.Caption = msg_1.Col
If msg_1.Text = "0" Then
tx_cantidad.Text = ""
Else
tx_cantidad.Text = msg_1.Text
End If
tx_cantidad.SetFocus
End Sub

```

```

Private Sub
msg_1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Seleccione una casilla
para actualizar el pronóstico
correspondiente"
End Sub

```

```

Private Sub tx_cantidad_Change()
msg_1.Text = tx_cantidad.Text
End Sub

```

```

Private Sub
tx_cantidad_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Escriba en este
campo, la cantidad que desea se
incorpore en la casilla seleccionada"
End Sub

```

```

Private Sub
tx_descripcion_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencalcons.sb_mencalcons.Pan
els(1).Text = "Presenta la
descripción del Catálogo
Seleccionado"
End Sub

```

```

' Archivo: fm_consirem
' Fecha: Julio-Septiembre de 1999
' Propósito: Cálculo de consumos por
mantenimiento de SIREM. También
realiza cálculo de consumos
promedios.

```

Option Explicit

'Propósito: Se realiza la obtención  
del consumo promedio tomando  
la suma de consumos de tabla  
consumo y despues dividiendo entre

```

'número de meses.
Private Sub cm_conprom_Click()
Dim s_ISQL As String
Dim i_icotainf As Long
Dim i_icotasup As Long
Dim i_inummes As Long
Dim d_ldb As Database
Dim qfd As QueryDef
Dim i_Anio As Integer
Dim i_Ires As Integer
Dim i_ICota2 As Integer
Dim s_IDia As String
Dim s_IMes As String
Dim s_Anio As String
Dim i_Ires As Integer
If Not
gVerifica_fecha_valida(tx_fechaIntro.
Text) Then ' = vbNullString Then
MsgBox "De una fecha intro
valida", vbExclamation
Exit Sub
End If
If Not
gVerifica_fecha_valida(tx_fechaRef.T
ext) Then ' = vbNullString Then
MsgBox "De una fecha de
referencia valida", vbExclamation
Exit Sub
End If
If Val(tx_nummes.Text) > 0 And
Val(tx_nummes) < 100 Then
i_inummes =
Val(tx_nummes.Text)
Else
MsgBox "De un número en el
rango 1 a 99", vbExclamation
Exit Sub
End If
i_Ires = MsgBox("¿ Confirma
ejecución de cálculo consumo
promedio ?", vbExclamation +
vbYesNo)
If i_Ires <> vbYes Then
Exit Sub
End If
Set d_ldb =
OpenDatabase(s_gRutaBD, True)

'Se calcula
i_icotainf =
Month(tx_fechaRef.Text) +
Year(tx_fechaRef.Text) * 12
i_Anio = Year(tx_fechaRef.Text)
i_ICota = Month(tx_fechaRef.Text)
- 1
i_ICota2 = (i_ICota + i_inummes)
Mod 12
i_ICota2 = i_ICota2 + 1
If i_ICota2 <= i_ICota Then
i_Anio = i_Anio + 1
End If
i_icotasup = i_ICota2 + i_Anio *
12
s_IDia =
Str(Day(tx_fechaIntro.Text))
s_IMes =
Str(Month(tx_fechaIntro.Text))
s_Anio =
Str(Year(tx_fechaIntro.Text))

'Verificar qe fecha intro no este ya
s_ISQL = s_ISQL + "INSERT
INTO cons( fechaIntro, catalogo,
almacen, consumprom)"
s_ISQL = s_ISQL + " SELECT #"
+ s_IMes + "/" + s_IDia + "/" +
s_Anio + "#. cat, alm,
sum(consumo) / " + Str(i_inummes)
s_ISQL = s_ISQL + " FROM
consumo "
s_ISQL = s_ISQL + " WHERE
mes+anio*12 >= " + Str(i_icotainf) + "
and mes+anio*12 <= " +
Str(i_icotasup)
s_ISQL = s_ISQL + " GROUP BY
cat,alm"

'Se ejecuta temporalmente
Set qfd =
d_ldb.CreateQueryDef("", s_ISQL)
qfd.Execute
qfd.Close

'Se ponen a cero todos los
consumos promedio negativos
s_ISQL = "UPDATE cons SET
consumprom = 0 WHERE fechaIntro =
#" + s_IMes + "/" + s_IDia + "/" +
s_Anio + "# AND consumprom < 0"

'Se ejecuta temporalmente
Set qfd =
d_ldb.CreateQueryDef("", s_ISQL)
qfd.Execute
qfd.Close
d_ldb.Close
MsgBox "Cálculo de Consumos
Promedio terminado y Depositado
en la Base de Datos Para la
Generación de Reportes",
vbExclamation
End Sub

Private Sub
cm_conprom_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencalconsb.sb_mencalconsb.Pan
els(1).Text = "Al presionar este
botón, una vez obtenida la sumatoria
de consumos. Ejecuta el cálculo del
Promedio"
End Sub

'Proposito. realiza el cálculo de
consumo en base a la formula
' consumo = salida a usuario -
entradas por devolución
'Este cálculo se realiza para cada
mes especificado de cada
'catalogo y almacen. Los resultados
de depositan en tabla consumos.
'Las fechas que se toman son la
fecha de captura .
Private Sub cm_consumo_Click()
Dim d_ldb As Database
Dim qfd As QueryDef
Dim i_Ires As Integer
Dim s_icons As String

'Se limpia tabla temporal de suma
consumo usuarios
s_icons = "DELETE FROM
sum_usu"
d_ldb.Execute s_icons

'Se limpia tabla temporal de suma
consumo devoluciones
s_icons = "DELETE FROM
sum_dev"
d_ldb.Execute s_icons

'Se limpia tabla temporal de suma
final de salidas a usuario
s_icons = "DELETE FROM
sum_usu2"
d_ldb.Execute s_icons

'Se limpia tabla temporal de suma
final de entradas por devolución
s_icons = "DELETE FROM
sum_dev2"
d_ldb.Execute s_icons

'Se limpia tabla temporal de suma
consumos
s_icons = "DELETE FROM
consumo"
d_ldb.Execute s_icons

'Se cargan total de entradas por
devolucion con ADM de
mantenimiento
' se modificó la consulta para que
tome todas las ADM's y excluir las
cuentas
' 1137, 1138 y 1342
Set qfd =
d_ldb.QueryDefs("tot_dev")
qfd.Parameters(0) =
CDate(tx_fechaRef.Text)

```

```

qfd.Execute
qfd.Close

'Se cargan total de salidas a
usuarios con ADM de mantenimiento
' se modificó la consulta para que
tome todas las ADM's y excluir las
cuentas
' 1137, 1138 y 1342
Set qfd =
d_ldb.QueryDefs("tot_usu")
qfd.Parameters(0) =
CDate(tx_fecha.Text)
qfd.Execute
qfd.Close

'Se cargan totaliza totales
parciales de salidas a usuarios
Set qfd =
d_ldb.QueryDefs("tottot_usu")
qfd.Execute
qfd.Close

'Se cargan totaliza totales
parciales de entradas por devolución
Set qfd =
d_ldb.QueryDefs("tottot_dev")
qfd.Execute
qfd.Close

'Se calculan consumos
Set qfd =
d_ldb.QueryDefs("cal_consumo")
qfd.Execute
qfd.Close

'Se limpia de datos temporales
'Se limpia tabla temporal de suma
consumo usuarios
s_icons = "DELETE FROM
sum_usu"
d_ldb.Execute s_icons

'Se limpia tabla temporal de suma
consumo devoluciones
s_icons = "DELETE FROM
sum_dev"
d_ldb.Execute s_icons

'Se limpia tabla temporal de suma
final de salidas a usuario
s_icons = "DELETE FROM
sum_usu2"
d_ldb.Execute s_icons

'Se limpia tabla temporal de suma
final de entradas por devolución
s_icons = "DELETE FROM
sum_dev2"
d_ldb.Execute s_icons
d_ldb.Close
MsgBox "Termino cálculo de
consumos, está LISTO el sistema
para Calcular el CONSUMO
PROMEDIO". vbExclamation
End Sub

Private Sub
cm_consumo_MouseMove(Button
As Integer, Shift As Integer, X As
Integer, Y As Integer)
fm_mencalcon.sb_mencalcon.Pan
els(1).Text = "Al Presionar este
botón se inicia el proceso de suma
de Consumos en Preparación para
el Promedio"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Integer, Y As Integer)
fm_mencalcon.sb_mencalcon.Pan
els(1).Text = "Regresa al Menú de
Cálculo de Consumos"
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Integer, Y As Integer)
fm_mencalcon.sb_mencalcon.Pan
els(1).Text = "Posicione el cursor en
algún botón o campo para ver acción
que realiza"
End Sub

Private Sub
tx_fechaIntro_MouseMove(Button As
Integer, Shift As Integer, X As
Integer, Y As Integer)
fm_mencalcon.sb_mencalcon.Pan
els(1).Text = "Proporcione la Fecha
con la que desea se Registren los
PRONÓSTICOS de Mantenimiento"
End Sub

Private Sub tx_fechaRef_Change()
fm_mencalcon.sb_mencalcon.Pan
els(1).Text = "Proporcione la fecha
de referencia a partir de la cual
desea se Consideren los Cosumos
para el Promedio"
End Sub

Private Sub
tx_nummes_MouseMove(Button As
Integer, Shift As Integer, X As
Integer, Y As Integer)
fm_mencalcon.sb_mencalcon.Pan
els(1).Text = "Proporcione la
cantidad de meses que deberán
tomarse para Calcular el Pronóstico"
End Sub

'-----
'Archivo: fm_entusu
'Fecha: Julio-Septiembre de 1999
'Proposito: Reporte de entregas a
usuario
'-----

Option Explicit

Private Sub ck_cat_Click()
If ck_cat.Value = 1 Then
pc_borracat.Enabled = False
cm_altacat.Enabled = False
It_catalogos.Enabled = False
It_catinc.Enabled = False
Else
pc_borracat.Enabled = True
cm_altacat.Enabled = True
It_catalogos.Enabled = True
It_catinc.Enabled = True
End If
End Sub

Private Sub
ck_cat_MouseMove(Button As
Integer, Shift As Integer, X As
Integer, Y As Integer)
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al marcar esta casilla,
se reportan todos los catálogos que
tuvieron movimiento"
End Sub

Private Sub ck_exp_Click()
If ck_exp.Value = 1 Then
tx_exp.Enabled = False
cm_altaexp.Enabled = False
pc_borraexp.Enabled = False
It_expinc.Enabled = False
Else
tx_exp.Enabled = True
cm_altaexp.Enabled = True
pc_borraexp.Enabled = True
It_expinc.Enabled = True
End If
End Sub

Private Sub
ck_exp_MouseMove(Button As
Integer, Shift As Integer, X As
Integer, Y As Integer)
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Marque esta casilla,
para indicar al reporte que se
mostrarán todos los expedientes de
usuario que haya tenido movimiento"
End Sub
Private Sub cm_altacat_Click()
Dim i_lres As Integer
If It_catalogos.ListIndex <> -1
Then
It_catinc.AddItem
Left(It_catalogos.List(It_catalogos.Lis
tIndex), 7)
End If
End Sub

Private Sub
cm_altacat_MouseMove(Button As

```



```

s_IExped =
IIf(IsNull(rst_res.Fields(1)), "",
rst_res.Fields(1))
d_Isuma =
IIf(IsNull(rst_res.Fields(2)), 0,
rst_res.Fields(2))
*****
s_IBuffer = "1" +
s_gCadena_fija(s_ICat, 12)
s_IBuffer = s_IBuffer +
s_gCadena_fija(s_IExped, 12)
s_IBuffer = s_IBuffer +
s_gNumero_fijo(d_Isuma,
Long_num)
Print #i_ArchSalida,
s_IBuffer
rst_res.MoveNext
i_NumLinea =
i_NumLinea + 1
DoEvents
Wend
Print #i_ArchSalida, "-----"
-----
-----
-----
-----
-----
-----
-----
-----
-----
Print #i_ArchSalida, "-----"
-----
-----
-----
-----
-----
-----
-----
-----
-----
Print #i_ArchSalida, ""
Print #i_ArchSalida, ""
Pagina." + Str(i_NumPagina)
Print #i_ArchSalida, ""
i_NumPagina =
i_NumPagina + 1
DoEvents
Wend
Close #i_ArchSalida
rst_res.Close
qfd.Close
d_ldb.Close
MsgBox "Generación de reporte
terminada", vbExclamation
End Sub

Private Sub
cm_generar_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione este botón
iniciar el proceso de Generación del
Reporte de Salidas a Usuario"
End Sub

Private Sub cm_imprimir_Click()
Call
glprime_archivo(tx_salusu.Text)
End Sub

Private Sub
cm_imprimir_MouseMove(Button As

```

```

Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione este botón
para mandar a impresión el reporte
generado"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione este botón
para regresar al Menú de Reportes"
End Sub

Private Sub Form_Load()
Dim d_ldb As Database
Dim t_ltb As Recordset
Dim s_ICriterio As String

' Carga de conjuntos y catalogos
para mostrar
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
s_ICriterio = "SELECT catalogo,
descripcion FROM catalogo ORDER
BY 1"
Set t_ltb =
d_ldb.OpenRecordset(s_ICriterio)
If t_ltb.RecordCount > 0 Then
It_catalogos.Clear
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
It_catalogos.AddItem
t_ltb.Fields(0) + "-" + t_ltb.Fields(1)
t_ltb.MoveNext
Wend
End If
t_ltb.Close
d_ldb.Close
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

Private Sub
Frame1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

```

```

Private Sub
Frame2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un boton para ver su función"
End Sub

Private Sub
It_catalogos_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Lista total de Catálogos
disponible. Seleccione uno para
agregar al reporte"
End Sub

Private Sub
It_catinc_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Muestra lista de
catalogos a analizar por Usuario.
Seleccione una para Borrario"
End Sub

Private Sub
It_expinc_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Lista de expedientes de
Usuario que serán incluidos en el
Reporte. Seleccione uno para
borrario"
End Sub

Private Sub pc_borracat_Click()
Dim i_lres As Integer
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
If It_catinc.ListIndex <> -1 Then
It_catinc.RemoveItem
End If
End If
End Sub

Private Sub
pc_borracat_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Presione el bote para
eliminar del Reporte el catálogo
Seleccionado"
End Sub

Private Sub pc_borraexp_Click()
Dim i_lres As Integer

```

```

i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
    If It_expinc.ListIndex <> -1 Then
        It_expinc.RemoveItem
    End If
End If
End Sub

```

```

Private Sub
pc_borraexp_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Borra de la lista de
expedientes de usuario, el que esté
seleccionado"
End Sub

```

```

Private Sub
bx_exp_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione un número
de expediente que desee agregar a
la lista del Reporte"
End Sub

```

```

Private Sub
tx_fechafin_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione en este
campo la fecha final hasta la cual se
buscarán las salidas de los
empleados"
End Sub

```

```

Private Sub
tx_fechaini_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione en este
campo la fecha a partir de la cual se
buscarán las salidas de los
empleados"
End Sub

```

```

Private Sub
tx_saiusu_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione en este
campo, la trayectoria donde desee
que se guarde el reporte generado,
o el que desee imprimir"
End Sub

```

```

'Archivo: fm_gen
'Fecha: Julio-Septiembre de 1999
'Proposito: Carga de consumos
promedio desde archivo gen

```

Option Explicit

```

'Proposito: Valida información y sube
a Access. Carga de acuerdo
Sub ICarga_gen()
Dim d_idb As Database
Dim r_iConss As Recordset
Dim d_idb As Database
Dim t_iConss As Recordset
Dim w_iWs As Workspace
Dim s_iIRut As String
Dim s_iNom As String
'Para control barras de progreso
Dim i_iNum As Long
Dim i_iCount As Long
Dim d_iRes As Double
Dim d_iporc As Double
Dim i_iRes As Integer
Dim r_iConss As Recordset
Dim d_iFecha As Date
PB_1.Value = 0
DoEvents
Call gObtenRutas(tx_archivo.Text,
s_iRut, s_iNom)
If True Then
    On Error GoTo termina

```

```

' Open the external dBASE
database.
Set d_idb =
OpenDatabase(s_iRut, False, False,
"dBASE III.")

```

```

' Abre tabla EXISTEN
Set r_iConss =
d_idb.OpenRecordset(s_iNom)
If r_iConss.RecordCount > 0
Then

```

```

    i_iNum =
r_iConss.RecordCount
    r_iConss.MoveLast
    r_iConss.MoveFirst
    On Error GoTo termina2
    Set w_iWs =
DBEngine.CreateWorkspace("CON",
"admin", "", dbUseJet)
    Set d_idb =
w_iWs.OpenDatabase(s_gRutaBD,
True)

```

```

    Set t_iConss =
d_idb.OpenRecordset("conss",
dbOpenTable)
    t_iConss.Index =
"PrimaryKey"
    d_iFecha =
CDate(tx_fecharef.Text)
    Set r_iCount =
d_idb.OpenRecordset("select
count(*) from conss where fechainro
= #" + Str(Month(d_iFecha)) + "/" +
Str(Day(d_iFecha)) + "/" +
Str(Year(d_iFecha)) + "#")
    If r_iCount.Fields(0) > 0 Then

```

```

MsgBox "Datos con la
fecha intro ya existen"
r_iCount = 0
d_iporc = 10
On Error GoTo termina3
w_iWs.BeginTrans
While Not r_iConss.EOF
    t_iConss.Seek "=",
CDate(tx_fecharef.Text),
r_iConss.Fields("catalogo"),
r_iConss.Fields("cod_alm")
    If t_iConss.NoMatch Then
        With t_iConss
            gl_itb.AddNew
            gl_itb.Fields("usuario") =
s_login

```

```

            gl_itb.Fields("basedatos") =
s_gRutaBD
            gl_itb.Fields("tabla") =
"conss"
            gl_itb.Fields("fecha_hora") = Date +
Time

```

```

            gl_itb.Fields("Numtrans") = 1
            gl_itb.Update
            AddNew
            Fields("fechainro") =
CDate(tx_fecharef.Text)
            Fields("catalogo") =
r_iConss.Fields("catalogo")
            Fields("almacen") =
r_iConss.Fields("cod_alm")
            Fields("consumprom") =
r_iConss.Fields("consumprom")
            Update
        End With
    Else

```

```

        i_lres = MsgBox("Hay
elementos repetidos, cancela?",
vbYesNo + vbExclamation)
        If i_lres = vbYes Then
            w_iWs.Rollback
            t_iConss.Close
            d_idb.Close
            w_iWs.Close
            r_iConss.Close
            d_idb.Close
            Exit Sub
        End If
    End If
    r_iConss.MoveNext
    i_iCount = i_iCount + 1
    d_iRes = Cdbl(i_iCount /
i_iNum) * 100#
    If d_iRes > d_iporc Then
        PB_1.Value = d_iporc
        d_iporc = d_iporc + 10
    End If
    DoEvents
Wend

```

```

    If vbYes = MsgBox("¿
Confirma adición de consumos
promedio ? Cargando " +
Str(!_iCount) + " registros",
vbYesNo) Then
    w_!Ws CommitTrans
    Else
    w_!Ws Rollback
    End If
    t_!Conss.Close
    d_!ldb.Close
    w_!Ws.Close
    End If
    r_!Conss.Close
    d_!ldbASE.Close
    End If
    PB_1.Value = 0
    MsgBox "Los Consumos Fueron
Cargados en la Tabla de Consumos
de Mantenimiento, con la Fecha
indicado"
    Exit Sub
termina:
    MsgBox "Consumos: El archivo
especificado tiene algun problema
verifique", vbExclamation
    Exit Sub
termina2:
    MsgBox "Consumos: No se
puede cargar la base de datos
principal", vbExclamation
    r_!Conss.Close
    d_!ldbASE.Close
    Exit Sub
termina3:
    MsgBox "Consumos: Error en
carga de datos ", vbExclamation
    t_!Conss.Close
    d_!ldb.Close
    w_!Ws.Close
    r_!Conss.Close
    d_!ldbASE.Close
    Exit Sub
End Sub

Private Sub cm_carga_Click()
    If Not
gVerifica_fecha_valida(tx_fechaRef.T
ext) Then ' = vbNullString Then
        MsgBox "Especifique una fecha
de entrada valida", vbExclamation
        Exit Sub
    End If
    fm_gen.Enabled = False
    Call iCarga_gen
    fm_gen.Enabled = True
End Sub

Private Sub
cm_carga_MouseDown(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
1).Text = "Al Presionar este Boton,
Inicia la Validación y Carga del
Archivo Indicado con la Fecha
indicada a la BD"
End Sub

```

```

Private Sub cm_salir_Click()
    Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
1).Text = "Al Presionar este botón,
Regresa el Menú para Carga de
Información"
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
1).Text = "Posicione el mouse en un
boton o Campo, para obtener Ayuda
sobre su función"
End Sub

Private Sub
PB_1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
1).Text = "Muestra el Avance en la
carga de Consumos a la Base de
Datos"
End Sub

Private Sub
tx_archivo_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    fm_mencarga.SB_mencarga.Panels(
1).Text = "Proporcione el Nombre
del Archivo GEN que desea Cargar
con CONSUMLOS"
End Sub

Private Sub tx_fechaRef_Change()

    fm_mencarga.SB_mencarga.Panels(
1).Text = "Proporcione la Fecha con
la que se van a ingresar los
Consumos del archivo GEN"
End Sub

'-----
'Archivo: fm_genrep
'Fecha: Julio-Septiembre de 1999
'Proposito: Módulo generador de
reporte en base a configuración de
reportes.
'Adicionalmente invoca generación
de reportes de pedidos y traspasos
localizados en módulo tras_ped.bas
'-----

Option Explicit

```

```

Dim b_gCorriendo As Boolean
'Indica si proceso de generación esta
corriendo
Dim d_gFechaReferencia As Date
'Fecha de referencia para calculos

'Proposito: este procedimiento
actualiza consnspron con pronosticos
Sub iActualizaPro(FechaRef As
Date)
    Dim t_!ICP As Recordset ' Tabla
Consspron
    Dim t_!IPR As Recordset ' Tabla
pronosticos
    Dim d_!ldb As Database
    Dim s_!iCat As String
    Dim s_!iReg As String
    Dim t_!imes As Integer
    Dim t_!Anio As Integer
    Set d_!ldb =
OpenDatabase(s_gRutaBD, True)
    Set t_!ICP =
d_!ldb.OpenRecordset("consp2",
dbOpenTable)
    Set t_!IPR =
d_!ldb.OpenRecordset("pronosticos",
dbOpenTable)
    t_!IPR.Index = "PrimaryKey"
    t_!ICP.MoveLast
    t_!ICP.MoveFirst
    i_!imes = Month(FechaRef)
    i_!Anio = Year(FechaRef)

    'Para cada consumo promedio se
sustituye con pronostico si existe
    While Not t_!ICP.EOF
        s_!iCat =
t_!ICP.Fields("catalogo")
        s_!iReg = t_!ICP.Fields("region")
        t_!IPR.Seek "=", s_!iCat, s_!Reg,
i_!imes, i_!Anio
        If Not t_!IPR.NoMatch Then
            t_!ICP.Edit
            t_!ICP.Fields("sumcon") =
t_!IPR.Fields("pronos")
            t_!ICP.Update
            g_!itb.AddNew
            g_!itb.Fields("usuario") =
s_!login
            g_!itb.Fields("basedatos") =
s_gRutaBD
            g_!itb.Fields("tabla") =
"consp2"
            g_!itb.Fields("fecha_hora") =
Date + Time
            g_!itb.Fields("Numtrans") = 3
            g_!itb.Update
        End If
        t_!ICP.MoveNext
    Wend
    t_!ICP.Close
    t_!IPR.Close
    d_!ldb.Close
End Sub

'Proposito: este procedimiento
actualiza consnspron con pronosticos
Sub iActTem(FechaRef As Date)
    Dim t_!ICP As Recordset ' Tabla
Consspron

```



```

Dim t_IPR As Recordset 'Tabla
pronicosticos
Dim d_kdb As Database
Dim s_ICat As String
Dim s_Reg As String
Dim i_lmes As Integer
Dim i_lanio As Integer
Dim sqlFind As String
Set d_kdb =
OpenDatabase(s_gRutaBD, True)
Set t_ICP =
d_kdb.OpenRecordset("conspptx",
dbOpenTable)
Set t_IPR =
d_kdb.OpenRecordset("pronicosticos",
dbOpenTable)
t_IPR.Index = "PrimaryKey"
t_ICP.MoveLast
t_ICP.MoveFirst
i_lmes = Month(FechaRef)
i_lanio = Year(FechaRef)

'Para cada consumo promedio se
sustituye con pronostico si existe
While Not t_ICP.EOF
s_ICat =
t_ICP.Fields("catalogo")
s_Reg = t_ICP.Fields("region")
t_IPR.Seek "=", s_ICat, s_Reg,
i_lmes, i_lanio
If Not t_IPR.NoMatch Then
t_ICP.Edit
t_ICP.Fields("sumcon") =
t_IPR.Fields("pronos")
t_ICP.Update
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"conspptx"
gl_ltb.Fields("fecha_hora") =
Date + Time
gl_ltb.Fields("Numtrans") = 3
gl_ltb.Update
End If
t_ICP.MoveNext
Wend
t_ICP.Close
t_IPR.Close
d_kdb.Close
End Sub

Private Sub cm_genera_Click()
Const Long_num = 8
Const Cte_Mto = 1 'Es
información mto en SIREM
Const Cte_TP = 2 'TP es
construcción telefonía pública
Const Cte_PE = 3 'PE es
construcción planta exterior
Const Cte_PRO = 4 'PRO es
pronicosticos de almacenes
Const Cte_MTP = 5 'MTP es
mantenimiento de telefonía pública
Dim s_lDefRep As String
Dim s_lFecExi As String
Dim s_lFecPed As String
Dim s_lFecConM As String

```

```

Dim s_lFecConCPE As String
Dim s_lIcons As String 'Consultas
Dim d_kdb As Database
Dim t_IDR As Recordset '
Definición de reporte
Dim t_IRC As Recordset '
Catalogos y conjuntos de reporte
Dim s_lStrId As String

'Queries para apertura de consultas
Dim qfd As QueryDef
Dim qfd_1 As QueryDef
Dim qfd_2 As QueryDef
Dim qfd_3 As QueryDef
Dim qfd_4 As QueryDef
Dim qfd_5 As QueryDef 'Para
abrir tot consumos TP
Dim qfd_6 As QueryDef 'Para
abrir tot consumos PE (pend)
Dim qfd_7 As QueryDef 'Para
abrir tot consumos pronicosticos
Dim qfd_8 As QueryDef 'Para
abrir tot consumos mto TP

'Recordsets para ver los resultados
de los queries(consultas)
Dim rst As Recordset
Dim rst_1 As Recordset 'Contiene
totalizado de existencias
Dim rst_2 As Recordset 'Contiene
totalizado de pedidos
Dim rst_3a As Recordset 'Contiene
totalizados de consumos sirem
Dim rst_3b As Recordset 'Contiene
totalizados de consumos const. PE
Dim rst_3c As Recordset 'Contiene
totalizados de consumos const. TP
Dim rst_3d As Recordset 'Contiene
totalizados de consumos en
pronicosticos
Dim rst_3e As Recordset 'Contiene
totalizados de consumos mto. TP
Dim rst_4 As Recordset 'Contiene
producto cartesiano de cat_tem x
region

Dim f_lFecha As Date
Dim i As Integer
Dim s_lCatAnt As String
Dim s_lCatCon As String
Dim s_lregion As String

'Variables para cálculo de estimación
de existencias
Dim d_lExistencia As Double
Dim d_lConsumo(Cte_Mto To
Cte_MTP) As Double '1=mto,
2=tp, 3=pe, 4=pro, 5=MTP
Dim d_lPedido(-1 To 6) As Double
Dim d_lExis_est(Cte_Mto To
Cte_MTP, 0 To 6) As Double '
1=mto, 2=tp, 3=pe, 4=pro, 5 = MTP
Dim d_lIEE As Double

'Variables de configuración de
reporte
Dim t_lICod_Form As Long
'Numero de formula a utilizar en
existencias estimadas
Dim s_lDes As String

```

```

Dim l_inummes As Long
'Numero de meses a calcular

'A continuación se listan variables
binarias que contienen configuración
'de reporte
Dim b_lkoncpe As Boolean
Dim b_lkoncpt As Boolean
Dim b_lkonman As Boolean
Dim b_llexisact As Boolean
Dim b_llexisest As Boolean
Dim b_lpedant As Boolean
Dim b_lstock As Boolean
Dim b_lusapro As Boolean
Dim b_lpedidos As Boolean
Dim b_lSoloAlmacenes As Boolean
Dim b_lpron As Boolean
Dim b_lkonmtp As Boolean

'Variables para el formateo del
archivo generado
Dim s_lDescripcion As String
Dim s_lHeader0 As String
Dim s_lHeader As String
Dim s_lHeader2 As String
Dim s_lFormato As String
Dim i_gArchSalida As Integer
Dim i_lNumPagina As Integer
Dim i_lNumLinea As Integer
Dim i_li As Integer
Dim s_lBuffer As String

'Variables auxiliares para fecha de
referencia usada
Dim i_lDia As Integer
Dim i_lmes As Integer
Dim i_lAnio As Integer

Dim i_lres As Integer
i_lres = MsgBox("¿ Confirma
generación de reporte dinámico ?",
vbQuestion + vbYesNo)
If i_lres <= vbYes Then
Exit Sub
End If
PB_1.Value = 0
DoEvents
Set d_kdb =
OpenDatabase(s_gRutaBD, True)
Set t_IDR =
d_kdb.OpenRecordset("def_rep",
dbOpenDynaset)
s_lStrId = "rep_id = " +
cb_reporte.Text + " "
t_IDR.FindFirst s_lStrId
If t_IDR.NoMatch Then
MsgBox "Identificador de
reporte no reconocido".
vbExclamation
Exit Sub
End If
s_lDes =
t_IDR.Fields("descripcion")
t_lCod_Form =
t_IDR.Fields("form_calc")
l_inummes =
t_IDR.Fields("num_meses")
b_lkoncpe = t_IDR.Fields("consc")
b_lkoncpt = t_IDR.Fields("constp")

```

```

b_Iconman =
t_IDR.Fields("conss")
b_lexisact =
t_IDR.Fields("exis_actuall")
b_lexisest =
t_IDR.Fields("exis_estim")
b_lpedant =
t_IDR.Fields("ped_ant")
b_lstock = t_IDR.Fields("stock")
b_lusapro =
t_IDR.Fields("uso_pron")
b_lpedidos =
t_IDR.Fields("pedidos")
b_lSoloAlmacenes =
t_IDR.Fields("solo_alm")
b_lpron = t_IDR.Fields("pron")
b_lconmtp =
t_IDR.Fields("conmtp")
If Not
gVerifica_fecha_valida(tx_fechaefe.
Text) Then ' = vbNullString Then
MsgBox "De una fecha de
referencia valida", vbExclamation
t_IDR.Close
d_ldb.Close
Exit Sub
Else
d_gFechaReferencia =
CDate(tx_fechaefe.Text)
End If
If Not
gVerifica_fecha_valida(cb_existencia
s Text) Then ' = vbNullString Then
MsgBox "Especifique una fecha
valida de existencias"
t_IDR.Close
d_ldb.Close
Exit Sub
End If
If Not
gVerifica_fecha_valida(cb_pedidos.T
ext) Then ' = vbNullString Then
MsgBox "Especifique fecha
valida en pedidos"
t_IDR.Close
d_ldb.Close
Exit Sub
End If
If b_Iconman Then
If Not
gVerifica_fecha_valida(cb_consumo
s.Text) Then ' = vbNullString Then
MsgBox "Especifique fecha
valida para Consumos Mitto"
t_IDR.Close
d_ldb.Close
Exit Sub
End If
End If
'Si no hay ningun consumo a usar,
avisar y salir del módulo
If Not b_Iconman And Not
b_lconctp And Not b_lconcpce And
Not b_lpron And Not b_lconmtp
Then
MsgBox " Se necesita saber de
que consumos calcular"
t_IDR.Close
d_ldb.Close
b_gCorriendo = False
Exit Sub
End If
b_gCorriendo = True
i_IDia = Day(d_gFechaReferencia)
i_lmes =
Month(d_gFechaReferencia)
i_lAnio =
Year(d_gFechaReferencia)
'Se limpia tabla temporal
s_lcons = "DELETE FROM
cat_tem"
d_ldb.Execute s_lcons
*****
'Se limpia tabla copy_ped para
pruebas
s_lcons = "DELETE FROM
copy_ped"
*****
'Se limpia consumos mtto con
pronosticos
s_lcons = "DELETE FROM
consspron"
d_ldb.Execute s_lcons
'Se limpia tabla
s_lcons = "DELETE FROM
consp2"
d_ldb.Execute s_lcons
'Se cargan datos de catalogos
Set qfd =
d_ldb.QueryDefs("ini_cat_tem1")
qfd.Parameters(0) =
cb_reporte.Text
qfd.Execute
qfd.Close
'Se cargan datos de conjuntos
Set qfd =
d_ldb.QueryDefs("ini_cat_tem2")
qfd.Parameters(0) =
cb_reporte.Text
qfd.Execute
qfd.Close
PB_1.Value = 10
DoEvents
'Se crean totales de existencias
actuales
If b_lSoloAlmacenes Then
'No se consideran depositos
auxiliares de calculo
Set qfd =
d_ldb.QueryDefs("ini_alm_dist")
qfd.Execute
qfd.Close
Set qfd =
d_ldb.QueryDefs("solo_almacenpi")
qfd.Execute
qfd.Close
Set qfd_1 =
d_ldb.QueryDefs("tot_exisalm")
qfd_1.Parameters(0) =
cb_existencias.Text
Else
Set qfd_1 =
d_ldb.QueryDefs("tot_exis")
qfd_1.Parameters(0) =
cb_existencias.Text
End If
Set rst_1 =
qfd_1.OpenRecordset(dbOpenSnap
shot)
PB_1.Value = 30
DoEvents
'Se crean totales de pedidos
Set qfd =
d_ldb.QueryDefs("up_ped_del")
qfd.Parameters(0) =
d_gFechaReferencia "Fecha
referencia
qfd.Parameters(1) =
cb_pedidos.Text 'Fecha de entrada
de pedidos a seleccionar
qfd.Execute
qfd.Close
PB_1.Value = 40
DoEvents
*****
'Se crean totales de consumos
If b_Iconman Then
If b_lusapro Then
'Se crean consumos con
pronosticos
Set qfd =
d_ldb.QueryDefs("crea_consspron")
qfd.Parameters(0) =
cb_consumos.Text
qfd.Execute
qfd.Close
'Se totaliza
Set qfd =
d_ldb.QueryDefs("tot_consspron")
qfd.Execute
qfd.Close
'Se sustituyen pronosticos
existentes
IActualizaPro
(d_gFechaReferencia)
'Se crean totales de
consumos con pronosticos
Set qfd_3 =
d_ldb.QueryDefs("tot_consp2")
'Al abrirlo se ejecuta y vemos
resultados
Set rst_3a =
qfd_3.OpenRecordset(dbOpenSnap
shot)
Else
Set qfd_3 =
d_ldb.QueryDefs("tot_conss")
qfd_3.Parameters(0) =
cb_consumos.Text
'Al abrirlo se ejecuta y vemos
resultados
Set rst_3a =
qfd_3.OpenRecordset(dbOpenSnap
shot)

```



```
'Busca en existencias
s_IStrId = " k2 = " +
s_IcAtCon + " and r2 = " + s_lregion
+ " "
rst_1.FindFirst s_IStrId
If Not rst_1.NoMatch Then
d_IExistencia =
rst_1.Fields("sumdisp")
Else
d_IExistencia = 0
End If
```

**CONSTRUCCION PLANTA EXTERIOR**

```
'USAR b_Iconcpe
If b_Iconcpe Then
s_IStrId = " k1 = " +
s_IcAtCon + " and r1 = " + s_lregion
+ " "
rst_3b.FindFirst s_IStrId
If Not rst_3b.NoMatch
Then
If
IsNull(rst_3b.Fields("sumcon"))
Then
d_IConsumo(Cte_PE) = 0
Else
d_IConsumo(Cte_PE) =
rst_3b.Fields("sumcon")
End If
Else
d_IConsumo(Cte_PE)
= 0
End If
End If
```

**CONSTRUCCION TELEFONIA PUBLICA**

```
'USAR b_Iconcnp
If b_Iconcnp Then
'USAR b_lusapro?
s_IStrId = " k1 = " +
s_IcAtCon + " and r1 = " + s_lregion
+ " "
rst_3c.FindFirst s_IStrId
If Not rst_3c.NoMatch
Then
If
IsNull(rst_3c.Fields("s1"))
Then
d_IConsumo(Cte_TP) = 0
Else
d_IConsumo(Cte_TP) =
rst_3c.Fields("s1")
End If
Else
d_IConsumo(Cte_TP)
= 0
End If
End If
```

**'Busca en consumos mantenimiento TELEFONIA PUBLICA**

```
'USAR b_Iconmtp
If b_Iconmtp Then
s_IStrId = " k1 = " +
s_IcAtCon + " and r1 = " + s_lregion
+ " "
rst_3e.FindFirst s_IStrId
If Not rst_3e.NoMatch
Then
```

```
If
IsNull(rst_3e.Fields("sumcon"))
Then
```

```
d_IConsumo(Cte_MTP) = 0
Else
```

```
d_IConsumo(Cte_MTP) =
rst_3e.Fields("sumcon")
End If
Else
```

```
d_IConsumo(Cte_MTP) = 0
End If
End If
```

**'Busca en consumos POR MANTENIMIENTO**

```
If b_Iconman Then
s_IStrId = " k1 = " +
s_IcAtCon + " and r1 = " + s_lregion
+ " "
rst_3a.FindFirst s_IStrId
If Not rst_3a.NoMatch
Then
```

```
If
IsNull(rst_3a.Fields("sumcon"))
Then
```

```
d_IConsumo(Cte_Mto) = 0
Else
```

```
d_IConsumo(Cte_Mto) =
rst_3a.Fields("sumcon")
End If
Else
```

```
d_IConsumo(Cte_Mto) = 0
End If
End If
```

**'Busca en consumos POR pronosticos**

```
If b_lpron Then
s_IStrId = " k1 = " +
s_IcAtCon + " and r1 = " + s_lregion
+ " "
rst_3d.FindFirst s_IStrId
If Not rst_3d.NoMatch
Then
```

```
If
IsNull(rst_3d.Fields("sumpron"))
Then
```

```
d_IConsumo(Cte_PRO) = 0
Else
```

```
d_IConsumo(Cte_PRO) =
rst_3d.Fields("sumpron")
End If
Else
```

```
d_IConsumo(Cte_PRO) = 0
End If
End If
```

```
'Busca en pedidos. Se
buscan hasta 6 meses adelante
For i_li = -1 To 6
d_IPedido(i_li) = 0
```

```
'Se hace calculo de
existencias solo para meses
presente y futuros
```

```
If i_li >= 0 Then
If i_li = 0 Then
d_IEE =
```

```
d_IExistencia
If b_Iconman Then
```

```
d_IExis_est(Cte_Mto, i_li) =
d_gCalcula_existencia(i_IcCod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_Mto))
End If
If b_Iconcnp Then
```

```
d_IExis_est(Cte_TP, i_li) =
d_gCalcula_existencia(i_IcCod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_TP))
End If
If b_Iconcpe Then
```

```
d_IExis_est(Cte_PE, i_li) =
d_gCalcula_existencia(i_IcCod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_PE))
End If
If b_lpron Then
```

```
d_IExis_est(Cte_PRO, i_li) =
d_gCalcula_existencia(i_IcCod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_PRO))
End If
If b_Iconmtp Then
```

```
d_IExis_est(Cte_MTP, i_li) =
d_gCalcula_existencia(i_IcCod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_MTP))
End If
Else
If b_Iconman Then
d_IEE =
```

```
d_IExis_est(Cte_Mto, i_li - 1)
```

```
d_IExis_est(Cte_Mto, i_li) =
d_gCalcula_existencia(i_IcCod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_Mto))
End If
If b_Iconcnp Then
d_IEE =
```

```
d_IExis_est(Cte_TP, i_li - 1)
```

```

d_IExis_est(Cte_TP, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_TP))
End If
If b_lconcpce Then
d_IEE =
d_IExis_est(Cte_PE, i_li - 1)

d_IExis_est(Cte_PE, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_PE))
End If
If b_lpron Then
d_IEE =
d_IExis_est(Cte_PRO, i_li - 1)

d_IExis_est(Cte_PRO, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_PRO))
End If
If b_lconmtp Then
d_IEE =
d_IExis_est(Cte_MTP, i_li - 1)

d_IExis_est(Cte_MTP, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE, d_IPedido(i_li),
d_IConsumo(Cte_MTP))
End If
End If
End If
Next i_li

'Datos iniciales
'Es VITAL que el orden de
if's sea el mismo
'que el de los
encabezados de arriba
s_Buffer =
s_gCadena_fija(s_CatCon, 7)
s_Buffer = s_Buffer +
s_IDescripcion
s_Buffer = s_Buffer + " " +
s_gCadena_fijaEsp(s_Iregion, 7)
If b_lexisad Then
'Existencias actuales
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IExistencia,
Long_num)
End If
If b_lconman Then
'Consumo mantenimiento
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IConsumo(Cte_M
to), Long_num)
End If
If b_lconman And b_Istock
Then 'Stock = consumo * 2
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IConsumo(Cte_M
to) * 2, Long_num)
End If
If b_lpron Then
'Pronosticos
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IConsumo(Cte_P
RO), Long_num)
End If
If b_lconmtp Then
'Consumo Mlti TP
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IConsumo(Cte_M
TP), Long_num)
End If
If b_lconctp Then
'Consumo TP
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IConsumo(Cte_T
P), Long_num)
End If
If b_lconcpce Then
'Consumo PE
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IConsumo(Cte_P
E), Long_num)
End If
For i_li = 0 To i_inummes
If b_lpedidos Then
s_Buffer = s_Buffer
+ s_gNumero_fijo(d_IPedido(i_li),
Long_num)
End If
If b_lexisest Then
'Existencias estimadas
If b_lconman Then
s_Buffer =
s_Buffer +
s_gNumero_fijo(d_IExis_est(Cte_Mt
o, i_li), Long_num)
End If
If b_lpron Then
s_Buffer =
s_Buffer +
s_gNumero_fijo(d_IExis_est(Cte_PR
O, i_li), Long_num)
End If
If b_lconmtp Then
s_Buffer =
s_Buffer +
s_gNumero_fijo(d_IExis_est(Cte_MT
P, i_li), Long_num)
End If
If b_lconctp Then
s_Buffer =
s_Buffer +
s_gNumero_fijo(d_IExis_est(Cte_TP,
i_li), Long_num)
End If
If b_lconcpce Then
s_Buffer =
s_Buffer +
s_gNumero_fijo(d_IExis_est(Cte_PE
, i_li), Long_num)
End If
End If
Next i_li
If b_lpedad Then
'Pedidos anteriores
s_Buffer = s_Buffer +
s_gNumero_fijo(d_IPedido(-1),
Long_num)
End If
Print #i_gArchSalida,
s_Buffer

rst_4.MoveNext
i_NumLinea =
i_NumLinea + 1
DoEvents
Wend
Print #i_gArchSalida, "-----"
Print #i_gArchSalida, "
Print #i_gArchSalida, "
Pagina:" + Str(i_NumPagina)
Print #i_gArchSalida, "
i_NumPagina =
i_NumPagina + 1
DoEvents
Wend
PB_1.Value = 100
Close #i_gArchSalida
MsgBox "Reporte generado ",
vbInformation
End If

'Se cierran datos
rst_1.Close
qfd_1.Close
If b_lconmtp Then 'Cierra si se
uso
rst_3e.Close
End If
If b_lpron Then 'Cierra
pronosticos si se uso
rst_3d.Close
qfd_7.Close
End If
If b_lconctp Then 'Cierra si se
uso
rst_3c.Close
qfd_5.Close
End If
If b_lconman Then 'Cierra si se
uso
rst_3a.Close
qfd_3.Close
End If
If b_lconcpce Then 'Cierra si se
uso
rst_3b.Close
qfd_6.Close
End If
rst_4.Close
qfd_4.Close
d_lcb.Close
PB_1.Value = 0
DoEvents
b_gCorriendo = False
End Sub

Private Sub
cm_genera_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```



```

    l_ibt.MoveNext
Wend
Else
    cb_pedidos.AddItem ("No hay
Pedidos")
End If
l_ibt.Close
Set l_ibt =
d_idb.OpenRecordset("select distinct
fechaintro from consa order by
fechaintro", dbOpenDynaset)
If l_ibt.RecordCount <> 0 Then
    l_ibt.MoveLast
    l_ibt.MoveFirst
    While Not l_ibt.EOF
        If Day(l_ibt("fechaintro")) < 10
Then
            l_dia = "0" +
CStr(Day(l_ibt("fechaintro")))
        Else
            l_dia =
CStr(Day(l_ibt("fechaintro")))
        End If
        If Month(l_ibt("fechaintro")) <
10 Then
            l_mes = "0" +
CStr(Month(l_ibt("fechaintro")))
        Else
            l_mes =
CStr(Month(l_ibt("fechaintro")))
        End If
        cb_consumos.AddItem l_dia
+ "/" + l_mes + "/" +
CStr(Year(l_ibt("fechaintro")))
        l_ibt.MoveNext
    Wend
Else
    cb_consumos.AddItem ("No
hay Consumos")
End If
l_ibt.Close
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un botón o campo para ver su
función"
End Sub

Private Sub
Label1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Seleccione el Reporte
que desea Generar"
End Sub

Private Sub
Label13_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane

```

```

ls(1).Text = "Seleccione una Fecha
de Existencias para el Reporte de
las cargadas en el Sistema"
End Sub

Private Sub
Label14_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione una Fecha
de Referencia a partir de la cual se
harán las proyecciones en el
reporte"
End Sub

Private Sub
Label3_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Seleccione una Fecha
de Pedidos, de las existentes en el
sistema"
End Sub

Private Sub
Label5_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Seleccione una fecha
de Consumos de Mantenimiento se
los que están registrados en el
sistema"
End Sub

Private Sub
Label7_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione una Fecha
de Referencia a partir de la cual se
harán las proyecciones en el
reporte"
End Sub

Private Sub
PB_1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Indica el Progreso en la
Generación de los Reportes
Principal y de Traspasos y Pedidos"
End Sub

Private Sub
bx_archimprim_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane

```

```

ls(1).Text = "Proporcione el nombre
del archivo que desea Imprimir"
End Sub

Private Sub
bx_archped_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione el Nombre
del Archivo donde se generará el
Reporte de Pedidos"
End Sub

Private Sub
bx_archsalida_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione el Nombre
del Archivo donde va a ser
Generado el Reporte Solicitado"
End Sub

Private Sub
bx_pedida_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione el Nombre
del Archivo donde se depositará el
Archivo Plano de Pedidos para
análisis"
End Sub

Private Sub
bx_retras_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione el Nombre
del Archivo donde va a generarse el
Reporte de Traspasos"
End Sub

Private Sub
bx_trapia_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Proporcione el Nombre
del Archivo donde se generará el
Archivo Plano de Traspasos para su
manipulación en excel"
End Sub

-----
'Archivo: fm_inmat
'Fecha: Julio-Septiembre de 1999
'Propósito: Captura de materiales
para cnsrucción de líneas en áreas
de Compañía Telefónica.
-----

```

```

Option Explicit
Dim w_gws As Workspace
Dim d_gdb As Database
Dim t_gPA As Recordset

Private Sub cm_actualiza_Click()
Dim s_lAreaTM As String
Dim i_lCat As Integer
Dim l_ires As Integer
l_ires = MsgBox(" Confirme
actualización ?", vbYesNo)
If l_ires = vbYes Then
s_lAreaTM = CStr(tx_areatm.Text)
For i_lCat = 0 To msg_1.Rows - 1
msg_1.Row = i_lCat
msg_1.Col = 0
t_gPA.Seek "=", s_lAreaTM,
msg_1.Text
If t_gPA.NoMatch Then
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
gl_ltb.Fields("tabla") =
"Inmat"
gl_ltb.Fields("fecha_hora")
= Date + Time
gl_ltb.Fields("Numtrans") =
1
gl_ltb.Update
t_gPA.AddNew
t_gPA.Fields("areatm") =
s_lAreaTM
t_gPA.Fields("catalogo") =
msg_1.Text
msg_1.Col = 1
t_gPA.Fields("factor") =
Val(msg_1.Text)
t_gPA.Update
Else
t_gPA.Edit
msg_1.Col = 1
t_gPA.Fields("factor") =
Val(msg_1.Text)
t_gPA.Update
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
gl_ltb.Fields("tabla") =
"Inmat"
gl_ltb.Fields("fecha_hora")
= Date + Time
gl_ltb.Fields("Numtrans") =
3
gl_ltb.Update
End If
Next i_lCat
w_gws.CommitTrans
msg_1.Clear
msg_1.Rows = 0
Frame1.Enabled = False
Frame1.Visible = False
tx_areatm.Enabled = True
cm_control.Enabled = True
End If
End Sub

```

```

Private Sub cm_agrega_Click()
Dim i_li As Integer
Dim l_cat As String
If l_catalogo.ListIndex <> -1 Then
For i_li = 0 To msg_1.Rows - 1
msg_1.Row = i_li
msg_1.Col = 0
l_cat =
Left(l_catalogo.List(l_catalogo.ListI
ndex), 7)
If msg_1.Text = l_cat Then
MsgBox "Catalogo ya
existe", vbExclamation
Exit Sub
End If
Next i_li
msg_1.AddItem l_cat
msg_1.Row = msg_1.Rows - 1
msg_1.Col = 1
msg_1.Text = 0
End If
End Sub

Private Sub
cm_agrega_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al Presionar
el Botón, Agrega a la lista de
Factores, el Catálogo que se
encuentra seleccionado"
End Sub

Private Sub cm_cancela_Click()
w_gws.Rollback
msg_1.Clear
msg_1.Rows = 0
Frame1.Enabled = False
Frame1.Visible = False
tx_areatm.Enabled = True
cm_control.Enabled = True
End Sub

Private Sub cm_control_Click()
Dim s_lAreaTM As String
Dim d_ldb As Database
Dim t_ltb As Recordset
Dim s_lCriterio As String
If tx_areatm.Text = vbNullString
Or tx_areatm.Text = "" Then
MsgBox "Proporcionar Area TM
valida", vbExclamation
Exit Sub
End If
s_lAreaTM = CStr(tx_areatm.Text)

' Carga de conjuntos y catalogos
para mostrar
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
s_lCriterio = "SELECT
catalogo,factor FROM Inmat
WHERE areatm = '" & s_lAreaTM & "'
ORDER BY 1"
Set t_ltb =
d_ldb.OpenRecordset(s_lCriterio)
If t_ltb.RecordCount > 0 Then

```

```

msg_1.Clear
msg_1.Rows = 0
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
msg_1.AddItem
t_ltb.Fields(0)
msg_1.Row = msg_1.Rows -
1
msg_1.Col = 1
msg_1.Text = t_ltb.Fields(1)
t_ltb.MoveNext
Wend
End If
t_ltb.Close
d_ldb.Close
w_gws.BeginTrans
Frame1.Enabled = True
Frame1.Visible = True
tx_areatm.Enabled = False
cm_control.Enabled = False
End Sub

Private Sub
cm_control_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Presione el
Boton para buscar un área de
Construcción que se desea editar"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Regresa al
Menú: Datos para Cálculo de
Consumos"
End Sub

Private Sub Form_Load()
Dim d_ldb As Database
Dim t_ltb As Recordset
Dim s_lCriterio As String
Dim l_cat_des As String

' Carga de conjuntos y catalogos
para mostrar
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
s_lCriterio = "SELECT catalogo,
descripcio FROM catalogo ORDER
BY Catalogo"
Set t_ltb =
d_ldb.OpenRecordset(s_lCriterio)
If t_ltb.RecordCount > 0 Then
l_catalogo.Clear
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF

```



```

        l_cat_des =
l_lb.Fields("Catalogo") + " - " +
l_lb.Fields("Descripcion")
        It_catalogo.AddItem
l_cat_des
        l_lb.MoveNext
        Wend
        End If
        t_lb.Close
        d_lb.Close

        Inicializa espacio de trabajo
CONREP para transac
        Set w_gws =
DBEngine.CreateWorkspace("LNMA
T", "admin", "", dbUseJet)
        Set d_gdb =
w_gws.OpenDatabase(s_gRutaBD,
True)
        Set t_gPA =
d_gdb.OpenRecordset("Inmat",
dbOpenTable)
        t_gPA.Index = "PrimaryKey"
        End Sub

        Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

        fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Posicione el
Mouse en algún boton o Campo
para Obtener ayuda de su función"
        End Sub

        Private Sub
Frame1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

        fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Posicione el
Mouse en algún boton o Campo
para Obtener ayuda de su función"
        End Sub

        Private Sub
It_catalogo_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

        fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Se Muestra
la lista Total de Materiales que tiene
el sistema, para que se seleccionen
a los que desee dar un Factor"
        End Sub

        Private Sub msg_1_Click()
        If msg_1.Text = "0" Then
            tx_factor.Text = ""
        Else
            tx_factor.Text = msg_1.Text
        End If
        tx_factor.SetFocus
        End Sub

        Private Sub
msg_1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

        desea Agregar Factores por
Material"
        End Sub

        Private Sub tx_factor_Change()
        If msg_1.Rows > 0 Then
            msg_1.Text = tx_factor.Text
        End Sub

        Private Sub
tx_factor_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

        fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Proporcione
aquí el Factor que desea de aplique
al Catálogo seleccionado"
        End Sub

        -----
        'Archivo: fm_MenArchivo
        'Fecha: Julio-Septiembre de 1999
        'Proposito: Módulo del Menú de
        'Archiv para invocar a las
        'aplicaciones que abren o crean
        'bases de datos
        'para trabajar. Hace uso de menues
        'y de botones de uso continuo
        -----

        Private Sub Command4_Click()
        End
        End Sub

        Private Sub cmd_abrir_Click()
        With dlgCommonDialog
            To Do
                'set the flags and attributes of
                the
                'common dialog control
                .DialogTitle = "Apertura de base
                de datos"
                .Filter = "Access DBs
                (*.mdb)|*.mdb"
                .ShowOpen
                If Len(FileName) = 0 Then
                    Exit Sub
                End If
                s_gRutaBD = FileName
                MainForm.Text1.Text =
                s_gRutaBD
                txt_nombreBD.Text =
                s_gRutaBD
                cmd_Comenzar.Enabled = True
                cmd_Comenzar.Caption =
                "&Iniciar"
                men_regresa.Caption =
                "&Iniciar"
                End With
                MsgBox ("Ha Seleccionado un
                Base de Datos, Precione " + "" +
                "Iniciar" + "" + "" para Trabajar con
                ella.")
            End Sub

        Private Sub cmd_atras_Click()
    
```

```

Unload Me
End Sub

Private Sub
cmd_abrir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
    SB_menarchivo.Panels(1).Text =
    "Permite seleccionar una base de
    datos existente"
End Sub

Private Sub cmd_Comenzar_Click()
    Unload Me
End Sub

Private Sub
cmd_Comenzar_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
    SB_menarchivo.Panels(1).Text =
    "Utilizar el sistema con Base de
    Datos seleccionada"
End Sub

Private Sub cmd_nueva_Click()
    With dlgCommonDialog
        'To Do
        'set the flags and attributes of
        the
        'common dialog control
        .DialogTitle = "Creación de
        nueva base de datos "
        .Filter = "Access DBs
        (*.mdb)|*.mdb"
        .ShowOpen
        If Len(FileName) = 0 Then
            Exit Sub
        End If
        If Dir(FileName) <> "" Then
            MsgBox "Atención: Ese
            nombre ya esta siendo ocupado por
            otra base. Creación cancelada",
            vbExclamation
            Exit Sub
        End If
        s_gRutaBD = .FileName
        Call gCrea_bd(s_gRutaBD)
        #MainForm.Text1.Text =
        s_gRutaBD
        txt_nombreBD.Text =
        s_gRutaBD
        cmd_Comenzar.Enabled = True
        cmd_Comenzar.Caption =
        "&Iniciar"
        men_regresa.Caption =
        "&Iniciar"
    End With
    MsgBox ("A Creado una nueva
    Base de Datos de Trabajo. Precione
    INICIO para empezar")
End Sub

Private Sub
cmd_nueva_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
    SB_menarchivo.Panels(1).Text =
    "Crea una nueva base de datos"
End Sub

Private Sub cmd_salir_Click()
    Unload Me
    If vbYes = MsgBox("¿ Confirma
    salida de sistema ?", vbQuestion +
    vbYesNo) Then
        End
    End If
End Sub

Private Sub
cmd_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
    SB_menarchivo.Panels(1).Text =
    "Al precionar sale del sistema y
    abandona la sesión"
End Sub

Private Sub Form_Load()
    If s_gadmin Then
        cmd_nueva.Enabled = True
    Else
        cmd_nueva.Enabled = False
    End If
    If s_gRutaBD = "" Then
        cmd_Comenzar.Enabled =
        False
        cmd_Comenzar.Caption =
        "&Iniciar"
        cmd_Comenzar.ToolTipText =
        "Comenzar a trabajar con Base de
        Datos nueva"
        Picture2.Visible = False
    Else
        fm_MenArchivo.men_regresa.Enabled =
        True

        fm_MenArchivo.men_regresa.Caption =
        "Regresar"
        cmd_Comenzar.Caption =
        "Regresa al Menú Anterior"
        cmd_Comenzar.Enabled = True
        cmd_Comenzar.ToolTipText =
        "Regresa al Menú Anterior"
        Picture2.Visible = True
        End If
        txt_nombreBD = s_gRutaBD
    End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
    SB_menarchivo.Panels(1).Text =
    "Posicione el mouse en algún boton
    para ver la acción que realiza"
End Sub

Private Sub men_abrir_Click()
    cmd_abrir_Click
End Sub

Private Sub men_nueva_Click()
    cmd_nueva_Click
End Sub

Private Sub men_regresa_Click()
    cmd_atras_Click
End Sub

Private Sub men_salir_Click()
    cmd_salir_Click
End Sub

Private Sub Picture2_Click()
    Unload Me
End Sub

Private Sub
Picture2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
    SB_menarchivo.Panels(1).Text =
    "Utilizar el sistema con Base de
    Datos seleccionada"
End Sub

Private Sub
txt_nombreBD_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
    SB_menarchivo.Panels(1).Text =
    "Muestra Base de Datos
    seleccionada para trabajar"
End Sub

'-----
'Archivo: fm_Mencalcons
'Fecha: Julio-Septiembre de 1999
'Proposito: Módulo del Menú de
Cálculo de consumos para invocar a
las aplicaciones sobre los consumos
'Hace uso de menues y de botones
de uso continuo
'-----

Private Sub cmd_consProm_Click()
    cms_consTelPub.Visible = False
    cmd_regresa.Visible = False
    cmd_consProm.Visible = False
    cmd_Construc_telpub.Visible =
    False
    fm_consirem.Show vbModal
    cms_consTelPub.Visible = True
    cmd_regresa.Visible = True
    cmd_consProm.Visible = True
    cmd_Construc_telpub.Visible =
    True
End Sub

Private Sub
cmd_consProm_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
    sb_mencalcons.Panels(1).Text =
    "Cálculo de consumo promedio
    según consumos de SIREM"
End Sub

Private Sub
cmd_Construc_telpub_Click()
    cms_consTelPub.Visible = False
    cmd_regresa.Visible = False
    cmd_consProm.Visible = False
    cmd_Construc_telpub.Visible =
    False
End Sub

```

```

    fm_tp.Show vbModal
    cms_consTelPub.Visible = True
    cmd_regresa.Visible = True
    cmd_consProm.Visible = True
    cmd_Construc_telpub.Visible =
True
End Sub

Private Sub
cmd_Construc_telpub_MouseMove(
Button As Integer, Shift As Integer, X
As Single, Y As Single)
    sb_mencalcons.Panels(1).Text =
"Cálculo de proyecciones de
Construcción según factores de
Telefonía Pública"
End Sub

Private Sub cmd_regresa_Click()
Unload Me
End Sub

Private Sub
cmd_regresa_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
    sb_mencalcons.Panels(1).Text =
"Regresa al Menú anterior"
End Sub

Private Sub
cms_consTelPub_Click()
    cms_consTelPub.Visible = False
    cmd_regresa.Visible = False
    cmd_consProm.Visible = False
    cmd_Construc_telpub.Visible =
False
    fm_conmtp.Show vbModal
    cms_consTelPub.Visible = True
    cmd_regresa.Visible = True
    cmd_consProm.Visible = True
    cmd_Construc_telpub.Visible =
True
End Sub

Private Sub
cms_consTelPub_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
    sb_mencalcons.Panels(1).Text =
"Cálculo de consumos de
mantenimiento de Telefonía Pública"
End Sub

Private Sub Form_Load()
Text1.Text = s_gRutaBD
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
    sb_mencalcons.Panels(1).Text =
"Posicione el cursor en algún boton
para ver acción que realiza"
End Sub

Private Sub
men_const_tel_pub_Click()
    cms_consTelPub.Visible = False
    cmd_regresa.Visible = False
    cmd_consProm.Visible = False
    cmd_Construc_telpub.Visible =
False
    fm_tp.Show vbModal
    cms_consTelPub.Visible = True
    cmd_regresa.Visible = True
    cmd_consProm.Visible = True
    cmd_Construc_telpub.Visible =
True
End Sub

Private Sub
men_Consumo_prom_Click()
    cms_consTelPub.Visible = False
    cmd_regresa.Visible = False
    cmd_consProm.Visible = False
    cmd_Construc_telpub.Visible =
False
    fm_consirem.Show vbModal
    cms_consTelPub.Visible = True
    cmd_regresa.Visible = True
    cmd_consProm.Visible = True
    cmd_Construc_telpub.Visible =
True
End Sub

Private Sub
men_Manto_Tel_Pub_Click()
    cms_consTelPub.Visible = False
    cmd_regresa.Visible = False
    cmd_consProm.Visible = False
    cmd_Construc_telpub.Visible =
False
    fm_conmtp.Show vbModal
    cms_consTelPub.Visible = True
    cmd_regresa.Visible = True
    cmd_consProm.Visible = True
    cmd_Construc_telpub.Visible =
True
End Sub

'-----
'Archivo: fm_Mencarga
'Fecha: Julio-Septiembre de 1999
'Propósito: Módulo del Menú de
Carga de información para invocar a
las aplicaciones que permiten cargar
los archivos externos.
'Hace uso de menus y de botones
de uso continuo
'-----

Private Sub cmd_car_exiped_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen Visible =
False
    cmd_carmov.Visible = False
    cmd_config.Visible = False
    cmd_regreso.Visible = False
    fm_cep.Show vbModal
    cmd_car_exiped.Visible = True
    cmd_pronosticos.Visible = True
    cmd_cargaconsngen Visible = True
    cmd_carmov Visible = True
    cmd_config.Visible = True
    cmd_regreso.Visible = True
End Sub

Private Sub
cmd_car_exiped_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
    SB_mencarga.Panels(1).Text =
"Extracción y Carga de archivos de
Existencias y Pedidos desde
SIREM"
End Sub

Private Sub
cmd_cargaconsngen_Click()
    cmd_car_exiped.Visible = False
    cmd_pronosticos.Visible = False
    cmd_cargaconsngen.Visible =
False
    cmd_carmov.Visible = False
    cmd_config.Visible = False
    fm_gen.Show vbModal
    cmd_car_exiped.Visible = True
    cmd_pronosticos.Visible = True
    cmd_cargaconsngen Visible = True
    cmd_carmov.Visible = True
    cmd_config.Visible = True
    cmd_regreso.Visible = True
End Sub

Private Sub
cmd_cargaconsngen_MouseMove(Bu
tton As Integer, Shift As Integer, X
As Single, Y As Single)
    SB_mencarga.Panels(1).Text =
"Carga de consumos promedio
desde archivo dbf"
End Sub

Private Sub cmd_carmov_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
    cmd_carmov.Visible = False
    cmd_config.Visible = False
    cmd_regreso.Visible = False
    fm_cconn.Show vbModal
    cmd_car_exiped.Visible = True
    cmd_pronosticos.Visible = True
    cmd_cargaconsngen.Visible = True
    cmd_carmov.Visible = True
    cmd_config.Visible = True
    cmd_regreso.Visible = True
End Sub

Private Sub
cmd_carmov_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
    SB_mencarga.Panels(1).Text =
"Permite Extraer y Cargar el archivo
de movimientos de SIREM"
End Sub

Private Sub cmd_config_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False

```

```

cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
fm_cconf.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub
cmd_config_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
SB_mencarga.Panels(1).Text =
"Permite configurar conexión para
la transferencia de archivos y sus
nombres"
End Sub

Private Sub cmd_pronosticos_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
fm_cconf.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub cmd_pronosticos_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
fm_prono.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub cmd_pronosticos_MouseMove(Button
n As Integer, Shift As Integer, X As
Single, Y As Single)
SB_mencarga.Panels(1).Text =
"Permite la captura de pronósticos
de material"
End Sub

Private Sub cmd_regreso_Click()
Unload Me
End Sub

Private Sub
cmd_regreso_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
SB_mencarga.Panels(1).Text =
"Permite regresar al menu Principal
del Sistema"
End Sub

Private Sub Form_Load()
If s_gadmin Then
cmd_config.Enabled = True
Else
cmd_config.Enabled = False
End If
Text1 = s_gRutaSD
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
SB_mencarga.Panels(1).Text =
"Posicione el mouse en un boton
para ver acción que realiza"
End Sub

Private Sub
men_configuración_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
fm_cconf.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub men_Consumos_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
fm_gen.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub men_Exis_ped_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
fm_cep.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub
men_Movtos_SIREM_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
End Sub

Private Sub
frm_cconm.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub men_pronosticos_Click()
cmd_car_exiped.Visible = False
cmd_pronosticos.Visible = False
cmd_cargaconsngen.Visible =
False
cmd_carmov.Visible = False
cmd_config.Visible = False
cmd_regreso.Visible = False
fm_prono.Show vbModal
cmd_car_exiped.Visible = True
cmd_pronosticos.Visible = True
cmd_cargaconsngen.Visible = True
cmd_carmov.Visible = True
cmd_config.Visible = True
cmd_regreso.Visible = True
End Sub

Private Sub men_regresa_Click()
Unload Me
End Sub

'-----
'Archivo: fm_mendat_calcons
'Fecha: Julio-Septiembre de 1999
'Proposito: Módulo del Menú de
Carga de datos para el Cálculo de
consumos para invocar a las
aplicaciones que permiten cargar
dichos datos.
'Hace uso de menues y de botones
de uso continuo
'-----

Private Sub
cmd_fact_x_linea_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_inmat.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub
cmd_fact_x_linea_MouseMove(Button
n As Integer, Shift As Integer, X As
Single, Y As Single)
sb_mendat_calcons.Panels(1).Text =
"Captura de factores de
Construcción por Linea a Construir"
End Sub

Private Sub cmd_plan_anual_Click()

```

```

cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_planc.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub
cmd_plan_anual_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

sb_mendat_calcons.Panels(1).Text = "Captura de plan anual de construcción por Región"
End Sub

Private Sub cmd_pronosticos_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_procap.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub
cmd_pronosticos_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

sb_mendat_calcons.Panels(1).Text = "Captura manual de pronósticos por material y región"
End Sub

Private Sub cmd_regresa_Click()
Unload Me
End Sub

Private Sub
cmd_regresa_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

sb_mendat_calcons.Panels(1).Text = "Regresar al menú anterior"
End Sub

Private Sub
cmd_adm_manto_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_planc.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub cmd_confrepord_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_planc.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub cmd_confrepord_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
sb_menrepor.Panels(1).Text = "Genera reporte de avance de pedidos"
End Sub

Private Sub cmd_confrepord_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
End Sub

cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub
cmd_adm_manto_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

sb_mendat_calcons.Panels(1).Text = "Actualización del catalogo de ADM (Proyectos) de mantenimiento"
End Sub

Private Sub Form_Load()
Text1 = s_gRutaBD
End Sub

Private Sub
Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

sb_mendat_calcons.Panels(1).Text = "Posicione el Mouse en algún boton para ver la acción que realiza"
End Sub

Private Sub
men_ADM_Manto_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_adms.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub men_fact_linea_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_inmat.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub men_plan_anual_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_pronosticos.Visible = False
cmd_regresa.Visible = False
fm_planc.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

Private Sub
men_regresa_Click()
Unload Me
End Sub

Private Sub Mn_pronostico_Click()
cmd_plan_anual.Visible = False
cmd_fact_x_linea.Visible = False
cmd_adm_manto.Visible = False
cmd_regresa.Visible = False
fm_procap.Show vbModal
cmd_plan_anual.Visible = True
cmd_fact_x_linea.Visible = True
cmd_adm_manto.Visible = True
cmd_pronosticos.Visible = True
cmd_regresa.Visible = True
End Sub

'-----
'Archivo: fm_menReportes
'Fecha: Julio-Septiembre de 1999
'Propósito: Módulo del Menú de Reportes para invocar a las aplicaciones que permiten configurar y generar los reportes del sistema.
'Hace uso de menues y de botones de uso continuo
'-----

Private Sub cmd_avpedidos_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepord.Visible = False
cmd_genrepord.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_avan.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepord.Visible = True
cmd_genrepord.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub

Private Sub
cmd_avpedidos_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
sb_menrepor.Panels(1).Text = "Genera reporte de avance de pedidos"
End Sub

Private Sub cmd_confrepord_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepord.Visible = False
cmd_genrepord.Visible = False
cmd_consusu.Visible = False
End Sub

```

```
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_cnrp.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub
cmd_confrepor_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_menrepor.Panels(1).Text =
"Permite configurar el reporte
principal"
End Sub
```

```
Private Sub cmd_conjuntos_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_conjcat.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub
cmd_conjuntos_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_menrepor.Panels(1).Text =
"Genera reportes de los conjuntos
de catálogos"
End Sub
```

```
Private Sub cmd_consusu_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_entusu.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepor.Visible = True
```

```
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub
cmd_consusu_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_menrepor.Panels(1).Text =
"Reporte de Material entregado a
Usuarios"
End Sub
```

```
Private Sub cmd_regiones_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_regalm.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub
cmd_regiones_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_menrepor.Panels(1).Text =
"Reporte de regiones por Almacén
Distribuidor"
End Sub
```

```
Private Sub cmd_regresa_Click()
Unload Me
End Sub
```

```
Private Sub
cmd_regresa_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_menrepor.Panels(1).Text =
"Regresa al menú anterior"
End Sub
```

```
Private Sub cmd_genrepor_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
```

```
Fm_genrep.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub
cmd_genrepor_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_menrepor.Panels(1).Text =
"Generación del reporte principal
según configuración"
End Sub
```

```
Private Sub Form_Load()
Text1.Text = s_RutaBD
End Sub
```

```
Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
sb_menrepor.Panels(1).Text =
"Posicione el mouse en un boton
para ver su función"
End Sub
```

```
Private Sub men_avanped_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepor.Visible = False
cmd_genrepor.Visible = True
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_avan.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub men_confrepor_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_cnrp.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepor.Visible = True
```

```
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub men_conjuntos_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepopor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_conjcat.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepopor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub
men_entregasusu_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepopor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_entusu.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepopor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
Private Sub men_regiones_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepopor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
fm_regalm.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepopor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
```

```
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub

Private Sub men_regreso_Click()
Unload Me
End Sub
```

```
Private Sub men_repprin_Click()
cmd_conjuntos.Visible = False
cmd_regiones.Visible = False
cmd_confrepopor.Visible = False
cmd_genrepor.Visible = False
cmd_consusu.Visible = False
cmd_avpedidos.Visible = False
cmd_regresa.Visible = False
Label1.Visible = False
Label2.Visible = False
Fm_genrep.Show vbModal
cmd_conjuntos.Visible = True
cmd_regiones.Visible = True
cmd_confrepopor.Visible = True
cmd_genrepor.Visible = True
cmd_consusu.Visible = True
cmd_avpedidos.Visible = True
cmd_regresa.Visible = True
Label1.Visible = True
Label2.Visible = True
End Sub
```

```
-----
'Archivo: fm_mentablas
'Fecha: Julio-Septiembre de 1999
'Propósito: Módulo del Menú de
Tablas para invocar a las
aplicaciones que permiten configurar
y cargar las tablas base del sistema.
'Hace uso de menús y de botones
de uso continuo
-----
```

```
Private Sub
cmd_alamacenes_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_alma.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub
cmd_alamacenes_MouseMove(Butt
on As Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1) Text =
"Mantenimiento al Catálogo de
almacenes"
End Sub
```

```
Private Sub
cmd_bitacora_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1).Text =
"Botón para consultar la bitácora de
actualizaciones a la base de datos"
End Sub
```

```
Private Sub cmd_borrar_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_borra.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub
cmd_borrar_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Integer)
sb_mentablas.Panels(1).Text =
"Borra y respaldo en archivo las
existencias, pedidos y movimientos"
End Sub
```

```
Private Sub cmd_carcat_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_cata.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub
cmd_carcat_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1) Text =
"Permite hacer la carga de catálogos
del sistema"
End Sub
```

```
Private Sub cmd_regiones_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_regfact.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
```

```
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub
cmd_regiones_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1).Text =
"Mantenimiento a las regiones que
pueden surtir por traspaso"
End Sub
```

```
Private Sub cmd_regresa_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
Unload Me
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub
cmd_regresa_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1).Text =
"Regresa al Menú Anterior"
End Sub
```

```
Private Sub
cmd_resp_pronos_Click()
End Sub
```

```
Private Sub
cmd_resp_pronos_MouseMove(Butt
on As Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1).Text =
"Respaldo y borra los cálculos de
consumos por periodo"
End Sub
```

```
Private Sub cmd_usuarios_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_usuarios.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub
cmd_usuarios_MouseMove(Button
```

```
As Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1).Text =
"Botón para atener acceso a
Agregar, borrar o deshabilitar
Usuarios del sistema"
End Sub
```

```
Private Sub Form_Load()
Text1.Text = s_gRutaBD
End Sub
```

```
Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
sb_mentablas.Panels(1).Text =
"Posicione el raton en un botón para
saber la acción que realiza"
End Sub
```

```
Private Sub men_almacenes_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_alma.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub men_carcat_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_cata.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub men_reg_fact_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_regfact.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub men_regresa_Click()
```

```
Unload Me
End Sub
```

```
Private Sub men_resp_Cons_Click()
cmd_carcat.Visible = False
cmd_regiones.Visible = False
cmd_regresa.Visible = False
cmd_alamacenes.Visible = False
cmd_borrar.Visible = False
cmd_usuarios.Visible = False
fm_probor.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

```
Private Sub men_respdatos_Click()
fm_borra.Show vbModal
cmd_carcat.Visible = True
cmd_regiones.Visible = True
cmd_regresa.Visible = True
cmd_alamacenes.Visible = True
cmd_borrar.Visible = True
cmd_usuarios.Visible = True
End Sub
```

---

```
'Archivo: fm_planc
'Fecha: Julio-Septiembre de 1999
'Proposito: Captura de plan anual
por región
```

---

```
Option Explicit
Dim regiones(1 To 32) As String
Dim i_gNumRegion As Integer
Dim w_gws As Workspace
Dim d_gdb As Database
Dim t_gPA As Recordset
```

```
Private Sub cb_area_anio_Click()
Dim s_lAreaTM As String
Dim i_lAnio As String
Dim i_lregion As Integer
Dim i_lmes As Integer
Dim i_area_anio As String
i_area_anio =
CStr(cb_area_anio.Text)
s_lAreaTM = Left(i_area_anio,
1)
bx_areatm.Text =
Left(i_area_anio, 1)
i_lAnio = Right(i_area_anio, 4)
bx_anio.Text =
Right(i_area_anio, 4)
For i_lregion = 1 To
i_gNumRegion
For i_lmes = 1 To 12
msg_1.Col = i_lmes
msg_1.Row = i_lregion
t_gPA.Seek "=",
s_lAreaTM, regiones(i_lregion),
i_lmes, i_lAnio
If t_gPA.NoMatch Then
```



```

        MsgBox "No existe año
registrado a esta area ",
vbExclamation
    Exit Sub
Else
    msg_1.Text =
t_gPA.Fields("cantidad")
    End If
    Next i_mes
    Next i_region
    w_gws.BeginTrans
    Frame1.Enabled = True
    Frame1.Visible = True
    tx_areatm.Enabled = False
    tx_anio.Enabled = False
    cm_busca(0).Enabled = False
    msg_1.Row = 1
    msg_1.Col = 1
    lb_row.Caption = msg_1.Row
    lb_col.Caption = msg_1.Col
End Sub

Private Sub cm_actualiza_Click()
Dim s_lAreaTM As String
Dim i_lAnio As Integer
Dim i_lregion As Integer
Dim i_lmes As Integer
Dim i_lres As Integer
i_lres = MsgBox("Confirme
actualización ?", vbYesNo)
If i_lres = vbYes Then
s_lAreaTM = CStr(tx_areatm.Text)
i_lAnio = Val(tx_anio.Text)
For i_lregion = 1 To
i_gNumRegion
For i_lmes = 1 To 12
msg_1.Col = i_lmes
msg_1.Row = i_lregion
t_gPA.Seek "=", s_lAreaTM,
regiones(i_lregion), i_lmes, i_lAnio
If Not t_gPA.NoMatch Then
t_gPA.Edit
t_gPA.Fields("cantidad") =
Val(msg_1.Text)
t_gPA.Update
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
gl_ltb.Fields("tabla") =
"planan"
gl_ltb.Fields("fecha_hora")
= Date + Time
gl_ltb.Fields("Numtrans") =
3
gl_ltb.Update
End If
Next i_mes
Next i_region
w_gws.CommitTrans
cb_area_anio.Clear
Dim t_IPA As Recordset
Set d_gdb =
OpenDatabase(s_gRutaBD, True)
Set t_IPA =
d_gdb.OpenRecordset("Select
distinct areaTM,anio from planan",
dbOpenSnapshot)
t_IPA.MoveLast
t_IPA.MoveFirst
While Not t_IPA.EOF
cb_area_anio.AddItem
t_IPA.Fields("areaTM") + " " +
CStr(t_IPA.Fields("anio"))
t_IPA.MoveNext
Wend
Frame1.Enabled = False
Frame1.Visible = False
tx_areatm.Enabled = True
tx_anio.Enabled = True
cm_busca(0).Enabled = True
cb_area_anio.Enabled = True
MsgBox ("El registro a sido
grabado con los datos capturados")
End If
End Sub

Private Sub
cm_actualiza_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
frm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Presione
este botón para Grabar los cambios
que ha efectuado en el área de
Construcción que está editando"
End Sub

Private Sub cm_busca_Click(Index
As Integer)
Dim s_lAreaTM As String
Dim i_lAnio As String
Dim i_lregion As Integer
Dim i_lmes As Integer
cb_area_anio.Enabled = False
If tx_areatm.Text = vbNullString
Or tx_areatm.Text = "" Then
MsgBox "Proporcionar Area TM
valida", vbExclamation
cb_area_anio.Enabled = True
Exit Sub
End If
If Not
gVerifica_anio_valido(tx_anio.Text)
Then ' = vbNullString Or tx_anio.Text
= "" Then
MsgBox "Proporcionar Año
valido", vbExclamation
cb_area_anio.Enabled = True
Exit Sub
End If
w_gws.BeginTrans
s_lAreaTM =
CStr(tx_areatm.Text)
i_lAnio = Val(tx_anio.Text)
For i_lregion = 1 To
i_gNumRegion
For i_lmes = 1 To 12
msg_1.Col = i_lmes
msg_1.Row = i_lregion
t_gPA.Seek "=",
s_lAreaTM, regiones(i_lregion),
i_lmes, i_lAnio
If t_gPA.NoMatch Then
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") =
"planan"
gl_ltb.Fields("fecha_hora") = Date +
Time
gl_ltb.Fields("Numtrans") = 1
gl_ltb.Update
t_gPA.AddNew
t_gPA.Fields("areatm")
= s_lAreaTM
t_gPA.Fields("region") =
regiones(i_lregion)
t_gPA.Fields("mes") =
i_lmes
t_gPA.Fields("anio") =
i_lAnio
msg_1.Text = 0
t_gPA.Update
Else
MsgBox "Este año ya
esta registrado para esta area",
vbExclamation
w_gws.Rollback
Exit Sub
End If
Next i_mes
Next i_region
Frame1.Enabled = True
Frame1.Visible = True
tx_areatm.Enabled = False
tx_anio.Enabled = False
cm_busca(0).Enabled = False
msg_1.Row = 1
msg_1.Col = 1
lb_row.Caption = msg_1.Row
lb_col.Caption = msg_1.Col
End Sub

Private Sub
cm_busca_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
frm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Presione el
Boton para Crear la Planificación del
Área indicada para el Año que se
indica"
End Sub

Private Sub cm_cancela_Click()
w_gws.Rollback
Frame1.Enabled = False
Frame1.Visible = False
tx_areatm.Enabled = True
tx_anio.Enabled = True
cm_busca(0).Enabled = True
cb_area_anio.Enabled = True
MsgBox ("Ha cancelado la
edición. NO SE GRABARON LOS
CAMBIOS")
End Sub

Private Sub
cm_cancela_MouseMove(Button As

```

Integer, Shift As Integer, X As Single, Y As Single)

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Cancela las
modificaciones que se han Hecho en
la Presente Edicio y permite Escoger
una nueva o Crear otra"
End Sub
```

```
Private Sub cm_salir_Click()
Unload Me
End Sub
```

```
Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Regresa a la
pantalla Menu Datos para Cálculo
de Consumos"
End Sub
```

```
Private Sub Form_Load()
Dim i_li As Integer
Dim d_ldb As Database
Dim t_ltb As Recordset
```

```
msg_1.Cols = 13
msg_1.Row = 0
msg_1.Col = 1
msg_1.Col = 1
msg_1.Col = 1
msg_1.Text = "Enero"
msg_1.Col = 2
msg_1.Text = "Febrero"
msg_1.Col = 3
msg_1.Text = "Marzo"
msg_1.Col = 4
msg_1.Text = "Abril"
msg_1.Col = 5
msg_1.Text = "Mayo"
msg_1.Col = 6
msg_1.Text = "Junio"
msg_1.Col = 7
msg_1.Text = "Julio"
msg_1.Col = 8
msg_1.Text = "Agosto"
msg_1.Col = 9
msg_1.Text = "Septiembre"
msg_1.Col = 10
msg_1.Text = "Octubre"
msg_1.Col = 11
msg_1.Text = "Noviembre"
msg_1.Col = 12
msg_1.Text = "Diciembre"
msg_1.Col = 0
```

```
'Carga de Regiones
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set t_ltb =
d_ldb.OpenRecordset("select region
from region order by region",
dbOpenSnapshot)
t_ltb.MoveLast
t_ltb.MoveFirst
If t_ltb.RecordCount > 0 Then
msg_1.Rows =
t_ltb.RecordCount + 1
```

```
i_li = 1
While Not t_ltb.EOF
msg_1.Row = i_li
msg_1.Text =
t_ltb.Fields("region")
regiones(i_li) =
t_ltb.Fields("region")
t_ltb.MoveNext
i_li = i_li + 1
Wend
i_gNumRegion = i_li - 1
Else
i_gNumRegion = 0
End If
d_ldb.Close
msg_1.Row = 1
msg_1.Col = 1
```

```
'Inicializa espacio de trabajo
CONREP para transac
Set w_gws =
DBEngine.CreateWorkspace("PLAN
C", "admin", "", dbUseJet)
Set d_gdb =
w_gws.OpenDatabase(s_gRutaBD,
True)
Set t_gPA =
d_gdb.OpenRecordset("planan",
dbOpenTable)
t_gPA.Index = "PrimaryKey"
Set t_ltb =
d_gdb.OpenRecordset("Select
distinct areaTM,anio from planan",
dbOpenSnapshot)
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
cb_area_anio.AddItem
t_ltb.Fields("areaTM") + " " +
CStr(t_ltb.Fields("anio"))
t_ltb.MoveNext
Wend
End Sub
```

```
Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Posicione el
Mouse en algún boton o Campo
para Obtener ayuda de su función"
End Sub
```

```
Private Sub
Frame1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Posicione el
Mouse en algún boton o Campo
para Obtener ayuda de su función"
End Sub
```

```
Private Sub msg_1_Click()
lb_row.Caption = msg_1.Row
lb_col.Caption = msg_1.Col
```

```
If msg_1.Text = "0" Then
bx_cantidad.Text = ""
Else
bx_cantidad.Text = msg_1.Text
End If
bx_cantidad.SetFocus
End Sub
```

```
Private Sub
msg_1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Área en la
que debe capturar el PLAN de
CONSTRUCCIÓN para cada Región
en cada MES del Año indicado"
End Sub
```

```
Private Sub Picture1_Click()
MsgBox ("Presione la flecha para
buscar un elemento ")
End Sub
```

```
Private Sub
Picture1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Para Buscar
un Área de Construcción en un
determinado Año, Busque esta
haciendo Click en la Flecha de la
derecha"
End Sub
```

```
Private Sub Picture2_Click()
Dim s_lAreaTM As String
Dim i_Anio As Integer
Dim i_lregion As Integer
Dim i_lmes As Integer
Dim i_lres
i_lres = MsgBox(" Confirme la
eliminación de los registros ? ",
vbYesNo)
```

```
If i_lres = vbYes Then
s_lAreaTM = CStr(bx_areaTM.Text)
i_Anio = Val(bx_anio.Text)
For i_lregion = 1 To
i_gNumRegion
For i_lmes = 1 To 12
msg_1.Col = i_lmes
msg_1.Row = i_lregion
t_gPA.Seek "=", s_lAreaTM,
regiones(i_lregion), i_lmes, i_Anio
If Not t_gPA.NoMatch Then
t_gPA.Edit
t_gPA.Fields("cantidad") =
Val(msg_1.Text)
t_gPA.Delete
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos")
= s_gRutaBD
"planan"
gl_ltb.Fields("tabla") =
```

```

        gl_fib.Fields("fecha_hora")
    = Date + Time
        gl_fib.Fields("Numtrans") =
2
        gl_fib.Update
    End If
    Next i_lmes
    Next i_region
    w_gws.CommitTrans
    cb_area_anio.Clear
    Dim t_IPA As Recordset
    Set d_gdb =
OpenDatabase(s_gRutaBD, True)
    Set t_IPA =
d_gdb.OpenRecordset("Select
distinct areaTM,anio from planan",
dbOpenSnapshot)
    t_IPA.MoveLast
    t_IPA.MoveFirst
    While Not t_IPA.EOF
        cb_area_anio.AddItem
t_IPA.Fields("areaTM") + " " +
CStr(t_IPA.Fields("anio"))
        t_IPA.MoveNext
    Wend
    Frame1.Enabled = False
    Frame1.Visible = False
    tx_areatm.Enabled = True
    tx_anio.Enabled = True
    cm_busca(0).Enabled = True
    cb_area_anio.Enabled = True
    MsgBox ("El registro fue eliminado
de la Base de Datos")
    End If
End Sub

```

```

Private Sub
Picture2_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Presione el
Bote para eliminar la Planificación de
esta área en el Año indicado"
End Sub

```

```

Private Sub
tx_anio_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Proporcione
el Año a que corresponde la
Planificación de esta ÁREA"
End Sub

```

```

Private Sub
tx_areatm_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Proporcione
el área de Construcción que Quiere
Planificar, para dar la Alta"
End Sub

```

```

Private Sub tx_cantidad_Change()

```

```

msg_1.Text = tx_cantidad.Text
End Sub

```

```

Private Sub
tx_cantidad_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Indica la
cantidad Pronosticada que se
grabará en la celda seleccionada"
End Sub

```

```

-----
'Archivo: fm_probor
'Autor Gil Tapia (Hildebrando, S.A.
de C V)
'Fecha: Julio-Septiembre de 1997
'Proposito: Borra de base de datos
pronosticos
-----

```

```

Option Explicit
Private Sub
ck_anio_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al Marcar
esta Casilla, se Toma el Año como
condición de Borrado"
End Sub

```

```

Private Sub
ck_cat_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al Marcar
esta Casilla, se Toma el Catálogo
como condición de Borrado"
End Sub

```

```

Private Sub
ck_mes_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al Marcar
esta Casilla, se Toma el Mes como
condición de Borrado"
End Sub

```

```

Private Sub
ck_region_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Al Marcar
esta Casilla, se Toma la Región
como condición de Borrado"
End Sub

```

```

Private Sub cm_borrar_Click()
Dim i_lres As Integer
Dim w_lws As Workspace
Dim d_ldb As Database
Dim s_lcad As String
Dim d_lfecha As Date
Dim s_lcadena As String
Dim b_lophab As Boolean
On Error GoTo ATIENDE_ERROR
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo +
vbExclamation)
If i_lres = vbYes Then
    Set w_lws =
DBEngine.CreateWorkspace("PRON
B", "admin", "", dbUseJet)
    Set d_ldb =
w_lws.OpenDatabase(s_gRutaBD,
True)
    b_lophab = False
    s_lcadena = "Delete from
pronosticos where "
    If ck_cat.Value = 1 Then
        If cb_cat.Text = "" Or
cb_cat.Text = vbNullString Then
            MsgBox "Especifique
catalogo"
            d_ldb.Close
            Exit Sub
        End If
        s_lcadena = s_lcadena + "
catalogo = " + cb_cat.Text + " "
        b_lophab = True
    End If
    If ck_region.Value = 1 Then
        If cb_region.Text = "" Or
cb_region.Text = vbNullString Then
            MsgBox "Especifique
region"
            d_ldb.Close
            Exit Sub
        End If
        If b_lophab Then
            s_lcadena = s_lcadena + "
and region = " +
UCase(cb_region.Text) + " "
        Else
            s_lcadena = s_lcadena + "
region = " + UCCase(cb_region.Text)
+ " "
        End If
        b_lophab = True
    End If
    If ck_mes.Value = 1 Then
        If cb_mes.Text = "" Or
cb_mes.Text = vbNullString Then
            MsgBox "Especifique mes(
1 a 12 )"
            d_ldb.Close
            Exit Sub
        End If
        If b_lophab Then
            s_lcadena = s_lcadena + "
and mes = " + cb_mes.Text + " "
        Else
            s_lcadena = s_lcadena + "
mes = " + cb_mes.Text + " "
        End If
        b_lophab = True
    End If

```



```

Select Case Index
Case 0 'Alta
    If cb_catalogo.Text = "" Or
IsNull(cb_catalogo.Text) Then
        MsgBox "Asigne un
catalogo", vbExclamation
        Exit Sub
    End If
    If cb_region.Text = "" Or
IsNull(cb_region.Text) Then
        MsgBox "Asigne una
region", vbExclamation
        Exit Sub
    Else
        cb_region.Text =
UCase(cb_region.Text)
    End If
    If cb_mes.Text = "" Or
IsNull(cb_mes.Text) Then
        MsgBox "Asigne un mes (1
a 12)", vbExclamation
        Exit Sub
    End If
    If cb_anio.Text = "" Or
IsNull(cb_anio.Text) Then
        MsgBox "Asigne un año
(XXXX)", vbExclamation
        Exit Sub
    End If
    tb_DR.Seek "=",
cb_catalogo.Text, cb_region.Text,
Val(cb_mes.Text), Val(cb_anio.Text)
    If Not tb_DR.NoMatch Then
        MsgBox "Atención: El
identificador ya existe use otro",
vbExclamation
        Exit Sub
    End If
    cb_catalogo.Enabled = False
    cb_region.Enabled = False
    cb_mes.Enabled = False
    cb_anio.Enabled = False
    d_lws.BeginTrans
    b_lHayTransaccion = True
    gl_ltb.AddNew
    gl_ltb.Fields("usuario") =
s_login
    gl_ltb.Fields("basedatos") =
s_gRutaBD
    gl_ltb.Fields("tabla") =
"pronosticos"
    gl_ltb.Fields("fecha_hora") =
Date + Time
    gl_ltb.Fields("Numtrans") = 1
    gl_ltb.Update
    tb_DR.AddNew

    'Carga a memoria
    tx_pronostico.Text = ""
    tx_pronostico.Enabled = True
    cm_control(0).Enabled =
False
    cm_control(1).Enabled =
True
    cm_control(2).Enabled =
True
    cm_control(3).Enabled =
True
    cm_control(4).Enabled =
False

Case 1 'Baja
    i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
    If i_lres = vbYes Then
        tb_DR.Delete
        gl_ltb.AddNew
        gl_ltb.Fields("usuario") =
s_login
        gl_ltb.Fields("basedatos")
= s_gRutaBD
        gl_ltb.Fields("tabla") =
"pronosticos"
        gl_ltb.Fields("fecha_hora")
= Date + Time
        gl_ltb.Fields("Numtrans") =
2
        gl_ltb.Update
        MsgBox "Pronostico
eliminado"
    End If
    d_lws.CommitTrans
    b_lHayTransaccion = False
    cb_catalogo.Text = ""
    cb_catalogo.Enabled = True
    cb_region.Text = ""
    cb_region.Enabled = True
    cb_mes.Text = ""
    cb_mes.Enabled = True
    cb_anio.Text = ""
    cb_anio.Enabled = True
    tx_pronostico.Text = ""
    tx_pronostico.Enabled =
False
    cm_control(0).Enabled =
True
    cm_control(1).Enabled =
False
    cm_control(2).Enabled =
False
    cm_control(3).Enabled =
False
    cm_control(4).Enabled =
True

Case 2 'Actualiza
    i_lres = MsgBox("Confirma
Actualización ?", vbYesNo)
    If i_lres = vbYes Then
        tb_DR.Fields("catalogo") =
cb_catalogo.Text
        tb_DR.Fields("region") =
cb_region.Text
        tb_DR.Fields("mes") =
cb_mes.Text
        tb_DR.Fields("anio") =
cb_anio.Text
        'Carga a TABLA
        tb_DR.Fields("pronos") =
tx_pronostico.Text
        tb_DR.Update
        gl_ltb.AddNew
        gl_ltb.Fields("usuario") =
s_login
        gl_ltb.Fields("basedatos") =
s_gRutaBD
        gl_ltb.Fields("tabla") =
"pronosticos"
        gl_ltb.Fields("fecha_hora") =
Date + Time
        gl_ltb.Fields("Numtrans") = 3
        gl_ltb.Update
        d_lws.CommitTrans
        b_lHayTransaccion = False
        cb_catalogo.Text = ""
        cb_catalogo.Enabled = True
        cb_region.Text = ""
        cb_region.Enabled = True
        cb_mes.Text = ""
        cb_mes.Enabled = True
        cb_anio.Text = ""
        cb_anio.Enabled = True
        tx_pronostico.Text = ""
        tx_pronostico.Enabled =
False
        cm_control(0).Enabled =
True
        cm_control(1).Enabled =
False
        cm_control(2).Enabled =
False
        cm_control(3).Enabled =
False
        cm_control(4).Enabled =
True

Case 3 'Cancela Cambios
    d_lws.Rollback
    b_lHayTransaccion = False
    cb_catalogo.Text = ""
    cb_catalogo.Enabled = True
    cb_region.Enabled = True
    cb_mes.Text = ""
    cb_mes.Enabled = True
    cb_anio.Text = ""
    cb_anio.Enabled = True
    tx_pronostico.Text = ""
    tx_pronostico.Enabled =
False
    cm_control(0).Enabled =
True
    cm_control(1).Enabled =
False
    cm_control(2).Enabled =
False
    cm_control(3).Enabled =
False
    cm_control(4).Enabled =
True

Case 4 'Busca
    cb_region.Text =
UCase(cb_region.Text)
    tb_DR.Seek "=",
cb_catalogo.Text, cb_region.Text,
Val(cb_mes.Text), Val(cb_anio.Text)
    If tb_DR.NoMatch Then
        MsgBox "No hay pronostico
con ese identificador",
vbExclamation
    Else
        d_lws.BeginTrans
        b_lHayTransaccion = True
        tb_DR.Edit
        cb_catalogo.Enabled = False
        cb_region.Enabled = False
        cb_mes.Enabled = False
        cb_anio.Enabled = False
        tx_pronostico.Text = ""
        tx_pronostico.Enabled = True
    End If
End If

```

```

False cm_control(0).Enabled = Set d_lws =
d_lws.OpenDatabase(s_gRutaBD,
True)
cm_control(1).Enabled = Set tb_DR =
True cm_control(2).Enabled = d_ldb.OpenRecordset("pronosticos",
True cm_control(3).Enabled = dbOpenTable)
True cm_control(3).Enabled = tb_DR.Index = "PrimaryKey"
False cm_control(4).Enabled = Set l_ttb =
d_ldb.OpenRecordset("select
'Carga a memoria catalogo from catalogo order by
bx_pronostico.Text = l_ttb.MoveLast
tb_DR.Fields("pronos") l_ttb.MoveFirst
End If While Not l_ttb.EOF
End Select cb_catalogo.AddItem
End Sub l_ttb("catalogo")
l_ttb.MoveNext
Wend
l_ttb.Close
Set l_ttb =
Private Sub d_ldb.OpenRecordset("select region
cm_control_MouseMove(Index As from region order by region",
Integer, Button As Integer, Shift As Integer, X As Single, Y As Single) dbOpenDynaset)
l_ttb.MoveLast
fm_mendat_calcons.sb_mendat_cal cons.Panels(1).Text = "Presione un l_ttb.MoveFirst
Boton para Actualizar la Base de Datos (Nuevo Registro, Borrar l_ttb.MoveFirst
Registro, Grabar Registro, Cancelar While Not l_ttb.EOF
Modificación o Buscar" cb_region.AddItem
End Sub l_ttb("region")
l_ttb.MoveNext
Wend
l_ttb.Close
Private Sub cmd_borrotot_Click() For l_mes = 1 To 12
fm_probor Show vbModal cb_mes.AddItem l_mes
End Sub Next
For l_anio = 1995 To 2100
Private Sub cmd_borrotot_MouseMove(Button cb_anio.AddItem l_anio
As Integer, Shift As Integer, X As Single, Y As Single) Next
End Sub
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
fm_mendat_calcons.sb_mendat_cal cons.Panels(1).Text = "Presione el Form_MouseMove(Button As
Boton para Hacer un Borrado Integer, Shift As Integer, X As Single, Y As Single)
colectivo de Pronósticos"
fm_mendat_calcons.sb_mendat_cal cons.Panels(1).Text = "Posicione el
End Sub Mouse en algún boton o Campo
para Obtener ayuda de su función"
Private Sub Command1_Click() End Sub
Unload Me
End Sub
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
fm_mendat_calcons.sb_mendat_cal cons.Panels(1).Text = "Regresa el
End Sub Menú Datos para Cálculo de
Consumos"
End Sub
Private Sub Form_Load()
'Inicializa espacio de trabajo
CONREP para transac
Dim l_ttb As Recordset
Dim l_mes As Integer
Dim l_anio As Integer
Set d_lws =
DBEngine.CreateWorkspace("PRON
", "admin", "", dbUseJet)

```

```

donde Registrar el Pronóstico del
Catálogo Indicado"
End Sub
Private Sub
Label3_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Seleccione
en la lista de la derecha, un mes
para el cual será válido el
pronóstico"
End Sub
Private Sub
Label4_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Selección un
Año para el cual va a Capturar o
Consultar un Pronóstico"
End Sub
Private Sub
tx_descripcion_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Presenta la
Descripción del Catálogo
seleccionado"
End Sub
Private Sub
tx_pronostico_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
fm_mendat_calcons.sb_mendat_cal
cons.Panels(1).Text = "Muestra la
Cantidad Pronosticada y Permite
modificarla"
End Sub
'-----
'Archivo: fm_prono
'Fecha: Julio-Septiembre de 1999
'Propósito: Carga de pronósticos de
archivos dBase hacia la base de
datos interna
'hecha en Access.
'-----
Option Explicit
Private Sub cb_region_Scroll()
fm_mencarga.SB_mencarga.Panels(
1).Text = "Seleccione Una REGIÓN
a la que desee agregar el
PRONÓSTICO"
End Sub
Private Sub cm_campos_Click()

```

```

Dim d_idBASE As Database
Dim r_ITabla As Recordset
Dim s_iRut As String
Dim s_iNom As String
Dim i_li As Integer
If tx_archivo.Text = vbNullString
Or Dir(tx_archivo.Text) = "" Then
MsgBox "Proporcione un
nombre de archivo ", vbExclamation
Exit Sub
End If
It_campos.Clear
tx_anoio.Text = ""
tx_mes.Text = ""
tx_cat.Text = ""
tx_pron.Text = ""

'Abrir archivo dbase
Call gObtenRutas(tx_archivo.Text,
s_iRut, s_iNom)
If True Then 'tabla_dbase then
On Error GoTo termina

' Open the external dBASE
database.
Set d_idBASE =
OpenDatabase(s_iRut, False, False,
"dBASE III;")

' Abre tabla EXISTEN
Set r_ITabla =
d_idBASE.OpenRecordset(s_iNom)
If r_ITabla.RecordCount > 0
Then
It_campos.Clear
For i_li = 0 To
r_ITabla.Fields.Count - 1
It_campos.AddItem
r_ITabla.Fields(i_li).Name
Next i_li
cm_cata.Enabled = True
cm_prono.Enabled = True
Else
MsgBox "La tabla no tiene
registros"
End If
r_ITabla.Close
d_idBASE.Close
Else 'Tabla plana ASCII
End If
Exit Sub
termina:
MsgBox "Pronosticos: El
archivo especificado tiene algun
problema verifique", vbExclamation
Exit Sub
End Sub

Private Sub
cm_campos_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Rastrea los Campos del
Archivo Indicado para Solicitar se
Asignen a los Datos Necesarios"
End Sub

'Proposito: Carga a pronosticos los
datos de la tabla externa
Private Sub cm_carga_Click()
Dim d_idBASE As Database
Dim r_ITabla As Recordset
Dim w_IWs As Workspace
'Para workspace
Dim d_idb As Database 'Para
base de datos
Dim i_IPro As Recordset 'Para
pronosticos
Dim i_iCat As Recordset 'Para
catalogos
Dim s_iRut As String
Dim s_iNom As String
Dim s_iRegion As String 'La
region a asignar
Dim s_iCat As String 'Nombre
de campo a asignar a catalogo
Dim s_iPron As String 'Nombre
de campo a asignar a pronostico
Dim i_Anio As Integer 'Año a
asignar
Dim i_iMesIni As Integer 'Mes
inicial a asignar
Dim i_iMesFin As Integer 'Hasta
mes final a asignar
Dim s_iBusca As String 'Buffer
para busqueda de datos
Dim i_iPos As Integer 'Variable
auxiliar
Dim i_li As Integer 'Variable a
auxiliar
Dim s_iCat_tab As String
'Nombre de campo a asignar a
catalogo

'Se verifican los datos para iniciar
transferencia
If cb_region.Text = vbNullString Or
cb_region.Text = "" Then
MsgBox "Especifique Region ",
vbExclamation
Exit Sub
Else
s_iRegion = cb_region.Text
End If
If tx_mes.Text = vbNullString Then
MsgBox "Especifique Mes ",
vbExclamation
Exit Sub
Else
'Se buscan mes inicial y final (si
es que los hay)
i_iPos = InStr(1, tx_mes.Text, "-"
)

If i_iPos > 0 Then
i_iMesIni =
Val(Mid(tx_mes.Text, 1, i_iPos - 1))
i_iMesFin =
Val(Mid(tx_mes.Text, i_iPos + 1))
Else
i_iMesIni = Val(tx_mes.Text)
i_iMesFin = Val(tx_mes.Text)
End If
End If
If tx_anoio.Text = vbNullString Then
MsgBox "Especifique Año ",
vbExclamation
Exit Sub
Else
i_Anio = Val(tx_anoio.Text)
End If
If tx_cat.Text = vbNullString Then
MsgBox "Especifique campo de
catalogo ", vbExclamation
Exit Sub
Else
s_iCat = tx_cat.Text
End If
If tx_pron.Text = vbNullString
Then
MsgBox "Especifique campo de
pronostico ", vbExclamation
Exit Sub
Else
s_iPron = tx_pron.Text
End If

'Abrir archivo dbase
Call gObtenRutas(tx_archivo.Text,
s_iRut, s_iNom)
If True Then 'tabla_dbase then
On Error GoTo termina

' Open the external dBASE
database.
Set d_idBASE =
OpenDatabase(s_iRut, False, False,
"dBASE III;")

' Abre tabla EXISTEN
Set r_ITabla =
d_idBASE.OpenRecordset(s_iNom)
If r_ITabla.RecordCount > 0
Then
r_ITabla.MoveLast
r_ITabla.MoveFirst
Set w_IWs =
DBEngine.CreateWorkspace("PRON
O", "admin", "", dbUseJet)
Set d_idb =
w_IWs.OpenDatabase(s_gRutaBD,
True)
Set i_IPro =
d_idb.OpenRecordset("pronosticos ",
dbOpenTable)
Set i_iCat =
d_idb.OpenRecordset("catalogo",
dbOpenTable)
i_IPro.Index = "PrimaryKey"
i_iCat.Index = "PrimaryKey"
If i_IPro.RecordCount > 0
Then
i_IPro.MoveLast
End If
On Error GoTo 0 'termina3
w_IWs.BeginTrans

'Proceso de carga a tabla
pronosticos
While Not r_ITabla.EOF
If Not
IsNull(r_ITabla.Fields(s_iCat)) Then
s_iCat_tab =
Mid(CStr(r_ITabla.Fields(s_iCat)), 1,
7)
i_iCat.Seek "=",
s_iCat_tab

```

```

If Not t_iCat.NoMatch
Then
    For i_li = i_mesIni To
i_mesFin
        s_iBusca = " catalogo
= " + r_iTabla.Fields(s_iCat) + "
        s_iBusca = s_iBusca
+ " and region = " + s_iRegion + "
        s_iBusca = s_iBusca
+ " and mes = " + Str(i_li)
        s_iBusca = s_iBusca
+ " and anio = " + Str(i_Anio)
        t_iPro.Seek "=",
r_iTabla.Fields(s_iCat), s_iRegion,
i_li, i_Anio
Then
    If t_iPro.NoMatch
Then
        With t_iPro
            gl_iTb.AddNew
gl_iTb.Fields("usuario") = s_login
gl_iTb.Fields("basedatos") =
s_gRutaBD
gl_iTb.Fields("tabla") = "pronosticos"
gl_iTb.Fields("fecha_hora") = Date +
Time
gl_iTb.Fields("Numtrans") = 1
gl_iTb.Update
AddNew
.Fields("catalogo") =
s_iCat_tab
.Fields("region") =
s_iRegion
.Fields("mes") = i_li
.Fields("anio") =
i_Anio
.Fields("pronos") =
CDbl(r_iTabla.Fields(s_iPron))
Update
End With
Else
With t_iPro
.Edit
.Fields("pronos") =
r_iTabla.Fields(s_iPron)
.Update
gl_iTb.AddNew
gl_iTb.Fields("usuario") = s_login
gl_iTb.Fields("basedatos") =
s_gRutaBD
gl_iTb.Fields("tabla") =
"pronosticos"
gl_iTb.Fields("fecha_hora") = Date +
Time
gl_iTb.Fields("Numtrans") = 3
gl_iTb.Update
End With
End If
Next i_li
End If
r_iTabla.MoveNext

DoEvents
Wend
If vbYes = MsgBox("¿
Confirma actualización de
pronosticos ?", vbYesNo) Then
w_iWs.CmdMTrans
Else
MsgBox ("Los pronósticos
fueron cargados en la tabla de
pronósticos por Región")
w_iWs.Rollback
End If
t_iPro.Close
d_iDb.Close
w_iWs.Close
Else
MsgBox "La tabla no tiene
registros"
End If
r_iTabla.Close
d_iDbASE.Close
Else "Tabla plana ASCII
End If
Exit Sub
termina:
MsgBox "Pronosticos: El
archivo especificado tiene algun
problema verifique", vbExclamation
Exit Sub
termina2:
MsgBox "Pronosticos: No se
puede cargar la base de datos
principal ", vbExclamation
t_iPro.Close
d_iDbASE.Close
Exit Sub
termina3:
MsgBox "Pronosticos: Error en
carga de datos ", vbExclamation
r_iTabla.Close
d_iDbASE.Close
Exit Sub
End Sub

Private Sub
cm_carga_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Al Precionar este boton,
comienza a Cargar los datos del
Archivo a la Base de Datos según se
hayan asignado los Campos"
End Sub

Private Sub cm_cata_Click()
If It_campos.ListIndex <> -1 Then
bx_cat.Text =
It_campos.List(It_campos.ListIndex)
Else
MsgBox "Seleccione un campo
"
End If
End Sub

Private Sub
cm_cata_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "El campo que Tiene
Seleccionado, lo Asigna para que de
el se tomen los datos de
CATÁLOGO"
End Sub

Private Sub cm_prono_Click()
If It_campos.ListIndex <> -1 Then
bx_pron.Text =
It_campos.List(It_campos.ListIndex)
Else
MsgBox "Seleccione un campo
"
End If
End Sub

Private Sub
cm_prono_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "El campo que Tiene
Seleccionado, lo Asigna para que de
el se tomen los datos de
PRONÓSTICO"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Regresa al Menú Para
Carga de Información"
End Sub

Private Sub cmd_buscaarch_Click()
With dlgComm1
.DialogTitle = "Busqueda de
archivo dbf de pronósticos"
.Filter = "d*BaselIII (*.dbf)".dbf
.ShowOpen
If Len(.FileName) = 0 Then
Exit Sub
End If
bx_archivo.Text = .FileName
End With
End Sub

Private Sub
cmd_buscaarch_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

fm_mencarga.SB_mencarga.Panels(
1).Text = "Abre cuadro de diálogo
para buscar el archivo en la PC"
End Sub

Private Sub Form_Load()
Dim d_iDb As Database
Dim t_iTb As Recordset

```



```

Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set t_ltb =
d_ldb.OpenRecordset("select region
from region order by region",
dbOpenSnapshot)
t_ltb.Clear
cb_region.Clear
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
t_ltb.Region.AddItem
t_ltb.Fields("region")
cb_region.AddItem
t_ltb.Fields("region")
t_ltb.MoveNext
Wend
t_ltb.Close
d_ldb.Close
t_ltb.Campos.Clear
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Posicione el MOUSE en
un Campo o Boton para Obtener
Ayuda"
End Sub

```

```

Private Sub
Frame1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Posicione el MOUSE en
un Campo o Boton para Obtener
Ayuda"
End Sub

```

```

Private Sub
bx_anio_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Teclee el Año al que
Pertenece los PRONÓSTICOS que
está Cargando"
End Sub

```

```

Private Sub
lx_archivo_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Indique el Nombre de
Archivo que desea se Cargue y que
contiene dos Datos del
PRONÓSTICO"
End Sub

```

```

Private Sub
tx_mes_MouseMove(Button As

```

```

Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_mencarga.SB_mencarga.Panels(
1).Text = "Teclee el Mes al que
Pertenece los PRONÓSTICOS que
está Cargando"
End Sub

```

```

'-----
'Archivo: fm_regalm
'Fecha: Julio-Septiembre de 1999
'Proposito: Carga regiones a las que
solo se considerará almacén
principal
'-----

```

```

Option Explicit
Dim d_gdb As Database
Dim t_gtb As Recordset
Private Sub cm_agrega_Click()
Dim i As Integer
If t_gtb.ListIndex <> -1 Then
For i = 0 To t_regalm.ListCount
- 1

```

```

If
t_regalm.List(t_regalm.ListIndex) =
t_regalm.List(i) Then
MsgBox "Ya existe region"
Exit Sub
End If
Next i

```

```

t_regalm.AddItem
t_regalm.List(t_regalm.ListIndex)
gl_ltb.AddNew
gl_ltb.Fields("usuario") =
s_login
gl_ltb.Fields("basedatos") =
s_gRutaBD
gl_ltb.Fields("tabla") = "region"
gl_ltb.Fields("fecha_hora") =
Date + Time
gl_ltb.Fields("Numtrans") = 1
gl_ltb.Update
t_gtb.AddNew
t_gtb.Fields("region") =
t_regalm.List(t_regalm.ListIndex)
t_gtb.Update

```

```

'Recarga de regalm
t_regalm.Clear
If t_gtb.RecordCount > 0 Then
t_gtb.MoveLast
t_gtb.MoveFirst
While Not t_gtb.EOF
t_regalm.AddItem
t_gtb.Fields("region")
t_gtb.MoveNext
Wend
End If
End Sub

```

```

Private Sub
cm_agrega_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al presionar este botón,
agrega a la lista, la Región
Seleccionada en el catálogo"
End Sub

```

```

Private Sub cm_safir_Click()
Unload Me
End Sub

```

```

Private Sub
cm_safir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Regresa al Menú de
Reportes"
End Sub

```

```

Private Sub Form_Load()

```

```

' Carga de Regiones
Set d_gdb =
OpenDatabase(s_gRutaBD, True)
Set t_gtb =
d_gdb.OpenRecordset("select region
from region order by region",
dbOpenSnapshot)
If t_gtb.RecordCount > 0 Then
t_gtb.MoveLast
t_gtb.MoveFirst
While Not t_gtb.EOF
t_regalm.AddItem
t_gtb.Fields("region")
t_gtb.MoveNext
Wend

```

```

End If
t_gtb.Close

' Carga de RegAlm
Set t_gtb =
d_gdb.OpenRecordset("regalm",
dbOpenTable)
t_gtb.Index = "RAL_IDX"
If t_gtb.RecordCount > 0 Then
t_gtb.MoveLast
t_gtb.MoveFirst
While Not t_gtb.EOF
t_regalm.AddItem
t_gtb.Fields("region")
t_gtb.MoveNext
Wend
End If
End Sub

```

```

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

```

```

fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un botón para ver su función"
End Sub

```

```

Private Sub
Frame1_MouseMove(Button As

```

```

Integer, Shift As Integer, X As
Single, Y As Single)

```

Integer, Shift As Integer, X As Single, Y As Single)

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Posicione el mouse en
un botón para ver su función"
End Sub
```

```
Private Sub
It_regalm_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Lista de Regiones que
van a tomar únicamente la
Existencia de los Almacenes
Distribuidores"
End Sub
```

```
Private Sub It_region_Click()
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Lista Total de Regiones
Existentes"
End Sub
```

```
Private Sub pc_borra_Click()
Dim i_lres As Integer
If It_regalm.ListIndex <> -1 Then
i_lres = MsgBox("¿Confirma
eliminación de " +
It_regalm.List(i_lres).ListIndex) +
"?", vbYesNo)
```

```
If i_lres = vbYes Then
t_gtb.Seek "="
It_regalm.List(i_lres).ListIndex)
If Not t_gtb.NoMatch Then
t_gtb.Delete
gl_ttb.AddNew
gl_ttb.Fields("usuario") =
s_login
gl_ttb.Fields("basedatos")
= s_gRutaBD
gl_ttb.Fields("tabla") =
"region"
gl_ttb.Fields("fecha_hora")
= Date + Time
gl_ttb.Fields("Numtrans") =
2
gl_ttb.Update
End If
```

```
'Recarga de regalm
It_regalm.Clear
If t_gtb.RecordCount > 0
```

```
Then
t_gtb.MoveLast
t_gtb.MoveFirst
While Not t_gtb.EOF
It_regalm.AddItem
t_gtb.Fields("region")
t_gtb.MoveNext
Wend
End If
End If
End If
End Sub
```

```
Private Sub
pc_borra_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_menReportes.sb_menrepor.Pane
ls(1).Text = "Al presionar el bote,
Elimina de la lista la Región
Marcada"
End Sub
```

```
-----
'Archivo: fm_regfact
'Fecha: Julio-Septiembre de 1999
'Propósito: Captura de regiones
factibles. Para uso por reporte de
pedidos y reporte de traspasos.
-----
```

```
Option Explicit
Dim regiones(1 To 32) As String
Dim i_gNumRegion As Integer
Dim w_gws As Workspace
Dim d_gdb As Database
Dim t_gPA As Recordset
```

```
Private Sub cm_actualiza_Click()
Dim s_lregion As String
Dim i_lCat As Integer
Dim i_lres As Integer
i_lres = MsgBox("Confirma
Actualización de region " +
msg_1.Text + " ?", vbYesNo)
If i_lres = vbYes Then
s_lregion = CStr(Region.Text)
For i_lCat = 0 To msg_1.Rows -
```

```
1
msg_1.Row = i_lCat
msg_1.Col = 0
t_gPA.Seek "=", s_lregion,
msg_1.Text
If t_gPA.NoMatch Then
gl_ttb.AddNew
gl_ttb.Fields("usuario") =
s_login
gl_ttb.Fields("basedatos") =
s_gRutaBD
gl_ttb.Fields("tabla") =
"reg_pref"
gl_ttb.Fields("fecha_hora") = Date +
Time
gl_ttb.Fields("Numtrans") = 1
gl_ttb.Update
t_gPA.AddNew
t_gPA.Fields("region") =
s_lregion
t_gPA.Fields("region_pref") =
msg_1.Text
msg_1.Col = 1
t_gPA.Fields("prefer") =
Val(msg_1.Text)
t_gPA.Update
Else
t_gPA.Edit
```

```
msg_1.Col = 1
t_gPA.Fields("prefer") =
Val(msg_1.Text)
t_gPA.Update
gl_ttb.AddNew
gl_ttb.Fields("usuario") =
s_login
```

```
gl_ttb.Fields("basedatos") =
s_gRutaBD
gl_ttb.Fields("tabla") =
"reg_pref"
```

```
gl_ttb.Fields("fecha_hora") = Date +
Time
```

```
gl_ttb.Fields("Numtrans") = 3
gl_ttb.Update
End If
Next i_lCat
```

```
w_gws.CommitTrans
msg_1.Clear
msg_1.Rows = 0
Frame1.Enabled = False
Frame1.Visible = False
bx_region.Enabled = True
It_reg.Enabled = True
cm_control.Enabled = True
Else
Exit Sub
End If
End Sub
```

```
Private Sub
cm_actualiza_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)
```

```
fm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
Gragar los cambios que ha
realizado"
End Sub
```

```
Private Sub cm_agrega_Click()
Dim i_li As Integer
If It_region.ListIndex <> -1 Then
For i_li = 0 To msg_1.Rows - 1
msg_1.Row = i_li
msg_1.Col = 0
If msg_1.Text =
It_region.List(i_lres).ListIndex)
Then
MsgBox "Region ya
existe", vbExclamation
Exit Sub
End If
Next i_li
msg_1.AddItem
It_region.List(i_lres).ListIndex)
msg_1.Row = msg_1.Rows - 1
msg_1.Col = 1
msg_1.Text = 0
End If
End Sub
```

```
Private Sub
cm_agrega_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)
```

```

frm_mentablas.sb_mentablas.Panels(
1).Text = "Al presionar este botón,
se agrega la región seleccionada,
como Región Factible de Traspaso"
End Sub

Private Sub cm_cancela_Click()
w_gws.Rollback
msg_1.Clear
msg_1.Rows = 0
Frame1.Enabled = False
Frame1.Visible = False
tx_region.Enabled = True
lt_reg.Enabled = True
cm_control.Enabled = True
End Sub

Private Sub
cm_cancela_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

frm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
cancelar los cambios y volver al
principio"
End Sub

Private Sub cm_control_Click()
Dim s_region As String
Dim d_ldb As Database
Dim t_ltb As Recordset
Dim s_iCriterio As String
If tx_region.Text = vbNullString Or
tx_region.Text = "" Then
MsgBox "Proporcionar Region
valida", vbExclamation
Exit Sub
End If
s_region = CStr(tx_region.Text)

' Carga de conjuntos y catalogos
para mostrar
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
s_iCriterio = "SELECT
region_pref.prefer FROM reg_pref
WHERE region = " + s_region + "
ORDER BY 2 "
Set t_ltb =
d_ldb.OpenRecordset(s_iCriterio)
If t_ltb.RecordCount > 0 Then
msg_1.Clear
msg_1.Rows = 0
t_ltb.MoveLast
t_ltb.MoveFirst
While Not t_ltb.EOF
msg_1.AddItem
t_ltb.Fields(0)
msg_1.Row = msg_1.Rows -
1
msg_1.Col = 1
msg_1.Text = t_ltb.Fields(1)
t_ltb.MoveNext
Wend
End If
t_ltb.Close
d_ldb.Close
w_gws.BeginTrans

Frame1.Enabled = True
Frame1.Visible = True
tx_region.Enabled = False
lt_reg.Enabled = False
cm_control.Enabled = False
End Sub

Private Sub
cm_control_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

frm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
Buscar los datos de la región
seleccionado"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

frm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
Regresar al Menú de Tablas"
End Sub

Private Sub Form_Load()
Dim d_ldb As Database
Dim t_ltb As Recordset
Dim i_li As Integer

' Carga de Regiones
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set t_ltb =
d_ldb.OpenRecordset("select region
from region order by region",
dbOpenSnapshot)
t_ltb.MoveLast
t_ltb.MoveFirst
If t_ltb.RecordCount > 0 Then
While Not t_ltb.EOF
it_region.AddItem
t_ltb.Fields("region")
lt_reg.AddItem
t_ltb.Fields("region")
t_ltb.MoveNext
Wend
End If
t_ltb.Close
d_ldb.Close

Inicializa espacio de trabajo
REGPREF para transac
Set w_gws =
DBEngine.CreateWorkspace("REGP
REF", "admin", "", dbUseJet)
Set d_gdb =
w_gws.OpenDatabase(s_gRutaBD,
True)
Set t_gPA =
d_gdb.OpenRecordset("reg_pref",
dbOpenTable)
t_gPA.Index = "PrimaryKey"
End Sub

Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

frm_mentablas.sb_mentablas.Panels(
1).Text = "Posicione el raton en un
botón para saber la acción que
realiza"
End Sub

Private Sub
Label1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

frm_mentablas.sb_mentablas.Panels(
1).Text = "Posicione el raton en un
botón para saber la acción que
realiza"
End Sub

Private Sub lt_reg_Click()
tx_region.Text =
lt_reg.List(it_reg.ListIndex)
End Sub

Private Sub
lt_region_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

frm_mentablas.sb_mentablas.Panels(
1).Text = "Seleccione una Región a
la que desee asignarle los
Almacenes que son Factibles de
Trasasarle"
End Sub

Private Sub
lt_region_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

frm_mentablas.sb_mentablas.Panels(
1).Text = "Lista total de Regiones
cargadas en el sistema. Seleccione
una para agregarla como Región
Factible para la que está Editando"
End Sub

Private Sub msg_1_Click()
If msg_1.Text = "0" Then
tx_pref.Text = ""
Else
tx_pref.Text = msg_1.Text
End Sub

```

```

End If
tx_pref.SetFocus
End Sub

Private Sub
msg_1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mentablas.sb_mentablas.Panels(1).Text = "Muestra las Regiones que están como factibles de traspaso a la que se edita. Seleccione una para Borrar"
End Sub

Private Sub Picture1_Click()
Dim s_lregion As String
Dim i_lres As Integer
If msg_1.Row >= 0 And msg_1.Col > 0 Then
msg_1.Col = 0
i_lres = MsgBox("Confirma eliminación de region " + msg_1.Text + " ?", vbYesNo)
If i_lres = vbYes Then
s_lregion = CStr(tx_region.Text)
t_gPA.Seek "=", s_lregion, msg_1.Text
If Not t_gPA.NoMatch Then
t_gPA.Delete
gl_kb.AddNew
gl_kb.Fields("usuario") = s_login
gl_kb.Fields("basedatos") = s_gRutaBD
gl_kb.Fields("tabla") = "reg_pref"
gl_kb.Fields("fecha_hora") = Date + Time
gl_kb.Fields("Numtrans") = 2
gl_kb.Update
End If
If msg_1.Rows = 1 Then
' Caso especial para borrar último catalogo
msg_1.Rows = 0
Else
msg_1.RemoveItem
msg_1.Row
End If
End If
End Sub

Private Sub
Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mentablas.sb_mentablas.Panels(1).Text = "Presione el Bote para descartar la Región escogida en La izquierda"
End Sub

Private Sub tx_pref_Change()
If msg_1.Rows > 0 Then

```

```

msg_1.Text = tx_pref.Text
End If
End Sub

Private Sub
tx_pref_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mentablas.sb_mentablas.Panels(1).Text = "Indica a la Región recién agregada o escogida, la preferencia en forma ascendente"
End Sub

'-----
'Archivo: fm_tp
'Fecha: Julio-Septiembre de 1999
'Proposito: Cálculo de consumo por construcción en telefonía pública
'-----

Option Explicit

'Proposito: Inicializa tabla de consumos construcción TP (constp) y calcula el consumo por region de cada elemento necesario para 'construir líneas de telefonos en Construcción TP.
'Se hace un producto cartesiano de tablas Inmat (materiales usados 'en líneas) con tabla planan que contiene el plan anual en meses 'de líneas a construir por región.
Dim gCalculo_tp(i_lAnio As Integer)
Dim s_icons As String
Dim d_kdb As Database 'Base de datos Access
Dim q_lQfd As QueryDef 'Invoca definición de consulta (QueryDef)
Set d_kdb = OpenDatabase(s_gRutaBD, True)

'Se limpia tabla constp
Set q_lQfd = d_kdb.QueryDefs("ini_cons_tp")
q_lQfd.Parameters(0) = i_lAnio
'Año a borrar
q_lQfd.Execute
q_lQfd.Close

'Se hace producto cartesiano con resultado de consumos tp en tabla constp
Set q_lQfd = d_kdb.QueryDefs("cal_cons_tp")
q_lQfd.Parameters(0) = i_lAnio
'Mismo año con calculos frescos
q_lQfd.Execute
q_lQfd.Close
d_kdb.Close
End Sub

Private Sub cm_inicia_Click()
Dim i_lAnio As Integer
Dim i_lres As Integer

```

```

If Not
gVerifica_anio_valido(cb_anio.Text)
Then ' = vbNullString Then
MsgBox "Especifique año valido"
Exit Sub
Else
i_lAnio = Val(cb_anio.Text)
End If
i_lres = MsgBox(" Desea Iniciar el Cálculo ? ", vbYesNo)
If i_lres = vbYes Then
gCalculo_tp(i_lAnio)
MsgBox "Cálculo terminado", vbExclamation
End If
End Sub

Private Sub
cm_inicia_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencalcons.sb_mencalcons.Panels(1).Text = "Al presionar este botón, se inicia el Calculo de Consumos de TP"
End Sub

Private Sub cm_salir_Click()
Unload Me
End Sub

Private Sub
cm_salir_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencalcons.sb_mencalcons.Panels(1).Text = "Regresa al Menú de Cálculo de Consumos"
End Sub

Private Sub Form_Load()
Dim i_anio As Integer
For i_anio = 1990 To 2100
cb_anio.AddItem i_anio
Next
End Sub

Private Sub
Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencalcons.sb_mencalcons.Panels(1).Text = "Posicione el cursor en algún botón o campo para ver acción que realiza"
End Sub

Private Sub
Label3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm_mencalcons.sb_mencalcons.Panels(1).Text = "Seleccione el año que desea se calculen las cantidades de Construcción de TP"

```

```

End Sub
'-----
'Archivo: fm_tp
'Fecha: Julio-Septiembre de 1999
'Propósito: Proporciona la facilidad
de agragar, habilitar o borrar
usuarios del sistema.
'-----

Option Explicit
Dim d_lws As Workspace
Dim d_ldb As Database
Dim tb_DR As Recordset
Dim tb_bit As Recordset
Dim b_lHayTransaccion As Boolean

Private Sub cb_usuario_Click()

fm_mentablas.sb_mentablas.Panels(
1).Text = "Seleccione el Usuario que
desea Consultar o Modificar"
Dim s_iStrId As String
Dim s_lBlancos As String
s_lBlancos = " " 'cadena con 7
blancos
Text1.Text = cb_usuario.Text
s_iStrId = " usuario = " +
Text1.Text + ""
tb_DR.FindFirst s_iStrId
If tb_DR.NoMatch Then
MsgBox "No hay usuarios
con ese identificador",
vbExclamation
Else
d_lws.BeginTrans
b_lHayTransaccion = True
tb_DR.Edit
Text1.Enabled = False
Frame1.Enabled = True
Frame1.Visible = True
cm_control(0).Enabled =
False
cm_control(1).Enabled =
True
cm_control(2).Enabled =
True
cm_control(3).Enabled =
True
cm_control(4).Enabled =
False
cb_usuario.Visible = False
cmd_cancela.Visible = False

'Carga a memoria
tx_numemp.Text =
tb_DR.Fields("numemp")
tx_nombre.Text =
tb_DR.Fields("nombre")
tx_depto.Text =
tb_DR.Fields("depto")
If tb_DR.Fields("activo") Then
ck_activo.Value = 1
Else
ck_activo.Value = 0
End If
If tb_DR.Fields("admin")

ck_admin.Value = 1
Else
ck_admin.Value = 0
End If

'End If
End Sub

Private Sub
ck_activo_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mentablas.sb_mentablas.Panels(
1).Text = "Indicador para desahilitar
la clave de usuario que se registra"
End Sub

Private Sub
ck_admin_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

fm_mentablas.sb_mentablas.Panels(
1).Text = "Bandera que indica si el
usuario que se está registrando
tendrá perfil de administrador"
End Sub

Private Sub cm_control_Click(Index
As Integer)
Dim i_lres As Integer
Dim s_lCata As String
Dim s_iStrId As String
Dim s_lCriterio As String
Dim s_lLinea As String
Dim s_lBlancos As String
Dim l_lb As Recordset
s_lBlancos = " " 'cadena con 7
blancos
Select Case Index
Case 0 'Alta
If Text1.Text = "" Or
IsNull(Text1.Text) Then
MsgBox "Asigne un
identificador de Usuario",
vbExclamation
Exit Sub
End If
Text1.Enabled = False
s_iStrId = " usuario = " +
Text1.Text + ""
tb_DR.FindFirst s_iStrId
If Not tb_DR.NoMatch Then
MsgBox "Atención: El
identificador de usuario ya existe use
otro", vbExclamation
Exit Sub
End If
Text1.Enabled = False
d_lws.BeginTrans
b_lHayTransaccion = True
tb_bit.AddNew
tb_bit.Fields("usuario") =
s_login
tb_bit.Fields("basedatos") =
s_gRutaBDAdmin
"usuarios"
tb_bit.Fields("fecha_hora") =
Date + Time
tb_bit.Update
MsgBox "Usuario
eliminado"
End If
d_lws.CommitTrans
b_lHayTransaccion = False
Text1.Text = ""
Text1.Enabled = True
Frame1.Enabled = False
Frame1.Visible = False
cm_control(0).Enabled =
True
cm_control(1).Enabled =
False
cm_control(2).Enabled =
False
cm_control(3).Enabled =
False
cm_control(4).Enabled =
True
Text1.Enabled = True
Case 2 'Actualiza
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
tb_DR.Fields("usuario") =
Text1.Text
If IsNull(tx_numemp.Text) Or
tx_numemp.Text = "0" Or
tx_numemp.Text = "" Then
tb_bit.Fields("Numtrans") = 1
tb_bit.Update
tb_DR.AddNew

'Carga a memoria
tx_numemp.Text = ""
tx_nombre.Text = ""
tx_depto.Text = ""
ck_activo.Value = 0
ck_admin.Value = 0
Frame1.Enabled = True
Frame1.Visible = True
cm_control(0).Enabled =
False
cm_control(1).Enabled =
True
cm_control(2).Enabled =
True
cm_control(3).Enabled =
True
cm_control(4).Enabled =
False
Case 1 'Baja
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
tb_DR.Delete
tb_bit.AddNew
tb_bit.Fields("usuario") =
s_login
tb_bit.Fields("basedatos")
= s_gRutaBDAdmin
tb_bit.Fields("tabla") =
"usuarios"
tb_bit.Fields("fecha_hora")
= Date + Time
tb_bit.Fields("Numtrans") =
2
tb_bit.Update
MsgBox "Usuario
eliminado"
End If
d_lws.CommitTrans
b_lHayTransaccion = False
Text1.Text = ""
Text1.Enabled = True
Frame1.Enabled = False
Frame1.Visible = False
cm_control(0).Enabled =
True
cm_control(1).Enabled =
False
cm_control(2).Enabled =
False
cm_control(3).Enabled =
False
cm_control(4).Enabled =
True
Text1.Enabled = True
Case 2 'Actualiza
i_lres = MsgBox("Confirma
eliminación ?", vbYesNo)
If i_lres = vbYes Then
tb_DR.Fields("usuario") =
Text1.Text
If IsNull(tx_numemp.Text) Or
tx_numemp.Text = "0" Or
tx_numemp.Text = "" Then

```

```

        MsgBox "Debe teclear el
numero de empleado",
vbExclamation
        Exit Sub
    End If
    If IsNull(tx_nombre.Text) Or
tx_nombre.Text = "" Then
        MsgBox "Debe Agear el
Nombre del Usuario", vbExclamation
        Exit Sub
    End If
    If IsNull(tx_depto.Text) Or
tx_depto.Text = "" Then
        MsgBox "Debe indicar el
Departamento al que pertenece el
empleado", vbExclamation
        Exit Sub
    End If
    'Carga a TABLA
    tb_DR.Fields("numemp") =
tx_numemp.Text
    tb_DR.Fields("nombre") =
tx_nombre.Text
    tb_DR.Fields("depto") =
tx_depto.Text
    If ck_activo.Value = 1 Then
        tb_DR.Fields("activo") =
True
    Else
        tb_DR.Fields("activo") =
False
    End If
    If ck_admin.Value = 1 Then
        tb_DR.Fields("admin") =
True
    Else
        tb_DR.Fields("admin") =
False
    End If
    If tb_DR.Fields("usuario") <>
"" Then
        i_res = MsgBox("Confirma
Modificación ?", vbYesNo)
        If i_res = vbYes Then
            tb_DR.Update
            tb_bit.AddNew
            tb_bit.Fields("usuario") =
s_login
            tb_bit.Fields("basedatos") =
s_gRutaBDAdmin
            tb_bit.Fields("tabla") =
"usuarios"
            tb_bit.Fields("Fecha_hora") = Date +
Time
            tb_bit.Fields("Numtrans") = 3
            tb_bit.Update
            MsgBox "Usuario
Modificado"
        End If
    Else
        MsgBox "Usuario es
invalido"
        Exit Sub
    End If
    Text1.Text = ""
    Text1.Enabled = True
    d_lws.CommitTrans

    b_lHayTransaccion = False
    Frame1.Enabled = False
    Frame1.Visible = False
    cm_control(0).Enabled =
True
    cm_control(1).Enabled =
False
    cm_control(2).Enabled =
False
    cm_control(3).Enabled =
False
    cm_control(4).Enabled =
True
    Text1.Enabled = True
    End If
    Case 3 'Cancela Cambios
    d_lws.Rollback
    b_lHayTransaccion = False
    Text1.Text = ""
    Text1.Enabled = True
    Frame1.Enabled = False
    Frame1.Visible = False
    cm_control(0).Enabled =
True
    cm_control(1).Enabled =
False
    cm_control(2).Enabled =
False
    cm_control(3).Enabled =
False
    cm_control(4).Enabled =
True
    Text1.Enabled = True
    Case 4 'Busca
    cb_usuario.Visible = True
    cmd_cancela.Visible = True
    cm_control(0).Enabled = False
    cm_control(1).Enabled = False
    cm_control(2).Enabled = False
    cm_control(3).Enabled = False
    cm_control(4).Enabled = False
    Text1.Enabled = False
    Set d_lwb =
d_lws.OpenDatabase(s_gRutaBDAdmin, True)
    Set l_ltb =
d_lwb.OpenRecordset("select
usuario from usuarios order by
usuario", dbOpenDynaset)
    l_ltb.MoveLast
    l_ltb.MoveFirst
    cb_usuario.Clear
    While Not l_ltb.EOF
        cb_usuario.AddItem
l_ltb("usuario")
        l_ltb.MoveNext
    Wend
    End Select
    End Sub

    Private Sub
cm_control_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)

    frm_mentablas.sb_mentablas.Panels(
1).Text = "Con este menú de
botones puede Crear, Borrar o
Actualizar el Usuario editado.
También Cancela o Busca"

    End Sub

    Private Sub
cmd_cancela_Click()
    cm_control(0).Enabled = True
    cm_control(1).Enabled = False
    cm_control(2).Enabled = False
    cm_control(3).Enabled = False
    cm_control(4).Enabled = True
    cb_usuario.Visible = False
    cmd_cancela.Visible = False
    Text1.Enabled = True
    End Sub

    Private Sub
cmd_cancela_MouseMove(Button
As Integer, Shift As Integer, X As
Single, Y As Single)

    frm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
Cancelar la Búsqueda de Usuarios y
Regresar al Principio de la Forma"
    End Sub

    Private Sub
Command1_Click()
    Unload Me
    End Sub

    Private Sub
Command1_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    frm_mentablas.sb_mentablas.Panels(
1).Text = "Presione este botón para
Regresar al Menú de Tablas"
    End Sub

    Private Sub
Form_Load()
    'Inicializa espacio de trabajo USU
para transac
    Dim l_ltb As Recordset
    Set d_lws =
DBEngine.CreateWorkspace("USU",
"admin", "", dbUseJet)
    Set d_lwb =
d_lws.OpenDatabase(s_gRutaBDAdmin, True)
    Set tb_DR =
d_lwb.OpenRecordset("usuarios",
dbOpenDynaset)
    Set tb_bit =
d_lwb.OpenRecordset("bitacora",
dbOpenDynaset)
    cb_usuario.Visible = False
    cmd_cancela.Visible = False
    b_lHayTransaccion = False
    End Sub

    Private Sub
Form_MouseMove(Button As
Integer, Shift As Integer, X As
Single, Y As Single)

    frm_mentablas.sb_mentablas.Panels(
1).Text = "Posicione el raton en un
botón para saber la acción que
realiza"
    End Sub

```

Private Sub Form\_Unload(Cancel As Integer)

```

    If b_HayTransaccion Then
        d_hws.Rollback
        Exit Sub
    Else
        End If
        tb_DR.Close
        tb_bit.Close
        d_idb.Close
        d_hws.Close
    End Sub

```

Private Sub Frame1\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm\_mentablas.sb\_mentablas.Panels(1).Text = "Posicione el raton en un botón para saber la acción que realiza"

Private Sub Text1\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm\_mentablas.sb\_mentablas.Panels(1).Text = "Proporcione la Calve del Usuario a dar de alta"

Private Sub tx\_depto\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm\_mentablas.sb\_mentablas.Panels(1).Text = "Campo para colocar el departamento al que pertenece el Empleado que se está registrando"

Private Sub tx\_nombre\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm\_mentablas.sb\_mentablas.Panels(1).Text = "Campo para colocar el nombre del usuario que se está registrando"

Private Sub tx\_numemp\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

fm\_mentablas.sb\_mentablas.Panels(1).Text = "Campo para colocar el número de Empleado que se está registrando"

-----  
 'Archivo: SQLs

'Fecha: Julio-Septiembre de 1999  
 'Proposito: Modulo con llamadas de SQL utilizadas en el sistema  
 'Estas llamadas son para MS Access

Option Explicit  
 Dim d\_gBD As Database

Sub gCrea\_queries(s\_lBaseDatos As Database)  
 Dim q\_lNuevo As QueryDef  
 Dim s\_SQL As String  
 With s\_lBaseDatos  
 s\_SQL = "INSERT INTO consumo ( cat, alm, anio, mes, consumo ) "

s\_SQL = s\_SQL + "SELECT sum\_usu2.cat, sum\_usu2.alm, sum\_usu2.anio, sum\_usu2.mes, if(isnull(sum\_dev2.suma),sum\_usu2.suma,sum\_usu2.suma - sum\_dev2.suma) "

s\_SQL = s\_SQL + "From sum\_usu2 LEFT JOIN sum\_dev2 " s\_SQL = s\_SQL + "ON (sum\_usu2.mes = sum\_dev2.mes) " s\_SQL = s\_SQL + "AND (sum\_usu2.anio = sum\_dev2.anio) "

s\_SQL = s\_SQL + "AND (sum\_usu2.cat = sum\_dev2.cat) " s\_SQL = s\_SQL + "AND (sum\_usu2.alm = sum\_dev2.alm);"

Set q\_lNuevo = .CreateQueryDef("cal\_consumo", s\_SQL)

s\_SQL = "PARAMETERS [anio\_calc] Long;" s\_SQL = s\_SQL + "INSERT INTO constp ( catalogo, region, mes, anio, consumo ) "

s\_SQL = s\_SQL + "SELECT Inmat.catalogo, planan.region, planan.mes, planan.anio, factor " cantidad "

s\_SQL = s\_SQL + "From Inmat, planan " s\_SQL = s\_SQL + "Where Inmat.areatm = planan.areatm " s\_SQL = s\_SQL + " and Inmat.areatm = '1' and planan.areatm = '1' and planan.anio = anio\_calc;"

Set q\_lNuevo = .CreateQueryDef("cal\_cons\_tp", s\_SQL)

s\_SQL = "SELECT DISTINCT llave\_extra, descripcion, region " s\_SQL = s\_SQL + "From cat\_tem, region, catalogo " s\_SQL = s\_SQL + "Where cat\_tem.catalogo = catalogo.catalogo "

s\_SQL = s\_SQL + "AND cat\_tem.tipo\_cat = '1' " s\_SQL = s\_SQL + "UNION select distinct llave\_extra, descripcion, region "

s\_SQL = s\_SQL + "From cat\_tem, region conjunto "

s\_SQL = s\_SQL + "Where cat\_tem.llave\_extra = conjunto.conj\_id " s\_SQL = s\_SQL + "AND cat\_tem.tipo\_cat = '2' " s\_SQL = s\_SQL + "ORDER BY llave\_extra, region; "

Set q\_lNuevo = .CreateQueryDef("cat\_tem\_reg", s\_SQL)

s\_SQL = "SELECT DISTINCT llave\_extra, region " s\_SQL = s\_SQL + "From cat\_tem, region " s\_SQL = s\_SQL + "ORDER BY llave\_extra, region; "

Set q\_lNuevo = .CreateQueryDef("cat\_tem\_x\_region", s\_SQL)

s\_SQL = "PARAMETERS [NombreRep] Text; " s\_SQL = s\_SQL + "SELECT con\_cat.conjunto, con\_cat.catalogo " s\_SQL = s\_SQL + "From con\_cat, conjunto, rep\_con, def\_rep "

s\_SQL = s\_SQL + "Where con\_cat.conjunto = conjunto.conj\_id "

s\_SQL = s\_SQL + "and conjunto.conj\_id = rep\_con.con\_cat\_id " s\_SQL = s\_SQL + "and rep\_con.tipo\_cat = '2' "

s\_SQL = s\_SQL + "and rep\_con.rep\_id = def\_rep.rep\_id " s\_SQL = s\_SQL + "and def\_rep.rep\_id = [NombreRep] "

s\_SQL = s\_SQL + "UNION select catalogo.catalogo, catalogo.catalogo "

s\_SQL = s\_SQL + "From catalogo, rep\_con, def\_rep " s\_SQL = s\_SQL + "Where catalogo.catalogo = rep\_con.con\_cat\_id "

s\_SQL = s\_SQL + "and rep\_con.tipo\_cat = '1' " s\_SQL = s\_SQL + "and rep\_con.rep\_id = def\_rep.rep\_id " s\_SQL = s\_SQL + "and def\_rep.rep\_id = [NombreRep]; "

Set q\_lNuevo = .CreateQueryDef("con\_x\_cat", s\_SQL)

s\_SQL = "SELECT k1, r1, tot\_ped2.ped\_prog, tot\_conss.sumconsu, tot\_exis.sumdisp, tot\_ped2.sumcant "

s\_SQL = s\_SQL + "From tot\_conss, tot\_exis, tot\_ped2 " s\_SQL = s\_SQL + "Where tot\_conss.k1 = tot\_exis.k2 " s\_SQL = s\_SQL + "and tot\_exis.k2 = tot\_ped2.k3 " s\_SQL = s\_SQL + "and tot\_conss.r1 = tot\_exis.r2 " s\_SQL = s\_SQL + "and tot\_exis.r2 = tot\_ped2.r3 "

```

s_ISQL = s_ISQL + "ORDER BY
k1, r1, ped_prog;"
Set q_INuevo =
.CreateQueryDef("concatena",
s_ISQL)
s_ISQL = "PARAMETERS
[FechaRep] DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO conspron ( catalogo,
almacen, consuprom, region )"
s_ISQL = s_ISQL + "SELECT
cons.catalogo, cons.almacen,
cons.consuprom, almacen.region "
s_ISQL = s_ISQL + "From cons,
almacen "
s_ISQL = s_ISQL + "Where
cons.fechaintro = [FechaRep]"
s_ISQL = s_ISQL + "and
cons.almacen =
almacen.cod_sirem;"
Set q_INuevo =
.CreateQueryDef("crea_conspron",
s_ISQL)
s_ISQL = "UPDATE almacen SET
alm_dist = 0;"
Set q_INuevo =
.CreateQueryDef("ini_alm_dist",
s_ISQL)
s_ISQL = "INSERT INTO almacen
( cod_saci, nombre, cod_sirem,
grupodist, region, subgrupo,
categoria, division, nuevadivis )"
s_ISQL = s_ISQL + "SELECT
almacen, nombre, cod_sirem,
grupodist, region, subgrupo,
categoria, division, nuevadivis "
s_ISQL = s_ISQL + "FROM
almacen;"
Set q_INuevo =
.CreateQueryDef("ini_almacen",
s_ISQL)
s_ISQL = "INSERT INTO cat_tem
( llave_extra, catalogo, tipo_cat )"
s_ISQL = s_ISQL + "SELECT
catalogo.catalogo,
catalogo.catalogo, tipo_cat "
s_ISQL = s_ISQL + "From
catalogo, rep_con, def_rep "
s_ISQL = s_ISQL + "Where
catalogo.catalogo =
rep_con.con_cat_id "
s_ISQL = s_ISQL + "and
rep_con.tipo_cat = '1'"
s_ISQL = s_ISQL + "and
rep_con.rep_id = def_rep.rep_id "
s_ISQL = s_ISQL + "and
def_rep.rep_id = [NombreRep];"
Set q_INuevo =
.CreateQueryDef("ini_cat_tem1",
s_ISQL)
s_ISQL = "PARAMETERS
[NombreRep] Text;"
s_ISQL = s_ISQL + "INSERT
INTO cat_tem ( llave_extra,
catalogo, tipo_cat )"
s_ISQL = s_ISQL + "SELECT
con_cat.conjunto, con_cat.catalogo,
tipo_cat "

```

```

s_ISQL = s_ISQL + "From
con_cat, conjunto, rep_con, def_rep"
s_ISQL = s_ISQL + "Where
con_cat.conjunto = conjunto.conj_id "
s_ISQL = s_ISQL + "and
conjunto.conj_id =
rep_con.con_cat_id "
s_ISQL = s_ISQL + "and
rep_con.tipo_cat = '2'"
s_ISQL = s_ISQL + "and
rep_con.rep_id = def_rep.rep_id "
s_ISQL = s_ISQL + "and
def_rep.rep_id = [NombreRep];"
Set q_INuevo =
.CreateQueryDef("ini_cat_tem2",
s_ISQL)
s_ISQL = "PARAMETERS
[anio_borrar] Long;"
s_ISQL = s_ISQL + "DELETE * *
s_ISQL = s_ISQL + "From constp "
s_ISQL = s_ISQL + "WHERE anio
= anio_borrar;"
Set q_INuevo =
.CreateQueryDef("ini_cons_tp",
s_ISQL)
s_ISQL = "INSERT INTO cons (
fechaintro, catalogo, almacen,
consuprom )"
s_ISQL = s_ISQL + "SELECT
[FechaEntry], catalogo, cod_alm,
consuprom "
s_ISQL = s_ISQL + "FROM
gen9707B;"
Set q_INuevo =
.CreateQueryDef("ini_cons",
s_ISQL)
s_ISQL = "PARAMETERS [Fecha]
DateTime, [Hoy] DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO tavan1 ( po_num, cod_sirem,
catalogo, sum_prog, sum_rec )"
s_ISQL = s_ISQL + "SELECT
po_num, cod_sirem, catalogo,
sum(qty_ordere), sum(qty_receiv) "
s_ISQL = s_ISQL + "From
ped_det "
s_ISQL = s_ISQL + "Where
request_da <= Hoy "
s_ISQL = s_ISQL + "AND
fechaintro = Fecha "
s_ISQL = s_ISQL + "GROUP BY
po_num, cod_sirem, catalogo;"
Set q_INuevo =
.CreateQueryDef("RAVAN1",
s_ISQL)
s_ISQL = "PARAMETERS [Fecha]
DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO tavan2 ( po_num, cod_sirem,
catalogo, sum_totprog )"
s_ISQL = s_ISQL + "SELECT
po_num, cod_sirem, catalogo,
sum(qty_ordere) "
s_ISQL = s_ISQL + "From
ped_det "
s_ISQL = s_ISQL + "Where
fechaintro = Fecha "

```

```

s_ISQL = s_ISQL + "GROUP BY
po_num, cod_sirem, catalogo;"
Set q_INuevo =
.CreateQueryDef("RAVAN2",
s_ISQL)
s_ISQL = "PARAMETERS [Fecha]
DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO Tavan3 ( catalogo, cod_sirem,
sum_disp )"
s_ISQL = s_ISQL + "SELECT
catalogo, cod_sirem,
sum(disponible) "
s_ISQL = s_ISQL + "From
existencias "
s_ISQL = s_ISQL + "Where
fechaintro = Fecha "
s_ISQL = s_ISQL + "GROUP BY
catalogo, cod_sirem;"
Set q_INuevo =
.CreateQueryDef("RAVAN3",
s_ISQL)
s_ISQL = "PARAMETERS [Fecha]
DateTime;"
s_ISQL = s_ISQL + "SELECT
tavan1.po_num, tavan1.cod_sirem,
tavan1.catalogo, catalogo.descripcion,
pedidos.vendor_num, sum_totprog,
if(sum_totprog > 0, sum_prog/sum_t
otprog*100.00, 0) AS Por1,
if(sum_totprog > 0, sum_rec/sum_tot
prog*100.00, 0) AS Por2, sum_disp "
s_ISQL = s_ISQL + "From
pedidos, catalogo, tavan1, TAVAN2,
TAVAN3 "
s_ISQL = s_ISQL + "Where
tavan1.po_num = pedidos.po_num "
s_ISQL = s_ISQL + "AND
tavan1.cod_sirem =
pedidos.cod_sirem "
s_ISQL = s_ISQL + "AND
pedidos.fechaintro = Fecha "
s_ISQL = s_ISQL + "AND
tavan1.po_num = tavan2.po_num "
s_ISQL = s_ISQL + "AND
tavan1.cod_sirem =
tavan2.cod_sirem "
s_ISQL = s_ISQL + "AND
tavan1.catalogo = tavan2.catalogo "
s_ISQL = s_ISQL + "AND
tavan1.catalogo = tavan3.catalogo "
s_ISQL = s_ISQL + "AND
tavan1.cod_sirem =
tavan3.cod_sirem "
s_ISQL = s_ISQL + "AND
tavan1.catalogo = catalogo.catalogo "
s_ISQL = s_ISQL + "AND
pedidos.po_date > #11/01/1996# "
s_ISQL = s_ISQL + "ORDER BY
tavan1.po_num, tavan1.cod_sirem,
tavan1.catalogo;"
Set q_INuevo =
.CreateQueryDef("RAVAN4",
s_ISQL)
s_ISQL = "UPDATE almacen SET
alm_dist = 2 "
s_ISQL = s_ISQL + "WHERE
categoria <> '01'"

```



```

s_ISQL = s_ISQL + "and region in
(SELECT region FROM regalm);"
Set q_INuevo =
.CreateQueryDef("solo_almacenPri",
s_ISQL)
s_ISQL = "PARAMETERS
mes_hoy Long, anio_hoy Long;"
s_ISQL = s_ISQL + "SELECT
cat_tem llave_extra AS k1,
conmtp.region AS r1, sum
(conmtp.consumoprom) AS sumcon
"
s_ISQL = s_ISQL + "From conmtmp,
cat_tem "
s_ISQL = s_ISQL + "Where
conmtp.catalogo = cat_tem.catalogo
"
s_ISQL = s_ISQL + "and
conmtp.mes = Mes_hoy "
s_ISQL = s_ISQL + "and
conmtp.anio = anio_hoy "
s_ISQL = s_ISQL + "GROUP BY
llave_extra, region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, region;"
Set q_INuevo =
.CreateQueryDef("Tot_conmtp",
s_ISQL)
s_ISQL = "SELECT llave_extra
AS k1, region AS r1, sum(sumcon)
AS sumconsu "
s_ISQL = s_ISQL + "From consp2,
cat_tem "
s_ISQL = s_ISQL + "Where
consp2.catalogo = cat_tem.catalogo
"
s_ISQL = s_ISQL + "GROUP BY
llave_extra, region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, region;"
Set q_INuevo =
.CreateQueryDef("tot_consp2",
s_ISQL)
s_ISQL = "PARAMETERS
[FechaRep] DateTime;"
s_ISQL = s_ISQL + "SELECT
cat_tem.llave_extra AS k1,
almacen.region AS r1,
sum(conss.consuprom) AS
sumconsu "
s_ISQL = s_ISQL + "From conss,
cat_tem, almacen "
s_ISQL = s_ISQL + "Where
conss.catalogo = cat_tem.catalogo "
s_ISQL = s_ISQL + "and
conss.almacen= almacen.cod_sirem
"
s_ISQL = s_ISQL + "and
conss.fechaintro = FechaRep "
s_ISQL = s_ISQL + "GROUP BY
llave_extra, almacen.region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, almacen.region;"
Set q_INuevo =
.CreateQueryDef("tot_conss",
s_ISQL)
s_ISQL = "INSERT INTO consp2 (
catalogo, region, sumcon ) "

```

```

s_ISQL = s_ISQL + "SELECT
catalogo, almacen.region,
sum(consspron.consuprom) "
s_ISQL = s_ISQL + "From
consspron, almacen "
s_ISQL = s_ISQL + "Where
consspron.almacen =
almacen.cod_sirem "
s_ISQL = s_ISQL + "GROUP BY
catalogo, almacen.region;"
Set q_INuevo =
.CreateQueryDef("tot_consspron",
s_ISQL)
s_ISQL = "PARAMETERS
mes_hoy Long, anio_hoy Long;"
s_ISQL = s_ISQL + "SELECT
cat_tem.llave_extra AS k1,
constp.region AS r1, sum
(constp.consumo) AS s1 "
s_ISQL = s_ISQL + "From constp,
cat_tem "
s_ISQL = s_ISQL + "Where
constp.catalogo = cat_tem.catalogo "
s_ISQL = s_ISQL + "and
constp.mes = Mes_hoy "
s_ISQL = s_ISQL + "and
constp.anio = anio_hoy "
s_ISQL = s_ISQL + "GROUP BY
llave_extra, region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, region;"
Set q_INuevo =
.CreateQueryDef("tot_constp",
s_ISQL)
s_ISQL = "PARAMETERS
(fecha_solic) DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO sum_dev ( cat, alm, anio, mes,
suma ) "
s_ISQL = s_ISQL + "SELECT cat,
alm, year(f_cap), month(f_cap),
sum(cant) "
s_ISQL = s_ISQL + "From
ment_de, adms "
s_ISQL = s_ISQL + "Where
F_CAP >= fecha_solic "
s_ISQL = s_ISQL + "and
ment_de.adm = adms.adm "
s_ISQL = s_ISQL + "GROUP BY
alm, cat, year(f_cap), month(f_cap) "
s_ISQL = s_ISQL + "ORDER BY
1, 2, 3, 4;"
Set q_INuevo =
.CreateQueryDef("tot_dev", s_ISQL)
s_ISQL = "PARAMETERS
[fecha_solic] DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO sum_dev ( cat, alm, anio, mes,
suma ) "
s_ISQL = s_ISQL + "SELECT cat,
alm, year(f_cap), month(f_cap),
sum(cant) "
s_ISQL = s_ISQL + "From
ment_de "
s_ISQL = s_ISQL + "Where
F_CAP >= fecha_solic "
s_ISQL = s_ISQL + "and
(ment_de.adm = " OR
isnull(ment_de.adm)) "

```

```

s_ISQL = s_ISQL + "and
ment_de.cta not in ('1137','1138') "
s_ISQL = s_ISQL + "GROUP BY
alm, cat, year(f_cap), month(f_cap) "
s_ISQL = s_ISQL + "ORDER BY
1, 2, 3, 4;"
Set q_INuevo =
.CreateQueryDef("tot_dev2",
s_ISQL)
s_ISQL = "PARAMETERS
[FechaRep] DateTime;"
s_ISQL = s_ISQL + "SELECT
cat_tem.llave_extra AS k2,
almacen.region AS r2,
sum(existencias.disponible) AS
sumdisp "
s_ISQL = s_ISQL + "From
existencias, cat_tem, almacen "
s_ISQL = s_ISQL + "Where
existencias.catalogo =
cat_tem.catalogo "
s_ISQL = s_ISQL + "and
almacen.cod_sirem =
existencias.cod_sirem "
s_ISQL = s_ISQL + "and
existencias.fechaintro = FechaRep "
s_ISQL = s_ISQL + "GROUP BY
llave_extra, almacen.region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, almacen.region;"
Set q_INuevo =
.CreateQueryDef("Tot_exis", s_ISQL)
s_ISQL = "PARAMETERS
[FechaRep] DateTime;"
s_ISQL = s_ISQL + "SELECT
cat_tem.llave_extra AS k2,
almacen.region AS r2,
sum(existencias.disponible) AS
sumdisp "
s_ISQL = s_ISQL + "From
existencias, cat_tem, almacen "
s_ISQL = s_ISQL + "Where
existencias.catalogo =
cat_tem.catalogo "
s_ISQL = s_ISQL + "and
almacen.cod_sirem =
existencias.cod_sirem "
s_ISQL = s_ISQL + "and
existencias.fechaintro = FechaRep "
s_ISQL = s_ISQL + "and
almacen.alm_dist <> 2 "
s_ISQL = s_ISQL + "GROUP BY
llave_extra, almacen.region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, almacen.region;"
Set q_INuevo =
.CreateQueryDef("tot_exisalm",
s_ISQL)
s_ISQL = "SELECT
cat_tem.llave_extra, almacen.region,
sum(cantidad) "
s_ISQL = s_ISQL + "From
ped_det, cat_tem, almacen "
s_ISQL = s_ISQL + "Where
ped_det.catalogo = cat_tem.catalogo
"
s_ISQL = s_ISQL + "and
almacen.cod_sirem =
ped_det.cod_sirem "

```

```

s_ISQL = s_ISQL + "GROUP BY
llave_extra, almacen.region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, almacen.region;"
Set q_INuevo =
.CreateQueryDef("tot_ped", s_ISQL)
s_ISQL = "PARAMETERS
[FechaRep] DateTime;"
s_ISQL = s_ISQL + "SELECT
cat_tem llave_extra AS k3,
almacen.region AS r3,
ped_det.ped_prog,
sum(ped_det.cantidad) AS sumcant
"
s_ISQL = s_ISQL + "From
ped_det, cat_tem, almacen "
s_ISQL = s_ISQL + "Where
ped_det.catalogo = cat_tem.catalogo
"
s_ISQL = s_ISQL + "and
almacen.cod_sirem =
ped_det.cod_sirem "
s_ISQL = s_ISQL + "and
ped_det.fechaintro = FechaRep "
s_ISQL = s_ISQL + "GROUP BY
cat_tem llave_extra, almacen.region,
ped_det.ped_prog "
s_ISQL = s_ISQL + "ORDER BY
cat_tem llave_extra, almacen.region,
ped_det.ped_prog;"
Set q_INuevo =
.CreateQueryDef("tot_ped2",
s_ISQL)
s_ISQL = "PARAMETERS
mes_hoy Long, anio_hoy Long;"
s_ISQL = s_ISQL + "SELECT
cat_tem.llave_extra AS k1,
pronosticos.region AS r1, sum
(pronosticos.pronos) AS sumpron "
s_ISQL = s_ISQL + "From
pronosticos, cat_tem "
s_ISQL = s_ISQL + "Where
pronosticos.catalogo =
cat_tem.catalogo "
s_ISQL = s_ISQL + "and
pronosticos.mes = Mes_hoy "
s_ISQL = s_ISQL + "and
pronosticos.anio = anio_hoy "
s_ISQL = s_ISQL + "GROUP BY
llave_extra, region "
s_ISQL = s_ISQL + "ORDER BY
llave_extra, region;"
Set q_INuevo =
.CreateQueryDef("tot_pron", s_ISQL)
s_ISQL = "PARAMETERS
[fecha_solis] DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO sum_usu ( cat, alm, anio, mes,
suma ) "
s_ISQL = s_ISQL + "SELECT cat,
alm, year(f_cap), month(f_cap),
sum(cant) "
s_ISQL = s_ISQL + "From
msal_us.adms "
s_ISQL = s_ISQL + "Where
F_CAP >= fecha_solis "
s_ISQL = s_ISQL + "and
msal_us.adm = adms adm "
s_ISQL = s_ISQL + "GROUP BY
alm, cat, year(f_cap), month(f_cap) "

```

```

s_ISQL = s_ISQL + "ORDER BY
1, 2, 3, 4;"
Set q_INuevo =
.CreateQueryDef("tot_usu", s_ISQL)
s_ISQL = "PARAMETERS
[fecha_solis] DateTime;"
s_ISQL = s_ISQL + "INSERT
INTO sum_usu ( cat, alm, anio, mes,
suma ) "
s_ISQL = s_ISQL + "SELECT cat,
alm, year(f_cap), month(f_cap),
sum(cant) "
s_ISQL = s_ISQL + "From
msal_us "
s_ISQL = s_ISQL + "Where
F_CAP >= fecha_solis "
s_ISQL = s_ISQL + "and
(msal_us.adm = " OR
isnull(msal_us.adm)) "
s_ISQL = s_ISQL + "GROUP BY
alm, cat, year(f_cap), month(f_cap) "
s_ISQL = s_ISQL + "ORDER BY
1, 2, 3, 4;"
Set q_INuevo =
.CreateQueryDef("tot_usu2",
s_ISQL)
s_ISQL = "INSERT INTO
sum_dev2 ( cat, alm, anio, mes,
suma ) "
s_ISQL = s_ISQL + "SELECT cat,
alm, anio, mes, sum(suma) "
s_ISQL = s_ISQL + "From
sum_dev "
s_ISQL = s_ISQL + "GROUP BY
cat, alm, anio, mes; "
Set q_INuevo =
.CreateQueryDef("totot_dev",
s_ISQL)
s_ISQL = "INSERT INTO
sum_usu2 ( cat, alm, anio, mes,
suma ) "
s_ISQL = s_ISQL + "SELECT cat,
alm, anio, mes, sum(suma) "
s_ISQL = s_ISQL + "From
sum_usu "
s_ISQL = s_ISQL + "GROUP BY
cat, alm, anio, mes; "
Set q_INuevo =
.CreateQueryDef("totot_usu",
s_ISQL)
s_ISQL = "PARAMETERS
[FechaRep] DateTime, [FechaSel]
DateTime;"
s_ISQL = s_ISQL + "UPDATE
ped_det SET ped_prog =
iif(month(request_da) +
year(request_da)*12-
(month(FechaRep)+year(FechaRep)
*12)<0, -1,month(request_da) +
year(request_da)*12 -(month(
FechaRep)+year(FechaRep)*12)) "
s_ISQL = s_ISQL + "WHERE
fechaintro = FechaSel;"
Set q_INuevo =
.CreateQueryDef("up_ped_del",
s_ISQL)
End With
End Sub

```

```

Public Sub gCrea_bd(s_Inombre As
String)
Dim tdfNuevo As TableDef
Dim idxNuevo As Index
Dim t_fTabla As Recordset
Set d_gBD =
.CreateDatabase(s_Inombre,
dbLangGeneral)
"" Aqui se crean las tablas del
sistema junto con sus indices
"" primarios y secundarios
With d_gBD
Set tdfNuevo =
.CreateTableDef("adms")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo
.Fields.Append
.CreateField("adm", dbText, 12)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("adm")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("almacen")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("cod_sirem", dbText, 3)
.Fields.Append
.CreateField("cod_saci", dbText, 3)
.Fields.Append
.CreateField("nombre", dbText, 30)
.Fields.Append
.CreateField("grupodist", dbText, 2)
.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("subgrupo", dbText, 15)
.Fields.Append
.CreateField("categoria", dbText, 2)
.Fields.Append
.CreateField("division", dbText, 15)
.Fields.Append
.CreateField("nuevavdivis", dbText,
15)
.Fields.Append
.CreateField("alm_dist", dbInteger, 4)
' Crea el objeto Index en la
tabla Departamentos

```

```

Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo
idxNuevo.Fields.Append
idxNuevo.CreateField("cod_sirem")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("cat_tem")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("llave_extra", dbText, 7)
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("tipo_cat", dbText, 1)
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("catalogo")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("unidad", dbText, 3)
.Fields.Append
.CreateField("descripcion", dbText,
30)
.Fields.Append
.CreateField("condicion", dbText, 3)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("con_cat")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("conjunto", dbText, 7)
.Fields.Append
.CreateField("catalogo", dbText, 7)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("conjunto")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("conf_arch")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("nombre", dbText, 12)
.Fields.Append
.CreateField("dir_ip", dbText, 30)
.Fields.Append
.CreateField("puerto", dbInteger, 30)
.Fields.Append
.CreateField("clave_acceso", dbText,
20)
.Fields.Append
.CreateField("pasaporte", dbText,
20)
.Fields.Append
.CreateField("ruta_remota", dbText,
100)
.Fields.Append
.CreateField("ruta_local", dbText,
100)
.Fields.Append
.CreateField("formato_dbase",
dbBoolean)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("nombre")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("conmtp")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("mes", dbLong)
.Fields.Append
.CreateField("anio", dbLong)
.Fields.Append
.CreateField("consumoprom",
dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("conjunto")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("conj_id", dbText, 7)
.Fields.Append
.CreateField("descripcion", dbText,
50)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("conj_id")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("conf_arch")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("conj_id")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("conj_id")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("conmtp")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("mes", dbLong)
.Fields.Append
.CreateField("anio", dbLong)
.Fields.Append
.CreateField("consumoprom",
dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")

```

```

' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("region")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("consc")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("techaintro", dbDate)
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("cod_sirem", dbText, 3)
.Fields.Append
.CreateField("mes_anio", dbText, 6)
.Fields.Append
.CreateField("consumo", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("techaintro")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("almacen")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("consspron")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("almacen", dbText, 3)
.Fields.Append
.CreateField("consumprom",
dbDouble)
.Fields.Append
.CreateField("region", dbText, 15)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")

```

```

.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("sumcon", dbDouble)
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("conss")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("techaintro", dbDate)
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("almacen", dbText, 3)
.Fields.Append
.CreateField("consumprom",
dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("techaintro")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("almacen")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("consspron")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("almacen", dbText, 3)
.Fields.Append
.CreateField("consumprom",
dbDouble)
.Fields.Append
.CreateField("region", dbText, 15)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")

```

```

idxNuevo.Fields.Append
idxNuevo.CreateField("almacen")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("constp")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("mes", dbLong)
.Fields.Append
.CreateField("anio", dbLong)
.Fields.Append
.CreateField("consumo", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("region")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("consumo")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("cat", dbText, 7)
.Fields.Append
.CreateField("aim", dbText, 3)
.Fields.Append
.CreateField("anio", dbLong)
.Fields.Append
.CreateField("mes", dbLong)
.Fields.Append
.CreateField("consumo", dbDouble)
' Crea el objeto Index en la
tabla Departamentos

```

```

Set idxNuevo =
.CreateIndex("CON_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("def_rep")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("rep_id", dbText, 10)
.Fields.Append
.CreateField("descripcion", dbText,
70)
.Fields.Append
.CreateField("num_meses",
dbInteger)
.Fields.Append
.CreateField("exis_actual",
dbBoolean)
.Fields.Append
.CreateField("exis_estim",
dbBoolean)
.Fields.Append
.CreateField("cons", dbBoolean)
.Fields.Append
.CreateField("constp", dbBoolean)
.Fields.Append
.CreateField("consc", dbBoolean)
.Fields.Append
.CreateField("consx", dbBoolean)
.Fields.Append
.CreateField("consy", dbBoolean)
.Fields.Append
.CreateField("form_calc", dbInteger)
.Fields.Append
.CreateField("stock", dbBoolean)
.Fields.Append
.CreateField("ped_ant", dbBoolean)
.Fields.Append
.CreateField("uso_pron", dbBoolean)
.Fields.Append
.CreateField("pedidos", dbBoolean)
.Fields.Append
.CreateField("solo_alm", dbBoolean)
.Fields.Append
.CreateField("pron", dbBoolean)
.Fields.Append
.CreateField("conmtp", dbBoolean)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("rep_id")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("existencias")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("fechaintr", dbDate)
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("cod_sirem", dbText, 3)
.Fields.Append
.CreateField("on_hand", dbDouble)
.Fields.Append
.CreateField("reservados",
dbDouble)
.Fields.Append
.CreateField("disponible", dbDouble)
.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("categoria", dbText, 2)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintr")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("cod_sirem")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("expusu")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("exped", dbText, 12)
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("Inmat")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("areatm", dbText, 1)
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("factor", dbDouble, 3)
' Crea el objeto Index en la
tabla Departamentos
Set idxNuevo =
.CreateIndex("Primarykey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("areatm")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("majuste")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("fechaintr", dbDate)
.Fields.Append
.CreateField("alm", dbText, 12)
.Fields.Append
.CreateField("cat", dbText, 12)
.Fields.Append
.CreateField("ser", dbText, 12)
.CreateField("t_cap", dbDate)
.Fields.Append
.CreateField("odo", dbText, 12)
.Fields.Append
.CreateField("uni", dbText, 12)
.Fields.Append
.CreateField("cant", dbDouble)
' Crea el objeto Index en la
tabla
Set idxNuevo =
.CreateIndex("AJU_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo

```

```

idxNuevo.Fields.Append
idxNuevo.CreateField("f_cap")
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
Indexes.Append idxNuevo
' Crea el objeto Index en la
tabla
Set idxNuevo =
.CreateIndex("AJUID_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintro")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("matusu")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("cal", dbText, 7)
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("ment_de")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("fechaintro", dbDate)
Fields.Append
.CreateField("un", dbText, 12)
Fields.Append
.CreateField("ar", dbText, 12)
Fields.Append
.CreateField("pob", dbText, 12)
Fields.Append
.CreateField("naf", dbText, 12)
Fields.Append
.CreateField("cta", dbText, 12)
Fields.Append
.CreateField("scta", dbText, 12)
Fields.Append
.CreateField("detscta", dbText, 12)
Fields.Append
.CreateField("a_r", dbText, 12)
Fields.Append
.CreateField("alm", dbText, 12)
Fields.Append
.CreateField("cat", dbText, 12)
Fields.Append
.CreateField("serie", dbText, 12)
.CreateField("depto", dbDate)
Fields.Append
.CreateField("mot_s", dbText, 12)
Fields.Append
.CreateField("f_cap", dbDate)
Fields.Append
.CreateField("f_u_c", dbDate)
Fields.Append
.CreateField("zona", dbText, 12)
Fields.Append
.CreateField("cdo", dbText, 12)
Fields.Append
.CreateField("uni", dbText, 12)
Fields.Append
.CreateField("doc", dbText, 12)
Fields.Append
.CreateField("adm", dbText, 12)
Fields.Append
.CreateField("remi", dbText, 12)
Fields.Append
.CreateField("exp", dbText, 12)
Fields.Append
.CreateField("cant", dbDouble)
Fields.Append
.CreateField("saldo", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("END_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("f_cap")
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
Indexes.Append idxNuevo
' Crea el objeto Index en la
tabla
Set idxNuevo =
.CreateIndex("ENDID_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintro")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("ment_oc")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("fechaintro", dbDate)
Fields.Append
.CreateField("ar", dbText, 12)
Fields.Append
.CreateField("pob", dbText, 12)
Fields.Append
.CreateField("naf", dbText, 12)
Fields.Append
.CreateField("cta", dbText, 12)
Fields.Append
.CreateField("scta", dbText, 12)
Fields.Append
.CreateField("detscta", dbText, 12)
Fields.Append
.CreateField("a_r", dbText, 12)
Fields.Append
.CreateField("alm", dbText, 12)
Fields.Append
.CreateField("cat", dbText, 12)
Fields.Append
.CreateField("serie", dbText, 12)
.CreateField("depto", dbDate)
Fields.Append
.CreateField("mot_s", dbText, 12)
Fields.Append
.CreateField("f_cap", dbDate)
Fields.Append
.CreateField("f_u_c", dbDate)
Fields.Append
.CreateField("zona", dbText, 12)
Fields.Append
.CreateField("cdo", dbText, 12)
Fields.Append
.CreateField("uni", dbText, 12)
Fields.Append
.CreateField("doc", dbText, 12)
Fields.Append
.CreateField("adm", dbText, 12)
Fields.Append
.CreateField("remi", dbText, 12)
Fields.Append
.CreateField("exp", dbText, 12)
Fields.Append
.CreateField("cant", dbDouble)
Fields.Append
.CreateField("saldo", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("ENO_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("f_cap")
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
Indexes.Append idxNuevo
' Crea el objeto Index en la
tabla
Set idxNuevo =
.CreateIndex("ENOID_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.

```

```

idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintro")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("ment_tr")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("fechaintro", dbDate)
.Fields.Append
.CreateField("un", dbText, 12)
.Fields.Append
.CreateField("ar", dbText, 12)
.Fields.Append
.CreateField("pob", dbText, 12)
.Fields.Append
.CreateField("naf", dbText, 12)
.Fields.Append
.CreateField("cta", dbText, 12)
.Fields.Append
.CreateField("scta", dbText, 12)
.Fields.Append
.CreateField("detscta", dbText, 12)
.Fields.Append
.CreateField("a_s", dbText, 12)
.Fields.Append
.CreateField("a_r", dbText, 12)
.Fields.Append
.CreateField("alm", dbText, 12)
.Fields.Append
.CreateField("cat", dbText, 12)
.Fields.Append
.CreateField("ser", dbText, 12)
.Fields.Append
.CreateField("f_cap", dbDate)
.Fields.Append
.CreateField("f_u_c", dbDate)
.Fields.Append
.CreateField("odo", dbText, 12)
.Fields.Append
.CreateField("uni", dbText, 12)
.Fields.Append
.CreateField("doc", dbText, 12)
.Fields.Append
.CreateField("cant", dbDouble)
.Fields.Append
.CreateField("saldo", dbDouble)
' Crea el objeto Index en ta
tabla Departamentos.
Set idxNuevo =
.CreateIndex("ENT_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("f_cap")
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
.Indexes.Append idxNuevo
' Crea el objeto Index en la
tabla
Set idxNuevo =
.CreateIndex("ENTID_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintro")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("movi")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("un", dbText, 12)
.Fields.Append
.CreateField("ar", dbText, 12)
.Fields.Append
.CreateField("po", dbText, 12)
.Fields.Append
.CreateField("naf", dbText, 12)
.Fields.Append
.CreateField("cta", dbText, 12)
.Fields.Append
.CreateField("scta", dbText, 12)
.Fields.Append
.CreateField("xor", dbText, 12)
.Fields.Append
.CreateField("a_s", dbText, 12)
.Fields.Append
.CreateField("a_r", dbText, 12)
.Fields.Append
.CreateField("alm", dbText, 12)
.Fields.Append
.CreateField("cat", dbText, 12)
.Fields.Append
.CreateField("ped", dbText, 12)
.Fields.Append
.CreateField("ser", dbText, 12)
.Fields.Append
.CreateField("depto", dbText, 12)
.Fields.Append
.CreateField("t_mov", dbText, 12)
.Fields.Append
.CreateField("mot_s", dbText, 12)
.Fields.Append
.CreateField("f_cap", dbDate)
.Fields.Append
.CreateField("f_u_c", dbDate)
.Fields.Append
.CreateField("zona", dbText, 12)
.Fields.Append
.CreateField("f_u_c", dbDate)
.Fields.Append
.CreateField("zona", dbText, 12)
.Fields.Append
.CreateField("doc", dbText, 12)
.Fields.Append
.CreateField("prov", dbText, 12)
.Fields.Append
.CreateField("uni", dbText, 12)
.Fields.Append
.CreateField("doc", dbText, 12)
.Fields.Append
.CreateField("adm", dbText, 12)
.Fields.Append
.CreateField("remi", dbText, 12)
.Fields.Append
.CreateField("exped", dbText, 12)
.Fields.Append
.CreateField("movto", dbText, 12)
.Fields.Append
.CreateField("c_mat", dbText, 12)
.Fields.Append
.CreateField("catego", dbText, 12)
.Fields.Append
.CreateField("cant", dbDouble)
.Fields.Append
.CreateField("saldo", dbDouble)
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("msal_tr")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo
.Fields.Append
.CreateField("fechaintro", dbDate)
.Fields.Append
.CreateField("un", dbText, 12)
.Fields.Append
.CreateField("ar", dbText, 12)
.Fields.Append
.CreateField("pob", dbText, 12)
.Fields.Append
.CreateField("naf", dbText, 12)
.Fields.Append
.CreateField("cta", dbText, 12)
.Fields.Append
.CreateField("scta", dbText, 12)
.Fields.Append
.CreateField("a_s", dbText, 12)
.Fields.Append
.CreateField("a_r", dbText, 12)
.Fields.Append
.CreateField("alm", dbText, 12)
.Fields.Append
.CreateField("cat", dbText, 12)
.Fields.Append
.CreateField("ser", dbText, 12)
.Fields.Append
.CreateField("f_cap", dbDate)
.Fields.Append
.CreateField("f_u_c", dbDate)
.Fields.Append
.CreateField("zona", dbText, 12)
.Fields.Append
.CreateField("cdo", dbText, 12)

```

```

.Fields.Append
CreateField("cant", dbDouble)
.Fields.Append
CreateField("saldo", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("SAT_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("f_cap")
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
' El índice en la tabla
principal debe ser
' Único para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
' Crea el objeto Index en la
tabla
Set idxNuevo =
.CreateIndex("SATID_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintro")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("msal_us")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("fechaintro", dbDate)
.Fields.Append
.CreateField("a_s", dbText, 12)
.Fields.Append
.CreateField("alm", dbText, 12)
.Fields.Append
.CreateField("cat", dbText, 12)
.Fields.Append
.CreateField("ser", dbText, 12)
.Fields.Append
.CreateField("depto", dbText, 12)
.Fields.Append
.CreateField("mot_s", dbText, 12)
.Fields.Append
.CreateField("f_cap", dbDate)
.Fields.Append
.CreateField("f_u_c", dbDate)
.Fields.Append
.CreateField("zona", dbText, 12)
.Fields.Append
.CreateField("cdo", dbText, 12)

```

```

.Fields.Append
CreateField("uni", dbText, 12)
.Fields.Append
.CreateField("doc", dbText, 12)
.Fields.Append
.CreateField("adm", dbText, 12)
.Fields.Append
.CreateField("exped", dbText, 12)
.Fields.Append
.CreateField("catego", dbText, 12)
.Fields.Append
.CreateField("saldo", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("SAU_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("f_cap")
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
' El índice en la tabla
principal debe ser
' Único para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
' Crea el objeto Index en la
tabla
Set idxNuevo =
.CreateIndex("SAUID_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintro")
' El índice en la tabla
principal debe ser
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("ped_det")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("fechaintro", dbDate)
.Fields.Append
.CreateField("po_num", dbText, 7)
.Fields.Append
.CreateField("cod_sirem", dbText, 3)
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("request_da", dbDate)
.Fields.Append
.CreateField("qty_ordere", dbDouble)

```

```

.Fields.Append
.CreateField("qty_to_rec", dbDouble)
.Fields.Append
.CreateField("qty_receiv", dbDouble)
.Fields.Append
.CreateField("cantidad", dbDouble)
.Fields.Append
.CreateField("ped_prog", dbInteger)
.Fields.Append
.CreateField("marca", dbText, 2)
.Fields.Append
.CreateField("region", dbText, 15)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PED_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("fechaintro")
idxNuevo.Fields.Append
idxNuevo.CreateField("po_num")
idxNuevo.Fields.Append
idxNuevo.CreateField("cod_sirem")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("request_da")
' El índice en la tabla
principal debe ser
' Único para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("pedidos")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("fechaintro", dbDate)
.Fields.Append
.CreateField("po_num", dbText, 7)
.Fields.Append
.CreateField("cod_sirem", dbText, 3)
.Fields.Append
.CreateField("po_date", dbDate)
.Fields.Append
.CreateField("pos_status", dbText,
10)
.Fields.Append
.CreateField("vendedor_num", dbText,
10)
.Fields.Append
.CreateField("sched_del", dbText, 1)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.

```



```

idxNuevo.Fields.Append
idxNuevo.CreateField("fechainitro")
idxNuevo.Fields.Append
idxNuevo.CreateField("po_num")
idxNuevo.Fields.Append
idxNuevo.CreateField("cod_sirem")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
CreateTableDef("planan")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("areatm", dbText, 1)
.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("mes", dbLong)
.Fields.Append
.CreateField("anio", dbLong)
.Fields.Append
.CreateField("cantidad", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("areatm")
idxNuevo.Fields.Append
idxNuevo.CreateField("region")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
CreateTableDef("pronosticos")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("catalogo", dbText, 7)
.Fields.Append
.CreateField("region", dbText, 15)
.Fields.Append
.CreateField("mes", dbLong)
.Fields.Append
.CreateField("anio", dbLong)

```

```

.Fields.Append
.CreateField("pronos", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("region")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("reg_pref")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("prefer", dbLong)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("region")
idxNuevo.Fields.Append
idxNuevo.CreateField("region_pref")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("regalm")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("region", dbText, 15)

```

```

' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("RAL_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("region")
' El índice en la tabla
principal no debe ser
' Unique.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("region")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("region", dbText, 15)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("region")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
.TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("rep_con")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
.Fields.Append
.CreateField("rep_id", dbText, 10)
.Fields.Append
.CreateField("con_cat_id", dbText, 7)
.Fields.Append
.CreateField("tipo_cat", dbText, 1)
' Crea el objeto Index en la
tabla Departamentos
Set idxNuevo =
.CreateIndex("con_cat_id")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("con_cat_id")
' El índice en la tabla
principal debe ser

```

```

' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("rep_id")
idxNuevo.Fields.Append
idxNuevo.CreateField("con_cat_id")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = True
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("sum_dev")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("cat", dbText, 7)
Fields.Append
.CreateField("alm", dbText, 3)
Fields.Append
.CreateField("anio", dbLong, 7)
Fields.Append
.CreateField("mes", dbLong, 7)
Fields.Append
.CreateField("suma", dbDouble)
' Crea el objeto Index en la
tabla Departamentos
Set idxNuevo =
.CreateIndex("SDEV_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("sum_usu")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("cat", dbText, 7)
Fields.Append
.CreateField("alm", dbText, 3)
Fields.Append
.CreateField("anio", dbLong)
Fields.Append
.CreateField("mes", dbLong)
Fields.Append
.CreateField("suma", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("SUSU_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("TAVAN1")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("po_num", dbText, 7)
Fields.Append
.CreateField("cod_sirem", dbText, 3)
Fields.Append
.CreateField("catalogo", dbText, 7)
Fields.Append
.CreateField("sum_prog", dbDouble)
Fields.Append
.CreateField("sum_rec", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("sum_usu2")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("cat", dbText, 7)
Fields.Append
.CreateField("alm", dbText, 3)
Fields.Append
.CreateField("anio", dbLong)
Fields.Append
.CreateField("mes", dbLong)
Fields.Append
.CreateField("suma", dbDouble)
' Crea el objeto Index en la
tabla Departamentos
Set idxNuevo =
.CreateIndex("SUSU2_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("sum_usu")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("cat", dbText, 7)
Fields.Append
.CreateField("alm", dbText, 3)
Fields.Append
.CreateField("anio", dbLong)
Fields.Append
.CreateField("mes", dbLong)
Fields.Append
.CreateField("suma", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("SUSU_IDX")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
idxNuevo.Fields.Append
idxNuevo.CreateField("cat")
idxNuevo.Fields.Append
idxNuevo.CreateField("alm")
idxNuevo.Fields.Append
idxNuevo.CreateField("anio")
idxNuevo.Fields.Append
idxNuevo.CreateField("mes")
' El indice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
idxNuevo.Unique = False
.Indexes.Append idxNuevo
End With
TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("TAVAN1")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
Fields.Append
.CreateField("po_num", dbText, 7)
Fields.Append
.CreateField("cod_sirem", dbText, 3)
Fields.Append
.CreateField("catalogo", dbText, 7)
Fields.Append
.CreateField("sum_prog", dbDouble)
Fields.Append
.CreateField("sum_rec", dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")

```

```

' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
    idxNuevo.Fields.Append
idxNuevo.CreateField("po_num")
idxNuevo.Fields.Append
idxNuevo.CreateField("cod_sirem")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
    idxNuevo.Unique = True
    .Indexes.Append idxNuevo
End With
    .TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("TAVAN2")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
    .Fields.Append
.CreateField("po_num", dbText, 7)
    .Fields.Append
.CreateField("cod_sirem", dbText, 3)
    .Fields.Append
.CreateField("catalogo", dbText, 7)
    .Fields.Append
.CreateField("sum_totprog",
dbDouble)
' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
    idxNuevo.Fields.Append
idxNuevo.CreateField("po_num")
idxNuevo.Fields.Append
idxNuevo.CreateField("cod_sirem")
idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
    idxNuevo.Unique = True
    .Indexes.Append idxNuevo
End With
    .TableDefs.Append tdfNuevo
Set tdfNuevo =
.CreateTableDef("TAVAN3")
With tdfNuevo
' Crea y agrega los objetos
Field a la
' colección Fields del objeto
TableDef nuevo.
    .Fields.Append
.CreateField("catalogo", dbText, 7)
    .Fields.Append
.CreateField("cod_sirem", dbText, 3)
    .Fields.Append
.CreateField("sum_disp", dbDouble)

```

```

' Crea el objeto Index en la
tabla Departamentos.
Set idxNuevo =
.CreateIndex("PrimaryKey")
' Crea y agrega el objeto
Field a la
' colección Fields del objeto
Index nuevo.
    idxNuevo.Fields.Append
idxNuevo.CreateField("catalogo")
idxNuevo.Fields.Append
idxNuevo.CreateField("cod_sirem")
' El índice en la tabla
principal debe ser
' Unique para formar parte de
un Relation.
    idxNuevo.Unique = True
    .Indexes.Append idxNuevo
End With
    .TableDefs.Append tdfNuevo
End With ' a base de datos
' Aqui se crean las consultas en
SQL del sistema
Call gCrea_queries(d_gBD)
' Ahora se cargan datos basicos
para operacion
Set t_ITabla =
d_gBD.OpenRecordset("region",
dbOpenTable)
With t_ITabla
    .Fields("region") = "CELAYA"
    .Update
    .AddNew
    .Fields("region") =
"CHIHUAHUA"
    .Update
    .AddNew
    .Fields("region") =
"GUADALAJARA"
    .Update
    .AddNew
    .Fields("region") =
"HERMOSILLO"
    .Update
    .AddNew
    .Fields("region") =
"MAZATLAN"
    .Update
    .AddNew
    .Fields("region") = "MERIDA"
    .Update
    .AddNew
    .Fields("region") = "METRO"
    .Update
    .AddNew
    .Fields("region") =
"MONTERREY"
    .Update
    .AddNew
    .Fields("region") = "PUEBLA"
    .Update
    .AddNew
    .Fields("region") =
"VILLAHERMOSA"
End With
t_ITabla.Close

```

```

Set t_ITabla =
d_gBD.OpenRecordset("conf_arch",
dbOpenTable)
With t_ITabla
    .AddNew
    .Fields("nombre") =
"existencias"
    .Update
    .AddNew
    .Fields("nombre") =
"movimientos"
    .Update
    .AddNew
    .Fields("nombre") = "pedidos"
    .Update
End With
t_ITabla.Close
d_gBD.Close
MsgBox "Creación de tablas,
indices y consultas terminada. Ahora
cargue datos necesarios",
vbInformation
End Sub

```

---

'Archivo: tras\_ped  
'Fecha: Julio-Septiembre de 1999  
'Propósito: Modulo que genera los  
reportes de pedidos y traspasos.  
'Se basa en código de generación de  
reportes dinámicos y de allí  
'genera los reportes deseados.

---

```

Option Explicit
Const Long_num = 7
Const Cte_Mto = 1 'Es
información mto en SIREM
Const Cte_TP = 2 'TP es
construccion telefonía pública
Const Cte_PE = 3 'PE es
construccion planta exterior
Const Cte_PRO = 4 'PRO es
pronosticos de almacenes
Const Cte_MTP = 5 'MTP es
mantenimiento de telefonía pública
Type tipo_lista
region As String
prefer As Integer
reg_pref As String
revisado As Boolean
End Type
Type tipo_traspaso 'Contiene los
traspasos obtenidos
region_actual As String
region_pref As String
catalogo As String
mes As Integer
cantidad As Double
End Type
Type tipo_pedido 'Contiene
pedidos obtenidos
region_actual As String
catalogo As String
mes As Integer
cantidad As Double
End Type
Type tipo_impmed

```

```

region_actual As String
PED(0 To 6) As Double
impreso As Boolean
End Type
Dim s_DefRep As String
Dim s_IFecExi As String
Dim s_IFecPed As String
Dim s_IFecConM As String
Dim s_IFecConCPE As String
Dim s_Icons As String 'Consultas
Dim d_ldb As Database
Dim t_IDR As Recordset
Definición de reporte
Dim t_IRC As Recordset
Catálogos y conjuntos de reporte
Dim t_IPref As Recordset 'Contiene
las preferencias de regiones para
traspasos
Dim s_IStird As String

'Queries para apertura de consultas
Dim qfd As QueryDef
Dim qfd_1 As QueryDef
Dim qfd_2 As QueryDef
Dim qfd_3 As QueryDef
Dim qfd_4 As QueryDef
Dim qfd_5 As QueryDef 'Para
abrir tot consumos TP
Dim qfd_6 As QueryDef 'Para
abrir tot consumos PE (pend)
Dim qfd_7 As QueryDef 'Para
abrir tot consumos pronosticos
Dim qfd_8 As QueryDef 'Para
abrir tot consumos mtto TP

'Recordsets para ver los resultados
de los queries(consultas)
Dim rst As Recordset
Dim rst_1 As Recordset 'Contiene
totalizado de existencias
Dim rst_2 As Recordset 'Contiene
totalizado de pedidos
Dim rst_3a As Recordset 'Contiene
totalizados de consumos sirem
Dim rst_3b As Recordset 'Contiene
totalizados de consumos const. PE
Dim rst_3c As Recordset 'Contiene
totalizados de consumos const. TP
Dim rst_3d As Recordset 'Contiene
totalizados de consumos en
pronosticos
Dim rst_3e As Recordset 'Contiene
totalizados de consumos mtto. TP
Dim rst_4 As Recordset 'Contiene
producto cartesiano de cat_tem x
region
Dim f_fecha As Date
Dim i As Integer
Dim s_ICatAnt As String
Dim s_ICatCon As String
Dim s_iregion As String

'Variables para cálculo de estimación
de existencias
Dim d_Existencia(1 To 32) As
Double 'existencias para
hasta 32 regiones
Dim d_Consumo(1 To 32, Cte_Mto
To Cte_MTP) As Double
1=mtto, 2=tp, 3=pe, 4=pro, 5=MTP

Dim d_IPedido(1 To 32, -1 To 6) As
Double
Dim d_IExis_est(1 To 32, Cte_Mto
To Cte_MTP, 0 To 6) As Double
i=mtto, 2=tp, 3=pe, 4=pro, 5 = MTP
Dim d_IEE As Double
Dim i_NumRegion As Integer
Dim s_IReg(1 To 32) As String

'Variables de configuración de
reporte
Dim t_ICod_Form As Long
'Numero de formula a utilizar en
existencias estimadas
Dim s_IDes As String
Dim t_Inummes As Long
'Numero de meses a calcular

'A continuación se listan variables
binarias que contienen configuración
'de reporte
Dim b_Iconcpe As Boolean
Dim b_Iconcp As Boolean
Dim b_Iconman As Boolean
Dim b_lexisact As Boolean
Dim b_lexisest As Boolean
Dim b_ipedant As Boolean
Dim b_istock As Boolean
Dim b_lusapro As Boolean
Dim b_ipedidos As Boolean
Dim b_ISoloAlmacenes As Boolean
Dim b_Ipron As Boolean
Dim b_Iconmtp As Boolean

'Variables para el formateo del
archivo generado
Dim s_IDescripcion As String
Dim s_IHeader0 As String
Dim s_IHeader As String
Dim s_IHeader2 As String
Dim s_IFormato As String
Dim i_gArchSalida As Integer
'Archivo de traspasos
Dim i_gArchPed As Integer
'Archivo de pedidos
Dim i_gTraPlano As Integer
'Archivo de traspasos plano
Dim i_gPedPlano As Integer
'Archivo de pedidos plano
Dim i_NumPagina As Integer
Dim i_NumLinea As Integer
Dim i_li As Integer
Dim s_IBuffer As String

'Variables auxiliares para fecha de
referencia usada
Dim i_Dia As Integer
Dim i_mes As Integer
Dim i_Anio As Integer
Dim gLista_factibles() As tipo_lista
Dim gMaxLista As Integer
Dim gPedidos(1 To 120) As
tipo_pedido
Dim gTraspasos(1 To 120) As
tipo_traspaso
Dim i_gIndPed As Integer
Dim i_gIndTra As Integer
Dim d_gFechaReferencia As Date

'Proposito: este procedimiento
actualiza conspron con pronosticos
Sub ActualizaPro(FechaRef As
Date)
Dim t_ICP As Recordset 'Tabla
Conspron
Dim t_IPR As Recordset 'Tabla
pronosticos
Dim d_ldb As Database
Dim s_ICat As String
Dim s_IReg As String
Dim i_mes As Integer
Dim i_Anio As Integer
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set t_ICP =
d_ldb.OpenRecordset("conspr2",
dbOpenTable)
Set t_IPR =
d_ldb.OpenRecordset("pronosticos",
dbOpenTable)
t_IPR.Index = "PrimaryKey"
t_ICP.MoveLast
t_ICP.MoveFirst
i_mes = Month(FechaRef)
i_Anio = Year(FechaRef)

'Para cada consumo promedio se
sustituye con pronostico si existe
While Not t_ICP.EOF
s_ICat =
t_ICP.Fields("catalogo")
s_IReg = t_ICP.Fields("region")
t_IPR.Seek "=", s_ICat, s_IReg,
i_mes, i_Anio
If Not t_IPR.NoMatch Then
t_ICP.Edit
t_ICP.Fields("sumcon") =
t_IPR.Fields("pronos")
t_ICP.Update
gl_ttb.AddNew
gl_ttb.Fields("usuario") =
s_login
gl_ttb.Fields("basedatos") =
s_gRutaBD
gl_ttb.Fields("tabla") =
"conspr2"
gl_ttb.Fields("fecha_hora") =
Date + Time
gl_ttb.Fields("Numtrans") = 3
gl_ttb.Update
End If
t_ICP.MoveNext
Wend
t_ICP.Close
t_IPR.Close
d_ldb.Close
End Sub

'Proposito: Se registra un traspaso
Sub gPon_traspaso(region_actual
As String, region_dest As String,
catalogo As String, mes As Integer,
cant_res As Double)
i_gIndTra = i_gIndTra + 1

gTraspasos(i_gIndTra) region_actual
= region_actual

```

```

gTrasposos(i_gIndTra).region_pref =
region_dest
gTrasposos(i_gIndTra).catalogo =
catalogo
gTrasposos(i_gIndTra).mes = mes
gTrasposos(i_gIndTra).cantidad =
cant_res
End Sub

'Proposito: Se registra un pedido
Sub gPon_pedido(region_actual As
String, catalogo As String, mes As
Integer, cantidad As Double)
i_gIndPed = i_gIndPed + 1

gPedidos(i_gIndPed).region_actual =
region_actual
gPedidos(i_gIndPed).catalogo =
catalogo
gPedidos(i_gIndPed).mes = mes
gPedidos(i_gIndPed).cantidad =
cantidad
End Sub

'Proposito: Imprime los trasposos
generados en forma de lista
Sub gImprime_trasposos()
Dim i As Integer
If i_gIndTra > 0 Then
For i = 1 To i_gIndTra
With gTrasposos(i)
Print #i_gArchSalida,
s_gCadena_fija(region_actual, 15) +
s_gCadena_fija(region_pref, 15) +
s_gNombreMes(mes +
Month(d_gFechaReferencia)) + " | "
+ s_gNumero_fijo(cantidad, 10)
End With
'Adición a archivo plano de
trasposos
With gTrasposos(i)
Print #i_gTraPlano,
s_gCadena_fija(catalogo, 8) +
s_gCadena_fija(region_actual, 15) +
s_gCadena_fija(region_pref, 15) +
s_gNombreMes(mes +
Month(d_gFechaReferencia)) + " | "
+ s_gNumero_fijo(cantidad, 10) +
s_gCadena_fija(Format(d_gFechaR
eferencia, "dd/mm/yyyy"), 8)
End With
Next i
Print #i_gArchSalida, ""
End If
End Sub

'Proposito: organiza la información
de pedidos por region
'y de acuerdo a calendario de meses
de hasta 6 meses
Sub gImprime_pedidos()
Dim gResu(1 To 32) As tipo_impded
'Matriz para impresion de pedidos
Dim i, j As Integer
Dim s_iCad As String
If i_gIndPed > 0 Then
For i = 1 To i_NumRegion
gResu(i).region_actual = ""
For j = 0 To 6
gResu(i).PED(j) = 0
Next j
gResu(i).impreso = False
Next i
For i = 1 To i_gIndPed
With gPedidos(i)
Print #i_gPedPlano,
s_gCadena_fija(catalogo, 8) +
s_gCadena_fija(region_actual, 15) +
s_gNombreMes(mes +
Month(d_gFechaReferencia)) + " | "
+ s_gNumero_fijo(cantidad, 10) +
s_gCadena_fija(Format(d_gFechaR
eferencia, "dd/mm/yyyy"), 8)
End With
Next i
For i = 1 To i_NumRegion
For j = 1 To i_gIndPed
If s_iReg(i) =
gPedidos(j).region_actual Then
With gPedidos(j)
gResu(i).region_actual =
region_actual
gResu(i).PED(mes) =
cantidad
gResu(i).impreso =
True
End With
End If
Next j
Next i
For i = 1 To i_NumRegion
If gResu(i).impreso Then
s_iCad =
s_gCadena_fija(gResu(i).region_act
ual, 15)
For j = 0 To 6
s_iCad = s_iCad +
s_gNumero_fijo(gResu(i).PED(j), 8)
Next j
Print #i_gArchPed, s_iCad
End If
Next i
End Sub

'Proposito: Regresa el indice que
corresponde a la region solicitada.
'En caso de no encontrarlo se indica
con -1
Function busca_ind_region(region
As String) As Integer
Dim i_li As Integer
For i_li = 1 To i_NumRegion
If s_iReg(i_li) = region Then
busca_ind_region = i_li
Exit Function
End If
Next i_li
busca_ind_region = -1
End Function

'Proposito: Primero localiza el o los
almacenes prefendos del almacen
'actual para hacer trasposos.
'Segundo: Al encontrar un almacen
se
'procede a determinar si sus
existencias actuales son suficientes
'para hacer un traspaso hacia el
almacen actual. En caso positivo
'se hace la disminucion pertinente y
se informa para llevar a cabo el
'traspaso. En caso negativo se
continua con la busqueda. Se
detiene
'hasta encontrar algun disponible o
no haber almacenes asociados.
Function
hay_disponible(region_actual As
String, catalogo As String, mes As
Integer, tipo_cons As Integer,
faltante As Double, ByRef region_res
As String, ByRef cant_res As
Double) As Boolean
Dim i_INR As Integer
Dim i_li As Integer
Dim d_iExisE As Double
For i_li = 1 To gMaxLista
If gLista_factibles(i_li).region =
region_actual And Not
gLista_factibles(i_li).revisado Then
gLista_factibles(i_li).reg_pref
gLista_factibles(i_li).revisado
= True
i_INR =
busca_ind_region(region_res)
d_iExisE =
d_iExis_est(i_INR, tipo_cons, mes)
If d_iExisE > 0 Then
'se busca cantidad a retirar
If d_iExisE >= faltante
Then
cant_res = faltante
Else
cant_res = d_iExisE
End If
'se quita a existencia
estimada
d_iExis_est(i_INR,
tipo_cons, mes) = d_iExis_est(i_INR,
tipo_cons, mes) - cant_res
region_res =
gLista_factibles(i_li).reg_pref
hay_disponible = True
Exit Function
End If
End If
If gLista_factibles(i_li).region >
region_actual Then
hay_disponible = False
Exit Function
End If
Next i_li
hay_disponible = False
End Function

'Proposito: Deshabilita revision de
almacenes factibles
Sub gLimpia_factibles()
Dim i_li As Integer
For i_li = 1 To gMaxLista
gLista_factibles(i_li).revisado =
False
Next i_li
End Sub

```

```
'Proposito Se carga arreglo con
almacenes y sus almacenes
preferidos
'para realización de traspasos
Sub gObten_factibles()
Dim i As Integer
Set t_IPref =
d_Idx.OpenRecordset("select region,
prefer_region_pref from reg_pref
order by 1,2")
If t_IPref.RecordCount > 0 Then
t_IPref.MoveLast
t_IPref.MoveFirst
gMaxLista =
t_IPref.RecordCount
ReDim
gLista_factibles(t_IPref.RecordCount
)
i = 1
While Not t_IPref.EOF
gLista_factibles(i).region =
t_IPref.Fields(0)
gLista_factibles(i).prefer =
t_IPref.Fields(1)
gLista_factibles(i).reg_pref =
t_IPref.Fields(2)
gLista_factibles(i).revisado =
False
i = i + 1
t_IPref.MoveNext
Wend
End If
t_IPref.Close
End Sub
```

```
'Proposito. Dado el faltante de la
region actual se busca satisfacerla
'con la realización de traspasos de
otras regiones y pedidos.
'Nota: Los traspasos se realizan con
regiones preferidas
'previamente definidas en una tabla.
Sub gEvalua_trasp_ped(ByRef
existencia As Double, reg_actual As
String, catalogo As String, mes As
Integer, tipo_cons As Integer)
Dim reg_res As String 'Contiene
region preferida resultante para
traspaso
Dim cant_res As Double 'Contiene
cantidad resultante a traspasar
Dim faltante As Double
If existencia < 0 Then
gLimpiar_factibles
faltante = -existencia
While faltante > 0
If hay_disponible(reg_actual,
catalogo, mes, tipo_cons, faltante,
reg_res, cant_res) Then
Call
gPon_traspaso(reg_actual, reg_res,
catalogo, mes, cant_res)
faltante = faltante -
cant_res
Else
Call
gPon_pedido(reg_actual, catalogo,
mes, faltante)
faltante = 0
End If
```

```
Wend
If faltante > 0 Then
Call
gPon_pedido(reg_actual, catalogo,
mes, faltante)
faltante = 0
End If
existencia = faltante
End If
End Sub

'propósito: Calculo de existencias
para el mes i_li de todas las
regiones
Sub gCalcula_exis(i_li As Integer,
i_MaxR As Integer)
Dim i_NumR As Integer 'Regiones
For i_NumR = 1 To i_MaxR
If i_li = 0 Then
d_IEE = d_Existencia(i_NumR)
If b_Iconman Then
d_IExis_est(i_NumR,
Cte_Mto, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_Mto))
End If
If b_Iconcpt Then
d_IExis_est(i_NumR,
Cte_TP, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_TP))
End If
If b_Iconcpe Then
d_IExis_est(i_NumR,
Cte_PE, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_PE))
End If
If b_Ipron Then
d_IExis_est(i_NumR,
Cte_PRO, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_PRO))
End If
If b_Iconmtp Then
d_IExis_est(i_NumR,
Cte_MTP, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_MTP))
End If
Else
If b_Iconman Then
d_IEE =
d_IExis_est(i_NumR, Cte_Mto, i_li -
1)
d_IExis_est(i_NumR,
Cte_Mto, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_Mto),
d_IEE)
End If
```

```
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_Mto))
End If
If b_Iconcpt Then
d_IEE =
d_IExis_est(i_NumR, Cte_TP, i_li -
1)
d_IExis_est(i_NumR,
Cte_TP, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_TP))
End If
If b_Iconcpe Then
d_IEE =
d_IExis_est(i_NumR, Cte_PE, i_li -
1)
d_IExis_est(i_NumR,
Cte_PE, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_PE))
End If
If b_Ipron Then
d_IEE =
d_IExis_est(i_NumR, Cte_PRO, i_li -
1)
d_IExis_est(i_NumR,
Cte_PRO, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_PRO))
End If
If b_Iconmtp Then
d_IEE =
d_IExis_est(i_NumR, Cte_MTP, i_li -
1)
d_IExis_est(i_NumR,
Cte_MTP, i_li) =
d_gCalcula_existencia(i_Cod_Form,
i_li, i_IDia, d_IEE,
d_IPedido(i_NumR, i_li),
d_IConsumo(i_NumR, Cte_MTP))
End If
End If
Next i_NumR
End Sub

Sub
gImprime_cab_traspos(catalogo
As String)
Print #i_gArchSalida, ">>
Teléfonos de México, S.A. de C.V.
"
Print #i_gArchSalida, "
Gerencia de Control de inventarios"
Print #i_gArchSalida, "
Reporte de Traspasos"
Print #i_gArchSalida, ""
Print #i_gArchSalida, "Catalogo "
+ catalogo
Print #i_gArchSalida, ""
Print #i_gArchSalida, " DESTINO
| ORIGEN | MES |
CANTIDAD "
Print #i_gArchSalida, ""
```

```

-----
-----
-----
End Sub

Sub
glImpime_cab_pedidos(catalogo As
String)
Dim s_iCad As String
Print #i_gArchPed, ">>
Teléfonos de México, S.A. de C.V.
"
Print #i_gArchPed, "
Gerencia de Control de Inventarios"
Print #i_gArchPed, "
Reporte de Pedidos"
Print #i_gArchPed, ""
Print #i_gArchPed, "Catalogo : " +
catalogo
Print #i_gArchPed, ""
s_iCad = "REGION | "
For i = 0 To 6
s_iCad = s_iCad +
s.gNombreMes(i +
Month(d_gFechaReferencia)) + "]"
Next i
Print #i_gArchPed, s_iCad
Print #i_gArchPed, "
-----
-----
-----
End Sub

'Propósito: Genera las existencias
estimadas y en base a eso genera
'un programa de pedidos y traspasos
de tal forma que no se tengan
faltantes.
Public Sub gGen_analisis(id_rep As
String, fecha_exis As String,
fecha_ped As String, fecha_conm As
String, fecha_refe As String,
arch_salida As String, arch_pedidos
As String, tra_pla As String, ped_pla
As String)
Dim i_iNumR As Integer
Dim s_iCatalogo As String
Fm_genrep.PB_1.Value = 0
DoEvents
If arch_salida = "" Or arch_salida
= vbNullString Then
MsgBox "Proporcione nombre
de reporte para traspasos",
vbExclamation
Exit Sub
End If
If arch_pedidos = "" Or
arch_pedidos = vbNullString Then
MsgBox "Proporcione nombre
de reporte para pedidos",
vbExclamation
Exit Sub
End If
If tra_pla = "" Or tra_pla =
vbNullString Then
MsgBox "Proporcione nombre
de archivo plano para traspasos",
vbExclamation
Exit Sub
End If
If ped_pla = "" Or ped_pla =
vbNullString Then
MsgBox "Proporcione nombre
de archivo plano para pedidos",
vbExclamation
Exit Sub
End If
Set d_ldb =
OpenDatabase(s_gRutaBD, True)
Set t_IDR =
d_ldb.OpenRecordset("def_rep",
dbOpenDynaset)
s_iStrId = " rep_id = " + id_rep + "
"
t_IDR.FindFirst s_iStrId
If t_IDR.NoMatch Then
MsgBox "Identificador de
reporte no reconocido",
vbExclamation
Exit Sub
End If
s_IDes =
t_IDR.Fields("descripcion")
i_Cod_Form =
t_IDR.Fields("form_calc")
i_inummes =
t_IDR.Fields("num_meses")
b_lconcp = t_IDR.Fields("consc")
b_lconctp = t_IDR.Fields("constp")
b_lconman =
t_IDR.Fields("conss")
b_laxisact =
t_IDR.Fields("exis_actua")
b_laxisest =
t_IDR.Fields("exis_estim")
b_lpedant =
t_IDR.Fields("ped_ant")
b_lstock = t_IDR.Fields("stock")
b_lusapro =
t_IDR.Fields("uso_pron")
b_lpedidos =
t_IDR.Fields("pedidos")
b_lSoloAlmacenes =
t_IDR.Fields("solo_alm")
b_lpron = t_IDR.Fields("pron")
b_lconmtp =
t_IDR.Fields("conmtp")
If Not
gVerifica_fecha_valida(fecha_refe)
Then ' = vbNullString Then
MsgBox "De una fecha de
referencia valida", vbExclamation
t_IDR.Close
d_ldb.Close
Exit Sub
Else
d_gFechaReferencia =
CDate(fecha_refe)
End If
If Not
gVerifica_fecha_valida(fecha_exis)
Then ' = vbNullString Then
MsgBox "Especifique una fecha
valida de existencias"
t_IDR.Close
d_ldb.Close
Exit Sub
End If
If Not
b_lconman And Not
b_lconctp And Not
b_lconcp And Not
b_lpron And Not
b_lconmtp
Then
MsgBox " Se necesita saber de
que consumos calcular"
t_IDR.Close
d_ldb.Close
Exit Sub
End If
i_Dia = Day(d_gFechaReferencia)
i_mes =
Month(d_gFechaReferencia)
i_Anio =
Year(d_gFechaReferencia)

'Se limpia tabla temporal
s_lcons = "DELETE FROM
cat_tem"
d_ldb.Execute s_lcons

'Se limpia consumos mto con
pronosticos
s_lcons = "DELETE FROM
conspron"
d_ldb.Execute s_lcons

'Se limpia tabla
s_lcons = "DELETE FROM
consp2"
d_ldb.Execute s_lcons

'Se cargan datos de catalogos
Set qfd =
d_ldb.QueryDefs("ini_cat_tem1")
qfd.Parameters(0) = id_rep
qfd.Execute
qfd.Close

'Se cargan datos de conjuntos
Set qfd =
d_ldb.QueryDefs("ini_cat_tem2")
qfd.Parameters(0) = id_rep

```

```

qfd.Execute
qfd.Close
Fm_genrep.PB_1.Value = 10
DoEvents

'Se crean totales de existencias
actuales
If b_IsoloAlmacenes Then
'No se consideran depositos
auxiliares de calculo
Set qfd =
d_ldb.QueryDefs("ini_alm_dist")
qfd.Execute
qfd.Close
Set qfd =
d_ldb.QueryDefs("solo_almacenpri")
qfd.Execute
qfd.Close
Set qfd_1 =
d_ldb.QueryDefs("tot_exisalm")
qfd_1.Parameters(0) =
fecha_exis
Else
Set qfd_1 =
d_ldb.QueryDefs("tot_exis")
qfd_1.Parameters(0) =
fecha_exis
End If
Set rst_1 =
qfd_1.OpenRecordset(dbOpenSnap
shot)
Fm_genrep.PB_1.Value = 30
DoEvents

'Se crean totales de pedidos
Set qfd =
d_ldb.QueryDefs("up_ped_def")
qfd.Parameters(0) =
d_gFechaReferencia Fecha
referencia
qfd.Parameters(1) = fecha_ped
'Fecha de entrada de pedidos a
seleccionar
qfd.Execute
qfd.Close
Fm_genrep.PB_1.Value = 40
DoEvents

'Se crean totales de consumos
If b_Iconman Then
If b_lusapro Then
'Se crean consumos con
pronosticos
Set qfd =
d_ldb.QueryDefs("crea_consspron")
qfd.Parameters(0) =
fecha_conm
qfd.Execute
qfd.Close

'Se totaliza
Set qfd =
d_ldb.QueryDefs("tot_consspron")
qfd.Execute
qfd.Close

'Se sustituyen pronosticos
existentes
IActualizaPro
(d_gFechaReferencia)

'Se crean totales de
consumos con pronosticos
Set qfd_3 =
d_ldb.QueryDefs("tot_consp2")

'Al abrirlo se ejecuta y vemos
resultados
Set rst_3a =
qfd_3.OpenRecordset(dbOpenSnap
shot)
Else
Set qfd_3 =
d_ldb.QueryDefs("tot_conss")
qfd_3.Parameters(0) =
fecha_conm

'Al abrirlo se ejecuta y vemos
resultados
Set rst_3a =
qfd_3.OpenRecordset(dbOpenSnap
shot)
End If
End If
If b_Iconctp Then 'Obtención de
consumos construcción TP
'Usar rst_3c
Set qfd_5 =
d_ldb.QueryDefs("tot_consp")
qfd_5.Parameters(0) = i_lmes
'Mes
qfd_5.Parameters(1) = i_lanio
'Año
Set rst_3c =
qfd_5.OpenRecordset(dbOpenSnap
shot)
End If
If b_Iconcpe Then
Usar rst_3b and qfd_6
End If
If b_lpron Then 'Obtención de
consumos en pronosticos
'Usar rst_3d
Set qfd_7 =
d_ldb.QueryDefs("tot_pron")
qfd_7.Parameters(0) = i_lmes
'Mes
qfd_7.Parameters(1) = i_lanio
'Año
Set rst_3d =
qfd_7.OpenRecordset(dbOpenSnap
shot)
End If
If b_Iconmtp Then 'Obtención de
consumos promedio de mto TP
'Usar rst_3e
Set qfd_8 =
d_ldb.QueryDefs("tot_conmtp")
qfd_8.Parameters(0) = i_lmes
'Mes
qfd_8.Parameters(1) = i_lanio
'Año
Set rst_3e =
qfd_8.OpenRecordset(dbOpenSnap
shot)
End If
Fm_genrep.PB_1.Value = 70
DoEvents

'Se traen las regiones factibles de
traspaso
Call gObten_factibles

'Se crea Producto cartesiano
cat_tem x region
Set qfd_4 =
d_ldb.QueryDefs("cat_tem_reg")
Set rst_4 =
qfd_4.OpenRecordset(dbOpenSnap
shot)
If rst_4.RecordCount > 0 Then
i_lNumPagina = 1
i_gArchSalida = FreeFile()
Open arch_salida For Output
As #i_gArchSalida
i_gArchPed = FreeFile()
Open arch_pedidos For Output
As #i_gArchPed
i_gTraPlano = FreeFile()
Open tra_pla For Output As
#i_gTraPlano
Print #i_gTraPlano,
"CATALOGO | DESTINO |
ORIGEN | MES | CANTIDAD |
FECHA_REF |"
i_gPedPlano = FreeFile()
Open ped_pla For Output As
#i_gPedPlano
Print #i_gPedPlano,
"CATALOGO | DESTINO |
MES | CANTIDAD | FECHA_REF
|"
rst_4.MoveLast
rst_4.MoveFirst

'Se organizan datos y se
genera reporte
Fm_genrep.PB_1.Value = 80
DoEvents
s_ICatAnt = rst_4.Fields(0)
rst_4.MoveFirst
While Not rst_4.EOF
s_ICatCon = rst_4.Fields(0)
s_IDescription =
rst_4.Fields(1)
s_lregion = rst_4.Fields(2)
i_lNumRegion = 1
While s_lCatAnt = s_lCatCon
And Not rst_4.EOF
s_lReg(i_lNumRegion) =
rst_4.Fields(2)
s_lCatalogo =
rst_4.Fields(0)

'Busca en existencias
s_lStrId = "k2 = " +
s_lCatCon + " and (2 = " + s_lregion
+ "-"
rst_1.FindFirst s_lStrId
If Not rst_1.NoMatch Then

d_Existencia(i_lNumRegion) =
rst_1.Fields("sumdisp")
Else

d_Existencia(i_lNumRegion) = 0
End If

```



```

'Busca en consumos
CONSTRUCCION PLANTA
EXTERIOR
'USAR b_Iconcpe
If b_Iconcpe Then
  s_IStIrd = " k1 = " +
s_ICatCon + " and r1 = " + s_lregion
+ ""
  rst_3b.FindFirst s_IStIrd
  If Not rst_3b.NoMatch
Then
  If
IsNull(rst_3b.Fields("sumcon"))
Then
d_IConsumo(i_NumRegion,
Cte_PE) = 0
Else
d_IConsumo(i_NumRegion,
Cte_PE) =
rst_3b.Fields("sumcon")
End If
Else
d_IConsumo(i_NumRegion,
Cte_PE) = 0
End If
End If
'Busca en consumos
CONSTRUCCION TELEFONIA
PUBLICA
'USAR b_Iconcnp
If b_Iconcnp Then
  s_IStIrd = " k1 = " +
s_ICatCon + " and r1 = " + s_lregion
+ ""
  rst_3c.FindFirst s_IStIrd
  If Not rst_3c.NoMatch
Then
  If
IsNull(rst_3c.Fields("s1")) Then
d_IConsumo(i_NumRegion,
Cte_TP) = 0
Else
d_IConsumo(i_NumRegion,
Cte_TP) = rst_3c.Fields("s1")
End If
Else
d_IConsumo(i_NumRegion,
Cte_TP) = 0
End If
End If
'Busca en consumos
mantenimiento TELEFONIA
PUBLICA
'USAR b_Iconmtp
If b_Iconmtp Then
  s_IStIrd = " k1 = " +
s_ICatCon + " and r1 = " + s_lregion
+ ""
  rst_3e.FindFirst s_IStIrd
  If Not rst_3e.NoMatch
Then
  If
IsNull(rst_3e.Fields("sumcon")) Then
d_IConsumo(i_NumRegion,
Cte_MTP) = 0
Else
d_IConsumo(i_NumRegion,
Cte_MTP) = rst_3e.Fields("sumcon")
End If
Else
d_IConsumo(i_NumRegion,
Cte_MTP) = 0
End If
End If
'Busca en consumos POR
MANTENIMIENTO
If b_Iconman Then
  s_IStIrd = " k1 = " +
s_ICatCon + " and r1 = " + s_lregion
+ ""
  rst_3a.FindFirst s_IStIrd
  If Not rst_3a.NoMatch
Then
  If
IsNull(rst_3a.Fields("sumcon"))
Then
d_IConsumo(i_NumRegion,
Cte_Mto) = 0
Else
d_IConsumo(i_NumRegion,
Cte_Mto) =
rst_3a.Fields("sumcon")
End If
Else
d_IConsumo(i_NumRegion,
Cte_Mto) = 0
End If
End If
'Busca en consumos POR
pronosticos
If b_Ipron Then
  s_IStIrd = " k1 = " +
s_ICatCon + " and r1 = " + s_lregion
+ ""
  rst_3d.FindFirst s_IStIrd
  If Not rst_3d.NoMatch
Then
  If
IsNull(rst_3d.Fields("sumpron"))
Then
d_IConsumo(i_NumRegion,
Cte_PRO) = 0
Else
d_IConsumo(i_NumRegion,
Cte_PRO) =
rst_3d.Fields("sumpron")
End If
Else
d_IConsumo(i_NumRegion,
Cte_PRO) = 0
End If
End If
If
End If
End If
'Busca en pedidos: Se
buscan hasta 6 meses adelante
For i_li = -1 To 6
d_IPedido(i_NumRegion, i_li) = 0
Next i_li
rst_4.MoveNext
If Not rst_4.EOF Then
  s_ICatCon =
rst_4.Fields(0)
  s_IDescripcion =
rst_4.Fields(1)
  s_lregion =
rst_4.Fields(2)
End If
i_NumRegion =
i_NumRegion + 1
DoEvents
i_NumRegion =
i_NumRegion - 1
'Análisis de faltantes y
sobrantes
glmprime_cab_traspasos
(s_ICatalogo)
glmprime_cab_pedidos
(s_ICatalogo)
If b_Iconman Then
  Print #i_gArchSalida, "
Tomando Consumos SIREM"
  Print #i_gArchPed, "
Tomando Consumos SIREM"
  For i_li = 0 To 6 'Calculo
de 0 a 6 meses
    Call gCalcula_exis(i_li,
i_NumRegion)
    For i_NumR = 1 To
i_NumRegion 'Calculo para region
      Call
gEvalua_trasp_ped(d_IExis_est(i_In
umR, Cte_Mto, i_li),
s_IReg(i_NumR), s_ICatalogo, i_li,
Cte_Mto)
      Next i_NumR
      glmprime_traspasos
i_gIndTra = 0
      Next i_li
      glmprime_pedidos
End If
      i_gIndPed = 0
      i_gIndTra = 0
      If b_Iconcnp Then
        Print #i_gArchSalida, "
Tomando Consumos const. TP"
        Print #i_gArchPed, "
Tomando Consumos const. TP"
        For i_li = 0 To 6
          Call gCalcula_exis(i_li,
i_NumRegion)
          For i_NumR = 1 To
i_NumRegion
            Call
gEvalua_trasp_ped(d_IExis_est(i_In
umR, Cte_TP, i_li),
s_IReg(i_NumR), s_ICatalogo, i_li,
Cte_TP)
          Next i_NumR
        Next i_li
      End If
    End If
  End If

```





## APÉNDICE E

### Glosario

#### A

**Abastecimiento:** Compra y administración de los bienes y servicios necesarios para la producción.

**Active:** Activo

#### B

**Barra de Menús:** Despliega los comandos que se usarán para construir alguna aplicación.

**Base de Datos:** Es un conjunto de información relacionada con un asunto o con una finalidad.

**Bit:** Unidad básica de información en un sistema de numeración binaria.

**Bucles:** Ciclos

#### C

**Checksum:** (Suma de control) Verificación de errores en la que se suma y transmite el número de bits de una unidad de información.

**Costo de adquisición:** Precio de costo correspondiente al costo del pedido.

**Costo de almacenamiento:** Gastos resultantes de guardar y poseer la mercancía.

**Cursor:** Símbolo móvil en una pantalla que sirve como punto de contacto entre el usuario y los datos.

#### D

**Diagrama de Contexto:** El primer nivel de un DFD (Diagrama de Flujo de Datos) es el diagrama de contexto, en donde a través de una sola burbuja se representa al sistema completo, con datos de entrada y de salida respectivamente.

**Diagrama de Entidad-Relación:** El modelo Entidad-Relación surge con la idea de separar el diseño

conceptual de datos del diseño físico, es decir, con el fin de plasmar en primer lugar mediante técnicas de diagramación en el mundo real, con sus entidades o agrupaciones de datos con significado dentro del sistema y las relaciones entre ellas.

**Diagrama de Flujo de Datos:** (DFD) es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

**Diagrama de Funcionalidad:** Son diagramas que representan en forma gráfica la secuencia en que se realizan las operaciones de un determinado procedimiento y/o el recorrido de las formas materiales.

**Diagrama de Nivel Cero:** Se representan los procesos principales de la aplicación que se está modelando, los cuales al llevarse a cabo el refinamiento sucesivo, se irán explotando en diagramas de nivel inferior.

**Diagrama de Nivel n:** Son el resultado del refinamiento de C/U de las burbujas en distintos niveles para mostrar un mayor detalle.

## F

**Flujo de Datos:** Representan la información de procesos, se muestran por medio de flechas curvas, la cabeza de la flecha indica la dirección del flujo de datos.

## G

**Gráfica de Gantt:** Diagrama que indica los trabajos asignados para un periodo dado, a departamentos, personas, máquinas, etc. Es una herramienta de planificación y de control de los trabajos por realizar.

**GUI:** Interface gráfica de usuario.

## H

**Host:** Anfitrión En redes de computadoras y telecomunicaciones, computadora que realiza funciones centralizadas, como poner al alcance de las demás computadoras los programas y los archivos de datos disponibles.

## I

**Interface:** Conexión lógica o física que permite la comunicación interactiva entre dos o más partes.

**Internet:** Sistema de redes de computadora ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como entrada remota, transferencia de archivos, correo electrónico y grupo de noticias.

**Inventario:** Conjunto de productos materiales (materia prima, mercancías, muebles, productos semiterminados y terminados) disponibles en una empresa.

## LL

**Llave primaria:** Consiste en uno o más campos que identifican a cada registro almacenado en la tabla.

## M

**Mailbox:** Buzón de correo electrónico. En correo electrónico, lugar de almacenamiento donde se guardan los mensajes dirigidos a un usuario hasta que éste entra al sistema. Un mensaje en pantalla le informa al usuario que le espera correo.

**Mainframe:** Macrocomputadora. Computadora multiusuario concebida para cubrir los requerimientos de computación de grandes empresas. Al principio, el término macrocomputadora se refería al gabinete metálico que guardaba la unidad central de procesamiento (CPU) de las primeras computadoras.

**Microsoft Access:** Es un programa para el manejo de bases de datos relacionales que permiten el almacenamiento, agrupamiento y búsqueda rápida de todo tipo de datos.

**Modelos de Bases de Datos:** Los modelos de datos son herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes.

**Modelo de Red:** Si se permite la relación entre "hermanos", es decir, entre registros de un mismo padre lógico, se tendrá una base de datos en red.

**Modelo Jerárquico:** En un modelo jerárquico los datos están estructurados en forma arborescente y las relaciones entre los diferentes tipos de registros se resuelven mediante punteros o enlaces entre ellos.

**Modelo Relacional:** El modelo relacional de datos es definido por medio de una serie de reglas cuyo objetivo es lograr la independencia de la representación lógica de los datos de su almacenamiento físico.

## N

**Normalización:** Es el proceso de agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones.

**Normalización Primera Forma:** Cuando no contiene grupos que se repiten.

**Normalización Segunda Forma:** Los campos clave dependen de la totalidad de la clave.

**Normalización Tercera Forma:** Los campos no clave son independientes entre sí, es decir, no hay un campo no clave que depende de otro no clave.

## O

**Octetos:** Conjunto de ocho bits

**Offset:**- Compensación. En procedimiento de texto, espacio adicional que se agregan a las hojas para la encuadernación. Es sinónimo de margen interior.

## P

**Password:** Palabra o código utilizado como un medio de seguridad contra el acceso no autorizado a los datos.

**Ping:** Es un protocolo que se usa en el entorno del Protocolo de control de transmisión (TCP) para probar si un nodo o dispositivo remoto se comunica en la red.

**Procesos:** Se representan por medio de círculos, o burbujas en el diagrama. Representan las diversas funciones individuales que el sistema lleva a cabo.

**Productividad:** Cantidad producida en función del trabajo proporcionado y los gastos enfrentados.

**Programación Visual:** Se entiende el uso de expresiones visuales (tales como gráficas, dibujos, iconos, barra de menú, etc.) en el proceso de la programación de aplicaciones.

**Protocolo:** Conjunto de reglas y procedimientos predefinidos

## R

**Ruteo:** Es el proceso de organizar y definir las rutas de para la transferencia de información entre diferentes equipos.

**Router:** Ruteador. De manera semejante a un puente, conecta únicamente las redes de área local con protocolos idénticos. Un ruteador transfiere los datos solamente cuando éstos están destinados a la otra LAN.

**Relaciones:** Constituyen el enlace entre las diferentes entidades del sistema.

**Relación Uno a Uno:** Una entidad en A esta relacionada únicamente con una entidad de B y una entidad de B con solo una de A.

**Relación Uno a Muchos:** Una entidad de A esta relacionada con cualquier número de entidades de B, pero una entidad de B solo esta relacionada con una de A.

**Relación Muchos a Uno:** Una entidad de A esta relacionada únicamente con una entidad de B, pero una entidad de B esta relacionada con cualquier número de entidades en A.

**Relación Muchos a Muchos:** Una entidad A esta asociada con cualquier número de entidades B, y una entidad de B esta vinculada con cualquier número de entidades de A.

**Run time:** Versión de ejecución. Versión limitada de un programa de apoyo que acompaña a otro programa que no puede correr sin la versión de ejecución. Las primeras versiones de Excel se vendían con las versiones de ejecución de Windows de Microsoft para los usuarios que no tenían Windows.

## T

**Tablas:** Es un conjunto de datos sobre un tema específico. Las tablas organizan los datos en columnas (denominados campos) y filas (denominadas registros).

**TCP/IP:** (Transmission Control Protocol ) protocolo de control de transmisiones/protocolo Internet. Conjunto de protocolos para intercomunicar sistemas diferentes. Se ejecuta en un gran número de computadoras Vax y basadas en Unix, es utilizado por muchos fabricantes de hardware, desde los de computadoras personales hasta las macrocomputadoras.



**ToolBar:** (Barra de Herramientas) Proporciona un acceso rápido a los comandos comúnmente usados en el ambiente de programación.

**ToolBox:** ( Caja de herramientas ) Provee un conjunto de herramientas que se utilizan en el modo de diseño de controles sobre una forma.

## U

**Unix:** Sistema Operativo multiusuario y multitarea de AT&T que se ejecuta en una amplia variedad de sistemas de computación de micro a macrocomputadoras. El Unix esta escrito en C (también desarrollado por AT&T) que es un lenguaje diseñado para programación a nivel de sistemas. Es la transportabilidad inherente al C lo que permite que Unix pueda ejecutarse con tal cantidad de computadoras diferentes.

**Usuario:** Cualquier individuo que interactua con una computadora a nivel de una aplicación.

## V

**Ventana de Código:** Es el lugar donde se escribe el código en Visual Basic para la aplicación.

**Ventana de Propiedades:** Lista el conjunto de propiedades para la forma o el control seleccionado.

**Ventana de Proyecto:** Lista las formas y módulos en el proyecto actual, es decir, el proyecto con el que sé esta trabajando.

**Visual Basic:** Herramienta (Software) de desarrollo de aplicaciones generadas mediante lo que se denomina programación visual.

## W

**Windows95 :** Versión de 32 bits para el entorno operativo para gráficos de Microsoft, proporciona un ambiente similar al de Macintosh, en el cual cada aplicación activa se visualiza en una pantalla movable y redimensionable sobre una pantalla principal.