



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
CAMPUS ARAGÓN

38
2y

"AUTOMATIZACION DEL DEPARTAMENTO CONTABLE-
ADMINISTRATIVO DE ISCAR DE MEXICO, S.A. DE
C.V."

TESIS

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A:

CLAUDIA IVONNE LUGO ZAMORA

ASESOR DE TESIS: Ing. Silvia Vega Muytoy

SAN JUAN DE ARAGON, MEXICO.

1996.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
CAMPUS ARAGÓN

UNIDAD ACADÉMICA

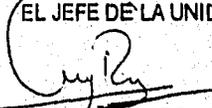
Ing. SILVIA VEGA MUYTOY
Jefe de la Carrera de Ingeniería
en Computación,
Presente.

En atención a la solicitud de fecha 20 de febrero del año en curso, por la que se comunica que la alumna CLAUDIA IVONNE LUGO ZAMORA, de la carrera de Ingeniero en Computación, ha concluido su trabajo de investigación intitulado "AUTOMATIZACIÓN DEL DEPARTAMENTO CONTABLE-ADMINISTRATIVO DE ISCAR DE MÉXICO, S.A. DE C.V.", y como el mismo ha sido revisado y aprobado por usted, se autoriza su impresión, así como la iniciación de los trámites correspondientes para la celebración del Examen Profesional.

Sin otro particular, le reitero las seguridades de mi distinguida consideración.

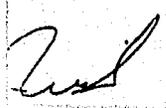
ATENTAMENTE
"POR MI RAZA HABLARA EL ESPÍRITU"
San Juan de Aragón, México, 21 de febrero de 1996.

EL JEFE DE LA UNIDAD


Lic. ALBERTO IBARRA ROSAS

c.c.p. Asesor de Tesis.
c.c.p. interesado.

AIR/vr



*Para alcanzar una meta, es necesario
participar activamente en ella.*

*La vida no es de aquellos que tienen problemas,
sino de aquellos que los enfrentan y resuelven.*

*Hay hombres que luchan un día y son buenos,
hay hombres que luchan un año y son mejores,
hay hombres que luchan muchos años y son muy buenos,
pero hay quienes luchan toda la vida,
y esos son los imprescindibles.*

*Las mejores cosas de la vida,
no son precisamente gratis. El precio que
tiene que pagar el hombre por ellas
es la aceptación de su responsabilidad,
de descubrir a cada momento lo que la vida
le exige como individuo.*

A mis padres:

*Por todo el amor y apoyo que de ellos he recibido .
Con infinito amor y respeto.*

A mi padre:

*Por respetar mis decisiones y aceptarme como soy,
aún cuando nuestras ideas son diferentes.
Gracias por creer en mí siempre.*

A mi madre:

*Porque decir madre es fácil.
No puedo pasar por alto todo lo que de ti aprendí.
mis ideas y anhelos se forjaron en tu persona.
Gracias por los años de comprensión y de cariño,
por los momentos de sacrificio y desvelo.
Porque de ti aprendí lo más valioso de la vida:
" el no dejarse vencer tan fácilmente ".
Porque este logro no es mío, es nuestro
Porque siempre me has impulsado a ser cada día mejor
y gracias a ello, he logrado una de mis más preciadas metas.*

A mi hermano:

*Por las horas de risas, juegos y cariño,
Con la firme esperanza de que este trabajo,
te pueda servir alguna vez,
como estímulo para mejorar tu vida.*

A mi abuelo:

*Porque a pesar de que ya no estás conmigo,
vivirás siempre en mi corazón.
Gracias por inculcar en mí, el deseo de salir adelante,
esforzándome cada día por ser mejor.*

A mi abuela:

*Por todos sus rezos y oraciones,
Por enseñarme a querer el mundo y la naturaleza,
Por todas sus enseñanzas.*

Mi agradecimiento especial a la :

Ing. Silvia Vega Maytoy.

Directora de Tesis

Por creer en mí, y brindarme ésta oportunidad.

Por depositar su confianza en mí.

Gracias por su apoyo incondicional y valiosos consejos.

A dios:

*Por haberme permitido llegar a la realización
de mi máximo anhelo en la vida.*

*A todas las personas que contribuyeron de alguna manera
para la realización de este trabajo.
Proporcionándome información ó la agilización de algún trámite.
Especialmente quiero agradecer a la Sra. Gloria Gallardo,
por su invaluable ayuda y por su apoyo incondicional.*

*A todas las personas que me brindan su amistad
y me consideran su amiga.
Especialmente a: Myrna Camales, Luis Manuel, Leticia,
Mercedes, Elisa, Alejandro, Arturo,
Porque sé que puedo contar con ustedes,
porque siempre han estado a mi lado
en los momentos más difíciles.
Les agradezco su amistad.*

*A todas las instituciones educativas,
por todos los conocimientos que de
ellas adquirí y que hoy me permiten
la realización de esta meta.*

*A mis maestros,
por su paciencia y dedicación.*

GRACIAS A TODOS.
Claudia Ivonne Lujo Zamora

**AUTOMATIZACION DEL DEPARTAMENTO
CONTABLE ADMINISTRATIVO DE
ISCAR DE MEXICO, S.A. DE C.V.**

INDICE

INTRODUCCION

CAPITULO I.LAS COMPUTADORAS Y LA ADMINISTRACION.

1.1 Componentes y funcionamiento de las computadoras	1
1.2 Conceptos de administración y organización de datos	
1.2.1 Conceptos de Administración de datos	17
1.2.2 Administración computarizada de datos	19
1.2.3 Alternativas de la organización de datos	27

CAPITULO II.BASES DE DATOS Y SISTEMAS DE INFORMACION.

2.1 Fundamentos de bases de datos	33
2.2 Ciclos de desarrollo de sistemas de Información	40
2.3 Fases del desarrollo de sistemas de información	47

Indice

CAPITULO III. ANALISIS DEL PROCESO DE INFORMACION DE ISCAR DE MEXICO, S.A. DE C.V.

3.1 Funciones principales de ISCAR DE MEXICO, S.A. DE C.V.	
3.1.1 Proceso de Facturación	59
3.1.2 Proceso de Inventarios	59
3.1.3 Proceso del Sistema Contable	60
3.2 Descripción de las funciones del sistema	60
3.3 Solución a la problemática del procesamiento de información	62
3.3.1 Requerimientos del sistema	63
3.3.2 Propuesta de solución	64

CAPITULO IV. DISEÑO E IMPLEMENTACION DEL SISTEMA

4.1 Justificación del software	68
4.2 Diseño del Sistema	
4.2.1 Explicación modular del Sistema	69
4.2.2 Diseño estructural de las bases de datos	74
4.3 Implantación del sistema	81

Indice

CAPITULO V.IMPLANTACION: PRUEBAS Y ANALISIS DE RESULTADOS.

5.1 Infraestructura de cómputo	85
5.2 Capacitación al usuario	86
5.3 Descripción general del proceso computarizado	86
5.4 Análisis de resultados	87
5.5 Mantenimiento del Sistema	89
5.6 Análisis costo-beneficio	89

CAPITULO VI.MANUAL DE OPERACION DEL SISTEMA	94
--	-----------

CONCLUSIONES	118
---------------------	------------

BIBLIOGRAFIA	120
---------------------	------------

APENDICE A	i
-------------------	----------

APENDIC B	ixii
------------------	-------------

INTRODUCCION.

Hace años, la capacidad de manejar la palabra impresa era el principal objetivo de la educación. En la actualidad, la capacitación en cualquier nivel, depende en gran medida del acercamiento de los individuos al lenguaje de las computadoras.

Esto implica la posibilidad de utilizar las computadoras como instrumento para la solución de problemas en muchos campos de nuestra vida cotidiana. Acercarnos al universo de la computación nos ayuda a desenvolvernos mejor en la sociedad en que vivimos, que sin lugar a dudas es una sociedad basada en la "información".

Hasta hace poco tiempo, casi todo el procesamiento de datos se realizaba en forma manual. El personal de oficina usaba papel, pluma y lápiz para mantener los registros. Este procesamiento de datos requería mano de obra intensiva. Los errores frecuentes causaban que las transacciones se registraran erróneamente, y que los registros de las compañías estuvieran mal representados. La información se recibía, a menudo, demasiado tarde para servir a cualquier propósito.

La era de las computadoras, se aplica sin lugar a dudas, en el procesamiento de datos en los negocios, así como en otras áreas de nuestra sociedad.

Introducción.

El uso de las computadoras tiene entre otras ventajas, en primer lugar, la rapidez para procesar instrucciones, es decir, para realizar cálculos y comparaciones lógicas. En segundo lugar, son exactas en sus procesos; rara vez comete un error electrónico del que no se dé cuenta por sí misma y lo señale al operador de la misma. Casi todos los errores en el procesamiento de datos por computadora están causados por programas con errores imputables a los humanos. En tercer lugar, son confiables. La combinación de éstos atributos, han aumentado notablemente la productividad del procesamiento de datos y en consecuencia, reduce su costo.

El objetivo principal del desarrollo del presente trabajo, es la automatización del departamento administrativo y contable de ISCAR DE MEXICO, S.A. DE C.V. En los capítulos que conforman el mismo, se pretenden reflejar los procedimientos que permitirán alcanzar el objetivo deseado.

El propósito de implantar procesos de automatización, es agilizar las transacciones diarias de la empresa, y darle confiabilidad a la información. Sin el uso de las computadoras, éstos procesos son muy lentos, y en algunos casos, la manipulación de la información genera errores.

Introducción.

En el capítulo I se habla del desarrollo que han tenido las computadoras a través del tiempo. Mencionando su historia y conceptos necesarios para su conocimiento.

El capítulo II nos introduce a los Sistemas de Información, explicando ¿ qué son ?, ¿ para qué nos sirven ? y ¿ qué los componen ?. Se habla acerca de conceptos tales como: bases de datos; así como la influencia de ellas en los procesos administrativos y la forma de administrarlas.

En el capítulo III se describen los procesos que se realizan en la empresa, se hace un análisis en torno a la problemática existente, y se genera la propuesta de solución.

El capítulo IV está dedicado al diseño e implementación del sistema, el cuál dará solución al problema en cuestión. Se realiza el diseño de las estructuras de las bases de datos, se delimita el problema, y se describe el funcionamiento de los programas que permiten realizar lo anterior.

En el capítulo V , se explica la implantación del sistema, las pruebas realizadas, y el análisis de resultados que permitirán la viabilidad del sistema.

El capítulo VI explica el funcionamiento del sistema. Se genera el manual de operación, como guía de manejo para lograr su óptimo funcionamiento.

En los apéndices se encontrarán los listados de programas que se utilizan en los procesos de automatización, así como los programas auxiliares.

CAPITULO I.

LAS COMPUTADORAS Y LA ADMINISTRACION.

CAPITULO I. Las Computadoras y la Administración.

Se incluye este capítulo como fuente de información sobre la terminología empleada en el entorno de las computadoras personales. En virtud de la facilidad de acceso y de la enorme cantidad de recursos, unidades periféricas, procesadores especializados, interfaces de entrada y salida, interfaces gráficas y sistemas que existen, se vuelve necesario hacer un resumen de todo este universo; por ello en este espacio se hace una breve descripción de la nomenclatura en uso, clasificada de acuerdo con la función que desempeña cada dispositivo utilizado en una microcomputadora.

CAPITULO I. Las Computadoras y la Administración.

Uno de los secretos mejor guardados durante la Segunda Guerra Mundial fue el del desarrollo de una máquina construida en la Universidad de Pensilvania, en los Estados Unidos. A pesar de que la máquina empleaba válvulas electrónicas (bulbos) y conexiones de alambre semejantes a las de cualquier aparato televisor, era aún muy complicada.

Se trataba de un calculador electrónico construido con 18 mil bulbos y cerca de medio millón de piezas diferentes, ocupaba una superficie de 200 metros cuadrados y era el dispositivo electrónico más grande y complicado jamás construido. Fueron J.P. Eckert, J. W. Mauchly y H. H. Goldstine quienes realizaron el proyecto de este calculador, el cual empezó a funcionar en febrero de 1946 con el nombre de ENIAC (Electronic Numerical Integrator and Calculator).

Esta primera máquina de cálculo electrónico fue construida para desarrollar esencialmente un trabajo único: cálculos relativos a las trayectorias balísticas. Posteriormente se usó también para otros trabajos científicos relacionados con la física nuclear. Para que resolviera un problema diferente, era necesario modificar manualmente la posición de los diversos interruptores y las conexiones de los cables eléctricos, empleando muchas personas por varios días.

Simultáneamente en otra Universidad de los Estados Unidos, Princeton, el científico de origen húngaro, John Von Neuman proyectó lo que hoy es universalmente reconocido como el prototipo de las computadoras modernas, la EDVAC (Electronic Discrete Variable Automatic Calculator), cuyo diseño estaba basado en el concepto de PROGRAMA ALMACENADO. Significa que la máquina podía almacenar, además de los datos, las instrucciones que regirían su propio funcionamiento.

La EDVAC, o máquina de Von Neuman fue construida para comenzar a funcionar en el año de 1952. Las ideas y los logros de Von Neuman inspiraron el desarrollo de todas las máquinas sucesivas. En Estados Unidos y Europa se proyectaron diferentes prototipos identificados por las siguientes siglas: EDSAC, UNIVAC, SEAC, etc.

Antes de describir la manera en que estas primeras computadoras han evolucionado hasta nuestros días, se describirá el proceso que dió origen a todo ello. Ahora se sabe que este proceso ha estado presente ya en las culturas más antiguas que se hayan investigado.

CAPITULO I. Las Computadoras y la Administración.

Se puede asegurar que en las primeras culturas que se desarrollaron sobre la tierra ya aparecen representaciones y sistemas numéricos, conceptos avanzados e instrumentos contables que al paso de los siglos, vienen a constituir la base de las matemáticas y de las máquinas mas avanzadas de nuestro tiempo. En cuanto el primer homo sapiens tuvo la necesidad de contar y agrupar elementos que constitúan su ambiente, surgieron entonces las nociones mas elementales de la aritmética.

Caldeos, babilonios y egipcios utilizaron cotidianamente números, cuentas y representaciones aritméticas. Los mayas, los mas avanzados científicos del pasado americano, descubrieron y utilizaron el concepto del cero, con lo que pudieron construir un poderoso sistema numérico que les permitió abordar problemas matemáticos y astronómicos.

Los Arabes introdujeron el sistema decimal, dieron luz al álgebra, de donde se pudo desarrollar la trigonometría, el calculo diferencial e integral, y otras muchas materias que forman las principales herramientas de los científicos modernos.

Muchas de estas nociones que se remontan a los tiempos más lejanos de la historia de la humanidad, han servido a los hombres de todas las épocas posteriores para enfrentar los problemas de sus días. De forma natural fueron surgiendo las máquinas que podían auxiliario a poner en práctica sus ideas. Así surgió primeramente el ábaco árabe y tablas que contenían los resultados de las operaciones que cada sociedad usaba mas comúnmente.

En 1642, el matemático y filósofo francés Blaise Pascal, de tan solo 19 años, inventó una máquina capaz de llevar la cuenta automática de sumas y restas elementales. En su máquina existían ruedas dentadas que representaban las unidades, decenas, centenas, etc. Sobre cada rueda estaban los dígitos del cero al nueve, lo que determinaba diez posiciones para cada rueda. La rotación completa de una rueda producían el avance de la rueda que queda a la izquierda en una sola posición. Por primera vez, una máquina efectúo automáticamente el acarreo, hasta entonces ejecutado solamente en la mente humana. Esta idea se exploto durante 300 años para construir muchos de los siguientes instrumentos de cálculo.

En 1671, el alemán G. W. Leibnitz proyecto una máquina con la que intento realizar multiplicaciones y divisiones mediante sumas y restas sucesivas, usando el principio de acarreo automático de Pascal. Desafortunadamente, la técnica de esa época no estaba en posibilidad de fabricar las piezas mecánicas con la facilidad y calidad indispensables para producir una serie de máquinas utilizables de manera práctica.

CAPITULO I. Las Computadoras y la Administración.

Solo después de la Revolución Industrial, con el perfeccionamiento de la técnicas de maquinado, fue posible realizar diversos dispositivos que resultaran prácticos y baratos. Al mismo tiempo aumento el interés hacia las máquinas calculadoras que ayudaban a economizar tiempo y dinero.

En 1804, el francés Joseph-Marie Jacquard puso en práctica la idea de automatizar ciertas fases del funcionamiento de las máquinas del tejido textil que un siglo antes había insinuado el mecánico Falcón: las telas mostrarían un dibujo correspondiente a un patrón registrado en una tarjeta perforada que la máquina podía identificar.

En 1822, el matemático inglés Charles Babbage ideó su máquina diferencial que podía realizar automáticamente operaciones aritméticas en secuencias diferentes. El diseño general de la máquina no fue valorado en su real magnitud, pues contaba con todos los elementos que la asemejan a un procesador moderno.

En 1887, no estaban interpretados todavía los datos del censo americano de 1880, elaborado manualmente por centenares de empleados. En previsión del siguiente censo, el experto en estadística Herman Hollerith inventó un sistema para representar el nombre, la edad, el sexo, la dirección y otros datos esenciales de cada ciudadano norteamericano, bajo la forma de agujeros hechos en una tarjeta de cartón que podía ser contada (procesada) por una máquina. con esta idea el Gobierno de los Estados Unidos proceso la información del censo de 1890 en solo dos años y medio. Motivado por el éxito del censo norteamericano y usando máquinas de Hollerith, se efectuó el primer censo ruso en 1896.

Para leer la información registrada en estas tarjetas, cada tarjeta era puesta en un mecanismo que contenía muchos contactos electrónicos en forma de aguja que se detenían al chocar con la tarjeta o pasaban por los agujeros de la tarjeta para cerrar un circuito eléctrico. El que cada aguja cerrara o no un circuito eléctrico tenía un significado relacionado con los datos tomados del censo. Así, todos los datos de una tarjeta podían ser contabilizados simultáneamente, y la tarjeta quedaba clasificada en bloques que se agrupaban atendiendo a edad, sexo, lugar de nacimiento o cualquier variable del individuo.

A principios de siglo Herman Hollerith se dedicó a promover el uso de su máquina de tarjeta perforada que permitía contabilizar y organizar rápida y económicamente grandes cantidades de datos. De 1890 a 1940, estas máquinas fueron desarrolladas y dotadas de otras funciones que las hicieron mas veloces.

CAPITULO I. Las Computadoras y la Administración.

Las máquinas de registro unitario perfeccionadas, o de tarjeta perforada reciben y reproducen datos, los clasifican, los subdividen, los suman o los restan, los multiplican o los dividen. Compara datos, ejecutan búsquedas, producen reportes impresos e incluso perforan en tarjetas los resultados de sus operaciones. Los datos son aceptados y emitidos por la máquina no solamente bajo la forma de números, sino también de letras; permitiendo así al personal usuario una comprensión mas fácil de los resultados y de la información.

Al mismo tiempo que se desarrollaron las máquinas contables impactando al mundo de los negocios y de la administración, continuaron los estudios de científicos e ingenieros para construir máquinas que calcularan a muy alta velocidad para resolver problemas mas complejos: previsión del clima, cálculo de órbitas de satélites y planetas, simulación de fenómenos atómicos, etc.

Así, en 1936 el alemán Konrad Zuse de 26 años construyó un calculador electromecánico, el Z1. Con el apoyo del gobierno alemán durante la Segunda Guerra Mundial construyó otros modelos mas avanzados. Estas máquinas fueron construidas introduciendo un principio fundamental de las computadoras modernas: la representación binaria de los números. Los calculadores Zuse fueron de las primeras máquinas que dejaron de ser cien por ciento mecánicas. En su construcción se incorporaron miles de relevadores o dispositivos electromecánicos que realizaban la tarea que hoy ejecutan los transistores ultraminiaturizados.

En 1944 el profesor Howard H. Aiken, de la Universidad de Harvard en los Estados Unidos, desarrolló, después de siete años de estudio un calculador automático, en el que se materializaron parte de las intuiciones de Babbage y la idea de tarjeta perforada de Jacquard. La máquina fue llamada MARK I y su funcionamiento se regía por instrucciones que se leían de una cinta de papel perforada. Sin intervención humana la Mak I daba los resultados de los cálculos perforándolos en una tarjeta o imprimiéndolos por medio de máquinas de escribir adaptadas especialmente. La Mark era en realidad un conjunto de muchas calculadoras conectadas entre sí. Contenía miles de relevadores que accionaban dispositivos mecánicos como: ruedas, contadores, levas, etc., y era capaz de sumar dos números de 23 cifras en 3 décimas de segundo, y multiplicar esos dos números entre sí en aproximadamente 6 segundos. La Mark I completo un importante capítulo en la historia del cálculo aritmético y termino el proceso iniciado por Pascal tres siglos antes.

CAPITULO I. Las Computadoras y la Administración.

La ENIAC era una máquina totalmente nueva en la cual se eliminaron todas las partes mecánicas en movimiento que representaban los números sustituyéndose por bulbos activados mediante impulsos electrónicos. Dado que los impulsos electrónicos en un bulbo se mueven miles de veces más rápido que en un electroimán o relevador, la ENIAC podía efectuar multiplicaciones miles de veces más rápido que el calculador electromecánico más sofisticado.

A la ENIAC se sucedió la EDVAC introduciendo el concepto de programa almacenado. La EDVAC junto con otra serie de máquinas desarrolladas en la misma época con técnicas y conceptos similares representan lo que se ha dado en llamar la primera generación de computadoras. Estas máquinas no sólo calculaban, sino que trabajan con miles o millones de datos, los organizaban y transformaban, creando así nueva información en tiempos muy reducidos, del orden de un milésimo de segundo.

Sin embargo, la característica que verdaderamente distinguió a estas máquinas de cualquier otra construida anteriormente, fue la capacidad de tomar decisiones lógicas o alterar su funcionamiento dependiendo del resultado de las operaciones y de los datos que ellas mismas manejaban. Estas computadoras podían entonces hacer un análisis sobre la información que les era dada, y sobre esta base, efectuar una operación en vez de otra. Naturalmente, esto era posible si la persona encargada de programar a la máquina le había comunicado de antemano como se tendría que comportar en los diferentes casos posibles.

La compañía Remington Rand desarrolló la UNIVAC I (Universal Automatic Computer), que fue la primera computadora de uso comercial, apareciendo en el año de 1951. Entre sus características se encontraba el uso de la cinta magnética para el suministro de datos, y almacenamiento de resultados; se inició el procesamiento de datos alfanuméricos, y el uso de un programa especial capaz de traducir en un lenguaje particular al lenguaje de máquina.

Esta generación de computadoras se caracterizó por el uso de tubos al vacío (bulbos) como componentes básicos de sus circuitos internos; por ella eran demasiado voluminosos (en realidad gigantescos), consumían mucha energía y producían tanto calor que fue preciso establecer rígidos controles en cuanto al aire acondicionado y temperatura. No eran tan confiables como se había esperado, eran rápidas pero no lo suficiente, y tenían capacidad de almacenamiento interno pero muy limitada.

CAPITULO I. Las Computadoras y la Administración.

Los transistores sustituyeron a los bulbos, reduciéndose las diferencias de las máquinas anteriores, surgieron las memorias de ferrita que permitieron reducir el tamaño, formando así la segunda generación de computadoras.

En 1964 apareció en el mercado la tercera generación de computadoras constituidas con circuitos integrados monolíticos, gracias a los cuales aumento considerablemente su velocidad de operación, incrementando a su vez su confiabilidad y disminuyendo su costo y tamaño. Una de las características fundamentales de estos nuevos equipos, fue la gran compatibilidad de sus componentes, permitiendo que hubiera una gran flexibilidad en la modificación o expansión de sistemas de computo sin alterar los sistemas básicos. Es importante señalar que las computadoras de esta generación se utilizaron, tanto para fines científicos como comerciales, sin embargo eran caras y solo podían ser manejadas por personal especializado.

A principios de la década de los 70's la manufactura de circuitos integrados llega a ser tan avanzada que se logran incorporar miles de componentes electrónicos en espacios de una fracción de pulgada, a esto se llama integración a gran escala de circuitos (LSI, Large Scale Integration).

Estos nuevos circuitos están mas densamente integrados que los sistemas anteriores y a partir de este tipo de componentes surgen las computadoras de la cuarta generación. Es aquí cuando aparecen los microprocesadores que son dispositivos electrónicos LSI que realizan las funciones de la unidad central de proceso de una gran computadora. Los microprocesadores son la base de las sofisticadas calculadoras programables de bolsillo, así como de las pequeñas y poderosas computadoras personales.

El nacimiento de las microcomputadoras tuvo lugar en los Estados Unidos, a partir de la comercialización de los primeros microprocesadores de 8 bits (Intel 8008, 8080) a comienzos de la década de los setenta. Las primeras de estas máquinas se conocían sencillamente como microcomputadoras y comenzaron a tener aceptación, primero exclusivamente en el mercado de los técnicos e ingenieros que deseaban o podían experimentar con esta nueva tecnología, que ponía al alcance de su mano un tipo de equipo que antes les era completamente inaccesible.

CAPITULO I. Las Computadoras y la Administración.

La empresa IMSAI (ya desaparecida) vendía una microcomputadora en forma de partes que el entusiasta experimentador armaba y probaba. Pronto otras compañías entraron a la competencia y algunas comenzaron a ofrecer equipos ya armados, más fáciles de utilizar, aunque seguía siendo absolutamente necesario un conocimiento especializado sobre la materia, o al menos, la disposición de dedicar muchas horas al día hasta obtenerlo.

Durante la década de los setentas aparecieron microcomputadoras de múltiples marca, y pronto se impusieron dos tendencias:

- La de los sistema Apple, que empleaban su propia tecnología basada en el microprocesador Mostek 6502, y un sistema operativo en particular, y

- La de casi todas las demás microcomputadoras, que empleaban el sistema operativo CP/M (Control Program/Monitor), basadas en los procesadores Intel 8080 y 8085, y luego en el Zilog Z80.

Durante varios años la situación se mantuvo con un ritmo de crecimiento rápido, pero frenado aún por la dificultad de empleo de estos dispositivos. Cuando en el mercado ya había adquirido proporciones importantes, y una vez que la nueva tecnología estaba ya aceptada, comenzó la verdadera explosión comercial masiva, con la introducción en 1981, de la Personal Computer de IBM. Esta máquina (basada en el microprocesador Intel 8088) tiene características interesantes, que hacen mucho mas amplio su campo de aplicaciones, sobre todo porque su nuevo sistema operativo estandarizado (MS-DOS, Microsoft Disk Operating System) y una capacidad mejorada de graficación, que la hacen mas atractiva y mas fácil de usar. La PC original ha pasado por varias transformaciones y mejoras, y los nuevos modelos siguen siendo los mas empleados. Existe una familia de sistemas de computadoras personales que se conocen con las nomenclaturas PC, XT, AT Y PS/2.

CAPITULO I. Las Computadoras y la Administración.

En la actualidad, la computación es una enorme industria que ha colocado decenas de millones de microcomputadoras en el mercado durante la última década, lo que a dado lugar también al nacimiento de tecnologías nuevas, tanto de diseño como de producción y empaque, con ello también se dado lugar al nacimiento de una terminología técnica nueva.

Originalmente, la computadora personal surgió para atender las necesidades de los usuarios individuales, en las ramas de procesamiento de palabras, hojas electrónicas de calculo, gráficas y aplicaciones varias, que en general no están directamente relacionadas con otras ramas de la operación de una oficina o industria.

Por otro lado, en una fábrica con varios centenares de empleados no es factible dedicar una computadora personal a las labores de mantenimiento de los inventarios, porque la capacidad de la pequeña máquina es insuficiente, porque para atender esta labor se requieren varios puntos de control y terminales de video, así como la capacidad de compartir y controlar los accesos a uno o varios discos magnéticos de gran capacidad, para efectos de consultas o movimientos.

Para este tipo de tareas se sigue requiriendo una microcomputadora, o incluso un mainframe. Por otro lado, una o varias computadoras personales si ofrecen una solución adecuada para las necesidades de planeación financiera o de procesamiento de textos de cada departamento de la fabrica, por ejemplo, una computadora personal incluye componentes de cada una de estas categorías:

- Componente electrónicos integrados (microprocesador, memoria, otros)

- Arquitectura global y canal

- Entrada/Salida e interfaces gráficas

- Interfaces para dispositivos periféricos

- Sistema Operativo

CAPITULO I. Las Computadoras y la Administración.

COMPONENTES DE LA COMPUTADORA

Una computadora digital es una máquina capaz de hacer operaciones aritméticas y lógicas utilizando representaciones discretas de los valores numéricos o lógicos de las variables. Todas las computadoras digitales, independientemente de su tamaño, son básicamente dispositivos electrónicos que pueden transmitir, almacenar y manipular información.

Básicamente existen dos tipos de datos: los datos numéricos y los alfanuméricos. Las aplicaciones científicas y técnicas exigen principalmente procesar datos numéricos, mientras que las aplicaciones comerciales implican generalmente procesamiento de datos, tanto numéricos como alfanuméricos.

Para procesar un conjunto particular de datos, la computadora debe disponer de un conjunto apropiado de instrucciones, al que se le llama programa, estas instrucciones se introducen en la computadora y luego se almacenan en una parte de la memoria; para lograr esto, la computadora se compone de cinco partes fundamentales a saber:

- a) Mecanismos de Entrada
- b) Mecanismos de Salida
- c) Unidad aritmética
- d) Memoria
- e) Unidad de Control

A. MECANISMOS DE ENTRADA.

Los mecanismos de entrada tienen por función introducir a la computadora la información necesaria para que la computadora pueda procesarla. Es la forma en que se establece una relación hombre-máquina. Algunos ejemplos de dispositivos de entrada son: monitor, teclado, cinta magnética, lector óptico, etc.

B. MECANISMOS DE SALIDA.

Los mecanismos de salida tienen por función obtener de la computadora la información o resultados del trabajo realizado por la computadora, es decir, son los medios de comunicación entre la computadora y el usuario. Entre ellos se encuentran, la impresora, la pantalla, cintas magnéticas, graficadores, etc.

C. UNIDAD ARITMETICA-LOGICA.

La unidad aritmética y lógica tiene como función principal realizar los cálculos aritméticos y lógicos de todas y cada una de las operaciones que se tengan que realizar dando a cada una prioridad y su solución lógica.

D. MEMORIA.

La memoria desarrolla la tarea de almacenar, ya sea temporal o permanentemente datos o instrucciones que requiera la unidad aritmética para funcionar de acuerdo a las necesidades del usuario. Para que una computadora pueda hacer operaciones mas complicadas que simples sumas, restas, multiplicaciones o divisiones es necesario que pueda guardar números e instrucciones (programas) para operar sin que una persona le este dando dichos números o instrucciones en forma continua durante un proceso.

CAPITULO I. Las Computadoras y la Administración.

Para hacer esta función, una computadora tiene una memoria principal RAM (Random Access Memory), en la cual se guardan temporalmente dichos datos, mientras se termina un proceso, y su capacidad dependerá de las necesidades del usuario de acuerdo a los programas, paquetes o sistemas que se requieran trabajar.

Los datos que se consideran importantes pero que no se puedan retener en el almacenamiento primario se envían al almacenamiento secundario el cual puede ser el disco duro de la computadora, disquetes flexibles, cintas magnéticas, etc. La cinta y el disco magnético son los medios principales de almacenamiento secundario que se usan en los sistemas de computación mas usuales. La cinta magnética se usa para almacenamiento secuencial, mientras que el disco magnético se usa para almacenamiento de acceso directo. El acceso secuencial a los datos es mucho mas lento que el acceso directo.

E. UNIDAD DE CONTROL.

La unidad de Control se encarga de coordinar a las cuatro partes anteriormente descritas para que operen como un todo armónico.

Todas las partes descritas, en su conjunto, forman lo que se conoce como la Unidad Central de Procesamiento (CPU), que es el eje de todas las actividades en un sistema de cómputo, en donde los programas se procesan permanentemente durante su ejecución.

La calidad de la Unidad Central de Proceso se mide en función de los siguientes aspectos a considerar:

- a. número de registros que tenga,
- b. la cantidad de bits que tenga cada registro,

CAPITULO I. Las Computadoras y la Administración.

c. la variedad de operaciones fundamentales que se puedan hacer con

dichos registros,

d. la velocidad a la que se puedan ejecutar dichas operaciones.

Esto resulta claro sin contar que en la actualidad, los coprocesadores tienen la misión principal de realizar más rápida todas las operaciones matemáticas y lógicas que realizan las computadoras.

EL SISTEMA OPERATIVO.

En la mayoría de las computadoras actuales, un programa de aplicación se ejecuta de una manera que determina tanto por el hardware de la computadora como por una colección de programas de control, de administración y de servicio llamado SISTEMA OPERATIVO.

El sistema operativo es el lenguaje de comunicación entre los usuarios y la computadora, el cual está constituido por diez partes fundamentales a saber:

1. CONTROL DE INTERRUPCIONES

Sistema de control que responde al medio ambiente externo.

CAPITULO I. Las Computadoras y la Administración.

2. CUADRO DE SERVICIO Y ASIGNADOR.

Determina que programas se van a ejecutar, y cuando, asignando memoria para su ejecución.

3. DESPACHADOR.

Lleva a cabo la mecanización del cuadro de servicio.

4. CONTROL E/S.

Controla las transferencias/comunicaciones de datos de/o para la computadora.

5. CONTROL DE ERROR Y PROCESADOR DE ERROR.

Controla y procesa los errores, de tal forma que se obtiene una operación libre de error y se mantiene la integración de los programas.

6. CONTROL DE PRIORIDAD.

Determina y controla el orden en el cual se ejecutaran los programas.

7. SUPERVISOR Y MONITOR.

Supervisa y sirve de monitor en la operación de los programas.

CAPITULO I. Las Computadoras y la Administración.

8. MANTENIMIENTO GENERAL.

Evita que el programador realice muchas tareas rutinarias.

9. CONTROL DE TAREA.

Controla cuando se requiere, la operación interna de las tareas para obtener datos, para llamar subrutinas, etc.

10. CONTROL DE TIEMPO REAL.

En los sistemas de tiempo real se controla la ejecución de tareas, de tal forma, que las operaciones se realizan de acuerdo con los requisitos de límite de tiempo externos.

El control de tareas es uno de los componentes más importantes de todos los arriba citados, ya que controla la ejecución de todas las tareas o programas. Aunque es la tarea principal, resulta ser la parte más pequeña del programa.

Los componentes estructurales descritos anteriormente, son para los sistemas de propósito general, aunque actualmente, estos sistemas ya están realizados de manera modular y de propósito especial. La tendencia evolutiva de los sistemas operativos empezó con la construcción de controles en cada programa de aplicación y el conjunto de subrutinas, rutinas utilitarias, así como el uso común de constantes.

Los sistemas primitivos estaban controlados, primeramente, desde la consola, y poco después, por tarjetas de control. El siguiente desarrollo evolutivo en el sistema operativo separó las partes de control de cada programa de aplicación en un programa separado, llamado antiguamente supervisor. El siguiente desarrollo añadió el control de muchos trabajos, entonces se introdujo la multiprogramación en un sentido limitado, y la tarea de entrada/salida y el procesamiento se realizaba de manera simultánea.

CAPITULO I. Las Computadoras y la Administración.

Esto fue seguido muy de cerca por el concepto de *TIEMPO COMPARTIDO* , en donde se hacia ya necesario controlar las operaciones mediante instrucciones de control, al principio estas estaban contenidas en tarjetas perforadas evolucionando hasta llegar a ser lenguajes de control.

En la última etapa ya se requería una mayor demanda en tiempo real, la cual consistía en realizar operaciones remotas para diversos trabajos y sistemas multiprocesadores que fueron complicando aun mas los sistemas operativos. Estos sistemas han crecido tanto en la actualidad que ya pueden controlar, crear y modificar todos los aspectos del ambiente en el que funcionan los programas de aplicación.

La base de datos es el componente estructural clave en el diseño de sistemas de información . Es la principal fuerza de integración del sistema de información de una organización. Debe lograrse un buen ajuste entre las necesidades de procesamiento y de toma de decisiones de la organización y la estructura y composición de la base de datos. Si no se logra esto, se desperdiciaron los esfuerzos del analista de sistemas en el diseño de los otros componentes estructurales

CAPITULO I. Las Computadoras y la Administración.

1.2. CONCEPTOS DE ADMINISTRACION Y ORGANIZACION DE DATOS

1.2.1. CONCEPTOS DE ADMINISTRACION DE DATOS

En las organizaciones grandes y complejas, muchos usuarios requieren simultáneamente tener acceso a la información. Los usuarios pueden ser ejecutivos, gerentes departamentales, personal de contabilidad y auditoría, vendedores, supervisores de manufactura, programadores y otras personas que necesiten estudiar la información acerca de la organización. Una base de datos consta de elementos de datos organizados en registros y archivos en forma tal que satisfagan los requerimientos de información de los usuarios. Los usuarios tienen acceso a la información derivada de esta base de datos mediante la función de administración de datos.

La administración de datos es el proceso de *almacenar y recuperar datos*. La administración de datos esta compuesta de tres tareas básicas:

1. Describir la organización real y la interrelación de los datos en una definición estándar de datos.
2. Almacenar físicamente los datos en un formato específico en un medio de almacenamiento dado
3. Recuperar los datos almacenados en una forma tal que proporcionen información válida a los usuarios del sistema.

El analista de sistemas tiene la responsabilidad de diseñar un sistema que cubra de manera eficaz las tres tareas de la administración de datos.

DESCRIPCIONES DE DATOS

Los datos representan objetos de la vida real. Los objetos acerca de los cuales se almacenan datos se denominan entidades. Una entidad puede ser un objeto tangible, como un empleado, un cliente o un artículo de inventario. Una entidad también podrá ser un evento, un proyecto de software o un centro de utilidades divisional. Una entidad tiene ciertos atributos que se desean registrar.

Un analista identifica todos los atributos de datos para una entidad, los cuales se necesitarán para proporcionar información a los usuarios. En ocasiones esta tarea puede realizarse revisando reportes o haciendo consultas al sistema.

Los analistas de sistemas con frecuencia deben anticipar las necesidades de los usuarios acerca de algunos atributos de los datos. Estos atributos podrán no ser necesarios para una consulta presente, pero probablemente podrán ser relevantes para necesidades de información futuras. El analista debe conocer lo suficiente acerca de la entidad para sugerir a los usuarios atributos adicionales al definir la descripción de los datos de un sistema.

ORGANIZACION FISICA DE DATOS

Los atributos tienen que almacenarse físicamente en un formato específico sobre un medio de almacenamiento para ser recuperados por los usuarios. El formato de los atributos incluye su tamaño y organización de los datos. La organización se refiere al orden físico de las entidades en un medio de almacenamiento. Los medios utilizados para almacenar datos vienen en una diversidad de formas físicas.

Una organización de datos más eficiente consiste en ordenar las entidades por atributos y luego intercalarlas para representar la ubicación de cada una de ellas. Esta forma de organización de datos se conoce como organización secuencial indexada. Para recuperar el registro primero se hace referencia a aquél que lo relaciona y luego se hace una búsqueda secuencial desde ese punto hasta encontrar el correcto.

RECUPERACION DE DATOS ALMACENADOS

Una vez que se definen los atributos y la organización física de una entidad, los usuarios pueden recuperar información acerca de la entidad

1.2.2. ADMINISTRACION COMPUTARIZADA DE DATOS.

Las computadoras son dispositivos excelentes para llevar a cabo el almacenamiento y recuperación de datos debido a su velocidad y a su capacidad de almacenamiento. Las tareas de la administración de datos no se realizan en forma diferente con una computadora, sino en forma más eficiente.

El analista de sistemas tiene que describir los atributos de datos para cada entidad y diseñar la organización física de los datos. La computadora se utiliza para recuperar en forma eficiente los datos en un formato deseado por los usuarios. Dos enfoques para la administración computarizada de datos son:

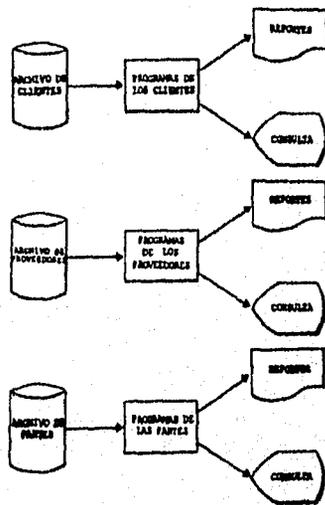
- 1) Ambientes tradicionales de procesamiento de archivos
- 2) Ambientes de administración de bases de datos

AMBIENTES TRADICIONALES PARA PROCESAMIENTO DE ARCHIVOS.

Los ambientes tradicionales para el procesamiento de archivos son sistemas orientados al proceso. Los datos fluyen de un programa a otro. Los archivos de datos se crean para satisfacer necesidades específicas de procesamiento. Cada programa o sistema de aplicación que se desarrolle contendrá los datos necesarios para cubrir sólo las necesidades de un departamento en particular o un grupo de usuarios.

CAPITULO I. Las Computadoras y la Administración.

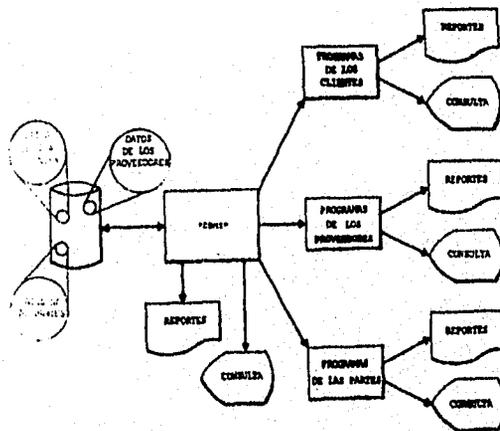
La siguiente figura ilustra un sistema tradicional de archivos para una organización . Cada archivo es accesado en forma separada por los programas de aplicación. Este sistema trabajará en forma eficaz hasta que los datos tengan que compartirse entre las aplicaciones.



AMBIENTE DE ADMINISTRACION DE BASES DE DATOS

En un ambiente de administración de bases de datos el enfoque está en los datos y no en los procedimientos utilizados para el procesamiento de los mismos. La base de datos se crea en forma separada de los programas que accesan los datos. En este ambiente, los datos se consideran como un recurso compartido. El recurso de los datos se implementa físicamente para que pueda ser compartido por varios usuarios.

La figura siguiente, muestra un ambiente de administración de bases de datos para la organización. Observe que cada programa interactúa con el sistema de administración de base de datos y no directamente con los datos. El sistema de administración de base de datos es responsable de proporcionar acceso a todos los datos.



RECUPERACION DE DATOS EN LOS DOS AMBIENTES

El ambiente tradicional de procesamiento de archivos accesa los datos directamente de la información intercambiada entre el programa de aplicación y el sistema operativo. Un ambiente de administración de base de datos requiere el uso de un sistema de administración de bases de datos.

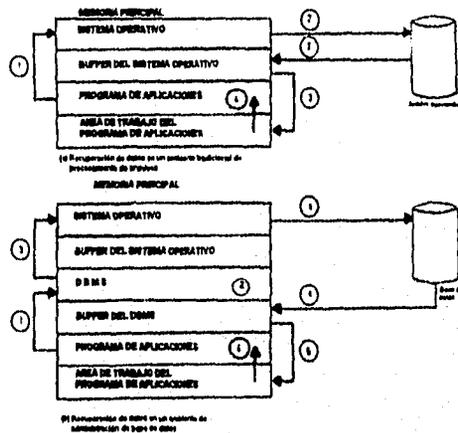
Un **sistema de administración de base de datos (DBMS)** es un conjunto de programas especializados diseñados para describir, proteger, almacenar y acceder la base de datos.

Las funciones de un DBMS son las siguientes:

1. Definir todos los datos empleados en el sistema y especificar las relaciones entre los datos en forma separada a los programas de aplicación que accesan los datos.
2. Proporcionar un método para dar de alta, dar de baja y modificar los datos que están en la base de datos.
3. Proteger el recurso de los datos de manera que sea seguro, confiable, consistente y correcto.
4. Permitir que los usuarios múltiples de la base de datos compartan los datos.
5. Permitir la recuperación de datos mediante un lenguaje entendible a los usuarios del sistema.

CAPITULO I. Las Computadoras y la Administración.

La figura ilustra la diferencia en el acceso de datos entre un sistema tradicional de procesamiento de archivos y un sistema de administración de base de datos. En un sistema tradicional de procesamiento de archivos, la recuperación de datos ocurre de la siguiente manera:



1. El programa de aplicación solicita leer un registro de un archivo de datos. El programa de aplicación contiene información acerca de la organización del archivo, de la forma en que se accederá, del dispositivo físico donde se encuentran los datos y de la estructura exacta de los campos del archivo.

2. El sistema operativo hace que los datos sean accesados, leídos y transferidos a un buffer en la memoria principal.

3. El sistema operativo transfiere los datos al área de trabajo para el programa de aplicación.

CAPITULO I. Las Computadoras y la Administración.

4. El programa de aplicación procesa los datos.

En el sistema de administración de base de datos, la recuperación de datos ocurre de la siguiente manera:

1. El programa de aplicación solicita leer un registro de la base de datos.

2. El DBMS verifica que los datos se haya definido previamente dentro de la base de datos. El DBMS identifica la ubicación de los datos.

3. El DBMS solicita que el sistema operativo lea los datos.

4. El sistema operativo hace que los datos sean accedados, leídos y transferidos a un buffer en la memoria principal utilizado por el DBMS.

5. El DBMS transfiere los datos desde su buffer hasta el área de trabajo del programa de aplicación. El DBMS también proporciona al programa de aplicación información de situación o condición, como el caso de "registro encontrado".

6. El programa de aplicación procesa los datos.

BENEFICIOS DEL AMBIENTE DE ADMINISTRACION DE BASE DE DATOS.

Un examen superficial de estos dos sistemas, podría hacer que uno piense que un DBMS no hace otra cosa que agregas pasos adicionales. Los pasos adicionales que se agregan a un DBMS en realidad facilitan a los usuarios la recuperación de los datos. El ambiente de base de datos ofrece varias ventajas importantes en comparación con el ambiente tradicional de procesamiento de archivos. Algunos de los beneficios principales se estudian en los párrafos siguientes:

1. **MEJORA EN LA INTEGRACION DE LOS DATOS.** En un sistema de administración de base de datos, los datos se organizan en una estructura lógica que permite definir relaciones múltiples entre las entidades de datos.

2. **AUMENTO EN LA ACCESIBILIDAD DE LOS DATOS.** La accesibilidad de los datos la capacidad de un usuario para obtener la información necesaria de la base de datos. Un DBMS generalmente contiene un lenguaje de consulta de alto nivel, como el de consulta. Por ejemplo (QBE), que permite que los datos se obtengan de la base de datos sin tener que escribir un programa de aplicación.

3. **MEJORA EN EL CONTROL DE LOS DATOS.** Una base de datos consta de un conjunto integrado de archivos. La redundancia de los datos puede minimizarse debido a que el recurso de los datos se controla mediante un conjunto único de programas. Como resultado de esto, es menos probable que se presenten inconsistencias en los datos.

4. **FACILIDAD EN EL DESARROLLO Y ADMINISTRACION DE APLICACIONES.** Los estudios demuestran que una vez que ha sido diseñado e implementado una base de datos, un programador puede codificar y depurar una nueva aplicación por lo menos de dos a cuatro veces más rápido que con archivos tradicionales. Con un sistema de administración de base de datos, el programador no tiene que estar consciente de la estructura real, la organización y localización del archivo. Al liberar al programador de aplicaciones de estas tareas se reduce el costo en el desarrollo de software. Una organización puede modificar la estructura de los datos de la base de datos sin tener que modificar los programas de aplicación para el acceso de los datos. Esta función reduce los costos de mantenimiento de los programas.

CAPITULO I. Las Computadoras y la Administración.

5. MEJORA EN LA SEGURIDAD DE LOS DATOS. La seguridad de los datos impide el acceso no autorizado de los datos. La mayoría de las organizaciones requieren alguna forma de protección contra un acceso no autorizado a los datos. Debido a que el DBMS puede controlar el acceso a las entidades de los datos, la función de seguridad se centraliza y se implementa fácilmente con el empleo de un sistema de administración de base de datos.

COMO DEBERA UNA ORGANIZACION CONSIDERAR UN SISTEMA DE ADMINISTRACION DE BASE DE DATOS ?

Ciertos factores acerca de una organización indican el grado de aplicación de un sistema de administración de base de datos. Una organización deberá considerar el empleo de un sistema de administración de base de datos si:

- a. Las necesidades de las aplicaciones están cambiando constantemente. Una organización que persigue nuevos mercados o desarrolla nuevas líneas de productos tendrá requerimientos de datos diferentes a los de una organización que tiene un producto más estático.
- b. Se requerirán consultas específicas e intermitentes del sistema de administración de base de datos.
- c. Muchos departamentos emplean esencialmente los mismos datos para satisfacer sus necesidades de información.
- d. Existe una necesidad de reducir los tiempos de terminación de los programas y disminuir los costos de desarrollo de los mismos.
- e. Existe la necesidad de mejorar la consistencia de los datos.

No todas las organizaciones deberán adoptar el enfoque de un sistema de administración de base de datos para el manejo de los datos. Una organización deberá utilizar un ambiente tradicional de procesamiento de archivos si:

CAPITULO I. Las Computadoras y la Administración.

- a. Existe muy poca necesidad de compartir los datos entre usuarios.
- b. La organización opera en un ambiente estático tanto interno como externo. Una organización que experimente poco o ningún cambio, generalmente sólo requiere un procesamiento de transacciones muy estandarizado. Este procesamiento se realiza de manera más eficiente con un sistema tradicional de procesamiento de archivos.
- c. La planeación de datos no puede coordinarse entre las fronteras departamentales. Un sistema tradicional de archivos no requiere que departamentos separados trabajen conjuntamente para planear sus necesidades futuras de datos. El empleo de un DBMS obliga a una organización a colaborar para crear la estructura de la base de datos.

Un ambiente de administración de base de datos requiere que una organización realice una inversión substancial de capital en el software del DBMS. También se requiere de una gran inversión oportuna al estructurar e introducir los datos en un formato de base de datos. Adicionalmente, un DBMS funciona de manera más eficaz, si una persona o un departamento es responsable de la planeación total e implementación de la base de datos. Un analista de sistemas debe examinar a una organización para asegurarse que el sistema de administración de base de datos proporcione la mejor alternativa de administración de datos.

1.2.3. ALTERNATIVAS DE LA ORGANIZACION DE DATOS

ORGANIZACION SECUENCIAL DE ARCHIVOS.

La forma más sencilla de organizar un conjunto de registros que forman un archivo consiste en la organización secuencial. Los registros se almacenan uno después de otro sin preocuparse del valor real de los datos contenidos en los registros. El primer registro se colocan en la primera posición del archivo, el segundo registro se coloca en la segunda posición, y así sucesivamente hasta que todos los registros estén colocados en el archivo. los registros también se pueden clasificar en un orden especificado previamente dentro de una organización secuencial de archivos.

CAPITULO I. Las Computadoras y la Administración.

Con una organización secuencial de registros de datos, los registros se colocan en el archivo empleando una clave o código para su secuencia. Por lo general, antes de cambiar o actualizar el archivo secuencial, todos los nuevos elementos se reúnen primeramente en lotes (se agrupan) y se ordenan en la misma secuencia. Para acceder cualquier registro de datos en el archivo secuencial, se deberá pasar por todos aquellos registros que preceden al registro en cuestión. La inserción de un registro de datos significa que se debe crear un nuevo archivo secuencial.

La secuencia del archivo generalmente se elige de acuerdo a algún atributo común denominado llave. La secuencia de un archivo puede modificarse seleccionando una llave diferente y ordenando los registros almacenados de acuerdo a los valores de la nueva llave. En algunos casos, el empleo de un atributo de datos como llave de ordenamiento no es suficiente para identificar a un registro almacenado dado. En este caso, se combinan uno o más atributos de datos adicionales para formar la llave.

La organización secuencial es un método eficiente de organización de datos cuando en cada corrida se procesa un gran volumen de registros y un porcentaje relativamente alto de los mismos. De esta forma, la organización secuencial ofrece un acceso rápido al siguiente registro de un archivo si la base de la recuperación es la misma que la base del ordenamiento físico del archivo. En un sistema de información que debe satisfacer una diversidad de necesidades de información, esto raras veces es el caso.

Si un grupo de registros almacenados se debe procesar empleando más de una llave para satisfacer las solicitudes de información, los registros almacenados se ordenan en diferentes archivos de trabajo. Los contenidos de cada archivo son los mismos, pero el ordenamiento es diferente para los distintos propósitos. Esta duplicación de archivos desperdicia espacio de almacenamiento y tiempo de procesamiento.

La organización secuencial de archivos es eficaz para archivos de gran volumen y una alta tasa de actividad que no tienen que accederse en forma aleatoria. La organización secuencial de archivos se puede utilizar tanto con discos magnéticos como con cintas magnéticas como medios de almacenamiento. Una desventaja principal de la organización secuencial de archivos es que un usuario no puede acceder directamente un registro individual del archivo.

ORGANIZACION DIRECTA DE ARCHIVOS

La organización directa de archivos permite el acceso aleatorio de los registros de los archivos. Esta tarea se realiza mediante una relación entre el campo llave del registro y la ubicación física del registro en un dispositivo de almacenamiento secundario. La organización directa infiere una relación directa entre un campo llave y una ubicación de almacenamiento. La organización directa de archivos se basa en el mantenimiento de una relación predecible entre el valor del campo llave de un registro y su ubicación en un dispositivo de almacenamiento secundario. Se pueden mantener dos relaciones diferentes predecibles: CONVERSION DIRECTA Y PARTICIONES (HASHING).

CONVERSION DIRECTA.

La conversión directa permite que los registros se encuentren rápidamente debido a que el mecanismo de acceso necesita moverse sólo una vez. Con la conversión directa, cada valor posible de la llave debe corresponder a una dirección única de almacenamiento. En consecuencia, el rango del valor de la llave dicta el número de ubicaciones físicas de almacenamiento que deben reservarse para el archivo de datos.

HASHING

El método de particiones o hashing (también denominado método de algoritmos, de asignación aleatoria o de transformación convierte una llave en una dirección relativa física del archivo. En este caso, el rango de las llaves en un archivo se comprime en un rango más pequeño de direcciones físicas. La dificultad principal que se encuentra en el hashing es el problema de los sinónimos (registros cuyas llaves dan aleatoriamente la misma dirección física).

CAPITULO I. Las Computadoras y la Administración.

Para minimizar los sinónimos que no pueden escribirse en el lugar donde pertenecen (registros de sobreflujo), se emplean dos técnicas. Con la primera técnica, el hashing es una dirección de pista, en lugar de una dirección de registro. De esta forma, no todos los sinónimos producirán sobreflujo. Una segunda técnica consiste en seleccionar un algoritmo de hashing o particiones que distribuya los registros uniformemente en el archivo. De los muchos algoritmos disponibles, el más popular parece ser el de división por números primos.

ORGANIZACION SECUENCIAL INDEXADA DE ARCHIVOS

La organización secuencial indexada de archivos es un intento por retener las ventajas de la organización secuencial de archivos, permitiendo al mismo tiempo el acceso aleatorio a los registros.

Un archivo secuencial indexado está formado en realidad por dos archivos:

1. Un archivo que contiene los datos clasificados y ordenados por un campo llave.
2. un archivo que tiene una serie de índices utilizados para el acceso de los datos.

La organización secuencial indexada de archivos, permite el acceso a los registros del archivo tanto en forma secuencial como aleatoria. Permite que el procesamiento secuencial comience en el primer (o cualquier otro) registro del archivo y el proceso de tantos otros registros subsecuentes como se desee. En forma alterna, la organización también permite que un usuario obtenga aleatoriamente un registro seleccionado haciendo una búsqueda en una serie de subíndices. Los datos se localizan con el método secuencial indexado creando un archivo físico separado, que están compuesto de una entrada de índice para cada enésimo registro de datos y no por cada registro de datos. Primero se hace una búsqueda secuencial hasta que se determina la ubicación aproximada del registro que se está buscando.

Luego, a partir de la ubicación indicada en el archivo de datos por la entrada del índice, se hace una búsqueda secuencial de la porción localizada del archivo de datos hasta que se encuentra una correspondencia con el valor llave deseado.

La organización secuencial indexada de archivos deberá utilizarse cuando sea necesario tener acceso a los datos tanto en forma secuencial como aleatoria. Un archivo secuencial indexado no se podrá acceder secuencialmente tan rápido como un archivo secuencial. Tampoco se podrá acceder aleatoriamente tan rápido como un archivo de organización directa. La ventaja más grande de la organización secuencial indexada es su flexibilidad.

RESUMEN

La característica dominante de nuestra sociedad, es la explosión de la información basada en la tecnología. Las computadoras están cambiando totalmente la forma en que trabajamos y vivimos. Es difícil predecir la naturaleza de la sociedad caracterizada por la información, pues los cambios son muy rápidos. El auge de la tecnología de la información conlleva la necesidad de administrar grandes sistemas. La información, a diferencia de los datos, consta de signos significativos que repercuten en el comportamiento. Hemos pasado de generar montones de datos, a producir información selectiva.

CAPITULO II.

BASES DE DATOS Y SISTEMAS DE INFORMACION.

Objetivo:

Explicar los conceptos básicos en se fundamenta la administración de la base de datos en un sistema de información administrativa.

CAPITULO II. Bases de datos y Sistemas de Información.

INTRODUCCION.

La información no siempre nace como tal. Lo más normal es que se obtenga de datos. Las organizaciones reúnen y almacenan datos para convertirlos en información, en el momento oportuno. Se da el nombre de base de datos, a la colección integrada de datos lógicamente organizados. La base de datos constituye un recurso de suma utilidad. Si se destruyera repentinamente, la organización quedaría paralizada. Un caso menos drástico, ocurre cuando se dificultan las operaciones, si es imposible recuperar los datos almacenados, como sucede frecuentemente con los sistemas manuales.

CAPITULO II. Bases de datos y Sistemas de Información.

2.1 FUNDAMENTOS DE BASES DE DATOS

LA BASE DE DATOS

El término base de datos describe una colección de información organizada en columnas y filas. Cada fila es una entrada, o registro, en la base de datos. Cada columna contiene los campos que componen cada registro. Al añadir un registro a una base de datos, se le proporciona un número secuencial que indique el orden en que fue añadido a la base de datos.

Una base de datos es un sistema de archivos de computadora que usa una organización de archivos particular para facilitar la actualización rápida de registros aislados, la actualización simultánea de registros relacionados, fácil acceso de los programas de aplicaciones a todos los registros, y acceso rápido a todos los datos almacenados que deben unirse para satisfacer un informe o consulta particular de rutina o de propósito especial.

Una base de datos facilita la actualización rápido de registros aislados y la actualización simultánea de registros relacionados; es decir, una base de datos permite la entrada de una transacción para actualizar en forma simultánea todos los registros afectados por esa transacción.

Las relaciones lógicas de los registros de una base de datos no están formadas por programas ni programadores. Por el contrario, los registros están relacionados entre sí de un modo extraordinariamente complejo, antes de ser creados. La descripción de las relaciones lógicas no se caracteriza por los programas que los procesen, tampoco las caracterizan las descripciones físicas de los registros que necesariamente se forman en los programas. El conjunto exacto de reglas que se aplican para establecer asociaciones entre los registros del sistema determinará la clase de base de datos. Hay tres tipos fundamentales de sistemas de bases de datos:

CAPITULO II. Bases de datos y Sistemas de Información.

- a. base de datos relacional
- b. base de datos jerárquica
- c. red o base de datos de enlace

BASE DE DATOS RELACIONAL

En el caso de una estructura de base de datos relacional, los usuarios pueden tomar archivos, y seleccionar sólo algunos de ellos. Pueden hacer esas operaciones fácilmente sin ser abrumados por la programación. En esencia, los datos son presentados en tablas bidimensionales. La decisión de recurrir a una base de datos relacional es elegir entre el poder de la computadora o el del ser humano.

Naturaleza de una "relación"

Las características de la relación deben cumplir con ciertas condiciones para que la computadora efectúe las operaciones sobre ella. El desarrollo de una relación comienza con el punto de vista del usuario. Dicha perspectiva es una representación lógica de los datos que se requieren para tomar una decisión o contestar una pregunta del usuario. Esa relación es una tabla donde las columnas son atributos y los renglones representan los registros individuales. Otra característica de una relación consiste en que en cualquier posición de renglón y columna existe exactamente un valor, nunca un grupo de valores. Se dice que esta tabla está normalizada.

CAPITULO II. Bases de datos y Sistemas de Información.

Formación de nuevas relaciones

La base de datos relacional (RDB) se ajusta a condiciones matemáticas muy precisas. El usuario no se limita simplemente a buscar las relaciones iniciales guardadas en ella. Puede construir fácilmente nuevas relaciones al seleccionar las de un menú desplegado en pantalla; después, puede escoger en el menú algunas operaciones realizadas en ellas. Sólo se necesitan tres operaciones en el m,todo relacional algebraico:

1. UNIR. La operación " unir " se limita a combinar dos relaciones para formar otra. Únicamente dos relaciones pueden combinarse a la vez, apareando los atributos seleccionados en cada una de las relaciones.

2. PROYECTAR. La operación " proyección " nos permite seleccionar cualquier conjunto de atributos (campos) a partir de una relación establecida.

3. SELECCIONAR. La operación seleccionar nos permite escoger un registro individual entre los que integran una relación.

BASE DE DATOS JERARQUICA

El modelo jerárquico es una estructura familiar, representa los datos como un árbol invertido de entidades interconectadas. El nodo superior de este modelo se denomina raíz. Otros nodos se ramifican desde la raíz. Ninguna ocurrencia de un nodo puede tener más de un nodo en su nivel superior. En un organigrama, esto significa que ningún empleado puede tener más de un supervisor, pero un supervisor puede tener un número ilimitado de empleados.

El modelo jerárquico por lo general lo entienden fácilmente las personas que emplean la base de datos. Sin embargo, este modelo no se puede utilizar para representar completamente a todas las bases de datos. Los modelos jerárquicos se adaptan bien para la representación de bases de datos que tienen relaciones de 1:1 y de 1:M entre las entidades. Además, este modelo funciona si cada nodo está enlazado con sólo un nodo de nivel superior, es decir, un padre.

CAPITULO II. Bases de datos y Sistemas de Información.

RED, O BASE DE DATOS DE ENLACE

Cualquier entidad en una red se puede relacionar con otra entidad en un número infinito de formas. El modelo de red no tiene regla estricta de un padre que gobierna al modelo jerárquico. Una red puede ser sencilla o compleja.

Una red se considera compleja si contiene relaciones de M:N. Los modelos de redes complejas pueden emplearse para mostrar todas las relaciones posibles entre las entidades de una base de datos. Sin embargo, los modelos de redes rápidamente pueden volverse difíciles de entender.

En una estructura tipo red, un hijo sí puede tener más de un padre, siempre y cuando cada uno este en un conjunto diferente. Una relación M:N es una red en sí misma. El modelo de red permite modelar en forma directa relaciones M:N, pero en este hay un nuevo elemento que se llama conector, que se puede representar como un registro que contiene datos que describen la asociación entre padres e hijos.

Todas las ocurrencias de un conector para un propietario se colocan en una cadena que parte del mismo y retorna a él, igual sucede con las ocurrencias de un miembro. De esta manera cada ocurrencia del conector esta en dos cadenas, en una de su padre y en una de su hijo. Esto hace que la estructura interna de un archivo sea muy compleja, pues tiene muchos apuntadores.

SISTEMA ADMINISTRADOR DE BASES DE DATOS

Los sistemas de bases de datos son parte esencial de los sistemas computacionales, debido a la facilidad que presentan para el manejo de grandes volúmenes de información. La información no siempre nace como tal. Lo más normal es que se obtenga de datos. Las organizaciones reúnen y almacenan datos para convertirlos en información en el momento oportuno. Se da el nombre de base de datos a la colección integrada de datos lógicamente organizados y relacionados entre sí. La base de datos constituye un recurso de suma utilidad.

CAPITULO II. Bases de datos y Sistemas de Información.

Antaño, con los sistemas manuales, los datos se guardaban en diversos tipos de archivos físicos; por ejemplo, los archiveros y las tarjetas de clasificación por llave. En esos archivos existía una enorme duplicidad de datos.

El hardware y el software han venido a reducir mucho la redundancia de datos. La base de datos puede ser considerada en la actualidad, conceptualmente, como una colección, prescindiendo de la dispersión o configuración física del equipo. Es decir, el sistema computacional podría poner casi instantáneamente a disposición de usuario cualesquiera información de la base de datos.

A cualquier organización, le interesa el rendimiento y la economía en el desarrollo de una base de datos. Debe comprobar las respuestas, en función de los costos a preguntas como las siguientes:

- 1. ¿Cuáles y cuántos datos deben almacenarse ?**
- 2. ¿Cuánto tiempo deben permanecer en la base de datos ?**
- 3. ¿Qué rapidez se desea que tenga la recuperación de datos y en qué forma debe realizarse?**
- 4. ¿Qué equipo y software deben seleccionarse ?**
- 5. ¿Cómo debe la compañía organizar la administración de los recursos de datos?**
- 6. ¿Cómo debe la información sacarse de la base de datos guardada en máquinas?**

Un ambiente de administración de base de datos requiere el uso de un sistema de administración de base de datos. Un sistema de administración de base de datos (DBMS) es un conjunto de programas especializados diseñados para describir, proteger, almacenar y acceder la base de datos.

CAPITULO II. Bases de datos y Sistemas de Información.

El DBMS (DATA BASE MANAGEMENT SYSTEM), Sistema administrador de base de datos, es un software que se encarga de controlar el uso de los datos dentro del sistema. Se puede decir que es la parte fundamental del sistema de base de datos, ya que sin éstos sólo serían un almacén de datos sin ningún control.

El Sistema administrador de bases de datos intercepta todas las solicitudes de acceso a la base de datos, revisa su validez, realiza las operaciones de entrada/salida a los archivos, obtiene un resultado y lo regresa al usuario.

Las funciones de un DBMS son las siguientes:

- 1. Definir todos los datos empleados en el sistema y especificar las relaciones entre los datos en forma separada a los programas de aplicación que accesan los datos.**
- 2. Proporcionar un método para dar de alta, dar de baja, y modificar los datos que están en la base de datos.**
- 3. Proteger el recurso de los datos de manera que sea seguro, confiable, consistente y correcto**
- 4. Permitir que los usuarios múltiples de la base de datos compartan los datos.**
- 5. Permitir la recuperación de datos mediante un lenguaje entendible a los usuarios del sistema.**

CAPITULO II. Bases de datos y Sistemas de Información.

BENEFICIOS DEL AMBIENTE DE ADMINISTRACION DE BASE DE DATOS

El ambiente de base de datos ofrece varias ventajas importantes en comparación con los sistemas tradicionales de procesamiento de archivos. Algunos de los beneficios principales se estudian en los párrafos siguientes:

a. Mejora en la integración de los datos. En un sistema de administración de base de datos, los datos se organizan en una estructura lógica que permite definir relaciones múltiples entre las entidades de datos.

b. Aumento en la accesibilidad de los datos. La accesibilidad de los datos es la capacidad de un usuario para obtener la información necesaria de la base de datos. Un DBMS, generalmente contiene un lenguaje de consulta de alto nivel, como el de Consulta por Ejemplo (QBE), que permite que los datos se obtengan de la base de datos sin tener que escribir un programa de aplicación.

c. Mejora en el control de los datos. Una base de datos consta de un conjunto integrado de archivos. La redundancia de los datos puede minimizarse debido a que el recurso de los datos se controla mediante un conjunto único de programas. Como resultado de esto, es menos probable que se presenten inconsistencias en los datos.

d. Facilidad en el desarrollo y administración de las aplicaciones. Los estudios demuestran que una vez que se ha diseñado e implementado una base de datos, un programador puede codificar y depurar una nueva aplicación por lo menos dos a cuatro veces más rápido que con archivos tradicionales. Con un sistema de administración de base de datos, el programador no tiene que estar consciente de la estructura real, la organización y la localización del archivo. Al liberar al programador de aplicaciones de estas tareas, se reduce el costo en el desarrollo del software.

e. Mejora en la seguridad de los datos. La seguridad de los datos impide el acceso no autorizado a los datos. La mayoría de las organizaciones requieren alguna forma de protección contra un acceso no autorizado a los datos. Debido a que el DBMS puede controlar el acceso a las entidades de los datos, la función de seguridad se centraliza y se implementa fácilmente con el empleo de un sistema administrador de base de datos.

2.2. CICLOS DE DESARROLLO DE SISTEMAS DE INFORMACION

SISTEMAS DE INFORMACION

Concepto de información

En un mundo que se concentra en los logros y las ventajas, la información puede ser el factor crítico que les permita a las organizaciones obtener una ventaja competitiva.

La información es un recurso crítico de las organizaciones, tan fundamental como la energía o las máquinas. Es el eslabón indispensable que une a todos los componentes de la organización para una mejor operación y coordinación y para su supervivencia en un ambiente competitivo y poco amigable.

La información la componen los datos que se han colocado en un contexto significativo y útil y se ha comunicado a un receptor, quien la utiliza para tomar decisiones. Implica la comunicación y recepción de la inteligencia o conocimiento. Evalúa y notifica, sorprende y estimula, reduce la incertidumbre, revela alternativas adicionales o ayuda a eliminar las irrelevantes.

La información es un signo o conjunto de signos que impulsan a la acción. Se distingue de los datos porque éstos no son estímulos de la acción, sino simplemente cadenas de caracteres o patrones sin interpretar. La toma de decisiones, un paso que conduce a la acción, se basa en la información.

Atributos de la información

Muchas personas tienden aún a creer que la información la forman listados de computadora. Otras afirman que los usuarios están sufriendo de una sobrecarga de información. Aunque a decir verdad, podemos estar inundados por una contaminación de listados y mala información, muchas personas carecen aún de información de calidad.

CAPITULO II. Bases de datos y Sistemas de Información.

La calidad de información descansa sólidamente sobre tres pilares: exactitud, oportunidad y relevancia. Estos son los atributos claves de la información.

EXACTITUD. Significa que la información esté libre de errores. Significa que la información es clara y refleja adecuadamente el sentido de los datos en los que se basa. Transmite una imagen clara al receptor, lo cual puede requerir una presentación en forma gráfica. La exactitud significa que la información está libre de tendencias o desviaciones. Este poder de la información, permite manipular o distorsionar la información para asegurar que las decisiones que se tomen reflejen sus mejores intereses.

OPORTUNIDAD. El hacer llegar la información a los receptores dentro del marco de tiempo necesario es otro atributo clave de la calidad de la información. La oportunidad en la información significa simplemente que los receptores la puedan obtener cuando la necesiten.

RELEVANCIA. La relevancia es el ultimo atributo clave de la calidad de información. En palabras sencillas, " la información responde de manera especifica al receptor sobre el que, por qué, donde, cuándo, quién y cómo?".

Concepto de Sistema

Un sistema se define como un conjunto de elementos y procedimientos que actúan de manera interrelacionada y conjunta con la finalidad de lograr un objetivo. Los sistemas varían en cuanto a los elementos, aspecto, volumen y objetivos, pero tienen características comunes: mantener la funcionalidad de los componentes para lograr determinadas metas mediante el cumplimiento de las funciones específicas de cada una de las partes que los conforman.

Sistema de información

Se da este nombre, al sistema que examina y recupera los datos provenientes del ambiente, que captura los datos a partir de las transacciones y operaciones, que filtra, organiza y selecciona los datos, y los presenta en forma de información, proporcionando medios para generar la información deseada.

CAPITULO II. Bases de datos y Sistemas de Información.

El sistema de información no sólo proporciona información para ayudar a la toma de decisiones, sino que además puede estar diseñado para toma de decisiones en el caso de problemas repetitivos. El moderno sistema de información ofrece un sistema que actúa sobre los datos, y la información para satisfacer la demanda de información.

Un sistema de información es un conjunto de elementos que conforman una actividad o procedimiento para obtener datos y procesarlos a fin de brindar información para la toma de decisiones.

Objetivos y funciones del sistema de información

Los objetivos del sistema de información son: obtener información oportuna, rápida y confiable a través de la explotación adecuada de las herramientas y técnicas con las que se cuenta para el procesamiento de datos, obteniéndose, beneficios como:

- 1. mayor oportunidad para la organización**
- 2. incremento de la eficiencia de operación**
- 3. reducción de costos, etc.**

Debe planearse de acuerdo a los recursos, ambiente, políticas, operaciones y planes de la organización.

CAPITULO II. Bases de datos y Sistemas de Información.

Las funciones principales de un sistema de información son:

- a. Recolección de datos.** Consiste en la recopilación de los datos a evaluar y analizar.
- b. Conversión de datos.** Para procesarlos, es necesario convertirlos a un código adecuado de acuerdo a los medios con los que cuenta el sistema para el proceso y almacenamiento de los mismos.
- c. Almacenamiento de datos.** Consiste en definir cómo se almacenarán los datos y en qué.
- d. Transmisión de datos.** Consiste en definir como se transmitirán los datos y con que serán transmitidos.
- e. Proceso de datos.** Calcula las operaciones lógico-matemáticas necesarias para proporcionar los resultados requeridos por un sistema de información a partir de los datos que se han suministrado.
- f. Recuperación de la información.** Se recuperan los datos que han sido procesados y analizados para una correcta toma de decisiones.

CICLO DE VIDA DEL SISTEMA DE INFORMACION

El concepto del ciclo de vida de un sistema de información, es medular en la investigación de sistemas. Durante su desarrollo, cada sistema se mueve a través de varias fases de un ciclo, después, del cual, sólo funciona por un cierto período de tiempo con un mínimo de mantenimiento. El sistema se deteriora gradualmente, hasta el punto en que deja de funcionar y se comienza un nuevo ciclo con el desarrollo de un nuevo sistema.

CAPITULO II. Bases de datos y Sistemas de Información.

El ciclo de vida del sistema de información muestra cinco fases principales, las cuales son:

a. Análisis de Sistemas

b. Diseño general de sistemas

c. Evaluación de sistemas

d. Diseño detallado de sistemas

e. Implantación de sistemas

Las primeras cuatro fases están dirigidas a proporcionar valores específicos para los componentes estructurales. La última fase se ocupa de que los componentes estructurales sean operacionales. La metodología del desarrollo de sistemas racionaliza y asigna una rutina al proceso de construcción de sistemas de información. Su sello distintivo lo forman sus fases discretas. La meta principal de la metodología del desarrollo de sistemas es reducir los inicios falsos, reciclamiento indebido, retrabajos y callejones sin salida. Además, aumenta la probabilidad de que el sistema que se construya e instale finalmente, sea el que los usuarios desean y necesiten. Pero de ninguna manera deberá ahogar la creatividad y la innovación. Proporciona un faro guía para el recorrido a lo largo de un sistema de información.

CAPITULO II. Bases de datos y Sistemas de Información.

Definición del ciclo de vida del desarrollo de sistemas

Análisis de Sistemas

Durante la fase del análisis de sistemas, los analistas de sistemas ayudan a los usuarios a identificar qué información se necesita. Sin embargo, en este punto los analistas de sistemas deben ser líderes y maestros. Con frecuencia los usuarios no tienen una idea completa de cuál es la información que realmente necesitan o cómo pueden obtenerla. En la mayoría de los casos, deben interactuar y determinar con los usuarios cuáles son sus verdaderas necesidades. Después de un buen número de entrevistas, observaciones y muestreos, los analistas de sistemas empiezan a conjuntar muchos hechos del estudio, para un análisis posterior.

Diseño General de sistemas

Antes de iniciar el diseño general de sistemas, la fase de análisis debe complementarse y los usuarios deben estar de acuerdo con los resultados. El diseño es el proceso de traducir los requerimientos definidos durante el análisis a varias alternativas de diseño para la consideración de los usuarios. También implica descubrir todas las fuerzas de diseño, para ver cómo impactarán e influirán en los diferentes diseños. Los usuarios pueden responder a las alternativas de diseño y empezar a trabajar hacia un consenso. Sin embargo, pueden ocurrir varias iteraciones entre el análisis y el diseño, en que cada iteración implique malabear y dar nueva forma a los componentes estructurales.

En la fase de diseño, los usuarios seleccionan dos o tres de los mejores diseños. Si se recopilaban y analizaban suficientes hechos del estudio en la fase de análisis, se reduce la probabilidad de tener que regresar a dicha fase. También se debe recalcar que durante el diseño los analistas concluyen la fase del diseño general con un entendimiento bastante claro de lo que se desea y de lo que tendrían que hacer para obtenerlo.

CAPITULO II. Bases de datos y Sistemas de Información.

Evaluación de Sistemas

La evaluación de sistemas, en realidad no es una fase discreta; en todo momento durante el trabajo en sistemas, se realiza una gran cantidad de evaluación, pero para resaltarla, se presenta como una fase separada. Esta fase implica la selección de la tecnología que soportará a los otros componentes estructurales, la evaluación de esta tecnología y los proveedores que la suministran, y un análisis completo de costo-eficacia de cada una de las alternativas propuestas del diseño de sistemas para determinar aquella con la mejor proporción de eficacia con relación al costo.

Diseño detallado de Sistemas

Es una descripción pormenorizada del sistema de información en forma verbal, con diagramas, gráficas y tablas, que son suficiente para que el que no está familiarizado con el proyecto pueda implantar el sistema propuesto. Se trata de una ampliación del diseño general, con las modificaciones necesarias.

Antes de iniciar este trabajo sustancia, los analistas deben asegurarse de que todas las partes hayan llegado a un consenso final en cuanto al sistema que desean que se implemente. Aquí a cada componente de construcción se le da una definición precisa y detallada. A decir verdad, sería un desperdicio excesivo de tiempo y dinero, por no mencionar la frustración y consternación, diseñar en detalle un sistema sólo para encontrar finalmente que nadie lo quiere y que no corresponde a las fuerzas del diseño.

Implantación de Sistemas

La implantación de sistemas es la capacitación y educación a los usuarios, la prueba y la conversión para hacer que el sistema sea operacional. Aquí es donde todo el trabajo de desarrollo y diseño llega a un clímax. Los analistas fijan las fechas límite o puntos de congelación, capacitan y coordinan al personal usuario, instruyen a los técnicos, prueban el nuevo sistema y eliminan errores, instalan nuevos procedimientos y formas, y ven si hay descuidos u omisiones.

CAPITULO II. Bases de datos y Sistemas de Información.

Poco después de su aceptación, el analista de sistemas deberá realizar un seguimiento para ver que el sistema está operando según lo esperado. En muchos casos los analistas pueden hacer algunos ajustes menores o afinaciones que resulten en mejoras significativas. Después de que el sistema haya estado operando durante cierto tiempo, las necesidades de los usuarios cambian y los analistas de sistemas se verán involucrados en la preparación de prototipos especiales para cubrir necesidades particulares.

En ocasiones será necesario efectuar algún trabajo de mantenimiento. Finalmente, el analista de sistemas se verá de nuevo fuertemente involucrado en un mantenimiento mayor del sistema, o en el desarrollo y diseño de uno nuevo, y entonces se repetirá el ciclo de vida del desarrollo de sistemas.

2.3 FASES DEL DESARROLLO DE SISTEMAS

Analicemos ahora, otro punto de vista en el desarrollo de sistemas.

ANALISIS

El objetivo de esta fase es determinar conjuntamente con el usuario las especificaciones funcionales del sistema, como son: Objetivos específicos del sistema, y actividades a desarrollar para que el sistema funcione adecuadamente.

Para ello, se analiza exhaustivamente cada uno de los elementos del sistema actual y su funcionamiento.

Esta etapa considera el estudio de las operaciones existentes para aprender lo que éstas llevan a cabo, por qué, trabajan de esa manera y que papel puede tener en las futuras actividades del procesamiento. Se identifican los datos y los procedimientos esenciales que se requieren para el nuevo enfoque.

CAPITULO II. Bases de datos y Sistemas de Información.

DISEÑO

Es el proceso de búsqueda de soluciones que satisfagan los objetivos perseguidos y determina cuál es la mejor. Esto se hace en base a la evaluación de las alternativas y a la graficación de las especificaciones para la opción seleccionada. Entre otras cosas, debe tenerse en cuenta: recursos financieros, recursos humanos, los diferentes procedimientos y el equipo disponible.

Se deben preparar nuevas especificaciones de diseño que incluyan la salida esperada, los datos necesarios de entrada y los procedimientos de procesamiento requeridos para convertir los datos de entrada en resultados de salida. Deben diseñarse formas, informes, archivos, diagramas de flujo del sistema y configurar el manual de procedimientos.

CONSTRUCCION Y PRUEBAS

Una vez rediseñados los procesos, y diseñados nuevos procesos que configuren el sistema, se procederá a desarrollarlos a través de los elementos de procesamiento de datos con los que se cuenta, y posteriormente se evaluará su funcionamiento, en base al manejo de datos ficticios para asegurar que los procedimientos no fallen. Para ello se pueden evaluar con datos históricos reales en presencia del usuario, lo cual permitirá realizar ajustes y correcciones en aquellos caso que así lo ameriten.

En caso de que se haya generado el nuevo sistema para sustituir a otro ya existente, se puede evaluar al mismo tiempo la eficacia del sistema anterior con el sistema actual y observar así las supuestas ventajas que se plantearan al diseñarlo. Este grupo de actividades se puede dividir en las siguientes etapas:

Programación: Esta etapa comprende la elaboración de los programas que cumplan con los objetivos del sistema en base a las especificaciones funcionales y de diseño proporcionadas por los analistas.

CAPITULO II. Bases de datos y Sistemas de Información.

Pruebas al sistema: Consiste en determinar la confiabilidad y la eficiencia del sistema desarrollado en la fase de programación. Para ello se realizan pruebas con datos ficticios y después con datos reales. Si el sistema sustituyera a uno anterior, entonces será recomendable realizar lo que se conoce como pruebas en paralelo.

Pruebas en paralelo: Este procedimiento es conveniente siempre que el sistema que se haya diseñado pretenda sustituir a uno existente, y consiste en poner en práctica los dos sistemas al mismo tiempo para comprobar los resultados y corregir errores.

Implantación: Se llega a esta etapa, cuando las pruebas realizadas al nuevo sistema son satisfactorias. Para ello debe entrenarse al personal que intervendrá en el sistema para que se adapte a los nuevos cambios, para lo cual, deberán desarrollarse manuales de uso del sistema y desarrollar cursos específicos de actualización. En esta etapa se procede a elaborar la información base del nuevo sistema y procesar los datos reales.

Documentación: Para que el usuario pueda interactuar con el nuevo sistema, es necesario que se disponga de documentos explicativos del funcionamiento y manejo del sistema. El analista encargado del diseño del sistema, debe entregar documentación sobre:

- a. Información histórica del sistema, planteamientos, diagramas de procesamiento, etc.
- b. Manuales de operación del sistema
- c. Instructivo de recolección de datos, de llenado de formas y de distribución.
- d. Carpeta de programas

CAPITULO II. Bases de datos y Sistemas de Información.

Mantenimiento: Durante la vida práctica de un sistema, pueden generarse ligeras modificaciones que implican una pequeña variación en el sistema actual a nivel de programas o procedimientos. Estos cambios deben ser aplicados en un tiempo breve, sin necesidad de reestructurar o modificar significativamente el sistema. Cuando esto ocurre, se dice que se está realizando un mantenimiento al sistema. Estos cambios, deberán registrarse en la documentación del sistema, para que ésta pueda estar actualizada y contenga información fehaciente sobre los elementos constitutivos del sistema.

Por lo tanto, podemos decir que el mantenimiento del sistema, es la etapa en donde se adapta al sistema actual a nuevas necesidades que no estaban contempladas en el momento de su concepción, pero que requieren de una nueva orientación total del sistema para su implantación.

NECESIDAD DE LA PLANEACION DE SISTEMAS

El sistema de información administrativa global consta de todas las actividades, equipo, personal e instalaciones que intervienen en generar información para dirigir a la organización en su conjunto. Podemos decir que dicho sistema se compone de un grupo integrado de subsistemas de información.

Sin embargo, tiene varias desventajas esta forma fragmentaria de realizar el sistema de información administrativa (MIS). Una de ellas se debe a la naturaleza inconexa de los subsistemas introducidos. Con frecuencia, existen sistemas individualistas que precinden de la forma en que interactúan con los sistemas en otras partes de la organización. Resultado de esto, ha sido la aparición de barreras de comunicación entre sistemas.

Otra gran desventaja del enfoque fragmentario, es el costo en cuanto al tiempo, los recursos y el dinero. Cuanto más tiempo se posponga un plan maestro, más caras resultarán las revisiones que se necesitarán para unificar y estandarizar el modo de aplicar el método al diseño de sistemas integrados.

De este modo, surgen las siguientes preguntas: ¿Por qué, se ha permitido un enfoque fragmentario?, ¿Qué, debe hacerse para mejorar el diseño de modo que pueda aplicarse un enfoque integrado y más satisfactorio?

CAPITULO II. Bases de datos y Sistemas de Información.

La respuesta a la primera pregunta es compleja, la razón principal, parece ser que no se toma en cuenta de que en la primera etapa del desarrollo de sistemas depende del alcance que tienen los sistemas computacionales y de información. Tampoco se ha comprendido la inversión que éstos representarán y el efecto que ejercerán sobre las operaciones de la organización.

En los últimos años, se ha advertido la necesidad de lograr la integración, realizando un plan maestro que coordine la implantación a corto y largo plazo del sistema de información administrativa.

La evidencia parece indicarnos que el desarrollo futuro de los sistemas se caracterizará por cinco tendencias:

1. La instalación del sistema de información administrativa empieza a ser reconocido como parte de la estrategia de la organización. Requiere la inversión de grandes cantidades de recursos humanos y financieros. Afecta profundamente a la organización y a la toma de decisiones. Asimismo, afecta la manera como la organización reacciona ante los cambios ambientales.
2. El reconocimiento de la interrelación de las decisiones en todo el ámbito de la organización culmina en un desarrollo mayor de modelos aplicables a ella en su totalidad.
3. Los adelantos tecnológicos del software y el hardware han hecho que cobre creciente importancia la base de datos, cuyo uso ha ido difundándose.
4. Una parte mayor de los costos de los sistemas de cómputo se destinará al diseño y al software, dedicándose una proporción menor al hardware.
5. La automatización de la toma de decisiones adquirirá más auge. Para que aumente dicha automatización se requieren más datos en todo el sistema.
6. Generalmente se acepta el control de las computadoras en el sistema de apoyo a las decisiones (DSS).

CAPITULO II. Bases de datos y Sistemas de Información.

La respuesta a la segunda pregunta es evidentemente, que deben definirse las políticas de la organización sobre la información, y preparar planes estratégicos, y a corto plazo. Existen siete razones fundamentales por las cuales se han de planear los sistemas de información:

1. Integrar los objetivos de los sistemas de información en los planes estratégicos, y a corto plazo.
2. Concentrarse en los objetivos del sistema de información administrativa.
3. Planear la asignación de recursos destinados a la introducción del sistema de información administrativa.
4. Garantizar un desarrollo eficiente y sistemático de los sistemas de información.
5. Sentar las bases para controlar los proyectos de los sistemas de información.
6. Equilibrar los costos y beneficios del sistema de información administrativa.
7. Aumentar la participación de los usuarios en el diseño de sistemas, con lo cual se evita la resistencia al cambio.

Los objetivos de los sistemas de información han de incorporarse en los planes de la organización para asegurarse de que los sistemas no sean descuidados ante la presión de las operaciones. Más aún, al incluirlos en los planes del sistema de información, en los estratégicos, y a corto plazo, la organización habrá de definir los sistemas futuros de información.

Al planear la instalación del sistema de información (MIS), se deben asignar de antemano los recursos. Con ello se asegura mejor que será introducido conforme vaya necesitándose. Este tipo de asignación permite un diseño y organización ordenados, evitándose así el diseño oportunista, no planeado y ascendente, lo mismo que una administración del mismo en circunstancias críticas.

CAPITULO II. Bases de datos y Sistemas de Información.

El costo de diseñar e instalar un sistema de información puede ser excesivo. Por consiguiente, son indispensables su planeación y control, la planeación de los objetivos técnicos y los presupuestos que sirven de criterios de control. Los beneficios siempre han de ser mayores que los costos, sin importar si se invierte mucho o poco en la instalación del sistema. La planeación requiere un análisis anticipado de costo/beneficio. Casi siempre se obtienen beneficios crecientes con las primeras cantidades invertidas y luego se merman los rendimientos a medida que crecen los gastos. La planeación obliga a asignar un valor a tales beneficios como mejoramiento de la planeación y control, mejoramiento de la toma de decisiones y un procesamiento de datos más automatizado, logrado con mayores gastos que se destinan al sistema de información.

OBJETIVOS DE LA PLANEACION CON EL SISTEMA DE INFORMACION

La planeación exige buscar y seleccionar, entre diversas alternativas, los cursos de acciones necesarios para alcanzar un objetivos. La planeación puede ser útil y debe comenzar sólo cuando los objetivos hayan sido escogidos correctamente. De ahí que no pueda efectuarse conforme a un plan maestro, a menos que los objetivos de la misma están detallados y se conozcan bien.

Nos referimos no a los objetivos específicos, sino a los objetivos globales de la planeación del sistema; en otras palabras, a las características de los sistemas de información que se introducirán a corto y a largo plazo.

La función de planeación de sistemas han de abarcar por eso el examen de los objetivos propuestos en función de los criterios de planeación cuyo fin es reducir al mínimo el número de sistemas, ampliar su alcance y colocarlos en el orden debido de desarrollo. Todas estas condiciones pueden expresarse con la siguiente lista de objetivos de la planeación de sistemas:

1. Evitar el traslape en el desarrollo de los principales elementos de los sistemas de gran aplicación en las líneas organizacionales, cuando no existen fuertes razones técnicas ni funcionales para una diferencia.

2. Ayudar a asegurar una base uniforme para determinar el orden de desarrollo en términos de potencial de ganancia, precedente natural y probabilidad de éxito.

CAPITULO II. Bases de datos y Sistemas de Información.

3. Minimizar el costo de integrar entre sistemas semejantes.
4. Reducir el número total de sistemas pequeños y aislados que es preciso instalar, operar y darles mantenimiento.
5. Asegurar la adaptabilidad de los sistemas al cambio de la organización y a su crecimiento sin grandes evaluaciones periódicos.
6. Pone las bases de un desarrollo coordinado de sistemas de información congruentes, generales, inter-organizacionales y que abarcan toda la organización.
7. Dar pautas y dirección para estudios y proyectos constantes de lo desarrollo de sistemas.

Conviene recalcar que los objetivos centrales de la planeación del sistema de información son asegurar:

- a) que el sistemas de información forme parte del plan estratégico, de la organización.
- b) que la asignación de recursos al sistema figure en el plan estratégico,.

PLANEACION ESTRATEGICA DEL SISTEMA DE INFORMACION

El plan del sistema de información ha de formar parte del plan estratégico, administrativo. Cuando se trata de un enfoque pasivo, el plan administrativo se elabora primero y luego el sistema que satisface las necesidades resultantes. Sin duda alguna es el método más fácil, aunque no el mejor. En el enfoque productivo, el sistema de información influye en el proceso de decisión y en la estructura de la organización. Se busca un buen sistema para lograr una ventaja competitiva más amplia, una planeación y mercadotecnia más eficaces del producto y una mejor utilización de los recursos humanos y financieros. Desde este punto de vista, la estrategia del sistema de información se elabora y se integra con la estrategia total de la organización.

CAPITULO II. Bases de datos y Sistemas de Información.

PASOS EN LA ELABORACION DEL PLAN ESTRATEGICO DEL SISTEMA DE INFORMACION

La elaboración del plan estratégico, del sistema de información requiere especificar los objetivos, las metas que deben cumplirse para alcanzarlos y los recursos necesarios. Para ello deberán seguirse los siguientes pasos:

- a. Examinar el plan a largo plazo y determinar su naturaleza.
- b. Estudiar el ambiente de la organización y los cambios tecnológicos que se prevean en los futuros sistemas de información.
- c. Estudiar la posición estratégica, de la organización; es decir, su ámbito actual, su estructura, su ventaja competitiva, su posición financiera, su nivel de riesgo en las nuevas actividades y sus políticas.
- d. Una vez que se ha examinado el ambiente interno y externo de la organización, se determinará el entorno en el que el sistema de información cumplirá su cometido.
- e. Definir las restricciones y especificar las políticas que se seguirán en la instalación del sistema de información durante la planeación.
- f. Establecer los objetivos estratégicos, del sistema de información, es decir, De que deberá ser capaz el sistema de información ?.
- g. Delinear el actual sistema de información e identificar provisionalmente los subsistemas que necesitan revisión con mayor urgencia.

CAPITULO II. Bases de datos y Sistemas de Información.

h. Descubrir nuevas aplicaciones e innovaciones que pueden aportar grandes beneficios.

i. Desarrollar nuevos conceptos alternos del sistema total de información. En otras palabras, se esbozan los posibles arreglos del sistema y se describen las principales características, como salidas, hardware y software, organización de las operaciones y capacidades de ayuda a las decisiones.

RESUMEN.

Toda organización almacena datos. Si cuenta con un sistema de información administrativa (MIS), seguramente tendrá una base de datos controlada en forma centralizada. La administración de datos dentro de la organización es una responsabilidad muy importante..

Los objetivos primarios del Sistema de Administración de base de datos (DBMS) son:

1. Controlar la redundancia.
2. Aumentar la independencia de datos.
3. Aumentar la velocidad y flexibilidad de acceso a la base de datos.
4. Reducir el costo del acceso.

5. Incrementar la compatibilidad de los datos.

6. Mejorar la integridad y seguridad de la base de datos.

Las bases de datos están diseñadas, primero, para ajustarse a las necesidades de los usuarios, y, segundo, para que correspondan al punto de vista del especialista en procesamiento de datos. De lo anterior se deduce, que la administración de datos es de gran interés.

CAPITULO III.

ANALISIS DEL PROCESO DE INFORMACION DE ISCAR DE MEXICO, S.A. DE C.V.

Objetivo:

Mostrar cómo el sistema de información administrativa interviene en la solución de problemas y en la toma de decisiones.

INTRODUCCION.

Los problemas y las decisiones se nos presentan en todos los ámbitos de la existencia, pero ¿Cómo sabemos cuando tenemos un problema?, ¿Cómo se relacionan las decisiones con los problemas?, ¿Qué tipo de información nos ayudará realmente a llegar a una solución final?

Por lo regular, un problema advierte muchas respuestas, algunas mejores que otras. La elección entre esas alternativas, es una toma de decisiones. Las decisiones más importantes, se presentan a lo largo del proceso de solución de problemas, a menudo sin darnos cuenta de ello.

Una idea importante, es que los objetivos se fundan principalmente en juicios de valor; no son el resultado de un proceso racional repetible. Además la selección de los criterios que se aplicarán a la solución del problema, son asimismo, un proceso basado en valores.

3.1 FUNCIONES PRINCIPALES DE ISCAR DE MEXICO, S.A. DE C.V.

El departamento de Sistemas y logística de Iscar de México es el encargado de atender los requerimientos administrativos y contables de la empresa; dichos procesos son: facturación, inventarios y cuentas por cobrar.

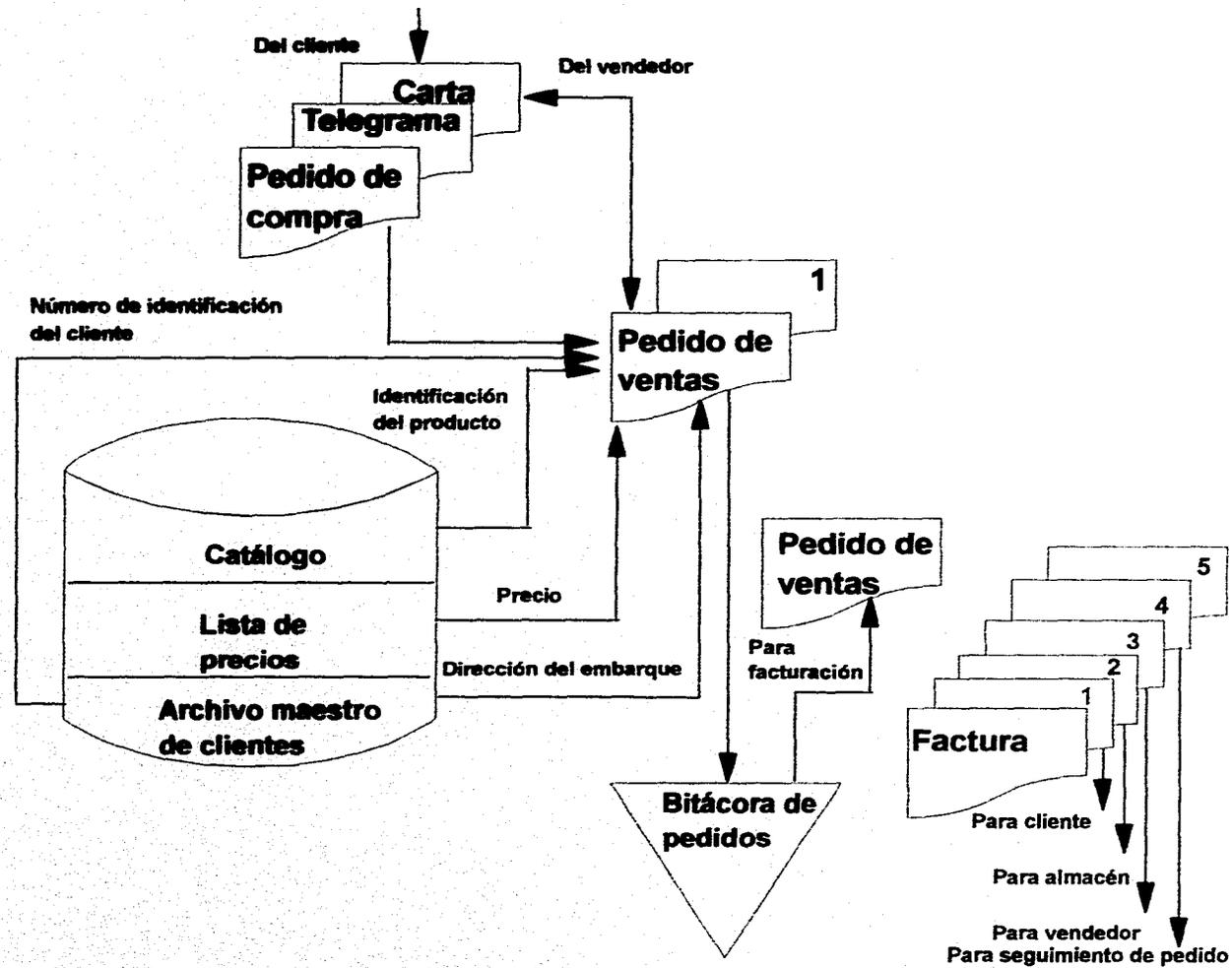
3.1.1. Proceso de Facturación.

La facturación es el proceso de mayor importancia. Sus objetivos fundamentales son asegurarse de que las órdenes se llenen en forma correcta y oportuna, notificando a los clientes y a la gerencia el estado y los retrasos. Aunque en sentido estricto se define como un sistema de mercadotecnia, en términos más generales puede decirse que está integrado por muchos elementos de la compañía (producción, embarque, contabilidad, etc.) y que participa en el surtido de los pedidos.

Esta información debe ser preparada en una forma estándar: los pedidos de venta de la compañía. Las facturas, modificadas según las circunstancias, se usan para avisar a la bodega que envíe los productos al departamento de embarque y que haga la factura al cliente. Cuando se embarcan (o no se embarcan), el grupo de seguimiento de pedidos y los vendedores, reciben una copia con comentarios de la factura (orden de embarque).

3.1.2 Proceso de Inventarios

Debido a la cantidad invertida y a la susceptibilidad de los artículos a ser robados y a registrarse indebidamente, el inventario es uno de los activos más significativos para muchas compañías. Este módulo mantiene los detalles completos de los artículos que salen y de los que son transferidos. La consulta del estado de las existencias permite a los usuarios conocer el inventario que se tiene, y cuál es su costo. Periódicamente se cuenta el inventario y se concilia con las cantidades que aparecen en los registros. Todas las variaciones se reportan para su investigación y balanceo.



CAPITULO III. Análisis del proceso de Información de Iscar de México, S.A. de C.V.

3.1.3. Proceso del Sistema Contable

Las cuentas por cobrar se originan con la venta de mercancías o la prestación de servicios a crédito y representan demandas contra los deudores, quienes generalmente son clientes. Es en el módulo de cuentas por cobrar en donde se registran y se da seguimiento a todas las cuentas de los clientes.

Cuando se realizan ventas a crédito, se asientan diversos datos en las cuentas apropiadas de los clientes. El módulo de cuentas por cobrar procesa facturas, mantiene una historia de ventas por cada cliente, e identifica a los clientes con cuentas vencidas. El módulo de cuentas por cobrar es central para la generación de facturas de los clientes, sus estados, cartas de cobranza y etiquetas de correspondencia.

3.2 Descripción de las funciones del sistema

Un sistema administrativo suele diseñarse a través de los siguientes pasos en el orden señalada:

- 1. Establecer la misión y los objetivos de la organización.**
- 2. Desarrollar subsistemas para resolver los problemas que entraña la consecución de los objetivos.**
- 3. Idear subsistemas del proceso de trabajo que especifiquen las actividades y su secuencia para resolver los problemas actuales.**
- 4. Desarrollar la entrada, la salida y la información intermedia que se requiera. La estructuración de las organizaciones se propone facilitar la solución de problemas, la toma de decisiones y su cumplimiento. Los patrones organizacionales les permiten agrupar a los que tengan habilidades semejantes u objetivos comunes, delegar la autoridad y la responsabilidad, e integrar los sistemas de decisión basados en la información.**

CAPITULO III. Análisis del proceso de Información de Iscar de México, S.A. de C.V.

El diseño del trabajo gira en torno a sistemas y subsistemas, pero los miembros de las empresas suelen agruparse conforme a sus especialidades. Una de las principales dificultades que afrontan las grandes empresas radica en que su estructura no es congruente con los subsistemas del proceso de trabajo. Un claro ejemplo de ello, lo encontramos en el sistema de procesamiento de pedidos.

Sus objetivos consisten en aceptar los pedidos de clientes y vendedores, hacer el seguimiento correspondiente para asegurarse de que sean surtidos con exactitud y prontitud y notificar los problemas a los clientes. Para que el sistema opere es preciso coordinar a los empleados de los departamentos de mercadotecnia, bodega, embarque y contabilidad.

El sistema de información administrativa nace de las necesidades de información de los gerentes y se basa en la captación de datos internos y externos. Los primeros son generados por los subsistemas de información que apoyan diversos sistemas de procesos.

Para facilitar el análisis, podemos clasificar los sistemas administrativos en tres grupos:

1. Sistemas administrativos generales

2. Sistemas administrativos de línea:

a. Mercadotecnia

b. Ingeniería

c. Producción

3. Sistemas administrativos de apoyo

- a. Recursos humanos
- b. Finanzas/contabilidad
- c. Administrativos

El departamento de Sistemas es el encargado de realizar el proceso de facturación, capturar los pedidos de los clientes, asignar el material que se tenga en existencia. Después de generar esta información, se procede a hacer la factura correspondiente, para generar los elementos necesarios para su análisis contable.

Este sistema puede resultar complejo en extremo dado que combina varias funciones de mercadotecnia, datos de costos y una amplia gama de informaciones. En los negocios, todos los planes principian con el pronóstico de ventas de la compañía. Y el pronóstico se basa en la ventaja competitiva de ella, el precio que fija, sus esfuerzos promocionales, las preferencias del cliente y el nivel de servicios que ofrece.

3.3 Solución a la problemática del procesamiento de la Información

Los factores más importantes en cuanto a carga de trabajo y control de la información manejada tanto en la manera de procesarse como en la cantidad que se maneja, se encuentra básicamente en el proceso de facturación de la empresa y en el proceso de cuentas por cobrar de los clientes.

CAPITULO III. Análisis del proceso de Información de Iscar de México, S.A. de C.V.

Una de las etapas fundamentales en el desarrollo de sistemas, es enfocar y entender el problema para el cual se busca la solución. En esta etapa se debe tener mucho cuidado, debido a que las solicitudes de los usuarios, no siempre expresan las necesidades reales, y es al final cuando se empiezan a dar cuenta de cuales son sus verdaderas necesidades. Esta etapa es parte esencial del desarrollo de sistemas, por ello es imprescindible mantener comunicación constante con los usuarios finales para dar una propuesta de solución que pueda satisfacer las necesidades requeridas.

La mejor forma de analizar esta problemática es considerar todos los casos posibles que se puedan presentar para poder determinar las necesidades del usuario. Para ello es necesario considerar:

- i) Identificar a las personas que estarán involucradas con el sistema y su relación organizacional dentro de la empresa.
- ii) Tener presente los recursos existentes con los que se cuenta, y determinar si será necesario la adquisición de nuevo equipo, así como la posibilidad de adquirirlo. En su defecto, adaptarse para poder operar de la mejor manera posible.
- iii) Para cumplir satisfactoriamente nuestro objetivo, se deberán identificar las áreas involucradas en el uso y distribución de la información que se necesite.

3.3.1. Requerimientos del Sistema

Los puntos propuestos están en base a los requerimientos de las personas encargadas del proceso de información de la compañía. Al tener trato directo con el personal involucrado directamente con todos los procedimientos se pudo obtener información confiable de los requerimientos de información para la automatización y además conocer a fondo todo el seguimiento que debe realizarse con la información generada por cada una de las transacciones de la compañía, incluyendo aquellos procedimientos manuales involucrados en ellos.

CAPITULO III. Análisis del proceso de Información de Iscar de México, S.A. de C.V.

Con la participación del usuario directo se logró la descripción de la problemática general, así como las consecuencias que estos problemas traerán en el desarrollo de sus actividades.

Con la información obtenida, se establecieron los puntos básicos para la creación del sistema, que de acuerdo con las necesidades debe tener las siguientes características:

- a. Poder contar con la información necesaria concentrada en una computadora personal, de tal forma que sea posible realizar alguna consulta de una forma rápida y eficaz.
- b. El acceso a la información debe realizarse de manera sencilla mediante el uso de campos clave.
- c. Llevar el control de las personas que tienen acceso al sistema, e identificar por quién fue atendido el cliente.
- d. Ser capaz de generar reportes con información de la situación que guardan las operaciones de la compañía.

Como puede verse el interés común entre el usuario y el analista de sistemas, es contar con una herramienta que les brinde información confiable y actualizada para atender eficientemente a los clientes y que simplifique el proceso de información, así como su manipulación.

3.3.2 Propuesta de Solución

Dado que ya fueron identificados los problemas, y con base en el análisis de las necesidades del usuario, podemos establecer los requerimientos del sistema. La propuesta de solución es:

1. Diseñar un sistema para el manejo de información que este controlado mediante el número de cliente como campo clave.

CAPITULO III. Análisis del proceso de Información de Iscar de México, S.A. de C.V.

2. El sistema será capaz de dar facilidad para añadir, editar o borrar registros de tal forma que la información quede libre de errores o en su defecto se encuentren datos duplicados.

3. Al término de las transacciones diarias, se generarán reportes con el estado actual de los movimientos del sistemas

4. Al final del día el sistema contará con una opción para respaldar la información que se genera diariamente a fin de evita pérdidas valiosas de información.

Las bases de datos a utilizar contendrán la información precisa para que la implantación del sistema sea adecuada. Dichas bases contienen:

1. Información general de los clientes de la empresa (número de cliente, razón social, dirección, condiciones de pago, etc.).

2. Información completa de todos los productos que maneja en stock la empresa.

3. Los datos de los empleados de la empresa.

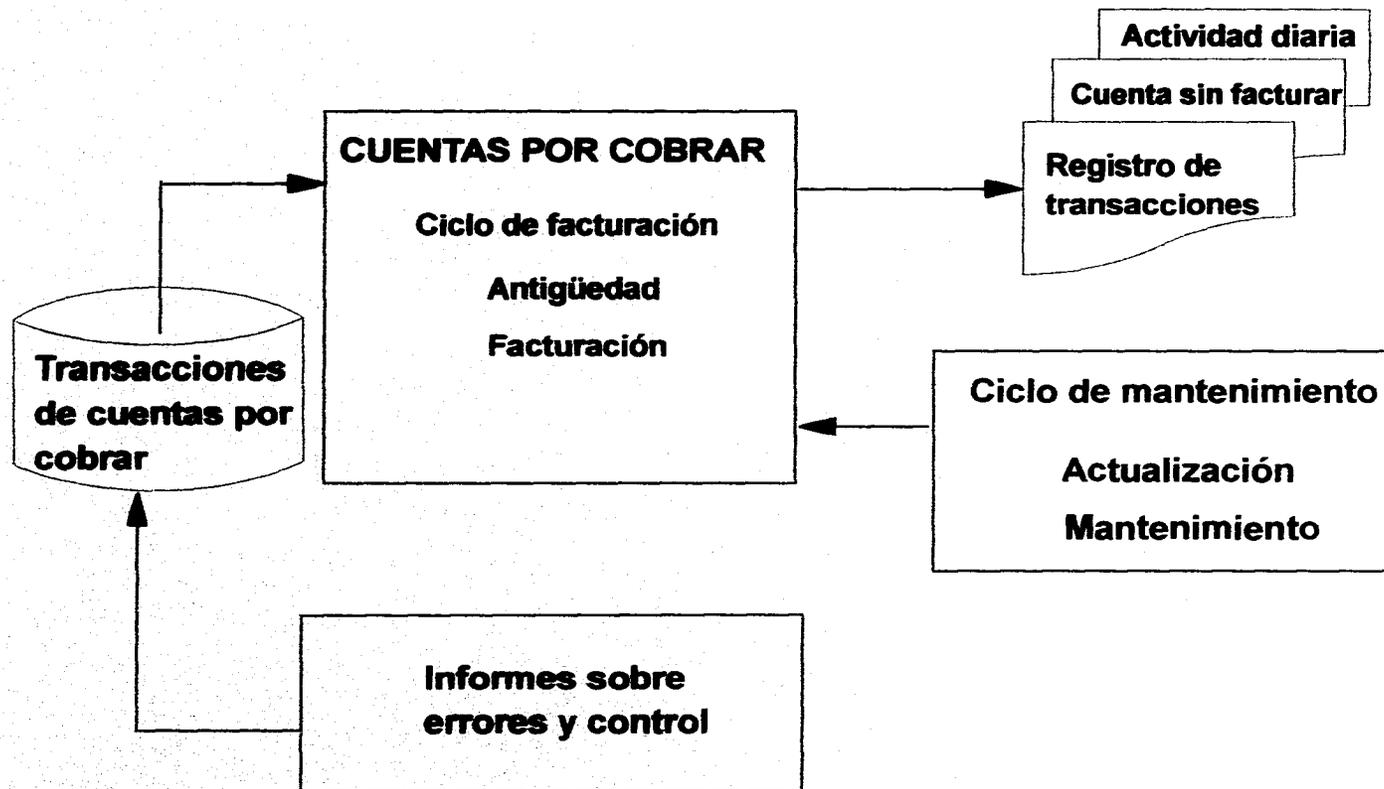
4. Los datos de los proveedores de la empresa.

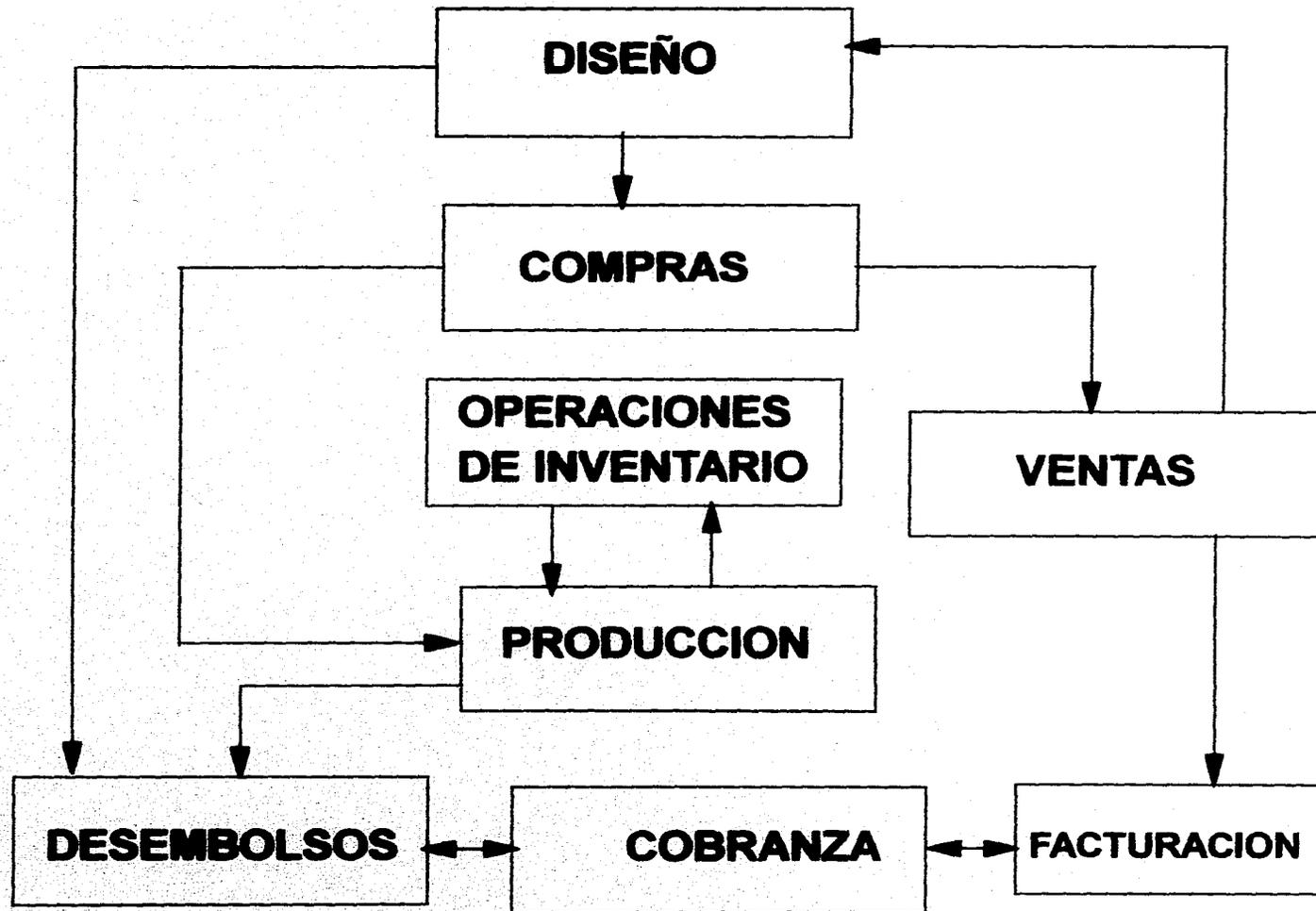
5. La información completa de cada una de las órdenes de compra de los clientes.

6. La información de la facturación generada diariamente en la empresa, para llevar el control de cuentas por cobrar.

REPRESENTACION DEL FLUJO DE INFORMACION DEL PROCESO CONTABLE

INFORMES SOBRE ADMINISTRACION Y FACTURACION





RESUMEN

La solución de problemas, es un proceso creativo encaminado a resolver una situación insatisfactoria. El problema se reconoce cuando una condición deseada, no corresponde a la actualmente existente, ó a la que se prevé. La fuente de problemas, es el cambio. Un problema puede originarse del cambio en los objetivos o en el ambiente. El enfoque de sistemas aplicado a la solución de problema, puede conceptualizarse a partir de tres niveles: ¿ cómo generar información?, ¿ cómo servirse de ella ?, y la aplicación que tiene. La toma de decisiones pone fin al análisis de diversos cursos de acción, al hacer una elección. los factores que configuran el proceso de decisión son: 1). tiempo, 2). costo, 3). análisis técnico, 4). características psicológicas, 5). factores organizacionales del comportamiento y 6). factores culturales. La necesidad del sistema de información administrativa, deriva de la dificultad de resolver problemas y de tomar decisiones.

CAPITULO IV.

DISEÑO E IMPLEMENTACION DEL SISTEMA

Objetivo:

Explicar el análisis y el trabajo creativo que se requieren para desarrollar el concepto general del Sistema de Información Administrativa.

CAPITULO IV. Diseño e Implementación del Sistema.

INTRODUCCION.

El diseño conceptual del sistema de información administrativa (MIS), esboza los principales componentes y sus relaciones dentro del sistema. El diseño detallado es una ampliación y extensión de las características generales del diseño que se incluyeron en el diseño conceptual. El diseño detallado del sistema de información administrativa se compone de especificaciones concernientes a todo lo relativo al sistema.

Diseñar un sistema requiere de muchas actividades simultáneas que interactúan entre sí. No existe una clara separación entre el fin del diseño conceptual y el inicio del diseño detallado. Cada diseñador supondrá la existencia de una situación distinta. No es posible prever y describir las reacciones organizacionales que pueden tener lugar en determinadas circunstancias. Por tanto, únicamente se puede presentar el enfoque técnico de la situación general. En la práctica, se requerirán modificaciones a medida que se desarrolle el diseño detallado.

CAPITULO IV. Diseño e Implementación del Sistema.

JUSTIFICACION DEL SOFTWARE

Existen algunos manejadores de bases de datos como son :

Dbase III Plus

Dbase IV

Foxbase

Foxpro

Clipper

Los cuatro primeros se caracterizan porque traducen las instrucciones a la computadora línea a línea y aunque poseen una gran versatilidad para realizar los programas, la velocidad de ejecución suele ser muy lenta al estar obligado a traducir simultáneamente dichas instrucciones.

El Clipper es un compilador que convierte el lenguaje interprete en un lenguaje de maquina, dotado de una gran rapidez de ejecución, ya que analiza las instrucciones de una sola vez y con absoluta autonomía.

De los manejadores de bases de datos anteriores, y a pesar de las ventajas que ofrece Clipper con respecto al Dbase IV, se eligió el Dbase IV para los fines que a la empresa convinieron, ya que se tiene proyectado desarrollar el sistema de la empresa con software utilizado en la Matriz ubicada en Israel y un cambio en el equipo de computo utilizado actualmente.

CAPITULO IV. Diseño e Implementación del Sistema.

Quizás el manejador seleccionado no sea el mas óptimos, pero fue la decisión tomada por el Departamento de Computo de Iscar LTD en Israel. Por tal razón se tuvo que desarrollar este sistema piloto en Dbase IV, por la compatibilidad con el software que se empleara en el desarrollo del nuevo sistema.

4.2 DISEÑO DEL SISTEMA

Una vez que se han delineado los problemas a los que se tiene que enfrentar el usuario, estos fueron analizados punto por punto. Para presentar la solución mas adecuada de acuerdo a las necesidades del proceso y al equipo con que se cuenta en la empresa. De esta manera realizar el diseño del sistema que solucionara los problemas delimitados durante la etapa de investigación .

Es en esta etapa cuando se realiza el diseño formal del sistema que será utilizado por el usuario final. Aquí se desarrolla la opción mas conveniente de las propuestas anteriores; una vez elegida esta opción se realizan las pruebas correspondientes para garantizar que el sistema trabaje adecuadamente y se cumplan los objetivos para los cuales fue diseñado. Si las pruebas no fueron satisfactorias se realizaran las modificaciones correspondientes para su implantación final.

4.2.1 EXPLICACION MODULAR DEL SISTEMA

A continuación se ilustrara mediante diagrama de bloques, la estructura modular del sistema. En cada diagrama se explica lo que ocurre en cada modulo del sistema, en donde cada cuadro representa un conjunto de instrucciones o función, las flechas indican el flujo de ejecución. En caso de existir alguna condición, posiblemente existan algunos módulos que contengan submódulos entre si.

CAPITULO IV. Diseño e Implementación del Sistema.

4.2.1. EXPLICACION DEL MODULO DE FACTURACION

El programa del sistema de facturación (ORDERS.PRG) inicia con la conexión al procedimiento externo (LIBRARY.PRG), programa que contiene los procedimientos que le permitirán al usuario realizar los movimientos que desee. Después inicia con la declaración de variables globales que utilizan durante la ejecución de todo el programa.

El primer procedimiento a ejecutar es Load_fid (contenido en el programa ORDERS), el cual tiene como función copiar el contenido de los campos de la base de datos ORDERS.DBF a las variables de memoria utilizadas.

Dstatus() (procedimiento de LIBRARY) checa el estado de la base de datos y muestra en pantalla el numero de registro actual.

Backgrnd() es el procedimiento del programa encargada de desplegar la información que se le requerirá al cliente para cargar su orden de compra.(Pertenece a ORDERS.PRG)

Show_data() el módulo o encargado de mostrar la información en pantalla del registro actual de la base de datos y de limpiar la misma en el caso de que se añada un nuevo registro a la base de datos.(contenida en Orders.prg)

Bar_def() (procedimiento de LIBRARY.PRG) Tiene como función mostrar en pantalla las opciones que podrá seleccionar el usuario para realizar sus movimientos.

Una vez que el usuario ha elegido lo que desea realizar se ejecutara el procedimiento Bar_pop (LIBRARY.PRG) que desarrollara la acción seleccionada por el usuario del menú de opciones que describiré a continuación.

CAPITULO IV. Diseño e Implementación del Sistema.

Add_new() (LIBRARY.PRG)

Tiene como función añadir un nuevo registro a la base de datos, suprimiendo la información del registro anterior en la pantalla. Si presiona la tecla F9 de búsqueda se ejecutara Sho_look() (LIBRARY.PRG) la cual tendrá como función mostrar en pantalla los registros de la base de datos si están disponibles.

El modulo siguiente Init_fid () (ORDERS.PRG) que inicializara los valores de las variables de memoria para la entrada de datos. Get_data() (ORDERS.PRG) muestra la pantalla para la captura de la información que introducirá el usuario. Si no da entrada de datos regresa el menú de opciones. En cada aplicación chequea que la entrada de datos sea correcta, en caso de no ser así mandara un mensaje de error y regresara al menú.

En el modo añadir limpia el área de campos de datos relacionados para su captura. Una vez que el usuario introdujo la información se ejecutara el procedimiento Sav_data() (LIBRARY.PRG) que se encargara de salvar el nuevo registro en la base datos, y el modulo Repl_fid (ORDERS.PRG) reemplazara el contenido de las variables de memoria en cada campo de la base de datos. Una vez finalizado este modulo regresara al menú de opciones.

Edit() (LIBRARY.PRG)

Tiene como función editar el registro actual. Si desea modificar el registro el programa ejecutara los mismos procedimientos que Add_new(), salvando la información modificada, para regresar al menú de opciones.

CAPITULO IV. Diseño e Implementación del Sistema.

Eraser() (LIBRARY.PRG)

Borra el registro actual de la base de datos, no sin antes preguntar al usuario si esta seguro de hacerlo. Si su respuesta es si el programa ejecutara la instrucción DELETE y saltara al registro siguiente de la base y ejecutara el procedimiento Sub_ret (LIBRARY.PRG) que se encargara de empaquetar los registros borrados y eliminarlos completamente de la base de datos.

Skip_rec() (LIBRARY.PRG)

Permitirá al usuario brincar registros hacia adelante o hacia atrás en uno o mas registros dependiendo de su elección en el orden de la base de datos indexada. Si desea brincar el primer o ultimo registro se ejecutaran las instrucciones TOP Y BOTTOM respectivamente, para ubicar el puntero al principio o al final del archivo

Find_rec() (LIBRARY.PRG)

Tiene como función hallar/buscar y mostrar datos del registro después de recuperarlo, concatenando cada una de las llaves proporcionadas por el usuario para su búsqueda, auxiliándose de las llaves declaradas en el ORDERS.PRG. Si la búsqueda concluyo satisfactoriamente mostrara el registro en la pantalla, en caso contrario mandara un mensaje de error y le pedirá que lo intente nuevamente.

List_rec() (LIBRARY.PRG)

Tiene como función mostrar en pantalla una lista de registro a partir del registro actual, si eligió un determinado grupo de registros solamente serán listados los que estén incluidos en este rango. Posteriormente regresara al registro actual.

CAPITULO IV. Diseño e Implementación del Sistema.

Barpop_d (LIBRARY.PRG)

Desarrolla la acción seleccionada por el usuario del menú destino Si selecciona la primera opción (Printer), la salida será a impresora y se auxiliara del modulo Prt_menu (LIBRARY.PRG) el cual activara el menú de opciones para impresión y posteriormente ejecutara el procedimiento Printout que seleccionara el reporte de salida dependiendo de la base de datos utilizada por cada programa.

Si elige la segunda opción (File) la salida será a un archivo específico dado por el usuario y se ejecutara el modulo Printout. Si se selecciona la tercera opción (Screen) se mostrara la información en pantalla (modulo Printout) y regresara al registro original. En los tres casos se ejecutara el modulo Barpop_r que se encargara de seleccionar el menú de reportes disponibles.

Kount() (LIBRARY.PRG)

Tendrá como función contar los registros existente en la base de datos y mostrarlo en pantalla, regresando al registro original antes de empezar la cuenta.

Indexer (ORDERS.PRG)

Crea y reconstruye el archivo indexado de la base de datos.

Helper.prg

Es un programa auxiliar que mostrara en pantalla de forma sintetizada la finalidad de cada opción del menú, para que el usuario se familiarice con el sistema.

CAPITULO IV. Diseño e Implementación del Sistema.

Para concluir el proceso de facturación, una vez que se ha capturado la orden de compra del cliente (ORDERS.PRG), el usuario tendrá que regresar al menú principal y seleccionar la opción PRINT INVOICES (INVOICES.PRG), en el cual se establecerán las condiciones de impresión y las relaciones entre bases de datos (ORDERS, GOODS, CUST, ACCT_REC), encabezados y pies de pagina, así como los cálculos para la impresión de la factura.

4.2.1.2 EXPLICACION MODULAR DEL MODULO DE INVENTARIOS

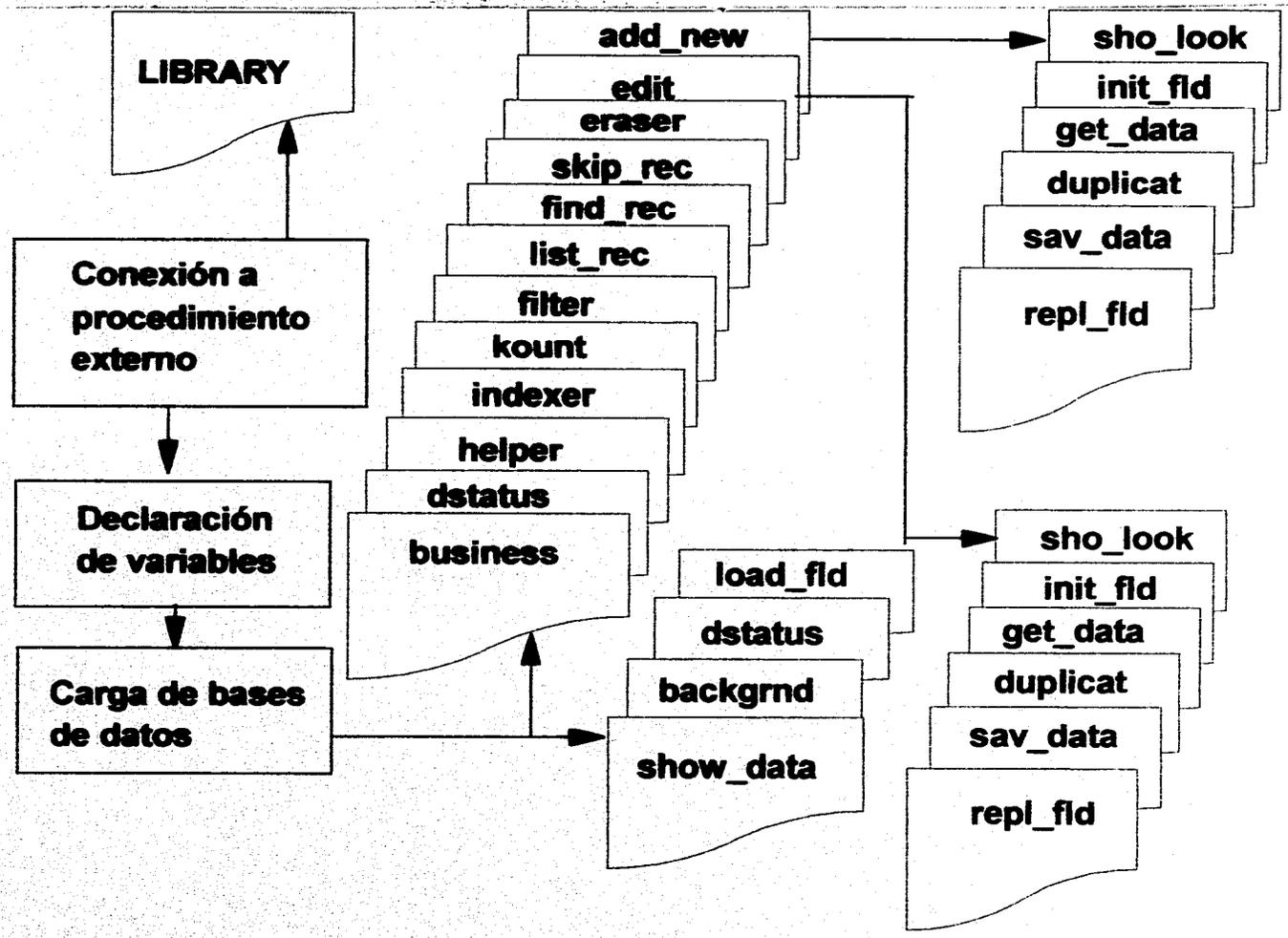
Todos los módulos que se explicaron en el punto anterior referente al modulo de facturación forman parte del modulo de inventarios, lo único que cambiara son los módulos pertenecientes a ORDERS.PRG, que cambiaran ahora por los correspondientes a GOODS.PRG, y la base de datos cambiara a GOODS.DBF, por lo que respecta al procedimiento externo LIBRARY.PRG, estará ligado de igual forma que el anterior punto.

4.2.1.3. EXPLICACION MODULAR DEL MODULO DE CUENTAS POR COBRAR

Los tres módulos fueron diseñados bajo el mismo concepto, razón por la cual, todos los módulos explicados anteriormente, también son aplicados en este, en consecuencia el programa en este caso es ACCT_REC.PRG y la base de datos correspondiente es ACCT_REC.DBF.

4.2.2. DISEÑO ESTRUCTURAL DE LAS BASES DE DATOS

Las bases de datos que se van a describir a continuación son las bases de datos de las que hace uso el sistema, ya sea para obtener información o para almacenarla. Se procuró que sólo tuvieran la información necesaria para el buen funcionamiento del sistema y de esta forma evitar el manejo de grandes volúmenes de información.



CAPITULO IV. Diseño e Implementación del Sistema.

A continuación se describirá la información que contiene y la estructura de las bases de datos.

A.EMPLOYEE.DBF

En esta base se almacena la información referente al Directorio de empleados. La siguiente es la estructura de la base de datos mostrando nombres de los campos, tipo y ancho.

Nombre del campo	Tipo	Ancho
lastname	caracter	15
firstname	caracter	10
initial	caracter	1
department	caracter	15
emp_id	caracter	11
phone	caracter	13
specialty	caracter	11
comments	caracter	40
awards	caracter	15
date_hired	date	8
full_time	logical	1
rate	numeric	4
salary	numeric	9
city	caracter	14
state	caracter	2

CAPITULO IV. Diseño e Implementación del Sistema.

B. CUST.DBF

Esta base de datos contiene información referente a todos los clientes de la empresa que conforma la cartera de la misma.

La estructura de la base de datos es la siguiente:

Nombre del campo	Tipo	Ancho
Cust_id	caracter	6
category	caracter	15
customer	caracter	30
address1	caracter	25
address2	caracter	20
city	caracter	20
state	caracter	2
zip	caracter	10
phone	caracter	13
terms	caracter	10
comments	caracter	20
contac	caracter	20
phone_cont	caracter	13
phone_ext	caracter	4
date_last	date	8

CAPITULO IV. Diseño e Implementación del Sistema.

C.VENDORS.DBF

Esta base de datos contiene la información referente a los distribuidores la cual contiene los datos correspondientes al mismo del vendedor. A continuación se muestra la estructura de la base de datos.

Nombre del campo	Tipo	Ancho
vendor_id	caracter	4
vendor	caracter	30
address1	caracter	30
address2	caracter	30
city	caracter	20
state	caracter	2
zip	caracter	10
phone	caracter	13
contact	caracter	30
phone_ext	caracter	4
terms	caracter	10
discount	numeric	2

CAPITULO IV. Diseño e Implementación del Sistema.

D. GOODS.DBF

Esta base de datos contiene toda la información referente a los productos que son manejados en la empresa, como son código de producto, descripción, precio, cantidad en existencia, entre otros, mismos que serán utilizados durante la ejecución del programa ORDERS.PRG.

La estructura de la base de datos es la siguiente:

Nombre del campo	Tipo	Ancho
part_id	caracter	10
date_order	date	8
part_name	caracter	30
descript	caracter	30
lead_time	numeric	3
price	numeric	8
qty_2order	numeric	4
qty_onhand	numeric	4
vendor_id	caracter	4
discontinuo	logical	1
cost	numeric	8
comments	caracter	30

CAPITULO IV. Diseño e Implementación del Sistema.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

E. ORDERS.DBF

En esta base de datos se almacenará la información correspondiente a cada una de las órdenes de compra de nuestros clientes en un registro individual. La información que contendrá cada registro será número de cliente, fecha de requisición, código del producto, cantidad, número de pedido, número de empleado.

La estructura de la base de datos es la siguiente:

EL campo INVOICED es muy importante pues de él depende si se factura ó no la orden de compra del cliente. Por default siempre estará en valor lógico T (verdadero).

Nombre del campo	Tipo	Ancho
cust_id	caracter	6
date_trans	date	8
part_id	caracter	10
part_qty	numeric	3
po_number	caracter	5
notes	memo	10
emp_id	caracter	11
invoiced	logical	1

CAPITULO IV. Diseño e Implementación del Sistema.

F. ACCT_REC.DBF

Esta base de datos contendrá información sobre el estado de cuenta actual de los clientes, tales como saldo, importe de la factura, número, antigüedad, último pago, y datos de la factura a pagar (número y fecha de factura).

La estructura de la base de datos es la siguiente:

Algunos de los campos de esta base de datos están relacionados con la bases de datos CUST.DBF. Como puede observarse de éstas estructuras, las bases de datos están relacionadas dependiendo del programa que las utilice.

Nombre del campo	Tipo	Ancho
cust_id	caracter	6
invoice_no	caracter	10
dat_of_bil	date	8
amt_1stbil	numeric	9
amt_1st_pd	numeric	9
oldbalance	numeric	9
amt_of_bil	numeric	9
amt_of_cur	numeric	9
amt_cur_pd	numeric	9
comments	caracter	30
dat_1stbil	date	8
invoic_old	caracter	10
notes	caracter	30

4.3 IMPLANTACION DEL SISTEMA

Diseñar un sistema de información administrativa puede ser un proyecto largo y costoso. Convertir los conceptos del diseño en un sistema funcional, puede incluso ser aun mas difícil. Esta conversión suele recibir el nombre de implantación del diseño. Esto significa realizar un cambio organizacional y técnico. A menudo pasan inadvertidos los problemas significativos del mismo, y de ahí el fracaso de la implantación técnica, la cual consiste en adquirir recursos, equipo y personal, transferir y capacitar al personal, la transición del antiguo sistema al nuevo.

PLANEACION DE LA IMPLANTACION

Cada fase de desarrollo del sistema de información administrativa deberá comenzar con un estudio o reformulación de un plan. Son demasiadas las actividades interrelacionadas que participan en todo esto, por lo cual no se pueden dirigir en forma diaria. Muchas de ellas pueden ser realizadas en paralelo y pese a ello deben combinarse.

IDENTIFICACION DE LAS TAREAS DE IMPLANTACION

Las principales tareas de la implantación suelen ser los siguientes aspectos:

- 1. Adquisición de personal**
- 2. Capacitación de personal**
- 3. Cambio de actitudes, patrones conductuales e interrelaciones**

CAPITULO IV. Diseño e Implementación del Sistema.

4. Adquisición y organización de las instalaciones y oficinas

5. Adquisición del hardware

6. Adquisición del software

7. Adquisición de formas y otros artículos

8. Generación de Archivos

9. Desarrollo de los procedimientos de pruebas

10. Pruebas del sistema

11. Operación del sistema

12. Evaluación del nuevo sistema de información

13. Mantenimiento del sistema

14. Documentación del sistema de trabajo

Estos puntos serán abordados en los capítulos siguientes.

CAPITULO IV. Diseño e Implementación del Sistema.

RESUMEN

El diseño conceptual del sistema de información administrativa, es una descripción preliminar de carácter general sobre sus elementos y relaciones estructurales. Es la fase más importante, pues determina el rendimiento general del sistema que se instale. Para éste, se requiere ante todo, efectuar un estudio del sistema total, investigar las necesidades, fijar los objetivos, y finalmente diseñar en forma apropiada los principales componentes del sistema de información administrativa. Cuando la estructura general del sistema de información ha sido establecida en el diseño conceptual, se hace la especificación de todas sus características. El diseño detallado requiere muchas horas-hombre de trabajo, de modo que entonces se realiza la revisión del plan del proyecto. Algunos de los principales pasos del diseño detallado son:

1. Convertir el costo, tiempo y funcionamiento en compromisos que optimicen el diseño general del sistema de Información administrativa.
2. Exponer las necesidades de información.
3. Desarrollar el sistema computacional.
4. Preparar el sistema de la base de datos.
5. Documentar el diseño detallado.

En teoría, el desarrollo del sistema de información administrativa parece ser un proceso sencillo.

CAPITULO V.

IMPLANTACION: PRUEBAS Y ANALISIS DE RESULTADOS

Objetivo:

Mostrar cómo convertir el diseño del sistema de información administrativa en un sistema de operación. Se incluyen la operación, la evaluación y el mantenimiento a fin de redondear el objetivo operacional.

CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

INTRODUCCION.

Implantar significa realizar un cambio organizacional y técnico. A menudo pasan inadvertidos los problemas significativos del cambio organizacional, y de ahí el fracaso de la implantación técnica. Esta consiste en adquirir recursos, equipo y personal; capacitar al personal; hacer la transacción del antiguo sistema al nuevo; probar el nuevo sistema; operarlo; evaluarlo y mantenerlo.

El costo relativo de implantación, puede ser alto si se se compara con el diseño. La razón de ello es el elevado precio de los paquetes de software, la capacitación, el equipo y suministros.

CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

5.1. INFRAESTRUCTURA DE COMPUTO

El departamento de sistemas de la empresa realmente es pequeño, ya que cuenta con sólo una computadora personal con las características siguientes.

Procesador 486 / 33 Mhz

Coprocesador matemático

Monitor color Super VGA

Disco duro de 180 Mbytes

4 Mb en memoria RAM

También cuenta con una impresora HP PAINT JET COLOR y otra IBM Personal Printer serie II de matriz de punto con una cabeza de impresión de 9 alambres, de carro ancho, que proporciona un adecuado avance de papel no continuo y de sobres, sin retirar el papel continuo, con diferentes velocidades de impresión para borrador o texto de calidad.

El software utilizado es el SISTEMA OPERATIVO versión 6.0.

En este momento, la empresa no quiso invertir más en este proyecto, porque se tiene planeado a corto plazo cambiar de sistema personal a un sistema AS/400 de IBM con plataforma cliente-servidor tan pronto como sea desarrollado el nuevo sistema con las instrucciones y diseño de la casa matriz de la empresa, por supuesto tomando como base el diseño del presente y la información que de él se genere.

CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

5.2. CAPACITACION AL USUARIO

Para obtener un mejor resultado al emplear el sistema, fue necesario impartir un curso de capacitación a todas las personas involucradas en el proceso.

Como primer punto se dio un curso de familiarización con el equipo de cómputo, el cual sirvió para que el personal implicado tuviera confianza y seguridad en el manejo del sistema y del equipo de cómputo.

Dicho curso, se dividió en cinco sesiones de dos horas cada una, con una simulación, en la última sesión, con el fin de tener un control exacto de todas las fallas que se pudieran presentar, así como el de tomar los tiempos estimados para las transacciones de la empresa.

Para hacer más fácil el curso se desarrolló un manual de operación (anexo en el capítulo VI), con el fin de auxiliar las dudas del usuario, y en caso de persistir, se procederá a dar una explicación detallada, para dar la respuesta adecuada al usuario.

5.3. DESCRIPCION GENERAL DEL PROCESO COMPUTARIZADO

Se describirá el procedimiento establecido para realizar el procedimiento de facturación mediante el sistema computarizado.

El procedimiento de facturación se dividió de la siguiente manera: Una persona encargada de dar seguimiento a la existencia de los productos requeridos por el cliente (ALMACEN), así como la requisición a nuestro proveedor, y otra de dar seguimiento a la captura de la orden de compra previa información de existencias.

CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

El encargado de almacén tendrá la función de actualizar el módulo de inventarios con la información generada de la órdenes de compra, requisiciones a proveedor y recepción de material , en coordinación con el encargado del sistema. Así como la de entregar el material requerido o embarcarlo al lugar donde el cliente se encuentre ubicado.

Una tercera persona tendrá a su cargo el módulo de cuentas por cobrar, cuya finalidad es la de llevar un control de los pagos de clientes generados diariamente. Por ser la primera vez que se realizaba este proceso de ésta forma, se realizó paralelamente, en forma manual y mediante el sistema computarizado, con la finalidad de poder tener un parámetro de comparación para determinar el grado de confiabilidad del sistema.

Una vez terminada la captura de información, el encargado del sistema procederá a respaldar la información generada en el módulo correspondiente del sistema, para evitar cualquier pérdida de información en caso de desastre.

5.4. ANALISIS DE RESULTADOS

Uno de los puntos más importantes que se registro, fué el ahorro de tiempo empleado en cada uno de los procesos de la empresa.

Anteriormente se tenía que llenar la factura de los clientes a máquina y entregarla un día después de la fecha de orden de compra del cliente. Ahora con el sistema computarizado, el cliente podrá recibir su factura en el momento mismo en que realiza su compra.

CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

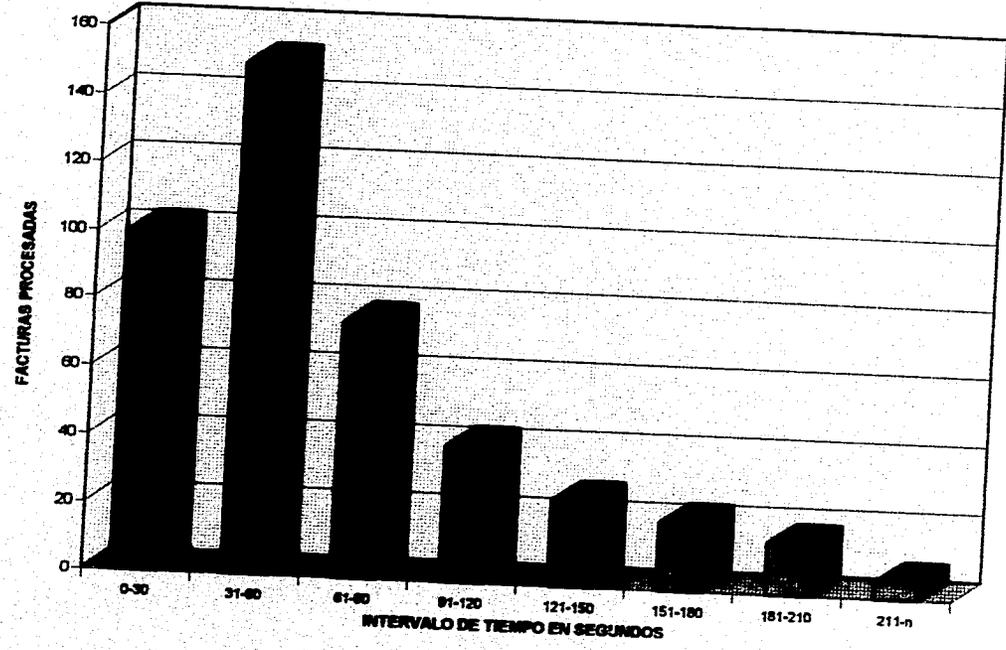
El porcentaje de tiempo empleado para las transacciones de la empresa fueron los siguientes:

Intervalos de tiempo	Ordenes de compra procesadas
menos de 30 seg.	100
mas de 30 seg. y menos de 60 seg.	150
mas de 61 seg y menos de 90 seg	75
mas de 91 seg y menos de 120 seg	40
mas de 121 seg y menos de 150 seg	25
mas de 151 seg y menos de 180 seg	20
mas de 181 seg y menos de 210 seg	15
en más de 210 segundos	5
Ordenes de compra procesadas:	430

Como se puede observar de la tabla, el mayor porcentaje, fué realizado en menos de un minuto, lo cual quiere decir que la mayoría fue procesada en tan sólo 60 segundos, y tomando en cuenta que anteriormente este proceso se llevaba uno o dos días, el resultado es bueno.

Otro punto importante en el que se logró un resultado significativo, fue la reducción en la transcripción de información para llevar el control de cuentas por cobrar de los clientes, así como la pérdida de tiempo en la captura de la información.

ESTADISTICAS DE TIEMPOS



CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

5.5. MANTENIMIENTO DEL SISTEMA

Una de las etapas más importantes dentro del desarrollo de sistemas es precisamente la etapa de mantenimiento, debido a que ésta marcará el ciclo de vida del sistema.

Para este sistema, en tanto no se cambien los requerimientos del usuario, el mantenimiento que se le debe proporcionar es:

Modificar dentro del programa fuente la actualización de las bases de datos que utiliza el sistema. (Depuración de las bases de datos)

5.6. ANALISIS COSTO-BENEFICIO

Al hablar de análisis costo-beneficio, se pretende llegar a la conclusión de saber si el sistema valió la pena, es decir si es viable; y un sistema se considera viable cuando los beneficios rebasan el costo del sistema. Es necesario mencionar que un sistema posiblemente no entregue el máximo de beneficios inmediatamente, pero puede ir retribuyéndolos a mediano plazo, los cuales pueden ser mucho mayores que es costo.

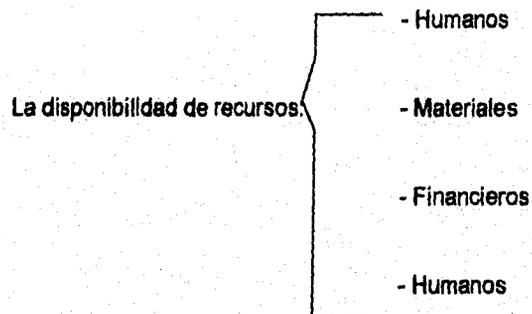
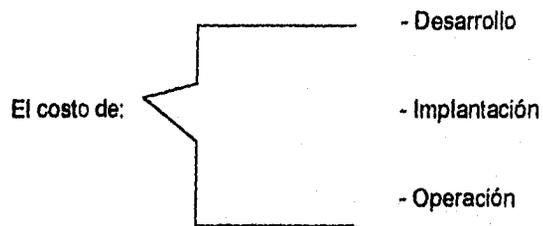
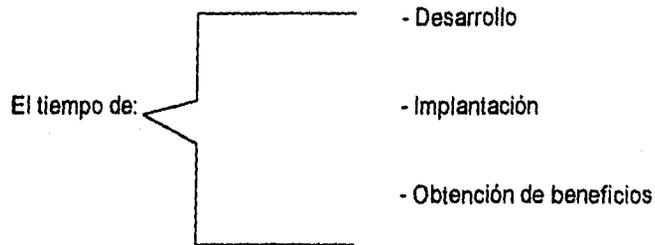
ESTUDIO DE VIABILIDAD

Antes de implantar un nuevo sistema es necesario determinar su viabilidad desde dos puntos de vista: el operativo y el económico. Para ello se realiza un análisis preliminar conocido como estudio de viabilidad, el cual se origina con base en un requerimiento realizado por la persona que lo solicita.

Posteriormente, se ordenan las ideas en forma adecuada para explotar las posibilidades de realización, estimar las condiciones del proyecto y tener una visión del beneficio supuesto del nuevo sistema.

CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

Para ello se deberá tener en cuenta lo siguiente:



CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

Los beneficios a obtener:

- Control interno
- Disponibilidad de información

Este análisis puede revelar que el sistema bajo estudio está operando satisfactoriamente y sólo requiere modificaciones ligeras. Los cambios recomendados se diseñarán para afinar el sistema a fin de que opere de manera óptima.

A continuación se analizará el costo promedio del sistema y los beneficios que entrega:

PROCEDIMIENTO	TIEMPO DESTINADO	COSTO
Análisis	10 hrs	\$ 120.00
Diseño	25 hrs	\$ 300.00
Construcción y pruebas	250 hrs	\$ 3,000.00
Implantación	30 hrs	\$ 360.00
Documentación	15 hrs	\$ 180.00
TOTAL	330 hrs.	N\$ 3,960.00

El costo por hora asignado es de 12 pesos y resulta ser una de las cotizaciones más económicas.

CAPITULO V. Implantación: Pruebas y Análisis de Resultados.

El costo total que se obtiene es por el tiempo asignado a cada módulo del sistema, lo que quiere decir que para los tres sistemas básicos el gran total sería de \$ 11,880.00. Si se solicitaran cotizaciones, este precio sería uno de los más bajos que podrían obtenerse.

El mantenimiento no se consideraría aún porque el sistema es nuevo y como tal no se cobraría, ya que iría implícito en la implantación.

Por estas razones, se llegó a la conclusión por el costo, que es viable, ya que los beneficios brindados rebasan el costo. Principalmente hay que tomar en cuenta la considerable disminución de tiempo y la agilización del proceso, lo cual es un beneficio que francamente es invaluable y nos inclina a asegurar la viabilidad del sistema.

Existen otras dos razones que nos empujan a declinar por su viabilidad y la cuales son la eliminación de tiempos muertos y el ahorro de tiempo en cada una de las operaciones de la empresa. Posiblemente no se logren apreciar los beneficios inmediatamente, pero es importante mencionar que a mediano plazo los beneficios obtenidos serán excelentes y rebasarán la inversión inicial. La comodidad del envío de la información y el mejoramiento en el procesamiento de la misma (tanto en velocidad, como en exactitud), es también una razón para aprobar la sistematización, lo cual es un poco difícil de evaluar. En conclusión el análisis costo-beneficio es satisfactorio.

RESUMEN

El estado de implantación del desarrollo del sistema de información administrativa, es la conversión del diseño en un sistema operacional. Equivale a fabricar un producto a partir de los dibujos de un diseño de ingeniería. La implantación suele ser un proyecto complejo que requiere una planeación cuidadosa y el control de los recursos. He aquí los pasos de la implantación:

1. Desarrollo y capacitación de la organización.
2. Adquisición.
3. Conversión
4. Pruebas
5. Operación
6. Evaluación.
7. Mantenimiento.

El desarrollo organizacional significa preparar a la empresa para el cambio. Quizás se necesiten nuevas actitudes. Es preciso que las personas involucradas deseen el cambio, en vez de resistirse a él. La capacitación es una extensión del desarrollo organizacional tendiente a mejorar los aspectos técnicos, lo mismo que las habilidades interpersonales. La adquisición se refiere a la obtención de todos los componentes del sistema de información administrativa, lo cual incluye el hardware y software. La conversión es la transición de un sistema antiguo a otro nuevo

CAPITULO VI.

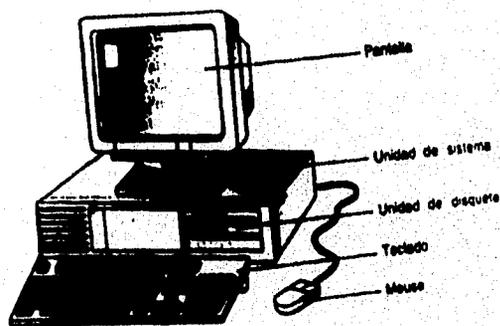
MANUAL DE OPERACION DEL SISTEMA.

MANUAL DE OPERACION

El manual de operación de cualquier sistema debe ser elocuente y preciso para poder brindar un mejor servicio a la persona que acuda a él. Este manual esta creado de tal forma que el usuario que requiera información del sistema encuentre explicación a sus dudas.

6.1. ACERCA DE SU COMPUTADORA

Su computadora consta de diversos elementos denominados hardware. El hardware permite ejecutar los programas de software que traducen las instrucciones que se envían a la computadora en un lenguaje que ésta pueda entender. Si no está familiarizado con los términos hardware, disquete y sistema operativo, es recomendable la lectura de este capítulo antes de utilizar el sistema.



CAPITULO VI. Manual de Operación del Sistema.

EL HARDWARE

El hardware que forma la mayor parte del sistema informático básico incluye monitor, un teclado y una unidad de sistema. La unidad del sistema contiene el procesador de la computadora, la memoria, los controladores de disco, los puertos y la tarjeta de video.

PROCESADOR Y MEMORIA

La unidad de proceso (CPU) y la memoria están colocadas en chips dentro de la unidad del sistema. La CPU es el lugar donde la computadora interpreta y procesa la información.

A menudo a la memoria de la computadora se le denomina RAM o memoria de acceso aleatorio. Las instrucciones que su computadora recibe y la información que procesa se guardan en la RAM durante la sesión de trabajo. La RAM de la computadora no es un lugar de almacenamiento permanente para la información: solamente está activa cuando la computadora está encendida. Cuando se apaga, la información se elimina de la RAM. Tiene que guardar su trabajo en un disco, un dispositivo de almacenamiento permanente, antes de apagar su computadora.

La memoria de la computadora se mide en kilobytes o megabytes de información. (Un byte es la cantidad de memoria necesaria para almacenar un carácter). Un kilobyte es igual a 1024 bytes, y un megabyte es igual a 1,048,576 bytes. Por lo tanto si el sistema tiene 640K de memoria. Puede almacenar 655,360 bytes en un momento dado. Por regla general, las necesidades de memoria del software que utiliza se indican en el exterior del paquete o puede obtener información de su distribuidor habitual.

MONITOR

El monitor tiene una pantalla que visualiza la información, es decir las instrucciones que el usuario envía a la computadora y el resultado e interpretación de esas instrucciones. La pantalla puede presentar esta información en uno ó varios colores. Un monitor de color no presentará el color a menos que disponga de la tarjeta de video apropiada.

CAPITULO VI. Manual de Operación del Sistema.

TECLADO

El teclado se utiliza para escribir las instrucciones para la computadora, y escribir la información que desea que procese la computadora. Todos los teclados tienen teclas normalizadas de letras, teclas de signos de puntuación y una barra espaciadora, de forma análoga a las teclas de una máquina de escribir. Asimismo, tienen teclas de función, teclas numéricas y de dirección, además de las teclas ALT, CTRL, SUPR y Enter o Retorno.

Teclado numérico

El teclado numérico es una forma rápida y sencilla de escribir todos los números, debido a que las teclas numéricas se agrupan en un bloque aparte. En muchos programas de software deberá presionar la tecla BLOQ NUM antes de utilizar el teclado numérico para escribirlos.

Teclas de dirección

En la mayor parte de los casos, las teclas de dirección le permiten moverse a través de la pantalla. Su función exacta está determinada por el tipo de software que se utilice. Las teclas de dirección están situadas en la parte inferior derecha a lado del teclado numérico, para el caso de teclados extendidos, en caso contrario la tecla bloq num debe estar desactivada y utilizar la tecla correspondiente del teclado numérico.

PUERTOS

Los puertos son conectores que están situados en la parte anterior o posterior de la unidad del sistema. Los puertos se utilizan para conectar el teclado, el monitor, la impresora, el mouse u otro tipo de hardware al sistema.

CAPITULO VI. Manual de Operación del Sistema.

HARDWARE ADICIONAL

Además de los componentes básicos, la mayor parte de los sistemas informáticos incluyen otro tipo de hardware como una impresora (que imprime la información que procesa), un mouse (para poder mover el puntero sobre la pantalla o seleccionar una opción de menú, simplemente apuntándolo y luego haciendo clic con uno de los botones) y posiblemente un módem (conecta el sistema a una línea telefónica de forma que pueda comunicarse con otra computadora).

USO DE DISQUETES Y UNIDADES DE DISCO

Un disquete, como la cinta de una grabadora, es un dispositivo de almacenamiento de uso repetido que contiene información, como programas y datos, en archivos. El espacio en un disco se mide en bytes. La información almacenada en un disco no se elimina cuando se apaga la computadora. La información se puede utilizar una y otra vez.

Las unidades de disco de la computadora mueven la información almacenada en los discos dentro y fuera de la memoria de acceso aleatorio. Hay dos clases de unidades de disco: una unidad de disco duro y una unidad de disquete. La unidad de disco duro contiene un disco fijo incorporado al sistema. En una unidad de disco duro, puede almacenar grandes cantidades de información, en lugar de almacenarlos en muchos disquetes.

Una unidad de disquete contiene un disquete que tiene menos capacidad de almacenamiento que un disco duro. Los disquetes se pueden sacar y se dispone de distintos tamaños. Muchos sistemas utilizan disquetes de 3 ½ pulgadas, y algunos de 5 1/4.

Cada unidad de disco tiene asignada una letra, de forma que pueda decir al sistema donde encontrar las instrucciones y la información. En muchos sistemas la unidad de disquetes se denomina A, y la unidad de disco duro se denomina unidad C.

CAPITULO VI. Manual de Operación del Sistema.

Para entrar al sistema , es necesario encender primero el sistema de cómputo. Una vez realizada esta operación, es necesario teclear desde la línea de comandos del sistema operativo, después del símbolo del sistema que puede ser C:\ lo siguiente:

```
c:\business
```

Para indicarle al sistema operativo que realice una tarea, debe escribir un comando como el anterior y luego presionar la tecla <ENTER>.

MANUAL DE USUARIO

Esta etapa comienza en el momento en el que el usuario ha entrado al sistema. Para comenzar el sistema despliega la pantalla siguiente:

```
ISCAR (MEXICO)  
ACCOUNTING SYSTEM
```

```
==== MAIN MENU ====  
Databases:  
EMPLOYEES  
CUSTOMERS  
VENDORS  
INVENTORY  
ORDERS  
ACCOUNTS RECEIVABLE  
AREA CODES  
Utilities:  
PRINT INVOICES  
BACK UP/RESTORE DATA  
QUIT TO DOS
```

CAPITULO VI. Manual de Operación del Sistema.

Esta ventana nos proporciona diversas opciones que el usuario puede seleccionar, dependiendo de la operación que desee realizar, las cuales se describirán a continuación.

EMPLOYEES

Si elije ésta opción se desplegará la siguiente pantalla:

EMPLOYEE DATABASE	
LAST NAME:	FIRST:
ADDRESS:	STATE:
CITY:	PHONE: () -
ZIP:	
DEPARTMENT:	TITLE:
EMPLOYEE No:	HIRE DATE:
	FULL TIME:
SALARY: \$	COMMISSION RATE: %
COMMENTS:	

== OPTION MENU ==

Add record
Edit record
Delete record

Next record
Previous record
Top record
Bottom record
Skip records
Find record

List records
Output reports
Group records
Count records
Index database
Help
Quit to MAIN MENU

Press first letter of menu choice, or highlight and press <ENTER>

CAPITULO VI. Manual de Operación del Sistema.

En la cual se descarga automáticamente la información contenida en la base de datos EMPLOYEES. En este momento el usuario estará en condiciones de seleccionar cualquiera de las opciones del menú y el sistema estará preparado para recibir información. Al introducir datos, el sistema checa que la información introducida sea válida, de no ser así le enviará un mensaje de error y regresará al menú de opciones en donde usted podrá intentar nuevamente la operación de desea realizar. De ser válida, se pasará a cada uno de los campos de ésta pantalla que se detallarán a continuación:

CAMPO	DESCRIPCION
LAST NAME:	Apellido paterno del empleado
FIRST:	Primer nombre del empleado
ADDRESS:	Dirección del empleado
CITY:	Ciudad
STATE:	Estado
ZIP:	Código postal
PHONE:	Télefono, empezando por la clave lada
DEPARTMENT:	Departamento o área en donde trabajará, la cual podrá escoger presionando la barra espaciadora que le mostrará las áreas validas para este campo.
TITLE:	Puesto
EMPLOYEE No:	Número de empleado
HIRE DATE:	Fecha de alta del empleado
FULL TIME:	Especifica si el empleado es de tiempo completo, en este caso introduzca "Y" En caso contrario Introduzca "N".
SALARY:	Sueldo del empleado
COMMENTS:	Comentarios que desee realizar sobre la persona en cuestión.

CAPITULO VI. Manual de Operación del Sistema.

A continuación se dará una breve explicación de las opciones del menú.

Add record

Esta opción le permitirá añadir un nuevo registro a la base de datos, la cual le pedirá la información antes mencionada. Una vez capturada la información requerida, el sistema procederá a preguntar al usuario si desea salvar la información, de ser así automáticamente realizará ésta operación, en caso contrario regresará al menú.

Edit record

Elija ésta opción si desea modificar o corregir la información capturada, la cual le permitirá hacer los cambios pertinentes ó necesarios para mantener la información actualizada. De igual forma que la opción anterior salvará automáticamente los datos modificados si así lo desea y regresará al menú de opciones, en caso contrario, no modificará la base de datos y regresará al menú.

Delete record

Esta opción sólo podrá ser utilizada cuando se requiera suprimir un registro de la base de datos siempre y cuando se tenga una razón justificada para hacerlo y pida autorización al encargado del sistema, quién será la única persona autorizada para ejecutar ésta opción. Ya que de no ser así, la información podría perderse en forma definitiva.

Next record

Permite al usuario visualizar el siguiente registro de la base de datos a partir del registro actual, mismo número que será indicado por el sistema en la parte superior derecha de la pantalla. Con la opción de modificar o editar dicho registro.

CAPITULO VI. Manual de Operación del Sistema.

Previous record

Permite visualizar el registro anterior de la base de datos a partir del registro actual. Además de permitirle al igual que en la opción anterior modificar este registro.

Top record

Permite visualizar el primer registro de la base de datos al usuario. Al mismo tiempo le dará la opción de modificar los datos correspondientes a este registro.

Bottom record

Permite visualizar el último registro de la base de datos y realizar las modificaciones necesarias al mismo si así lo desea.

Skip records

Al elegir ésta opción, estará en condiciones de brincar tantos registros como usted requiera para visualizar un registro de la base de datos. Introduzca un número positivo si desea brincar hacia adelante del registro actual, un número negativo, si desea brincar hacia atrás del registro actual.

Find record

El sistema desplegará la siguiente pantalla, en la que tendrá que introducir uno ó mas datos del registro a buscar. En caso de no existir alguno con la información proporcionada, el sistema mandará un mensaje de error y le pedirá que lo intente nuevamente con los datos correctos.

CAPITULO VI. Manual de Operación del Sistema.

----- ENTER FILTER CONDITION -----

DEPARTMENT:

TITLE:

Enter one or more conditions

Lists records

Desplegará en pantalla la lista de los registros existentes en la base de datos.

Output reports

Permitirá visualizar reportes impresos o en pantalla siempre y cuando estén disponibles al usuario y le mostrará la pantalla siguiente:

En cualquiera de éstas opciones se desplegará la pantalla siguiente:

===== REPORTS =====

Database report: EMPLOYEE
Mailing list: EMPLOYEE
Custom programmed report:
Exit to OPTION MENU

===== DESTINATION =====

Printer
File
Screen
Exit to OPTION MENU

CAPITULO VI. Manual de Operación del Sistema.

Si elige la opción **PRINTER**, mandará el reporte a impresora y le mostrará una pantalla en la que el sistema le pedirá diversas opciones dependiendo de la impresora, en caso contrario presione **ESC** para que el sistema tome tales opciones por default.

Si su respuesta es **FILE**, el sistema mandará el reporte a un archivo y le pedirá el nombre del mismo, el cuál no deberá exceder de ocho caracteres (letras) como máximo.

Si escogió **SCREEN**, enviará el reporte en la pantalla, en donde usted podrá visualizarlos. Si desea hacer una pausa en algun registro en especial bastará con presionar la tecla **PAUSE** y para seguir visualizando presione cualquier tecla.

Si desea salir elija **EXIT**.

Group records

En ésta opción podrá establecer las condiciones para trabajar con un determinado grupo de registros.

Count records

Tiene como función contar los registros existentes en la base de datos.

Index database

Se encarga de ordenar la base de datos, lo cual le permite localizar rápidamente la información o visualizar los registros en un cierto orden

CAPITULO VI. Manual de Operación del Sistema.

Help

En caso de existir alguna duda sobre cada opción, usted puede recurrir a esta opción que le proporcionará información sintetizada de éstas opciones.

Quit to MAIN MENU

Si desea regresar al menú principal elija ésta opción.

Cabe mencionar que usted puede seleccionar cada opción utilizando las teclas de movimiento de cursor ó simplemente presionando la primera letra de la opción que desee ejecutar.

CUSTOMERS

Si elige la opción Customers del menú principal aparecerá la siguiente pantalla:

CUSTOMER DATABASE	
CUSTOMER NO.:	CATEGORY:
NAME:	
ADDRESS:	
CITY:	
STATE:	ZIP:
PHONE: () -	
CONTACT:	
PHONE: () -	EXTENSION:
LAST CONTACT DATE: //	
TERMS:	COMMENT:

== OPTION MENU ==
Add record
Edit record
Delete record
Next record
Previous record
Top record
Bottom record
Skip records
Find record
List records
Output reports
Group records
Count records
Index database
Help
Quit to MAIN MENU

CAPITULO VI. Manual de Operación del Sistema.

En la cual se descarga automáticamente la información contenida en la base de datos CUSTOMERS. En este momento el usuario estará en condiciones de seleccionar cualquiera de las opciones de la barra de menu. Este módulo funcionará de la misma forma que la opción anterior, lo único que cambiará será la base de datos usada para cada módulo del sistema y los datos que pedirá al usuario para su captura.

En este caso los campos de detallarán a continuación:

CAMPO	DESCRIPCION
CUSTOMER No.:	Número de cliente comenzando por una letra.
CATEGORY:	Giro de la empresa
NAME:	Razón social
ADDRESS:	Domicilio Fiscal
CITY:	Ciudad
STATE:	Estado
ZIP:	Código postal
PHONE:	Teléfono
CONTACT:	Nombre de la persona a contactar
PHONE:	Teléfono del contacto
EXTENSION:	Si tiene extensión telefónica
LAST CONTACT DATE:	Fecha en que se hace el contacto con la persona
TERMS:	Condiciones de pago
COMMENTS:	Comentarios que desee hacer el usuario sobre el cliente

Las opciones que presenta la pantalla, se ejecutan de la misma forma en que fueron detalladas en la opción anterior, y de igual forma para cada una de las opciones, a excepción de las utilerías del menú principal.

VENDORS

Al seleccionar ésta opción se desplegará la pantalla siguiente:

VENDORS DATABASE	
VENDOR NUMBER:	
NAME: ADDRESS:	
CITY: STATE:	ZIP:
CONTACT: PHONE: () - TERMS: DISCOUNT: 0%	EXTENSION:
== OPTION MENU ==	
Add record Edit record Delete record	
Next record Previous record Top record Bottom record Skip records Find record	
List records Output reports Group records Count records Index database Help Quit to MAIN MENU	

Press first letter of menu choice, or highlight and press <ENTER>

CAPITULO VI. Manual de Operación del Sistema.

A continuación se describirán los campos:

CAMPO	DESCRIPCION
VENDOR NUMBER:	Número de distribuidor
NAME:	Razon social
ADDRESS:	Domicilio fiscal
CITY:	Ciudad
STATE:	Estado
ZIP:	Código postal
CONTACT:	Contacto
PHONE:	Teléfono
EXTENSION:	Extensión
TERMS:	Condiciones de pago
DISCOUNT:	Descuento

Las opciones del menú son iguales a los casos anteriores, solo cambia la base de datos, la cual es **VENDORS.DBF**.

CAPITULO VI. Manual de Operación del Sistema.

INVENTORY

La siguiente pantalla se desplegará si usted eligió esta opción:

INVENTORY (GOODS) DATABASE	
PART NO:	
NAME: DESCRIPTION: SALES PRICE: QUANTITY ON HAND: DISCONTINUED: N	== OPTION MENU == Add record Edit record Delete record <hr/> Next record Previous record Top record Bottom record Skip records Find record <hr/> List records Output reports Group records Count records Index database Help Quit to MAIN MENU
VENDOR NUMBER:	
COST: \$	
QUANTITY TO ORDER : (minimum/batch)	
LEAD TIME: (in days)	
COMMENTS:	

Press first letter of menu choice, or highlight and press <ENTER>

CAPITULO VI. Manual de Operación del Sistema.

A continuación se describirán los campos de ésta pantalla:

CAMPO	DESCRIPCION
PART NO:	Clave del producto
NAME:	Nombre del producto
DESCRIPTION:	Descripción del producto
SALES PRICE:	Precio de Lista
QUANTITY ON HAND:	Existencia
DISCONTINUED:	"Y" si el producto está descontinuado. "N" si no lo está.
VENDOR NUMBER:	Número de distribuidor
COST:	Costo del producto
QUANTITY TO ORDER:	Cantidad a ordenar
LEAD TIME:	Tiempo de entrega
COMMENTS:	Comentarios..

La base de datos usada es GOODS.dbf , y tiene las mismas funciones que las anteriores.

CAPITULO VI. Manual de Operación del Sistema.

A continuación se describirán los campos de ésta pantalla:

CAMPO	DESCRIPCION
PART NO:	Clave del producto
NAME:	Nombre del producto
DESCRIPTION:	Descripción del producto
SALES PRICE:	Precio de Lista
QUANTITY ON HAND:	Existencia
DISCONTINUED:	"Y" si el producto está descontinuado. "N" si no lo está.
VENDOR NUMBER:	Número de distribuidor
COST:	Costo del producto
QUANTITY TO ORDER:	Cantidad a ordenar
LEAD TIME:	Tiempo de entrega
COMMENTS:	Comentarios..

La base de datos usada es GOODS.dbf , y tiene las mismas funciones que las anteriores.

ORDERS

La opción orders desplegará la siguiente pantalla:

ORDERS TRANSACTIONS DATABASE		<p>== OPTION MENU ==</p> <p>Add record Edit record Delete record</p> <hr/> <p>Next record Previous record Top record Bottom record Skip records Find record</p> <hr/> <p>List records Output reports Group records Count records Index database Help Quit to MAIN MENU</p>
CUSTOMER ID:	P.O. NUMBER:	
ORDER DATE:		
<p>PART #:</p> <p>PART NAME:</p> <p>QTY. ORDERED: each PRICE: \$</p> <p>QTY. AVAILABLE: each MARGIN: %</p> <p>EMPLOYEE # - -</p> <p>INVOICED: Y</p> <p>NOTES: memo</p> <p>COMMENTS:</p>		

Press first letter of menu choice, or highlight and press <ENTER>

CAPITULO VI. Manual de Operación del Sistema.

Los campos se describirán a continuación:

CAMPO	DESCRIPCION
CUSTOMER ID:	Número de cliente
ORDER DATE:	Fecha
P.O. NUMBER:	Número de pedido
PART # :	Clave de producto
QTY. ORDERED:	Cantidad ordenada
EMPLOYEE:	Número de empleado
INVOICED:	Campo lógico que le permitirá imprimir la factura si selecciona "Y".
NOTES:	Notas que desee capturar

CAPITULO VI. Manual de Operación del Sistema.

Para realizar el procedimiento de facturación deberá seguir los siguientes pasos:

1. Elija la opción **Add_record** del menú de opciones y capture los datos de la orden de compra del cliente.
2. Una vez capturada la información, proceda a checar posibles errores en la captura, y en caso de existir seleccione la opción **Edit** y realice las modificaciones pertinentes.
3. Una vez que se ha percatado de la veracidad de los datos capturados y que haya introducido una "Y" en el campo **INVOICED**, regrese al menú principal y elija la opción **PRINT INVOICES**, la cual realizará el procedimiento de impresión

ACCOUNTS RECEIVABLE

Esta opción le permitirá entrar al sistema de cuentas por cobrar, la cual le mostrará la siguiente pantalla:

ACCOUNTS RECEIVABLE DATABASE		<p>== OPTION MENU ==</p> <p>Add record Edit record Delete record</p> <hr/> <p>Next record Previous record Top record Bottom record Skip records Find record</p> <hr/> <p>List records Output reports Group records Count records Index database Help Quit to MAIN MENU</p>													
INVOICE NO.:	BILLING DATE: / /														
CUSTOMER NO.:															
<table border="0"> <tr> <td style="text-align: center;">-- LAST INVOICE --</td> <td style="text-align: center;">----- AMOUNTS -----</td> </tr> <tr> <td>NUMBER:</td> <td>LAST BILL \$</td> </tr> <tr> <td>DATE: / /</td> <td>LAST PAID \$</td> </tr> <tr> <td>AGE: days</td> <td>OLD BALANCE \$</td> </tr> <tr> <td></td> <td>CURRENT ORDERS \$</td> </tr> <tr> <td colspan="2">=====</td> </tr> <tr> <td>CURRENT PMT.\$</td> <td>0.00 CURRENT BILL \$</td> </tr> </table>			-- LAST INVOICE --	----- AMOUNTS -----	NUMBER:	LAST BILL \$	DATE: / /	LAST PAID \$	AGE: days	OLD BALANCE \$		CURRENT ORDERS \$	=====		CURRENT PMT.\$
-- LAST INVOICE --	----- AMOUNTS -----														
NUMBER:	LAST BILL \$														
DATE: / /	LAST PAID \$														
AGE: days	OLD BALANCE \$														
	CURRENT ORDERS \$														
=====															
CURRENT PMT.\$	0.00 CURRENT BILL \$														
COMMENT: NOTE:															

Press first letter of menu choice, or highlight and press <ENTER>

CAPITULO VI. Manual de Operación del Sistema.

A continuación se describirán los campos de ésta pantalla:

CAMPO	DESCRIPCION
INVOICED NO:	Número de factura
BILLING DATE:	Fecha factura
CUSTOMER NO:	Numero de cliente
NUMBER:	Número de factura anterior del cliente
DATE:	Fecha factura anterior
AGE:	Antigüedad
LAST BILL:	Monto de la factura
LAST PAID:	Monto del pago del cliente
OLD BALANCE:	Saldo
CURRENTS ORDERS:	Monto de ordenes actuales
COMMENT:	Comentarios de usuario
NOTE:	Notas importantes

CAPITULO VI. Manual de Operación del Sistema.

Cuando desee realizar un pago del cliente seleccione la opción Add record del menú de opciones y capture la información requerida, en caso de visualizar errores seleccione la opción Edit y modifique los errores. Al igual que los otros módulos usted podrá realizar cualquiera de la opciones del menú que fueron descritas anteriormente.

BACKUP/RESTORE DATA

Una vez realizados todos los movimientos diarios, es conveniente respaldar la información en otro medio de almacenamiento diferente al usado por el sistema para prevenir pérdidas de información en caso de desastre, para lo cual pida al encargado del sistema realice esta operación cuando usted como usuario finalice sus actividades diarias dentro de la empresa para seguridad de la información.

CONCLUSIONES.

CONCLUSIONES

Definitivamente la automatización de los procesos de facturación, inventarios y cuentas por pagar, proporcionan muchas ventajas que incline a determinar que los sistemas en realidad son viables.

A continuación mencionaré alguna de las ventajas que me hacen formular esta opinión:

1. Uno de los principales beneficios que nos proporciona el sistema es la disminución de los tiempos en los procesos, y al hablar de procesos no sólo me refiero a los del sistema, sino a la las búsquedas y acceso a la información que los sistema de cómputo nos proporcionan.
2. Como ya se vio los tiempos de proceso son muy bajos comparados con los tiempos registrados en el proceso manual.
3. Otro punto importante es la generación de reportes actualizados que reflejan el estado actual del sistema, evitando de esta forma pérdidas de tiempo innecesarias.
4. Se reduce en gran medida los errores que se presentan en el proceso manual, pues el sistema sólo acepta valores reales y existentes
5. Se logró un porcentaje sin error superior al 98%, presentándose un margen de error que oscila alrededor del 0.2%.
6. El uso de reportes que se generan permitirán tener un mejor control de los procesos.

Son muchos los beneficios que la automatización de los procesos administrativos nos proporciona y hay que aprovecharlos al máximo mejorando su funcionamiento en base a la experiencia.

GLOSARIO.

GLOSARIO

Archivo.- Conjunto de registros que contienen datos

Automatización.- Aplicación de máquina que sustituyen al hombre en el proceso de información.

Bases de datos.- Es el conjunto de registros que contienen la información, y está estructurada en forma de tabla con renglones y columnas.

Bit.- Es la unidad más pequeña de información

Byte.- Es el conjunto de 8 bits, un carácter se presenta con un byte.

Carácter.- Símbolos que pueden ser alfabéticos.

Comando. Instrucción que contiene el Sistema operativo con el cual se puede realizar diversas tareas.

Compilador.- Es un programa que convierte las instrucciones escritas en un lenguaje de alto nivel a instrucciones o lenguaje de máquina.

Intérprete.- Son programas que traducen de un lenguaje a otro

Manual de usuario.- Es el documento que contiene toda la información necesaria para el buen uso de todos los módulos que componen el sistema.

Registro.- Es una colección de elementos de datos con los atributos asociados a ellos, relacionados por un nombre específico.

Sistema: Es el conjunto de programas supervisores y administradores que permiten la programación más eficiente.

Software.- Son todos aquellos programas, lenguajes, paquetes, etc. Utilizados por la computadora.

APENDICE A.

APENDICE A.

PROCEDURE Acct_rec

```
SET PROCEDURE TO c:\dbase\bases\Library

DO Set_env

SET COLOR TO &c_standard.
STORE "" TO invoice_no, cust_id, comments, notes, invoice_no, invoic_old
STORE 0 TO amt_of_cur, amt_cur_pd, amt_of_bil, amt_1stbil
STORE 0 TO amt_1st_pd, oldbalance
STORE 0 TO dat_of_bil, dat_1stbil

dbf = "ACCT_REC"
mlist = "NOT AVAILABLE"
cust_rpt = "N/A"
STORE "m->invoice_no" TO key, key1
STORE "NONE" TO key2, key3
keyname1 = "Invoice #:"
STORE "" TO keyname2, keyname3
list_flds = "INVOICE_NO, CUST_ID, DAT_OF_BIL, AMT_OF_BIL, OLDBALANCE"
STORE 0 TO balance, age

SELECT 1
USE c:\dbase\bases\Acct_rec ORDER invoice_no
USE c:\dbase\bases\Cust ORDER Cust_id IN 2
SET RELATION TO cust_id INTO Cust
GO TOP
record_num = RECNO()
DO Load_fid
CLEAR
DO Dstatus
DO Backgmd
DO Show_data

DO Bar_def
SET COLOR TO &c_popup.
ACTIVATE POPUP main_mnu
DO Sub_ret

RETURN

PROCEDURE Indexer

INDEX ON oldbalance TAG Oldbalance
INDEX ON cust_id TAG Cust_id
INDEX ON invoice_no TAG Invoice_no
GO TOP
RETURN
```

APENDICE A.

PROCEDURE Init_fld

```
STORE SPACE(10) TO invoice_no, invoic_old
cust_id = SPACE(6)
STORE 0 TO amt_of_cur, amt_of_bil, amt_lstbil, amt_lst_pd
STORE SPACE(30) TO comments, notes
STORE { / / } TO dat_of_bil, dat_lstbil
RETURN
```

PROCEDURE Load_fld

```
Invoice_no = invoice_no
cust_id = cust_id
amt_of_cur = amt_of_cur
amt_cur_pd = amt_cur_pd
dat_of_bil = dat_of_bil
amt_of_bil = amt_of_bil
comments = comments
notes = notes
dat_lstbil = dat_lstbil
amt_lstbil = amt_lstbil
amt_lst_pd = amt_lst_pd
oldbalance = oldbalance
invoic_old = invoic_old
RETURN
```

PROCEDURE Repl_fld

```
REPLACE Invoice_no WITH m->invoice_no, cust_id WITH m->cust_id,;
amt_of_cur WITH m->amt_of_cur, dat_of_bil WITH m->dat_of_bil,;
amt_of_bil WITH m->amt_of_bil, comments WITH m->comments
REPLACE notes WITH m->notes, dat_lstbil WITH m->dat_lstbil,;
amt_lstbil WITH m->amt_lstbil, amt_lst_pd WITH m->amt_lst_pd,;
invoic_old WITH m->invoic_old, oldbalance WITH m->oldbalance,;
amt_cur_pd WITH m->amt_cur_pd
RETURN
```

PROCEDURE Backgrnd

- ② 1,16 TO 3,41 DOUBLE COLOR &c_blue.
- ② 5, 1 TO 7,56 DOUBLE COLOR &c_red.
- ② 2,19 FILL TO 2,40 COLOR &c_red.
- ② 6, 2 FILL TO 6,55 COLOR &c_red.
- ② 9, 2 FILL TO 20,55 COLOR &c_red.
- ② 10, 1 TO 10,56 COLOR &c_red.
- ② 16, 1 TO 16,56 COLOR &c_red.
- ② 6, 1 TO 21,56 COLOR &c_red.
- SET COLOR TO &c_data.
- ② 2,20 SAY "CUENTAS POR COBRAR"
- ② 6, 3 SAY "FACTURA No.:"
- ② 6,26 SAY "FECHA FACTURA:"

APENDICE A.

```

@ 9, 3 SAY "CLIENTE NO.:"
@ 11, 3 SAY "-- DATOS FACTURA --"
@ 12, 3 SAY "NUMERO:"
@ 13, 3 SAY "FECHA:"
@ 14, 3 SAY "ANTIG:"
@ 11,28 SAY "----- CANTIDADES ----"
@ 12,28 SAY "TOTAL FACTURA $"
@ 14,16 SAY "DIAS"
@ 13,28 SAY "ULTIMO PAGO  $"
* @ 14,28 SAY "BALANCE ANT.  $"
* @ 15,28 SAY "ORDENES ACT.  $"
@ 16,45 SAY "======"
* @ 17, 3 SAY "ACTUAL PMT.$"
* @ 17,28 SAY "ACTUAL FACTURA $"
@ 19, 3 SAY "COMENTARIOS:"
@ 20, 3 SAY "NOTAS:"
SET COLOR TO &c_standard.
RETURN

```

PROCEDURE Show_data

```

age = IIF(oldbalance > 0, DATE() - dat_1stbil, 0)
SET COLOR TO &c_fields.
@ 6,16 SAY Invoice_no
@ 6,42 SAY dat_of_bil
@ 9,17 SAY cust_id
@ 9,24 SAY Cust->customer COLOR &c_yelowhit.
@ 12,11 SAY Invoice_no && old
@ 13,11 SAY dat_of_bil && 1stbil
age_color = "W"
IF ISCOLOR()
DO CASE
CASE m->age >= 60
age_color = c_red
CASE m->age >= 45
age_color = c_yellow
OTHERWISE
age_color = c_green
ENDCASE
ENDIF
@ 12,45 SAY amt_of_cur PICTURE "999,999.99" && 1stbil
@ 14,12 SAY m->age PICTURE "999" COLOR &age_color.
@ 13,45 SAY amt_1st_pd PICTURE "999,999.99"
bal_color = "W"
IF ISCOLOR()
DO CASE
CASE oldbalance >= 1000
bal_color = c_red
CASE oldbalance >= 100
bal_color = c_yelowhit
OTHERWISE
bal_color = c_green
ENDCASE
ENDIF

```

APENDICE A.

```

oldbalance= amt_of_cur - amt_lst_pd
@ 14,45 SAY m->oldbalance PICTURE "999,999.99" COLOR &bal_color.
@ 15,45 SAY amt_of_cur PICTURE "999,999.99"
@ 17,17 SAY amt_cur_pd PICTURE "999,999.99"
@ 17,45 SAY m->amt_of_bil PICTURE "999,999.99" COLOR &c_yelowhit.
@ 19,12 SAY comments
@ 20,12 SAY notes
IF ISCOLOR()
@ 22,1 SAY "números/texto amarillo son cálculos de la base de datos" ;
COLOR &c_yelowhit.
ELSE
@ 22,1 SAY "Dim texto/números c lculos de la base de datos" ;
COLOR &c_red.
ENDIF
SET COLOR TO &c_standard.
RETURN
PROCEDURE Get_data
SET COLOR TO &c_data.
@ 6,16 GET m->invoice_no ;
VALID Duplicat(m->invoice_no) ;
ERROR "Número de factura duplicado; Intente nuevamente" ;
MESSAGE "Teclee un número v lido " + ;
"(Id de cliente + año + mes)"
@ 6,42 GET m->dat_of_bil
@ 9,17 GET m->cust_id PICTURE "IXXXX" ;
VALID Lookupid((m->cust_id), "Cust", "Customer", 2) ;
ERROR "Identificador de cliente inv lido, Intente nuevamente" ;
MESSAGE "Teclee Identificador de cliente (empezando con una " + ;
"letra), o presione Esc para salir"
IF .NOT. PROMPT() = " Add record"
@ 9,24 SAY Cust->customer COLOR &c_yelowhit.
ELSE
@ 9,24 SAY SPACE(30)
ENDIF
@ 12,11 GET m->invoic_old
@ 13,11 GET m->dat_lstbil FUNCTION "D"
@ 12,45 GET m->amt_lstbil PICTURE "999,999.99"
@ 13,45 GET m->amt_lst_pd PICTURE "999,999.99"
@ 15,45 GET m->amt_of_cur PICTURE "999,999.99"
@ 17,17 GET m->amt_cur_pd PICTURE "999,999.99"
@ 19,12 GET m->comments FUNCTION "I"
@ 20,12 GET m->notes FUNCTION "I"
IF ISCOLOR()
@ 22,1 SAY "Yellow text/numbers from related database file or calc. " ;
COLOR &o_yelowhit.
ELSE
@ 22,1 SAY "Dim text/numbers from related database or calc. " ;
COLOR &c_red.
ENDIF
SET COLOR TO &c_standard.
ON KEY LABEL F9 DO Findcust WITH m->cust_id
RETURN

```

APENDICE A.

PROCEDURE Areacode

```
SET PROCEDURE TO c:\dbase\bases\Library
IF TYPE("c_save") = "U"
    PUBLIC c_save
    SET CONSOLE off
    c_save = SET("ATTRIBUTES")
    SET CONSOLE on
ENDIF
SET CONSOLE off
DO Set_env
SET CONSOLE on
SET COLOR TO &c_standard.
city = SPACE(20)
code = 0
STORE "CODES" TO dbf
STORE "NOT AVAILABLE" TO mlist
STORE "N/A" TO cust_rpt
STORE "m->city" TO key, key1
STORE "NONE" TO key2, key3
keyname1 = "City:"
STORE "" TO keyname2, keyname3
list_fids = "CITY, CODE"
mcode = 0
lookup_ok = .F.
SELECT 1
USE Codes ORDER City
GO TOP
record_num = RECNO()
DO Load_fid
CLEAR
DO Dstatus
DO Backgmd
DO Show_data
DO Bar_def
SET COLOR TO &c_popup.
ACTIVATE POPUP main_mnu
DO Sub_ret
RETURN
```

PROCEDURE Filter

```
choice = IIF(filters_on, "T", "Y")
DO Filt_ans
IF choice = "Y"
    mcode = 0
    ACTIVATE WINDOW alert
```

APENDICE A.

```
@ 0, 0 SAY "----- ENTER FILTER CONDITION ----"
@ 2, 0 SAY "Area code:" GET mcode PICTURE "999"
READ
DEACTIVATE WINDOW alert
IF 0 <> mcode
  SET FILTER TO code = mcode
ELSE
  ?? CHR(7)
  filters_on = .F.
  RETURN
ENDIF
GO TOP
filters_on = .NOT. EOF()
IF .NOT. filters_on
  ?? CHR(7)
  DO Show_msg WITH "No Areacode records match the filter condition"
  SET FILTER TO
  GO record_num
ENDIF
ELSE
  SET FILTER TO
  filters_on = .F.
ENDIF
RETURN
```

```
PROCEDURE Indexer
  INDEX ON code TAG Code
  INDEX ON city TAG City
  SET ORDER TO TAG City
  GO TOP
```

```
RETURN
```

```
PROCEDURE Init_fid
  city = SPACE(20)
  code = 0
```

```
RETURN
```

```
PROCEDURE Load_fid
```

```
  city = city
  code = code
```

```
RETURN
```

```
PROCEDURE Repl_fid
```

```
  REPLACE city WITH m->city, code WITH m->code
```

```
RETURN
```

```
PROCEDURE Backgmd
```

- Ⓢ 1,26 TO 3,53 DOUBLE COLOR &c_blue.
- Ⓢ 6,7 TO 8,38 DOUBLE COLOR &c_red.
- Ⓢ 9,7 TO 11,38 COLOR &c_red.
- Ⓢ 2,26 FILL TO 2,52 COLOR &c_blue.
- Ⓢ 6,7 FILL TO 11,38 COLOR &c_red.
- Ⓢ SET COLOR TO &c_data.
- Ⓢ 2,27 SAY "PERSONAL AREACODE SYSTEM"
- Ⓢ 7,10 SAY "CITY:"

APENDICE A.

```
@ 10,10 SAY "CODE:"  
SET COLOR TO &c_standard.  
RETURN
```

```
PROCEDURE Show_data
```

```
SET COLOR TO &c_fields.  
@ 7,17 SAY city  
@ 10,17 SAY code PICTURE "999"  
SET COLOR TO &c_standard.  
RETURN
```

```
PROCEDURE Get_data
```

```
SET COLOR TO &c_data.  
@ 7,17 GET m->city PICTURE "XXXXXXXXXXXXXXXXXXXXX"  
@ 10,17 GET m->code PICTURE "999"  
SET COLOR TO &c_standard.  
RETURN
```

```
PROCEDURE Colo_rese
```

```
PRIVATE old_color, c_messages, c_titles, c_box, c_info, c_fields  
old_color = c_save  
SET COLOR TO &old_color.  
old_color = STUFF(old_color, 1, AT("&",old_color)+2, "")  
comma = AT(",",old_color)  
c_messages = LEFT(old_color, comma-1)  
old_color = STUFF(old_color, 1, comma, "")  
comma = AT(",",old_color)  
c_titles = LEFT(old_color, comma-1)  
old_color = STUFF(old_color, 1, comma, "")  
comma = AT(",",old_color)  
c_box = LEFT(old_color, comma-1)  
old_color = STUFF(old_color, 1, comma, "")  
comma = AT(",",old_color)  
c_info = LEFT(old_color, comma-1)  
old_color = STUFF(old_color, 1, comma, "")  
comma = AT(",",old_color)  
c_fields = old_color  
SET COLOR OF MESSAGES TO &c_messages.  
SET COLOR OF TITLES TO &c_titles.  
SET COLOR OF BOX TO &c_box.  
SET COLOR OF INFORMATION TO &c_info.  
SET COLOR OF FIELDS TO &c_fields.  
RETURN
```

APENDICE A.

PROCEDURE Back_res

```
CLOSE DATABASES
PUBLIC mpath
STORE "" TO choice, answer
DO M_popdef
SET COLOR TO &c_normal.
CLEAR
SET COLOR TO &c_pop.
ACTIVATE POPUP backmenu
RETURN TO Business
PROCEDURE Bbackup
  ACTIVATE WINDOW backup
  @ 0,0 SAY "----- BACK UP DATA -----"
  @ 1,2 SAY "Insert a formatted floppy disk in drive A:"
  WAIT " Then press any key to begin backup..."
  RUN BACKUP &mpath.*.DB? A:
  RUN BACKUP &mpath.*.MDX A: /A
  ?? CHR(7)
  CLEAR
  ? ***** BACKUP PROCESS FINISHED *****
  WAIT
  DEACTIVATE WINDOW backup
  SET COLOR TO &c_normal.
  CLEAR
  SET COLOR TO &c_pop.
RETURN
PROCEDURE Back
DO CASE
CASE BAR() = 2
DO Sure
IF choice = "Y"
DO Bbackup
ENDIF
CASE BAR() = 3
DO Sure
IF choice = "Y"
DO Rrestore
ENDIF
CASE BAR() = 4
RETURN TO Business
ENDCASE
RETURN
PROCEDURE M_popdef
DEFINE POPUP backmenu FROM 7,20 TO 12,59 ;
MESSAGE "Press first letter of menu choice, " + ;
"or highlight then press <Enter>"
DEFINE BAR 1 OF backmenu ;
PROMPT "***** BACKUP - RESTORE MENU *****" SKIP

DEFINE BAR 2 OF backmenu ;
PROMPT " Back up your data to floppy disks"
```

APENDICE A.

```

DEFINE BAR 3 OF backmenu ;
  PROMPT " Restore your data from floppy disks"
DEFINE BAR 4 OF backmenu ;
  PROMPT " Exit to the Main Menu"
ON SELECTION POPUP backmenu DO Back
DEFINE WINDOW backup FROM 14,15 TO 20,65 COLOR &c_pop.
RETURN
PROCEDURE Rrestore
  ACTIVATE WINDOW backup
  @ 0,0 SAY "----- RESTORE DATA -----"
  @ 1,0 SAY "Insert first backup floppy disk in drive A:"
  WAIT " Then press any key to begin restoring..."
  I RESTORE A: &mpath.*.DB?
  I RESTORE A: &mpath.*.MDX
  ?? CHR(7)
  CLEAR
  ? "***** RESTORE PROCESS FINISHED *****"
  WAIT
  DEACTIVATE WINDOW backup
  SET COLOR TO &c_normal.
  CLEAR
  SET COLOR TO &c_pop.
RETURN
PROCEDURE Sure
  SET COLOR TO &c_normal.
  CLEAR
  choice = "N"
  ACTIVATE WINDOW backup
  @ 0,0 SAY "----- WARNING -----"
  @ 1,0 SAY " Are you sure you want to BACK UP or RESTORE ?"
  @ 2,0 SAY " Press Y to continue or N to stop."
  @ 3,10 SAY "CHOICE:" GET choice PICTURE "Y"
  ?? CHR(7)
  READ
  IF choice = "N"
    SET COLOR TO &c_pop.
    DEACTIVATE WINDOW backup
    RETURN
  ENDIF
  * Get DOS path for BACKUP or RESTORE programs
  IF UPPER(choice) = "Y"
    ?? CHR(7)
    mpath = "C:\dBASE\bases" + SPACE(10)
    CLEAR
    @ 0,0 SAY "----- ENTER SAMPLE FILES PATH -----"
    @ 1,0 SAY " Enter disk drive and full DOS path"
    @ 2,0 SAY " (the drive and path your databases are on)"
    @ 3,2 GET mpath VALID "" <> TRIM(mpath) ;
    MESSAGE "Enter a disk drive letter and full path"
    READ
    mpath = TRIM(mpath)
  ENDIF
  DEACTIVATE WINDOW backup
RETURN

```

APENDICE A.

PROCEDURE Business

```
CLEAR ALL
SET DEVELOPMENT on
SET DELIMITERS off
SET SCOREBOARD off
SET ESCAPE off
SET CLOCK off
SET STATUS off
SET TALK off
SET EXACT off
SET BELL off
SET CATALOG off
ON KEY LABEL F1 DO Helper
PUBLIC c_save
c_save = SET("ATTRIBUTES")
IF ISCOLOR()
c_normal = "W+/B,GR+/R,B"
c_pop = "B/W,GR+/R,W+/R"
red = "R/W"
blue = "B/W"
lt_blue = "W/BG"
ELSE
STORE "W+/N,N/W" TO c_normal, c_pop
STORE "W" TO red, blue
STORE "N/W" TO lt_blue
ENDIF
DO Main_def
**mstloop = .T.
DO WHILE BAR() <> 13
SET COLOR TO &c_normal.
CLEAR
DO Title
SET COLOR TO &c_pop.
ACTIVATE POPUP mainmenu
ENDDO
CLOSE ALL
CLEAR
RETURN
PROCEDURE Title
CLEAR
* Draw lines and box for menu with colors for effect
* 2,24 TO 5,53 DOUBLE COLOR &blue.
* 2,24 FILL TO 5,53 COLOR &blue.
SET COLOR TO &red.
* 3,27 SAY "A.T FURNITURE INDUSTRIES"
* 4,26 SAY "ACCOUNTING SYSTEM"
SET COLOR TO &c_normal.
RETURN
```

APENDICE A.

```
PROCEDURE Main_def
  DEFINE POPUP mainmenu FROM 7,27 TO 22,50;
  MESSAGE "Press first letter of menu choice, " + ;
    "or highlight and press <Enter>"
  DEFINE BAR 1 OF mainmenu PROMPT "===== MAIN MENU =====" SKIP
  DEFINE BAR 2 OF mainmenu PROMPT " Databases:" SKIP
  DEFINE BAR 3 OF mainmenu PROMPT " EMPLOYEES"
  DEFINE BAR 4 OF mainmenu PROMPT " CUSTOMERS"
  DEFINE BAR 5 OF mainmenu PROMPT " VENDORS"
  DEFINE BAR 6 OF mainmenu PROMPT " INVENTORY"
  DEFINE BAR 7 OF mainmenu PROMPT " ORDERS"
  DEFINE BAR 8 OF mainmenu PROMPT " ACCOUNTS RECEIVABLE"
  DEFINE BAR 9 OF mainmenu PROMPT " AREA CODES"
  DEFINE BAR 10 OF mainmenu PROMPT " Utilities:" SKIP
  DEFINE BAR 11 OF mainmenu PROMPT " PRINT INVOICES"
  DEFINE BAR 12 OF mainmenu PROMPT " BACK UP/RESTORE DATA"
  DEFINE BAR 13 OF mainmenu PROMPT " RETURN TO dBASE"
  DEFINE BAR 14 OF mainmenu PROMPT " QUIT TO DOS"
  ON SELECTION POPUP mainmenu DO Main
RETURN
PROCEDURE Main
DO CASE
  CASE BAR() = 3
    DO c:\dbase\bases\Employee
  CASE BAR() = 4
    DO c:\dbase\bases\Cust
  CASE BAR() = 5
    DO c:\dbase\bases\Vendors
  CASE BAR() = 6
    DO c:\dbase\bases\Goods
  CASE BAR() = 7
    DO c:\dbase\bases\Orders
  CASE BAR() = 8
    DO c:\dbase\bases\Acct_rec
  CASE BAR() = 9
    DO c:\dbase\bases\Areacode
  CASE BAR() = 11
    DO c:\dbase\bases\Invoices
  CASE BAR() = 12
    DO c:\dbase\bases\Back_res
  CASE BAR() = 13 .OR. BAR() = 14
    RELEASE ALL
    SET CLOCK ON
    IF BAR() = 13
      DO Colo_res
    RETURN TO MASTER
  ELSE
    QUIT
  ENDF
ENDCASE
RETURN
```

APENDICE A.

PROCEDURE Colo_rese

```
PRIVATE old_color, c_messages, c_titles, c_box, c_info, c_fields
old_color = c_save
SET COLOR TO &old_color.
old_color = STUFF(old_color, 1, AT("&",old_color)+2, "")
comma = AT(", ",old_color)
c_messages = LEFT(old_color, comma-1)
old_color = STUFF(old_color, 1, comma, "")
comma = AT(", ",old_color)
c_titles = LEFT(old_color, comma-1)
old_color = STUFF(old_color, 1, comma, "")
comma = AT(", ",old_color)
c_box = LEFT(old_color, comma-1)
old_color = STUFF(old_color, 1, comma, "")
comma = AT(", ",old_color)
c_info = LEFT(old_color, comma-1)
old_color = STUFF(old_color, 1, comma, "")
comma = AT(", ",old_color)
c_fields = old_color

SET COLOR OF MESSAGES TO &c_messages.
SET COLOR OF TITLES TO &c_titles.
SET COLOR OF BOX TO &c_box.
SET COLOR OF INFORMATION TO &c_info.
SET COLOR OF FIELDS TO &c_fields.
RETURN
```

APENDICE A.

PROCEDURE Cust

```
SET PROCEDURE TO Library
DO Sel_env
SET COLOR TO &c_standard.
STORE "" TO cust_id, category, customer, address1, address2, city, state
STORE "" TO zip, phone, contact, phone_cont, phone_ext, date_last, terms
STORE "" TO comments
```

```
STORE "CUST" TO dbf,
STORE "N/A" TO cust_rpt
STORE "m->cust_id" TO key, key1
STORE "NONE" TO key2, key3
keyname1 = "Customer #:"
STORE "" TO keyname2, keyname3, mcategory, mcity, mstate
list_flds = "CUST_ID, CONTACT, PHONE_CONT, PHONE_EXT"
```

```
SELECT 1
USE Cust ORDER Cust_id
GO TOP
```

```
USE Codes ORDER City IN 2
```

```
record_num = RECNO()
```

```
DO Load_fld
```

```
CLEAR
DO Dstatus
DO Backgmd
DO Show_data
```

```
DO Bar_def
```

```
SET COLOR TO &c_popup.
ACTIVATE POPUP main_mnu
DO Sub_ret
```

```
RETURN
```

PROCEDURE Filter

```
choice = IIF(filters_on, "T", "Y")
DO Fil_ens
IF choice = "Y"
  mcategory = SPACE(15)
  mcity = SPACE(20)
  mstate = SPACE(2)
  STORE SPACE(10) TO mzip, mterms
  ACTIVATE WINDOW alert
```

APENDICE A.

```

@ 0, 0 SAY "----- ENTER FILTER CONDITION -----"
@ 1, 0 SAY "CATEGORY:" GET mcategory FUNCTION "I" ;
MESSAGE "Enter a customer category"
@ 2, 0 SAY "CITY: " GET mcity PICTURE "XXXXXXXXXXXXXXXXXXXX"
@ 3, 0 SAY "STATE: " GET mstate PICTURE "II"
@ 3,15 SAY "ZIP: " GET mzip
@ 4, 0 SAY "TERMS: " GET mterms FUNCTION "I"
@ 5, 0 SAY "Enter one or more filter conditions"
READ
DEACTIVATE WINDOW alert
subset = " "
mcategory = TRIM(mcategory)
mcity = UPPER(TRIM(mcity))
mstate = TRIM(mstate)
mzip = TRIM(mzip)
mterms = TRIM(mterms)
subset = subset + IIF("'" <> mcategory, ;
[category = mcategory .AND. ], "")
subset = subset + IIF("'" <> mcity, ;
[UPPER(TRIM(mcity)) = mcity .AND. ], "")
subset = subset + IIF("'" <> mstate, ;
[state = mstate .AND. ], "")
subset = subset + IIF("'" <> mzip, ;
[zip = mzip .AND. ], "")
subset = subset + IIF("'" <> mterms, ;
[terms = mterms .AND. ], "")
IF "" = TRIM(subset)
DO Wambell
filters_on = .F.
ELSE
subset = SUBSTR(subset,1,LEN(subset)-6)
SET FILTER TO &subset.
GO TOP
filters_on = .NOT. EOF()
IF .NOT. filters_on
DO Wambell
DO Show_msg WITH "No Customer records match the filter condition"
SET FILTER TO
GO record_num
ENDIF
ENDIF
ELSE
SET FILTER TO
filters_on = .F.
ENDIF
RETURN

```

APENDICE A.

PROCEDURE Indexer

```
INDEX ON cust_id TAG Cust_id
GO TOP
RETURN
```

PROCEDURE Init_flg

```
STORE SPACE(30) TO customer, address1
STORE SPACE(20) TO city, contact, comments
STORE SPACE(10) TO zip, terms
STORE SPACE(13) TO phone, phone_cont
state = "TX"      && Could be any state or blank
cust_id = SPACE(6)
category = SPACE(15)
address2 = SPACE(25)
phone_ext = SPACE(4)
date_last = { / / }
RETURN
```

PROCEDURE Load_flg

```
cust_id = cust_id
category = category
customer = customer
address1 = address1
address2 = address2
city = city
state = state
zip = zip
phone = phone
contact = contact
phone_cont = phone_cont
phone_ext = phone_ext
date_last = date_last
terms = terms
comments = comments
RETURN
```

PROCEDURE Repl_flg

```
REPLACE cust_id WITH m->cust_id, category WITH m->category, ;
customer WITH m->customer, address1 WITH m->address1, ;
address2 WITH m->address2, city WITH m->city, state WITH m->state
REPLACE zip WITH m->zip, phone WITH m->phone, ;
contact WITH m->contact, phone_cont WITH m->phone_cont, ;
phone_ext WITH m->phone_ext, date_last WITH m->date_last, ;
terms WITH m->terms, comments WITH m->comments
RETURN
```

APENDICE A.

PROCEDURE Backgmd

```
@ 1,18 TO 3,41 DOUBLE COLOR &c_blue.  
@ 5, 2 TO 7,56 DOUBLE COLOR &c_red.  
@ 2,19 FILL TO 2,40 COLOR &c_blue.  
@ 6, 3 FILL TO 6,55 COLOR &c_red.  
@ 9, 3 FILL TO 19,55 COLOR &c_red.  
@ 15, 2 TO 15,56 COLOR &c_red.  
@ 8, 2 TO 20,56 COLOR &c_red.  
SET COLOR TO &c_data.  
@ 2,20 SAY "CUSTOMER DATABASE"  
@ 6, 4 SAY "CUSTOMER NO:"  
@ 6,30 SAY "CATEGORY:"  
@ 9, 4 SAY "NAME:"  
@ 10, 4 SAY "ADDRESS:"  
@ 12, 4 SAY "CITY:"  
@ 13, 4 SAY "STATE:"  
@ 13,16 SAY "ZIP:"  
@ 14, 4 SAY "PHONE:"  
@ 18, 4 SAY "CONTACT:"  
@ 17, 4 SAY "PHONE:"  
@ 17,27 SAY "EXTENSION:"  
@ 18, 4 SAY "LAST CONTACT DATE:"  
@ 19, 4 SAY "TERMS:"  
@ 19,27 SAY "COMMENT:"  
SET COLOR TO &c_standard.  
RETURN
```

PROCEDURE Show_data

```
SET COLOR TO &c_fields.  
@ 6,17 SAY cust_id  
@ 6,40 SAY category  
@ 9,13 SAY customer  
@ 10,13 SAY address1  
@ 11,13 SAY address2  
@ 12,13 SAY city  
@ 13,13 SAY state  
@ 13,21 SAY zip  
@ 14,13 SAY phone  
@ 16,13 SAY contact  
@ 17,13 SAY phone_cont  
@ 17,38 SAY phone_ext  
@ 18,23 SAY date_last  
@ 19,13 SAY terms  
@ 19,36 SAY comments  
SET COLOR TO &c_standard.  
RETURN
```

APENDICE A.

```
PROCEDURE Get_data
SET COLOR TO &c_data.
@ 6,17 GET m->cust_id PICTURE "I99999";
    VALID Duplicat(&key.);
    ERROR "Invalid customer ID number; please re-enter";
    MESSAGE "Enter a six digit customer ID beginning " + ;
        "with a letter, or Esc to quit"
@ 6,40 GET m->category;
    PICTURE "@M ARCHITECTS, CONSULTANTS, CONTRACTORS, LEGAL";
    MESSAGE "Press spacebar for Category choices"
@ 9,13 GET m->customer FUNCTION "I";
    MESSAGE "Enter name of customer"
@ 10,13 GET m->address1
@ 11,13 GET m->address2
@ 12,13 GET m->city PICTURE "XXXXXXXXXXXX"
@ 13,13 GET m->state PICTURE "II"
@ 13,21 GET m->zip
@ 14,13 GET m->phone PICTURE "(999)999-9999"
@ 16,13 GET m->contact FUNCTION "I";
    MESSAGE "Enter name of contact"
@ 17,13 GET m->phone_cont PICTURE "(999)999-9999"
@ 17,38 GET m->phone_ext PICTURE "9999";
    MESSAGE "Enter phone extension"
@ 18,23 GET m->date_last FUNCTION "D";
    MESSAGE "Enter date that customer was last contacted"
@ 19,13 GET m->terms PICTURE "@M CASH, NET 30, NET 45";
    MESSAGE "Press spacebar to see Terms choices"
@ 19,38 GET m->comments FUNCTION "I";
    MESSAGE "Enter any comments"
SET COLOR TO &c_standard.
ON KEY LABEL F9 DO Findcode WITH m->city
RETURN
```

APENDICE A.

PROCEDURE Employee

```
SET PROCEDURE TO Library
DO Set_env
SET COLOR TO &c_standard.
STORE "" TO lastname, firstname, initial, address1, address2, city, state
STORE "" TO zip, phone, emp_id, specialty, degree, awards, comments
STORE "" TO department, title
STORE 0 TO laborgrade, yrs_exper, salary, rate
STORE .T. TO exempt, full_time
date_hired = { / / }
STORE "EMPLOYEE" TO dbf,mlist
STORE "" TO cust_rpt
key = "m->lastname+m->firstname"
key1 = "m->lastname"
key2 = "m->firstname"
key3 = "NONE"
keyname1 = "Lastname:"
keyname2 = "Firstname:"
keyname3 = ""
list_fds = "LASTNAME, FIRSTNAME, DEPARTMENT, PHONE"
```

```
SELECT 1
USE Employee ORDER Names
GO TOP
USE Codes ORDER City IN 2
record_num = RECNO()
DO Load_fid
SET COLOR TO &c_standard.
CLEAR
DO Dstatus
DO Backgmd
DO Show_data
```

```
DO Bar_def
```

```
SET COLOR TO &c_popup.
ACTIVATE POPUP main_mnu
DO Sub_ret
*
```

```
RETURN
```

APENDICE A.

PROCEDURE Filter

DO Filt_ans

```
IF choice = "Y"
  STORE SPACE(15) TO department, title
  STORE SPACE(11) TO specialty
  STORE SPACE(3) TO degree
  ACTIVATE WINDOW alert
  @ 0,0 SAY "----- ENTER FILTER CONDITION -----"
  @ 1,1 SAY "DEPARTMENT: " GET m->department FUNCTION "!"
  @ 2,1 SAY "TITLE " GET m->title FUNCTION "!"
  @ 3,1 SAY "SPECIALTY " GET m->specialty FUNCTION "!"
  @ 4,1 SAY "DEGREE " GET m->degree FUNCTION "!"
  @ 5,1 SAY "Enter one or more conditions"
  READ
  DEACTIVATE WINDOW alert
  subset = ""
  subset = subset + IIF("" <> TRIM(m->department), ;
    [department = TRIM("&department.") .AND., ""])
  subset = subset + IIF("" <> TRIM(m->title), ;
    [title = TRIM("&title.") .AND., ""])
  subset = subset + IIF("" <> TRIM(m->specialty), ;
    [specialty = TRIM("&specialty.") .AND., ""])
  subset = subset + IIF("" <> TRIM(m->degree), ;
    [degree = TRIM("&degree.") .AND., ""])
  IF "" = TRIM(subset)
    DO Wambell
    filters_on = .F.
  ELSE
    subset = SUBSTR(subset,1,LEN(subset)-6)
    SET FILTER TO &subset.
    GO TOP
    filters_on = .NOT. EOF()
    IF .NOT. filters_on
      DO Wambell
      DO Show_msg WITH "No Employee records match the filter condition"
      SET FILTER TO
      GO record_num
    ENDIF
  ENDIF
ELSE
  SET FILTER TO
  filters_on = .F.
ENDIF
RETURN
```

PROCEDURE Indexer

```
INDEX ON department+lastname+firstname+initial TAG Dept
INDEX ON department+STR(salary,8,2) TAG Status
```

APENDICE A.

```
INDEX ON STR(yrs_exper,4,1)    TAG Years
INDEX ON emp_id              TAG Emp_id
INDEX ON lastname+firstname+initial TAG Names
GO TOP
RETURN
```

```
PROCEDURE Init_flg
```

```
Initial = ""
STORE SPACE(20) TO address1, address2
STORE SPACE(10) TO firstname, zip
STORE SPACE(15) TO lastname, department, title, awards
STORE SPACE(11) TO emp_id, specialty
STORE 0 TO laborgrade, yrs_exper, salary, rate
STORE .T. TO exempt, full_time
city = SPACE(14)
state = SPACE(2)
phone = SPACE(13)
degree = SPACE(3)
comments = SPACE(40)
date_hired = { / / }
```

```
RETURN
```

```
PROCEDURE Load_flg
```

```
lastname = lastname
firstname = firstname
initial = initial
emp_id = emp_id
address1 = address1
address2 = address2
city = city
state = state
zip = zip
phone = phone
department = department
title = title
laborgrade = laborgrade
exempt = exempt
full_time = full_time
date_hired = date_hired
specialty = specialty
yrs_exper = yrs_exper
degree = degree
salary = salary
rate = rate
awards = awards
comments = comments
RETURN
```

APENDICE A.

PROCEDURE Repl_fld

```
REPLACE emp_id WITH m->emp_id, lastname WITH m->lastname, ;
  firstname WITH m->firstname, initial WITH m->initial, ;
  address1 WITH m->address1, address2 WITH m->address2, ;
  city WITH m->city, state WITH m->state, zip WITH m->zip, ;
  phone WITH m->phone, department WITH m->department
REPLACE title WITH m->title, laborgrade WITH m->laborgrade, ;
  exempt WITH m->exempt, full_time WITH m->full_time, ;
  date_hired WITH m->date_hired, specialty WITH m->specialty, ;
  yrs_exper WITH m->yrs_exper, degree WITH m->degree, ;
  salary WITH m->salary, rate WITH m->rate, ;
  awards WITH m->awards, comments WITH m->comments
```

RETURN

PROCEDURE Backgmd

```
@ 1,18 TO 3,41 DOUBLE COLOR &c_blue.
@ 4, 1 TO 6,56 DOUBLE COLOR &c_red.
@ 2,19 FILL TO 2,40 COLOR &c_blue.
@ 4, 2 FILL TO 21,55 COLOR &c_red.
@ 11, 1 TO 11,56 COLOR &c_red.
@ 7, 1 TO 22,58 COLOR &c_red.
SET COLOR TO &c_data.
@ 2,20 SAY "EMPLOYEE DATABASE"
@ 5, 3 SAY "LAST NAME:"
@ 5,32 SAY "FIRST:"
@ 5,53 SAY ""
@ 8, 3 SAY "ADDRESS:"
@ 9, 3 SAY "CITY:"
@ 9,32 SAY "STATE:"
@ 10, 3 SAY "ZIP:"
@ 10,32 SAY "PHONE:"
@ 12, 3 SAY "DEPARTMENT:"
@ 12,32 SAY "TITLE:"
@ 13,32 SAY "SPECIALTY:"
@ 14, 3 SAY "EMPLOYEE NO.:"
@ 14,32 SAY "HIRE DATE:"
@ 15,32 SAY "FULL TIME:"
@ 16,32 SAY "EXEMPT:"
@ 17,32 SAY "LABOR GRADE:"
@ 16, 3 SAY "SALARY: $"
@ 16,32 SAY "COMMISSION RATE:"
@ 16,54 SAY "%"
@ 19, 3 SAY "DEGREE:"
@ 19,32 SAY "YEARS EXPERIENCE:"
@ 20, 3 SAY "AWARDS:"
@ 21, 3 SAY "COMMENTS:"
SET COLOR TO &c_standard.
RETURN
```

APENDICE A.

PROCEDURE Show_data

```
SET COLOR TO &c_fields.
@ 5,14 SAY lastname
@ 5,39 SAY firstname
@ 5,52 SAY initial
@ 8,12 SAY address1
@ 8,34 SAY address2
@ 9,12 SAY city
@ 9,39 SAY state
@ 10,12 SAY zip
@ 10,39 SAY phone
@ 12,16 SAY department
@ 12,39 SAY title
@ 13,43 SAY specialty
@ 14,16 SAY emp_id
@ 14,43 SAY date_hired
@ 15,43 SAY full_time PICTURE "Y"
@ 16,43 SAY exempt PICTURE "Y"
@ 17,45 SAY laborgrade PICTURE "9"
@ 18,14 SAY salary PICTURE "999,999.99"
@ 18,50 SAY rate PICTURE "99.9"
@ 19,14 SAY degree
@ 19,50 SAY yrs_exper PICTURE "99.9"
@ 20,14 SAY awards
@ 21,14 SAY comments
SET COLOR TO &c_standard.
ON KEY LABEL F9 DO Findcode WITH m->city
RETURN
```

PROCEDURE Get_data

```
SET COLOR TO &c_data.
@ 5,14 GET m->lastname PICTURE "XXXXXXXXXXXXXXXXX";
MESSAGE "Enter employee last name"
@ 5,39 GET m->firstname PICTURE "XXXXXXXXXX"
@ 5,52 GET m->initial PICTURE "I"
@ 8,12 GET m->address1
@ 8,34 GET m->address2
@ 9,12 GET m->city PICTURE "XXXXXXXXXXXXXXXXX"
@ 9,39 GET m->state PICTURE "II"
@ 10,12 GET m->zip
@ 10,39 GET m->phone PICTURE "(999)999-9999"
@ 12,16 GET m->department PICTURE "@M SALES, EXECUTIVE";
MESSAGE "Press spacebar for Department options"
@ 12,39 GET m->title FUNCTION "I"
@ 13,43 GET m->specialty FUNCTION "I"
@ 14,16 GET m->emp_id PICTURE "999-99-9999"
@ 14,43 GET m->date_hired FUNCTION "D"
@ 15,43 GET m->full_time PICTURE "Y";
```

APENDICE A.

```
WHEN TRIM(m->department) <> "EXECUTIVE"  
@ 16,43 GET m->exempt PICTURE "Y" ;  
WHEN TRIM(m->department) <> "EXECUTIVE"  
@ 17,45 GET m->laborgrade PICTURE "9"  
@ 18,14 GET m->salary PICTURE "999,999.99"  
@ 18,50 GET m->rate PICTURE "99.9" ;  
WHEN TRIM(m->department) <> "EXECUTIVE"  
@ 19,14 GET m->degree PICTURE "III"  
@ 19,50 GET m->yrs_exper PICTURE "99.9"  
@ 20,14 GET m->awards FUNCTION "I"  
@ 21,14 GET m->comments  
SET COLOR TO &c_standard.  
ON KEY LABEL F9 DO Findcode WITH m->city  
RETURN
```

APENDICE A.

PROCEDURE Goods

```
SET PROCEDURE TO Library
DO Set_env
SET COLOR TO &c_standard.
STORE "" TO part_id, part_name, descript, vendor_id, comments
STORE 0 TO price, cost, qty_onhand, qty_2order, lead_time
discontinu = .F.
dbf = "GOODS"
mlist = "NOT APPLICABLE"
cust_rpt = "N/A"
STORE "m->part_id" TO key, key1
STORE "NONE" TO key2, key3
keyname1 = "Part ID:"
STORE "" TO keyname2, keyname3, mvendorid
list_flds = "PART_ID, PART_NAME, QTY_ONHAND"
```

```
SELECT 1
USE Goods ORDER Part_id
GO TOP
USE Vendors ORDER Vendor_id IN 2
record_num = RECNO()
DO Load_fid
CLEAR
DO Dstatus
DO Backgmd
DO Show_data
DO Bar_def
SET COLOR TO &c_popup.
ACTIVATE POPUP main_mnu
DO Sub_ret
*
```

RETURN

PROCEDURE Filter

```
choice = IIF(filters_on, "T", "Y")
DO Filtr_ens
IF choice = "Y"
mvendorid = SPACE(4)
ACTIVATE WINDOW alert
  @ 0, 0 SAY "----- ENTER FILTER CONDITION -----"
  @ 2, 0 SAY "VENDOR ID:" GET mvendorid FUNCTION "g"
  READ
DEACTIVATE WINDOW alert
```

APENDICE A.

```
IF "" = TRIM(mvendorid)
  filters_on = .F.
  DO Wambell
ELSE
  SET FILTER TO vendor_id = TRIM(mvendorid)
  GO TOP
  filters_on = .NOT. EOF() && Filter is turned on if .T.
  IF .NOT. filters_on
    DO Wambell
    DO Show_msg WITH "No Goods (inventory) records match the " + ;
      "filter condition"
  SET FILTER TO
  GO record_num
ENDIF
ENDIF
ELSE
  SET FILTER TO
  filters_on = .F.
ENDIF
RETURN
```

```
PROCEDURE Indexer
  INDEX ON vendor_id TAG Vendor_id
  INDEX ON part_id TAG Part_id
  GO TOP
  RETURN
```

```
PROCEDURE Init_fid
  part_id = SPACE(10)
  STORE SPACE(30) TO part_name, descript, comments
  STORE 0 TO qty_onhand, cost, price, qty_2order, lead_time
  vendor_id = SPACE(4)
  RETURN
```

```
PROCEDURE Load_fid
  part_id = part_id
  part_name = part_name
  descript = descript
  qty_onhand = qty_onhand
  cost = cost
  price = price
  qty_2order = qty_2order
  vendor_id = vendor_id
  lead_time = lead_time
  comments = comments
  RETURN
```

APENDICE A.

PROCEDURE Repl_fld

```
REPLACE part_id WITH m->part_id, part_name WITH m->part_name, ;
descript WITH m->descript, qty_onhand WITH m->qty_onhand, ;
cost WITH m->cost, price WITH m->price, ;
qty_2order WITH m->qty_2order, vendor_id WITH m->vendor_id, ;
lead_time WITH m->lead_time, comments WITH m->comments
RETURN
```

PROCEDURE Backgmd

```
@ 1,17 TO 3,46 DOUBLE COLOR &c_blue.
@ 5, 2 TO 7,30 DOUBLE COLOR &c_red.
@ 2,18 FILL TO 2,45 COLOR &c_blue.
@ 6, 3 FILL TO 6,29 COLOR &c_red.
@ 9, 3 FILL TO 18,54 COLOR &c_red.
@ 13, 3 TO 13,54 COLOR &c_red.
@ 8, 2 TO 19,55 COLOR &c_red.
SET COLOR TO &c_data.
@ 2,19 SAY "GOODS (INVENTORY) DATABASE"
@ 6, 4 SAY "PART NO.:"
@ 9, 4 SAY "NAME:"
@ 10, 4 SAY "DESCRIPTION:"
@ 11, 4 SAY "SALES PRICE:"
@ 12, 4 SAY "QUANTITY ON HAND:"
@ 12,32 SAY "DISCONTINUED:"
@ 14, 4 SAY "VENDOR NUMBER:"
@ 15, 4 SAY "COST: $"
@ 16, 4 SAY "QUANTITY TO ORDER:"
@ 16,29 SAY "(minimum/batch)"
@ 17, 4 SAY "LEAD TIME:"
@ 17,20 SAY "(in days)"
@ 18, 4 SAY "COMMENTS:"
SET COLOR TO &c_standard.
RETURN
```

PROCEDURE Show_data

```
SET COLOR TO &c_fields.
@ 6,15 SAY part_id
@ 9,17 SAY part_name
@ 10,17 SAY descript
@ 11,17 SAY price PICTURE "99,999.99"
@ 12,22 SAY qty_onhand PICTURE "9,999"
@ 12,46 SAY discontinu PICTURE "Y"
@ 14,19 SAY vendor_id
@ 15,16 SAY cost PICTURE "99,999.99"
@ 16,23 SAY qty_2order PICTURE "9,999"
@ 17,18 SAY lead_time PICTURE "999"
```

APENDICE A.

```
@ 18,16 SAY comments
SET COLOR TO &c_standard.
RETURN
```

PROCEDURE Get_data

```
* Show screen for data entry
SET COLOR TO &c_data.
@ 6,15 GET m->part_id FUNCTION "I" ;
    VALID Duplicat(&key.) ;
    ERROR "Duplicate part ID number, please re-enter" ;
    MESSAGE "Enter a part ID number, or Esc to quit"
@ 9,17 GET m->part_name FUNCTION "I" ;
    MESSAGE "Enter the name of the part"
@ 10,17 GET m->descript FUNCTION "I" ;
    MESSAGE "Enter a description of the part"
@ 11,17 GET m->price PICTURE "99,999.99" ;
    MESSAGE "Enter the selling price of this part"
@ 12,22 GET m->qty_onhand PICTURE "9,999" ;
    MESSAGE "Enter how many of these parts are in current inventory"
@ 12,46 GET m->discontin PICTURE "Y" ;
    MESSAGE "Is the part now discontinued (Y/N)"
@ 14,19 GET m->vendor_id FUNCTION "9" ;
    VALID Lookupid((m->vendor_id),"Vendors", "Vendor",1) ;
    ERROR "Invalid vendor ID number, please re-enter" ;
    MESSAGE "Enter a vendor ID number, or Esc to quit"
@ 15,16 GET m->cost PICTURE "99,999.99" ;
    MESSAGE "Enter the cost of the part"
@ 16,23 GET m->qty_2order PICTURE "9,999" ;
    MESSAGE "Enter the minimum quantity which can be ordered"
@ 17,16 GET m->lead_time PICTURE "999" ;
    MESSAGE "Enter the lead time before vendor " + ;
        "typically ships the parts"
@ 18,16 GET m->comments FUNCTION "I" ;
    MESSAGE "Enter any comments on this part"
SET COLOR TO &c_standard.
ON KEY LABEL F9 DO Findvend WITH m->vendor_id
RETURN
```

APENDICE A.

PROCEDURE Helper

```
IF ISCOLOR()
c_help = "B/BG,GR+/G"
ELSE
c_help = "W+/N,W/N"
ENDIF
DEFINE WINDOW Helpscr FROM 0,0 TO 24,79 PANEL COLOR &c_help.
ACTIVATE WINDOW Helpscr
CLEAR
DO CASE
CASE POPUP() = "MAINMENU"
DO Help_main
CASE POPUP() = "RPT_MNU"
DO Help_rept
CASE POPUP() = "DEST_MNU"
DO Help_dest
CASE POPUP() = "MAIN_MNU"
DO Help_optl
CASE POPUP() = "BACKMENU"
DO Help_back
ENDCASE
WAIT SPACE(24) + "Press any key to return to menu."
CLEAR
DEACTIVATE WINDOW Helpscr
RELEASE WINDOW Helpscr
RETURN
```

PROCEDURE Help_main

TEXT

HELP FOR THE MAIN MENU

Selection	Purpose
EMPLOYEES	Employee subapplication
CUSTOMERS	Customer subapplication
VENDORS	Vendors subapplication
INVENTORY	Inventory subapplication
ORDERS	Orders subapplication
ACCOUNTS RECEIVABLE	Accounts Receivable subapplication
AREA CODES	Area Codes subapplication
PRINT INVOICES	Print invoices from orders
BACK UP/RESTORE DATA	Back up or restore business data files
RETURN TO dBASE	Quit application and return to dot prompt or Control Center
QUIT TO DOS	Quit application and return to operating system

ENDTEXT

RETURN

APENDICE A.

PROCEDURE Help_opti

TEXT

HELP FOR THE OPTION MENU

Selection	Purpose
Add record	Add new record to database
Edit record	Edit the current record
Delete record	Delete current record from the database
Next record	skip to Next record
Previous record	skip to Previous record
Top record	go to Top of database
Bottom record	go to Bottom of database
Skip records	Skip fwd or backwards by an entered # of records
Find record	Find a record using key field search
List records	List data to screen beginning with current record
Output reports	Print or display reports and labels
Group records	Specify filter condition to view selective records
Count records	Count number of records in database
Index database	create/rebuild Indexes
Quit	Quit from subapplication screen to MAIN MENU

ENDTEXT

RETURN

PROCEDURE Help_rept

TEXT

HELP FOR THE REPORTS MENU

Selection	Purpose
Mailing List	Print mailing labels
Customized programmed report	Execute a custom report program (not available for some subapplications)
Exit to OPTION MENU	Return to the OPTION MENU without printing

ENDTEXT

RETURN

APENDICE A.

PROCEDURE Help_dest
* Display help screen for the Option Menu
TEXT

HELP FOR THE DESTINATION MENU

Selection	Purpose
Printer	List to the current printer
File	Output to a file for later printing
Screen	List report on the screen
Exit to OPTION MENU	Return to the OPTION MENU without printing

ENDTEXT
RETURN

PROCEDURE Help_back
TEXT
HELP FOR THE BACK UP/RESTORE MENU

Selection	Purpose
Back up your data	Save database files and indexes on to floppy disks floppy disks
Restore your data	Copy database files and indexes from floppy
Exit to the Main Menu	Return to the MAIN MENU

ENDTEXT
RETURN

APENDICE A.

INVOICE2.PRG

```
*****
CLEAR WIND
CLOSE DATABASE
SAVE SCREEN TO Invoice2
CLEAR

DEFINE WINDOW Pause FROM 15,00 TO 19,79 DOUBLE
DEFINE WINDOW invo_hlp FROM 3,00 TO 21,79 DOUBLE
ON ERROR DO Pause WITH "Line number in program "+Program()+": "+LTRIM(STR(LINE()))

IF EOF()
  SKIP -1
ENDIF

IF SET("TALK")="ON"
  SET TALK OFF
  lc_talk = "ON"
ELSE
  lc_talk = "OFF"
ENDIF
lc_escape = SET("ESCAPE")
lc_cursor = SET("CURSOR")
SET CURSOR OFF

SELECT 1
USE TRANSACT.DBF
USE CLIENT.DBF IN 2 ORDER CLIENT_ID
SET RELATION TO TRIM(CLIENT_ID) INTO B
SET FIELDS TO A->CLIENT_ID,A->ORDER_ID,A->DATE_TRANS,A->INVOICED,A->
TOTAL_BILL,B->CLIENT,B->LASTNAME,B->FIRSTNAME,B->ADDRESS,B->CITY,B->
STATE,B->ZIP,B->PHONE,B->CLIEN_HIST
GO TOP
lc_status = SET("STATUS")
IF lc_status = "ON"
  SET STATUS OFF
ENDIF
gc_msg = "F9:Line Items | PgDn:Next | PgUp:Previous | Ctrl-PgDn:Bottom | Ctrl-PgUp:Top"

DEFINE WINDOW wndow1 FROM 7,26 TO 20,78

DEFINE WINDOW Table FROM 9,1 TO 20,78 DOUBLE

ON KEY LABEL F1 DO invo_hlp
ON KEY LABEL F9 KEYBOARD CHR(23)+"L" && SEND CTRL-W + L FOR linetm
ON KEY LABEL F10 KEYBOARD CHR(23)+"M" && SEND CTRL-W + M FOR MENU
```

APENDICE A.

DO Defnmenu

gc_mdx = MDX(1)
gc_alias = ALIAS()

gn_sele = IIF(SELE()=10, SELE()-1, SELE())
SELE (gn_sele)
USE STOCK ORDER TAG ORDER_ID
SET FIELDS TO ORDER_ID, PART_ID, PART_NAME, DESCRIPT, QTY, ITEM_COST,
EXTENDED=QTY*ITEM_COST
SET CARRY TO STOCK->ORDER_ID

SELE &gc_alias.
gn_gorec = 0
gc_search = SPACE(200)
gc_seek = SPACE(LEN(ORDER_ID))
gl_newrec = .f.
gl_chgrec = .t.
gl_extloop = .f.
gl_rollback = .f.
@ 23,0
@ 23,CENTER(gc_messg,80) SAY gc_messg

DO WHILE .NOT. gl_extloop
gn_reco = RECNO()
gl_lineitn = .T.
BEGIN TRANSACTION

DO WHILE gl_lineitn
IF gl_chgrec
DO Show_get
ENDIF

DO Showbrow

DO Edit_get

SET CURSOR ON
READ
SET CURSOR OFF

gn_inkey = INKEY()
gn_readkey = READKEY()
gc_readvar = VARREAD()
gl_lineitn = .F.

ACTIVATE SCREEN
DO CASE
CASE gn_inkey = 76

APENDICE A.

```

DO LineItem
gl_lineitm = .T.
CASE gn_inkey = 77
ACTIVATE MENU Editmenu
CASE gn_readkey = 6 .OR. gn_readkey = 262 .OR. gn_readkey = 260 ;
.OR. gn_readkey = 4
IF .NOT. BOF()
SKIP -1
ENDIF
CASE gn_readkey = 7 .OR. gn_readkey = 263 .OR. gn_readkey = 5 ;
.OR. gn_readkey = 261
SKIP
IF EOF() .AND. .NOT. gl_newrec
CLEAR GETS
SET DELI OFF
@ 23,0
@ 23,25 SAY "====> Add new records (Y/N)?" GET gl_newrec PICT "Y"
READ
SET DELI ON
CLEAR GETS
@ 23,0
@ 23,CENTER( gc_messg, 80) SAY gc_messg
ENDIF
IF gl_newrec
DO Recappnd
ELSE
IF EOF()
SKIP -1
ENDIF
ENDIF
CASE gn_readkey = 34 .OR. gn_readkey = 290

GO TOP
CASE gn_readkey = 35 .OR. gn_readkey = 291

GO BOTTOM
CASE gn_readkey = 12 .or. gn_readkey = 270

gl_extloop = .T.
EXIT
ENDCASE
*
gl_chgrec = IIF(gn_reco = RECNO(), .F., .T.) && See if record # changed
*
IF gl_rollbck
gl_rollbck = .F.
ROLLBACK
ENDIF
ENDDO
*
END TRANSACTION

```

APENDICE A.

```
IF .NOT. ROLLBACK()
  DO Pause WITH "Undo not successful"
  ACTIVATE SCREEN
ENDIF
DO Chkdele
ENDDO
```

```
IF lc_status = "ON"
  SET STATUS
ENDIF
SET CURSOR &lc_cursor.
SET TALK &lc_talk.
```

```
ON KEY
ON ERROR
```

```
SELE 1
CLOSE DATABASE
```

```
RELEASE MENU Editmenu
RELEASE POPUPS records, go_to, exit
RELEASE WINDOWS table, pause, seek, search, bsearch, Invo_hlp, wndow1
RESTORE SCREEN FROM Invoice2
RELEASE lc_talk, lc_fields, lc_status, lc_escape
RELEASE SCREEN Invoice2
RETURN
```

PROCEDURE Show_get

```
① 1,0 TO 21,79 COLOR gr+/b
② 1,28 SAY " Master Systems Invoice " COLOR r+/b
③ 3,2 SAY "Client:"
④ 3,11 GET Client->Client PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
⑤ 3,56 SAY "Client No.:"
⑥ 3,69 GET Transact->Client_id PICTURE "IXXXX"
⑦ 4,2 SAY "Name:"
⑧ 4,11 GET Client->Lastname PICTURE "XXXXXXXXXXXXXXXXXXXX"
⑨ 4,28 GET Client->Firstname PICTURE "XXXXXXXXXXXXXXXXXXXX"
⑩ 4,56 SAY "Order No.:"
⑪ 4,69 GET Transact->Order_id PICTURE "XX-XXX"
⑫ 5,2 SAY "Address:"
⑬ 5,11 GET Client->Address PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
⑭ 5,56 SAY "Trans. Date.:"
⑮ 5,69 GET Transact->Date_trans
⑯ 6,2 SAY "City.:"
⑰ 6,11 GET Client->City PICTURE "XXXXXXXXXXXXXXXXXXXX"
⑱ 6,32 SAY "State.:"
⑲ 6,39 GET Client->State PICTURE "!"
⑳ 6,42 SAY "Zip.:"
㉑ 6,47 GET Client->Zip PICTURE "XXXXXXXXXX"
㉒ 7,2 SAY "Phone.:"
㉓ 7,11 GET Client->Phone PICTURE "(999)999-9999"
```

APENDICE A.

```
@ 7,26 SAY "Client History"  
@ 7,41 GET Client->Clie_hist WINDOW Wndow1  
CLEAR GETS  
RETURN
```

PROCEDURE Edit_get

```
@ 3,11 GET Client->Client PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"  
@ 3,69 GET Transact->Client_id PICTURE "IXXXXX";  
VALID client_id <> space(6);  
ERROR "You must enter a client number";  
WHEN client_id = space(6);  
MESSAGE "Enter a customer number"  
@ 4,11 GET Client->Lastname PICTURE "XXXXXXXXXXXXXXXXXXXX"  
@ 4,28 GET Client->Firstname PICTURE "XXXXXXXXXXXXXXXXXXXX"  
@ 4,69 GET Transact->Order_id PICTURE "XX-XXX";  
VALID order_id <> space(8);  
ERROR "You must enter a order number";  
WHEN ORDER_ID=SPACE(8);  
MESSAGE "Enter a new order number"  
@ 5,11 GET Client->Address PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"  
@ 5,69 GET Transact->Date_trans;  
DEFAULT date()  
@ 6,11 GET Client->City PICTURE "XXXXXXXXXXXXXXXXXXXX"  
@ 6,39 GET Client->State PICTURE "I"  
@ 6,47 GET Client->Zip PICTURE "XXXXXXXXXXXX"  
@ 7,11 GET Client->Phone PICTURE "(999)999-9999"  
@ 7,41 GET Client->Clie_hist WINDOW Wndow1  
RETURN
```

PROCEDURE Showbrow

```
SELE STOCK  
SET FILTER TO  
SEEK &GC_ALIAS.->ORDER_ID  
IF EOF()  
APPEND BLANK  
REPLACE STOCK->ORDER_ID WITH &GC_ALIAS.->ORDER_ID  
ENDIF  
SET FILTER TO &GC_ALIAS.->ORDER_ID = STOCK->ORDER_ID  
GO TOP  
KEYBOARD CHR(27)  
DO Browselt  
SELE &gc_alias.  
ACTIVATE SCREEN  
RETURN
```

PROCEDURE LineItem

```
In_key = INKEY()  
DO Keykill
```

APENDICE A.

```
SELE STOCK
APPEND BLANK
REPLACE STOCK->ORDER_ID WITH &GC_ALIAS.->ORDER_ID
GO TOP
DO Browseit
ACTIVATE SCREEN
DO Keyset
```

```
SELE &gc_alias.
RETURN
```

```
PROCEDURE Browseit
SET CURSOR ON
BROWSE;
FIELDS PART_ID, PART_NAME/12, DESCRIPT/20, QTY, ITEM_COST,
EXTENDED=QTY*ITEM_COST,STOCK->ORDER_ID /R;
WINDOW table COMPRESS NOMENU NOCLEAR
SET CURSOR OFF
RETURN
```

```
PROCEDURE Keyset
ON KEY LABEL F1 DO Invo_hlp
ON KEY LABEL F9 KEYBOARD CHR(23)+"L"
ON KEY LABEL F10 KEYBOARD CHR(23)+"M"
RETURN
```

```
PROCEDURE Keykill
ON KEY LABEL F1 DO Nothing
ON KEY LABEL F9 DO Nothing
ON KEY LABEL F10 DO Nothing
RETURN
```

```
PROCEDURE Nothing
in_key=INKEY()
RETURN
```

```
PROCEDURE Chkdele
IF DELETED()
DEFINE BAR 4 OF records PROMPT " Clear deletion mark";
MESSAGE "Mark/unmark this record for deletion"
ELSE
DEFINE BAR 4 OF records PROMPT " Mark record for deletion";
MESSAGE "Mark/unmark this record for deletion"
ENDIF
RETURN
```

```
PROCEDURE Recappnd
GO BOTT
```

APENDICE A.

```
morder = ORDER_ID
APPEND BLANK
REPLACE order_id WITH SUBSTR(morder,1,3) + LTRIM(STR(VAL(SUBSTR(morder,4))+1))

RETURN

PROCEDURE Get_recs
gn_pick = BAR() && Variable for bar testing
DO CASE
CASE gn_pick = 1

    gl_rollbck = .T.
CASE gn_pick = 3
DO Recappnd
CASE gn_pick = 4

IF DELETE()
RECALL
SELE STOCK
RECALL ALL
ELSE
DELETE
SELE STOCK
DELETE ALL
ENDIF
SELE (gc_alias)
CASE gn_pick = 5

ENDCASE
DO Chkdele
DEACTIVATE MENU
RETURN

PROCEDURE Get_goto

gn_pick = BAR() && Variable for bar testing
mpict = REPLICATE("9", LEN( LTRIM( STR( RECCOUNT() ) ) ) )
gc_search = gc_search + SPACE( 200 - LEN(gc_search) )
gc_seek = gc_seek + SPACE( LEN( ORDER_ID ) - LEN(gc_seek) )
DO CASE
CASE gn_pick = 1

GO TOP
CASE gn_pick = 2

GO BOTTOM
CASE gn_pick = 3

@ 4,39 GET gn_gorec RANGE 1, RECCOUNT() PICTURE mpict
```

APENDICE A.

```
READ
gn_gorec = IIF( gn_gorec=0, RECNO(), gn_gorec)
GO gn_gorec
CASE gn_pick = 4

skiprec = IIF( RECCOUNT() > 9, 10, 5)
SET DELI OFF
@ 5,39 GET skiprec PICTURE mpict
READ
SET DELI ON
SKIP skiprec
IF EOF()
SKIP -1
ENDIF
CASE gn_pick = 6
ACTIVATE WINDOW seek

IF " = ORDER()
tempmdx = TAG(1)
SET ORDER TO TAG &tempmdx.
ENDIF
@ 0,1 SAY "Enter search string for"
@ 1,1 SAY TAG(1)+":." GET gc_seek PICT "@S20";
MESSAGE "Cancel: Esc";
VALID LEN( TRIM( gc_seek)) > 0;
ERROR "No search condition entered"
READ
IF READKEY() <> 12
gc_seek = LTRIM( TRIM(gc_seek))
mrec = RECNO()
SEEK gc_seek
IF .NOT. FOUND()
GO mrec
ENDIF
ENDIF
DEACTIVATE WINDOW seek
ACTIVATE SCREEN
CASE gn_pick = 7

ACTIVATE WINDOW search
@ 0,1 SAY "Enter search string:" GET gc_search PICT "@S21";
MESSAGE "Cancel: Esc";
VALID LEN( TRIM( gc_search)) > 0;
ERROR "No search condition entered"
READ
DEACTIVATE WINDOW search
ACTIVATE SCREEN
IF READKEY() <> 12
gc_search = LTRIM( RTRIM( gc_search))
mrec = RECNO()
```

APENDICE A.

```
LOCATE REST FOR &gc_search.
IF .NOT. FOUND()
  GO mrec
ENDIF
ENDIF
CASE gn_pick = 8

  ACTIVATE WINDOW bsearch
  @ 0,1 SAY "Enter search string." GET gc_search PICT "@S21";
  MESSAGE "Cancel: Esc";
  VALID len(trim(gc_search)) > 0;
  ERROR "No search condition entered"
  READ
  DEACTIVATE WINDOW bsearch
  ACTIVATE SCREEN
  IF READKEY() <> 12
    gc_search = LTRIM( RTRIM( gc_search))
    mrec = RECNO()

DO WHILE .NOT. (BOF() .OR. &gc_search.)
  SKIP -1
ENDDO
IF BOF()
  GO mrec
ENDIF
ENDIF
ENDCASE
DEACTIVATE MENU
RETURN

PROCEDURE Get_exit
  CLEAR GETS

  gl_extloop = .T.
  DEACTIVATE MENU
  RETURN

PROCEDURE Pause
  PARAMETER lc_msg
  IF TYPE("lc_message")="U"
    gn_error=ERROR()
  ENDIF
  lc_msg = lc_msg
  lc_option="0"
  ACTIVATE WINDOW Pause
```

APENDICE A.

```
IF gn_error > 0
  IF TYPE("lc_message")="U"
    @ 0,1 SAY [An error has occurred !! - Error message: ]+MESSAGE()
  ELSE
    @ 0,1 SAY [Error # ]+lc_message
  ENDIF
ENDIF
@ 1,1 SAY lc_msg
WAIT " Press any key to continue..."
DEACTIVATE WINDOW Pause
RETURN
```

```
FUNCTION Center
PARAMETER lc_string, ln_width
RETURN ((ln_width/2)-(LEN(lc_string)/2))
```

```
PROCEDURE Invo_hlp
gc_readvar = VARREAD()
ACTIVATE WINDOW Invo_hlp
CLEAR
TEXT
```

Navigation HELP:

F1: Displays this message
F9: Takes you from the top of your invoice to the line items
F10: Takes you to the menu system at the top of the invoice

PgDn: Takes you to the next invoice
PgUp: Takes you to the previous invoice
Ctrl-PgDn: Takes you to the last invoice in your database
Ctrl-PgUp: Takes you to the first invoice in your database
Esc: Exits the invoice

```
ENDTEXT
@ 16,5 say "Press any key..."
x = INKEY(0)
CLEAR
TEXT
```

Line Item HELP:

F1: Displays this message

PgDn: Displays next screen of line items
PgUp: Displays previous screen of line items
Ctrl-PgDn: Takes you to the last screen of line items
Ctrl-PgUp: Takes you to the first screen of line items
Esc: Returns to header information part of invoice

APENDICE A.

```
ENDTEXT
@ 16.5 say "Press any key..."
x = INKEY(0)
DEACTIVATE WINDOW Invo_hlp
ACTIVATE SCREEN
RETURN
```

PROCEDURE Defnmenu

SET BORDER TO DOUBLE

DEFINE MENU editmenu

```
DEFINE PAD records OF editmenu PROMPT "Records" AT 0,2
ON PAD records OF editmenu ACTIVATE POPUP records
DEFINE PAD go_to OF editmenu PROMPT "Go To" AT 0,14
ON PAD go_to OF editmenu ACTIVATE POPUP go_to
DEFINE PAD exit OF editmenu PROMPT "Exit" AT 0,24
ON PAD exit OF editmenu ACTIVATE POPUP exit
```

DEFINE POPUP records FROM 1,0

```
DEFINE BAR 1 OF records PROMPT " Undo change to record";
MESSAGE "Undo Change to current record"
DEFINE BAR 2 OF records PROMPT "" SKIP
DEFINE BAR 3 OF records PROMPT " Add new records";
MESSAGE "Add records to the end of the database file"
DO Chkdelte && Define bar 4 done in procedure
DEFINE BAR 5 OF records PROMPT " Blank record" SKIP ;
MESSAGE "Erase the contents of all the fields in this record"
DEFINE BAR 6 OF records PROMPT " Lock record";
MESSAGE "" SKIP
DEFINE BAR 7 OF records PROMPT " Follow record to new position YES" SKIP
DEFINE POPUP go_to FROM 1,12
DEFINE BAR 1 OF go_to PROMPT " Top record";
MESSAGE "Move to the first record in this database file"
DEFINE BAR 2 OF go_to PROMPT " Last record";
MESSAGE "Move to the last record in this database file"
DEFINE BAR 3 OF go_to PROMPT " Record number";
MESSAGE "Move to the specified record number"
DEFINE BAR 4 OF go_to PROMPT " Skip";
MESSAGE "Move by skipping the specified number of records (minus for backwards)"
DEFINE BAR 5 OF go_to PROMPT "" SKIP
DEFINE BAR 6 OF go_to PROMPT " Index key search" SKIP FOR "" = gc_mdx;
MESSAGE "Use the index file to search for the specified string"
DEFINE BAR 7 OF go_to PROMPT " Forward search ";
MESSAGE "Search a field for the specified string from the current record forward"
DEFINE BAR 8 OF go_to PROMPT " Backward search ";
MESSAGE "Search a field for the specified string from the current record backward"
DEFINE BAR 9 OF go_to PROMPT " Match capitalization NO" SKIP
```

APENDICE A.

```
DEFINE POPUP exit FROM 1,22
  DEFINE BAR 1 OF exit PROMPT " Exit";
  MESSAGE "Save changes to current record and exit"
  DEFINE BAR 2 OF exit PROMPT " Transfer to Query Design " SKIP
```

```
ON SELECTION POPUP records DO Get_recs
ON SELECTION POPUP go_to DO Get_goto
ON SELECTION POPUP exit DO Get_exit
```

```
DEFINE WINDOW seek FROM 8,15 TO 11,44
DEFINE WINDOW search FROM 9,15 TO 11,60
DEFINE WINDOW bsearch FROM 10,15 TO 12,60
```

```
SHOW MENU Editmenu
RETURN
```

APENDICE A.

INVOICES.PRG

```
SET COLOR TO &c_normal.
CLEAR
answer = "N"
inv_month = 0
@ 6,6 to 16,73 DOUBLE
@ 8,31 SAY "INVOICE PRINTING"
@ 10,12 SAY "Are you sure you want to print invoices (Y/N) ?";
    GET answer PICTURE "Y"
@ 12,8 SAY "*****"
@ 13,8 SAY " This will process and alter current data in: Acct_rec "
@ 14,8 SAY "          Orders "
READ
IF answer = "N"
    RETURN TO Business
ELSE
    ?? CHR(7)
    inv_month = MONTH(DATE())-1
    @ 10,8 CLEAR TO 14,72
    @ 12,12 SAY "What month do want to process orders for (01-12)?";
        GET inv_month PICTURE "99" RANGE 1,12
    READ
ENDIF

SELECT 1
USE c:\dbase\bases\Orders ORDER Order
USE c:\dbase\bases\Cust ORDER Cust_id IN 2
USE c:\dbase\bases\Acct_rec ORDER Cust_id IN 3
USE c:\dbase\bases\Goods ORDER part_id IN 4
IF .NOT. FILE("Archiv_o.dbf")
    COPY STRUCTURE TO Archiv_o
ENDIF
USE Archiv_o IN 5
SET RELATION TO cust_id INTO Cust, cust_id INTO Acct_rec, part_id INTO Goods
GO TOP
DEFINE WINDOW err_wndo FROM 16,3 TO 23,60 COLOR &c_pop.
ON ESCAPE DO Stop_rpt
ON ERROR DO Err_msg
SET SPACE OFF
    _plineno = 0
    _pject = "NONE"
    _pageno = 1
    continu_on = .T.
    complete = .F.
    on_pg_line = 0
```

APENDICE A.

```
STORE 0 TO amt_of_bil, amt_of_cur, inv_amount, oldbalance
STORE 0 TO inv_count, ord_count, grand_tot, tot_price
STORE "" TO invoice_no, mcust_id, today, this_year, this_month
today = DTOC(DATE())
this_year = RIGHT(today,2)
this_month = LEFT(today,2)

on_pg_line = INT(_plength - 6)

ON PAGE AT LINE on_pg_line DO Page_brk

SET CONSOLE off
SET PRINTER on

PRINTJOB

SCAN FOR .NOT. Invoiced .AND. Inv_month = MONTH(date_trans) ;
  WHILE continu_on
    mcust_id = cust_id
    DO Pg_head
    DO Inv_head
    complete = .F.

SCAN FOR .NOT. invoiced .AND. Inv_month = MONTH(date_trans) ;
  WHILE cust_id = mcust_id .AND. continu_on
    DO Detail
  ENDSCAN
  complete = .T.
  SKIP -1
  DO Inv_calc
  EJECT
  DO Updat_ar
  DO Reinit
  ENDSCAN
  IF continu_on
    message = "Invoices were completely processed and printed for month " ;
      + STR(inv_month,2)
  ELSE

    message = "Invoices were NOT COMPLETED - stopped by user at " + TIME()
  ENDIF
  DO Rpt_end WITH message
  ON PAGE
ENDPRINTJOB
EJECT PAGE
ON PAGE
SET CONSOLE on
SET PRINTER off
```

APENDICE A.

```
CLEAR
answer = ""
@ 6, 6 to 15,74 DOUBLE
@ 8,33 SAY "ARCHIVE ORDERS"
@ 10,12 SAY "Do you want to archive the processed orders (Y/N) ?" ;
  GET answer PICTURE "Y"
@ 12,10 SAY "*****"
@ 13,10 SAY "(Copy processed orders to archive file and delete from Orders)"
READ
IF answer = "Y"
  ?? CHR(7)
  @ 10,12 SAY "COPYING processed orders...please wait"
  @ 12,10 SAY SPACE(81)
  @ 13,10 SAY SPACE(83)
  SELECT Archiv_o
  APPEND FROM Orders FOR Invoiced
  SELECT Orders
  SET TALK on
  DELETE ALL FOR Invoiced
  @ 10,10 SAY "ERASING processed orders...please wait"
  PACK
  SET TALK off
ELSE
  ?? CHR(7)
  @ 10, 10 CLEAR TO 14, 73
  @ 10,10 SAY "Processed orders will not be erased"
  @ 12,10 SAY "Please press spacebar to continue..."
  WAIT ""
ENDIF
ON ESCAPE
ON ERROR
CLOSE ALL
RETURN TO Business
```

PROCEDURE Detail

```
?? date_trans AT 0,
?? part_id AT 10,
?? Goods->part_name AT 21,
?? part_qty AT 53 PICTURE "999",
?? Goods->price AT 58 PICTURE "99,999.99",
* Extend price
tot_price = ROUND(part_qty * Goods->price,2)
?? tot_price AT 70 PICTURE "99,999.99"
?
amt_of_cur = amt_of_cur + tot_price
ord_count = ord_count + 1

REPLACE Invoiced WITH .T.
```

APENDICE A.

RETURN

PROCEDURE Err_msg

```
SET PRINTER off
SET CONSOLE on
ACTIVATE WINDOW err_wndo
CLEAR
? "----- ERROR WARNING -----"
IF .NOT. PRINTSTATUS()

    msg = "Printer error - fix paper or select ONLINE button"
    ELSE

    msg = MESSAGE()
ENDIF
? " Error Number ", LTRIM(STR(ERROR(),4))
? " " + msg
WAIT " Press spacebar to continue..."
DEACTIVATE WINDOW err_wndo
SET CONSOLE off
SET PRINTER on
RETURN
```

PROCEDURE Inv_calc

```
amt_of_bil = amt_of_cur + oldbalance
?? "-----" AT 69
?
?? "CURRENT ORDERS" AT 0,
?? "$" AT 66,
?? amt_of_cur PICTURE "999,999.99" AT 69
?
IF oldbalance <> 0
?? "-----" AT 69
? "+ OLD BALANCE"
?? oldbalance PICTURE "999,999.99" AT 69,
?
ENDIF
?? "======" AT 69
?
?? "TOTAL AMOUNT DUE" STYLE "B" AT 0,
?? "$" STYLE "B" AT 66,
?? amt_of_bil PICTURE "999,999.99" STYLE "B" AT 69
?
?? "======" AT 69

grand_tot = grand_tot + amt_of_bil
?
?
RETURN
```

APENDICE A.

PROCEDURE Inv_foot

?
? "TERMS: " AT 27, Cust->terms
?
? Acct_rec->notes AT 18

EJECT PAGE
RETURN

PROCEDURE Inv_head

invoice_no = cust_id + this_year + this_month

inv_count = inv_count + 1
?
?? "INVOICE NO.: " STYLE "B" AT 0,
?? invoice_no STYLE "B" FUNCTION "T" PICTURE "XXXXXXXX",
?? DATE() AT 69
?
?
?? "CUSTOMER NO.: " AT 0,
?? cust_id FUNCTION "T" PICTURE "XXXXXX"
?
?
?? Cust->customer AT 0
?
?? Cust->address1 AT 0, Cust->address2 AT LEN(TRIM(Cust->address1))+2
?
?? Cust->city PICTURE "@T XXXXXXXXXXXXXXXXXXXX" AT 0,
?? " ",
?? Cust->state, " ",
?? Cust->zip
?
?? "ATTENTION: " AT 0,
?? Cust->contact PICTURE "@T XXXXXXXXXXXXXXXXXXXX", "
?? Cust->phone_cont
?
? REPLICATE(CHR(205),80)
?
?
?? "PREVIOUS ACTIVITY:" STYLE "BU" AT 0
?
?? "INVOICE NO.:" AT 4, Acct_rec->invoic_old AT 15
?? "SENT:" AT 31, Acct_rec->dat_istbil AT 37
?
?? "AMOUNT \$" AT 4, Acct_rec->amt_istbil PICTURE "999,999.99" AT 15
?
?? "PAID \$" AT 4, Acct_rec->amt_istpd PICTURE "999,999.99" AT 15

APENDICE A.

```
?  
?? "-----" AT 15  
?  
?? "BALANCE $" AT 4  
oldbalance = Acct_rec->oldbalance  
?? oldbalance PICTURE "999,999.99" AT 15  
?  
?  
?? "CURRENT ACTIVITY:" STYLE "BU" AT 0  
? REPLICATE(CHR(196),80)  
? "Ordered" AT 0  
?? "Part no." AT 10  
?? "Part name" AT 21  
?? "Qty" AT 53  
?? "Price" AT 59  
?? "Total" AT 74  
? REPLICATE(CHR(196),80)  
?  
RETURN  
  
PROCEDURE Page_brk  
  
DO Inv_foot  
  
IF .NOT. EOF() .AND. .NOT. complete  
DO Pg_head  
ENDIF  
RETURN  
  
PROCEDURE Pg_head  
  
?  
? "Page ",  
?? _pageno PICTURE "999"  
?  
?  
? "ISCAR DE MEXICO, S.A. DE C.V." STYLE "B" AT 27  
?  
DEFINE BOX FROM 34 TO 45 HEIGHT 3 SINGLE  
?  
?? "INVOICE" STYLE "B" AT 36  
?  
?  
? REPLICATE(CHR(205),80)  
?  
RETURN
```

APENDICE A.

PROCEDURE Reinit

```
STORE 0 TO amt_of_cur, inv_amount
_pageno = 1
RETURN
```

PROCEDURE Rpt_end
PARAMETERS message

```
?
?
? "ISCAR DE MEXICO, S.A. DE C.V. "BU" AT 27
?
? "INVOICE SUMMARY PAGE" STYLE "B" AT 30
?
inv_date = CTOD(STR(inv_month,2)+RIGHT(DTOC(DATE()),6))
? "FOR MONTH of " AT 31, CMONTH(inv_date)
?
?
? REPLICATE(CHR(205),80
? DATE() AT 0,
?? TIME() AT 69
? REPLICATE(CHR(205),80)
?
?
?
?? "======" AT 67,
?
?? "GRAND TOTAL for " AT 0,
?? inv_count PICTURE "999",
?? " Invoices " AT 21,
?? "and ", ord_count PICTURE "9,999",
?? " orders:",
?? "$" AT 66,
?? grand_tot PICTURE "999,999.99",
?
?? "======" AT 67
?
?
?
? message AT 6
?
```

RETURN

PROCEDURE Stop_rpt
continu_on = .F.
RETURN

APENDICE A.

PROCEDURE Updat_ar

```
SELECT Acct_rec  
IF Orders->cust_id <> cust_id
```

APPEND BLANK

```
REPLACE cust_id WITH Orders->cust_id
```

ENDIF

```
REPLACE invoic_old WITH invoice_no, dat_1stbil WITH dat_of_bil ;  
amt_1st_pd WITH amt_cur_pd, amt_1stbil WITH amt_of_cur ;  
oldbalance WITH amt_1stbil - amt_1st_pd, comments WITH "" ;  
notes WITH "", invoice_no WITH m->invoice_no ;  
dat_of_bil WITH DATE(), amt_of_cur WITH m->amt_of_cur ;  
amt_of_bil WITH m->amt_of_bil
```

```
SELECT Orders
```

RETURN

APENDICE A.

PROCEDURE Orders

SET PROCEDURE TO c:\dbase\bases\Library

DO Set_env
SET NEAR on
SET COLOR TO &c_standard.

STORE "" TO cust_id, po_number, emp_id, part_id
STORE { / / } TO date_trans
part_qty = 0
invoiced = .F.

dbf = "ORDERS"
mlist = "NOT APPLICABLE"
STORE "N/A" TO cust_rpt
key = "m->cust_id+DTC(m->date_trans)+m->po_number"
key1 = "m->cust_id"
key2 = "m->date_trans"
key3 = "m->po_number"
keyname1 = "Cust ID:"
keyname2 = "Date of Order:"
keyname3 = "P.O. Number:"
list_flds = "CUST_ID,DATE_TRANS,PO_NUMBER,PART_ID,PART_QTY,Goods->PRICE"
STORE "" TO mcustid, mpartid, mempid

SELECT 1
USE c:\dbase\bases\Orders ORDER Order
USE c:\dbase\bases\Goods ORDER Part_id IN 2
USE c:\dbase\bases\Cust ORDER Cust_id IN 3
USE c:\dbase\bases\Employee ORDER Emp_id IN 4
SET RELATION TO part_id INTO Goods, cust_id INTO Cust, emp_id INTO Employee
GO TOP

record_num = RECNO()
DO Load_fld

CLEAR
DO Delstat
DO Backgmd
DO Show_data
DO Bar_def

APENDICE A.

```
SET COLOR TO &c_popup.  
ACTIVATE POPUP main_mnu  
DO Sub_ret  
RETURN
```

PROCEDURE Filter

```
choice = IIF(filters_on,"T","Y")  
DO Filt_ans  
IF choice = "Y"  
    mcustid = SPACE(6)  
    mpartid = SPACE(10)  
    mempid = SPACE(11)  
    ACTIVATE WINDOW alert  
  
    @ 0, 0 SAY "----- ENTER FILTER CONDITION -----"  
    @ 2, 0 SAY "CUST.I.D.:" GET mcustid FUNCTION "I";  
    MESSAGE "Enter a six digit customer ID beginning with a " +  
        " letter - Esc to quit"  
    @ 3, 0 SAY "PART I.D.:" GET mpartid FUNCTION "I"  
    @ 4, 0 SAY "EMPLOYEE I.D.:" GET mempid  
    @ 5, 0 SAY "Enter one or more conditions"  
    READ  
    DEACTIVATE WINDOW alert  
    subset = ""  
    mcustid = TRIM(mcustid)  
    mpartid = TRIM(mpartid)  
    mempid = TRIM(mempid)  
    subset = subset + IIF("" <> mcustid,"cust_id = '&mcustid.' .AND. ","")  
    subset = subset + IIF("" <> mpartid,"part_id = '&mpartid.' .AND. ","")  
    subset = subset + IIF("" <> mempid,"emp_id = '&mempid.' .AND. ","")  
    IF "" = TRIM(subset)  
        DO Wambell  
        filters_on = .F.  
    ELSE  
        subset = SUBSTR(subset,1,LEN(subset)-6)  
        SET FILTER TO &subset.  
        GO  
        filters_on = .NOT. EOF()  
        IF .NOT. filters_on  
            DO Wambell  
            DO Show_msg WITH "No Orders records match the Filter condition."  
            SET FILTER TO  
            GO record_num  
        ENDIF  
    ENDIF  
ELSE  
    SET FILTER TO  
    filters_on = .F.  
ENDIF  
RETURN
```

APENDICE A.

PROCEDURE Indexer

```
INDEX ON cust_id+DTOC(date_trans)+po_number TAG Order
GO TOP
RETURN
```

PROCEDURE Init_fid

```
cust_id = SPACE(6)
date_trans = DATE()
po_number = SPACE(5)
emp_id = SPACE(11)
part_id = SPACE(10)
part_qty = 0
Invoiced = .F.
RETURN
```

PROCEDURE Load_fid

```
cust_id = cust_id
date_trans = date_trans
po_number = po_number
emp_id = emp_id
part_id = part_id
part_qty = part_qty
Invoiced = Invoiced
RETURN
```

PROCEDURE Repl_fid

```
REPLACE cust_id WITH m->cust_id, po_number WITH m->po_number,;
date_trans WITH m->date_trans, emp_id WITH m->emp_id,;
part_id WITH m->part_id, part_qty WITH m->part_qty,;
Invoiced WITH m->Invoiced
RETURN
```

FUNCTION Prof_mgn

PARAMETERS cost,price

```
margin = ROUND((price-cost)/price*100,1)
RETURN margin
```

PROCEDURE Backgmd

- ① 1,18 TO 3,40 DOUBLE COLOR &c_blue.
- ② 5, 2 TO 8,56 DOUBLE COLOR &c_red.
- ③ 18, 2 TO 18,56 COLOR &c_red.
- ④ 9, 2 TO 18,56 COLOR &c_red.
- ⑤ 2,19 FILL TO 2,48 COLOR &c_blue.
- ⑥ 6, 3 FILL TO 7,55 COLOR &c_red.

APENDICE A.

```

@ 10, 3 FILL TO 17,55 COLOR &c_red.
@ 6, 3 FILL TO 17,55 COLOR &c_red.
SET COLOR TO &c_data.
@ 2,20 SAY "ORDERS TRANSACTIONS DATABASE"
@ 6, 4 SAY "CUSTOMER ID:"
@ 7, 4 SAY "ORDER DATE:"
@ 7,35 SAY "P.O. NUMBER:"
@ 10, 4 SAY "PART #:"
@ 11, 4 SAY "PART NAME:"
@ 12, 4 SAY "QTY. ORDERED:"
@ 12,25 SAY "each"
@ 12,35 SAY "PRICE: $"
@ 13, 4 SAY "QTY. AVAILABLE:"
@ 13,25 SAY "each"
@ 13,35 SAY "MARGIN:"
@ 13,53 SAY "%"
@ 14, 4 SAY "EMPLOYEE #:"
@ 15, 4 SAY "INVOICED:"
@ 17, 4 SAY "NOTES:"
SET COLOR TO &c_standard.
RETURN
    
```

PROCEDURE Show_data

```

SET COLOR TO &c_fields.
@ 6,18 SAY cust_id
@ 7,18 SAY date_trans
@ 7,48 SAY po_number
@ 10,18 SAY part_id
@ 12,21 SAY part_qty PICTURE "999"
@ 14,16 SAY emp_id
@ 15,14 SAY Invoiced PICTURE "Y"
@ 17,14 SAY Notes
IF .NOT. BAR() = 2
    @ 6,26 SAY Cust->Customer.
    @ 11,18 SAY Goods->Part_name.
    @ 12,44 SAY Goods->Price PICTURE "99,999.99" COLOR &c_yelowhit.
    @ 13,21 SAY Goods->Qty_onhand PICTURE "999" COLOR &c_yelowhit.
    @ 13,48 SAY Prof_mgn(Goods->Cost, Goods->Price);
        PICTURE "99.9" COLOR &c_yelowhit.
    @ 14,30 SAY TRIM(Employee->Firstname)+" "+ Employee->Lastname;
        COLOR &c_yelowhit.
ELSE
    @ 6,26 SAY SPACE(30) && CUSTOMER
    @ 11,18 SAY SPACE(20) && PARTNAME
    @ 12,44 SAY SPACE(8) && PRICE
    @ 13,21 SAY SPACE(3) && QTY ONHAND
    @ 13,48 SAY SPACE(4) && MARGIN
    @ 14,30 SAY SPACE(26) && EMPLOYEE
    
```

APENDICE A.

```

ENDIF
IF ISCOLOR()
  @ 20, 4 SAY "Yellow text/numbers - from related databases or calc." ;
  COLOR &c_yelowhit.
ELSE
  @ 20, 4 SAY "Dim text/numbers - from related databases or calc." ;
  COLOR &c_red.
ENDIF
SET COLOR TO &c_standard.
RETURN

PROCEDURE Get_data

SET COLOR TO &c_data.
@ 6,18 GET m->cust_id PICTURE "199999" ;
  VALID Lookupid(m->cust_id,"Cust","Customer", 2) ;
  ERROR "Invalid customer I.D. number, please re-enter." ;
  MESSAGE "Enter a six digit customer ID (beginning with a " +
    "letter) - Esc to quit"
@ 7,18 GET m->date_trans FUNCTION "D" ;
  MESSAGE "Enter date of this order"
@ 7,48 GET m->po_number FUNCTION "I" ;
  MESSAGE "Enter customer's P.O. number"
@ 10,18 GET m->part_id FUNCTION "I" ;
  VALID Lookupid(m->part_id,"Goods", "Part", 3) ;
  ERROR "Invalid part ID number, please re-enter." ;
  MESSAGE "Enter a part ID number, or Esc to quit"
@ 12,21 GET m->part_qty PICTURE "999" ;
  MESSAGE "Enter quantity of parts ordered"
@ 14,16 GET m->emp_id PICTURE "999-99-9999" ;
  VALID Lookupid(m->emp_id, "Employee", "Employee", 6) ;
  ERROR "Invalid employee ID number, please re-enter." ;
  MESSAGE "Enter an employee ID number, or Esc to quit"
@ 15,14 GET m->Invoiced PICTURE "Y" ;
  MESSAGE "Enter whether this order has been Invoiced " +
    "(usually done by system)"
@ 17,14 GET Notes WINDOW memo_windo ;
  MESSAGE "Enter notes into memo field, press " +
    "Ctrl-Home to enter (Ctrl-End to exit)"
IF NOT. BAR() = 2 && not Add mode
  @ 8,26 SAY Cust->Customer COLOR &c_yelowhit.
  @ 11,18 SAY Goods->Part_name COLOR &c_yelowhit.
  @ 12,44 SAY Goods->Price PICTURE "99,999.99" COLOR &c_yelowhit.
  @ 13,21 SAY Goods->Qty_onhand PICTURE "999" COLOR &c_yelowhit.
  @ 13,48 SAY Prof_mgn(Goods->Cost,Goods->Price) ;
    PICTURE "99.9" COLOR &c_yelowhit.
  @ 14,30 SAY TRIM(Employee->Firstname)+" "+ Employee->Lastname ;
    COLOR &c_yelowhit.
ELSE

```

APENDICE A.

```
@ 6,26 SAY SPACE(30)  && CUSTOMER
@ 11,18 SAY SPACE(20)  && PARTNAME
@ 12,44 SAY SPACE(9)   && PRICE
@ 13,21 SAY SPACE(3)   && QTY ONHAND
@ 13,48 SAY SPACE(4)   && MARGIN
@ 14,30 SAY SPACE(26)  && EMPLOYEE
ENDIF
IF ISCOLOR()
  @ 20, 4 SAY "Yellow text/numbers - from related databases or calc." ;
  COLOR &c_yelowhit.
ELSE
  @ 20, 4 SAY "Dim text/numbers - from related databases or calc." ;
  COLOR &c_red.
ENDIF
SET COLOR TO &c_standard.
ON KEY LABEL F9 DO Findcust WITH m->cust_id
ON KEY LABEL F10 DO Findpart WITH m->part_id
RETURN
.
```

APENDICE A.

PROCEDURE Vendors

SET PROCEDURE TO Library

DO Set_env

SET COLOR TO &c_standard.

discount = 0
STORE "" TO vendor_id, vendor, address1, address2, city, state
STORE "" TO zip, phone, contact, phone_ext, terms

STORE "VENDORS" TO dbf, mlist
cust_rpt = "N/A"
STORE "m->vendor_id" TO key, key1
STORE "NONE" TO key2, key3
keyname1 = "Vendor ID:"
STORE "" TO keyname2, keyname3
list_fds = "VENDOR_ID, VENDOR, PHONE"

SELECT 1
USE Vendors ORDER Vendor_id
GO TOP

USE Codes ORDER City IN 2

record_num = RECNO()
DO Load_fid

CLEAR
DO Dstatus
DO Backgmd
DO Show_data

DO Bar_def

SET COLOR TO &c_popup.
ACTIVATE POPUP main_mnu
DO Sub_ret

RETURN

APENDICE A.

PROCEDURE Filter

```

* Filter (group) data into subset
* Select subset to set up filter condition (Y=turn on, N=abort selection,
* T=turn off). If filter is already on, set default choice to T, show
* window. If filter is not on, set default choice to Y, show window.
choice = IIF(filters_on,"T","Y")
DO Fill_ans
IF choice = "Y"
  * Start process of choosing filter condition
  STORE SPACE(15) TO city,terms
  STORE SPACE(2) TO state
  STORE SPACE(10) TO zip
  ACTIVATE WINDOW alert
  * Get users filter condition selection(s)
  @ 0,0 SAY "----- ENTER FILTER CONDITION -----"
  @ 1,1 SAY "CITY: " GET m->city PICTURE "XXXXXXXXXXXXXXXX"
  @ 2,1 SAY "STATE: " GET m->state PICTURE "I"
  @ 3,1 SAY "ZIP: " GET m->zip
  @ 4,1 SAY "TERMS: " GET m->terms FUNCTION "I"
  READ
  DEACTIVATE WINDOW alert
  * Initialize filter condition variable to null (empty)
  PUBLIC subset1,subset2,subset3,subset4,subset5
  subset1 = ""
  * Process user's entries to build filter condition
  subset2 = subset1 + IIF( [] <> TRIM(m->city), ;
    [UPPER(city) = UPPER(TRIM(m->city)) .AND. ], [] )
  subset3 = subset2 + IIF( [] <> TRIM(m->state), ;
    [state = TRIM(state) .AND. ], [] )
  subset4 = subset3 + IIF( [] <> TRIM(m->zip), ;
    [zip = TRIM(zip) .AND. ], [] )
  subset5 = subset4 + IIF( "" <> TRIM(m->terms), ;
    [terms = TRIM(terms) .AND. ], [] )
  subset = subset5
  *
  * Check whether data entered into subset string
  IF "" = TRIM(subset)
    DO Wambell
    filters_on = .F.
  ELSE
    * If string is not empty, truncate the .AND. from end
    subset = SUBSTR(subset, 1, LEN(subset) - 6)
    * Filter on entered filter string condition
    SET FILTER TO &subset.
    * Activate filter by moving record pointer
    GO TOP
    * Check whether filter condition matches any records (no match=EOF)
    filters_on = .NOT. EOF() && Filter is turned on if .T.
    IF .NOT. filters_on
      * Turn off filter if no matching records found

```

APENDICE A.

```
SET FILTER TO  
filters_on = .F.  
ENDIF  
RETURN
```

```
PROCEDURE Indexer  
INDEX ON vendor_id TAG Vendor_id  
GO TOP  
RETURN
```

```
PROCEDURE Init_fid  
STORE SPACE(4) TO vendor_id, phone_ext  
STORE SPACE(30) TO vendor, address1, address2, contact  
terms = SPACE(15)  
discount = 0  
city = SPACE(20)  
state = "TN"  
zip = SPACE(10)  
phone = SPACE(13)  
RETURN
```

```
PROCEDURE Load_fid
```

```
vendor_id = vendor_id  
vendor = vendor  
address1 = address1  
address2 = address2  
city = city  
state = state  
zip = zip  
phone = phone  
contact = contact  
phone_ext = phone_ext  
terms = terms  
discount = discount  
RETURN
```

```
PROCEDURE Repl_fid  
REPLACE vendor_id WITH m->vendor_id, vendor WITH m->vendor, ;  
address1 WITH m->address1, address2 WITH m->address2, ;  
city WITH m->city, state WITH m->state, ;  
zip WITH m->zip, phone WITH m->phone, ;  
contact WITH m->contact, phone_ext WITH m->phone_ext, ;  
terms WITH m->terms, discount WITH m->discount  
RETURN
```

```
PROCEDURE Backgmd
```

```
Ⓢ 14, 5 TO 14, 52 COLOR &c_red.
```

APENDICE A.

```
@ 1,22 TO 3,53 DOUBLE COLOR &c_blue.  
@ 5, 4 TO 7,27 DOUBLE COLOR &c_red.  
@ 8, 4 TO 19,53 COLOR &c_red.  
@ 2,23 FILL TO 2,52 COLOR &c_blue.  
@ 6, 5 FILL TO 6,26 COLOR &c_red.  
@ 9, 5 FILL TO 18,52 COLOR &c_red.
```

```
SET COLOR TO &c_data.  
@ 2,28 SAY "VENDORS DATABASE"  
@ 6, 6 SAY "VENDOR NUMBER:"  
@ 9, 6 SAY "NAME:"  
@ 10, 6 SAY "ADDRESS:"  
@ 12, 6 SAY "CITY:"  
@ 13, 6 SAY "STATE:"  
@ 13,30 SAY "ZIP:"  
@ 15, 6 SAY "CONTACT:"  
@ 16, 6 SAY "PHONE:"  
@ 16,30 SAY "EXTENSION:"  
@ 17, 6 SAY "TERMS:"  
@ 18, 6 SAY "DISCOUNT:"  
@ 18,18 SAY "%"  
SET COLOR TO &c_standard.  
RETURN
```

PROCEDURE Show_data

```
SET COLOR TO &c_fields.  
@ 6,21 SAY vendor_id  
@ 9,15 SAY vendor  
@ 10,15 SAY address1  
@ 11,15 SAY address2  
@ 12,15 SAY city  
@ 13,15 SAY state  
@ 13,35 SAY zip  
@ 15,15 SAY contact  
@ 16,15 SAY phone  
@ 16,41 SAY phone_ext  
@ 17,15 SAY terms  
@ 18,16 SAY discount PICTURE "99"  
SET COLOR TO &c_standard.  
RETURN
```

PROCEDURE Get_data

```
SET COLOR TO &c_data.  
@ 6,21 GET m->vendor_id PICTURE "9999";  
VALID Duplicat(&key.);  
ERROR "Invalid vendor ID number; please re-enter";  
MESSAGE "Enter a four digit vendor ID number, or Esc to quit"  
@ 9,15 GET m->vendor FUNCTION "!";
```

APENDICE A.

```
MESSAGE "Enter vendor name"
@ 10,15 GET m->address1 FUNCTION "I"
@ 11,15 GET m->address2 FUNCTION "I"
@ 12,15 GET m->city PICTURE "XXXXXXXXXXXX"
@ 13,15 GET m->state PICTURE "I"
@ 13,35 GET m->zip
@ 15,15 GET m->contact FUNCTION "I" ;
MESSAGE "Enter name of vendor contact"
@ 16,15 GET m->phone PICTURE "(999)999-9999"
@ 16,41 GET m->phone_ext PICTURE "9999" ;
MESSAGE "Enter phone extension"
@ 17,15 GET m->terms FUNCTION "I" ;
MESSAGE "Enter vendor's terms of sale"
@ 18,16 GET m->discount PICTURE "99" ;
MESSAGE "Enter a discount rate (max. 99)"
SET COLOR TO &c_standard.
ON KEY LABEL F9 DO Findcode WITH m->city
RETURN
```

APENDICE B.

APENDICE B.

PROGRAMA AUXILIAR

PROGRAM: LIBRARY.PRG

PROCEDURE Add_new

```
@ 0,65 SAY SPACE(15) COLOR &c_yellow.
IF lookup_ok
  DO Sho_look WITH dbf
ENDIF
DO Init_fid
DO Get_data
READ

@ 0,0 SAY SPACE(51)

IF "" = TRIM(&key) .OR. READKEY() < 256
  RETURN
ELSE

  IF rec_is_dup

    rec_is_dup = .F.
    RETURN
  ELSE

    DO Sav_data
    GO record_num
  ENDIF
ENDIF
RETURN
```

PROCEDURE Bar_def

```
mesg = "Press first letter of Menu choice, or highlight and press <Enter>"
DEFINE POPUP main_mnu FROM 2,58 TO 22,78 MESSAGE mesg
DEFINE BAR 1 OF main_mnu PROMPT "== OPTION MENU ==" SKIP
DEFINE BAR 2 OF main_mnu PROMPT " Add record"
DEFINE BAR 3 OF main_mnu PROMPT " Edit record"
DEFINE BAR 4 OF main_mnu PROMPT " Delete record"
DEFINE BAR 5 OF main_mnu PROMPT "-----" SKIP
DEFINE BAR 6 OF main_mnu PROMPT " Next record"
DEFINE BAR 7 OF main_mnu PROMPT " Previous record"
DEFINE BAR 8 OF main_mnu PROMPT " Top record"
DEFINE BAR 9 OF main_mnu PROMPT " Bottom record"
DEFINE BAR 10 OF main_mnu PROMPT " Skip records"
DEFINE BAR 11 OF main_mnu PROMPT " Find record"
```

APENDICE B.

```
DEFINE BAR 12 OF main_mnu PROMPT "-----" SKIP
DEFINE BAR 13 OF main_mnu PROMPT " List records"
DEFINE BAR 14 OF main_mnu PROMPT " Output reports"
DEFINE BAR 15 OF main_mnu PROMPT " Group records" SKIP FOR dbf = "ACCT_REC"
DEFINE BAR 16 OF main_mnu PROMPT " Count records"
DEFINE BAR 17 OF main_mnu PROMPT " Index database"
DEFINE BAR 18 OF main_mnu PROMPT " Help"
DEFINE BAR 19 OF main_mnu PROMPT " Quit to MAIN MENU"
DEFINE POPUP dest_mnu FROM 13,10 TO 19,38 MESSAGE mesg
DEFINE BAR 1 OF dest_mnu PROMPT "===== DESTINATION =====" SKIP
DEFINE BAR 2 OF dest_mnu PROMPT " Printer"
DEFINE BAR 3 OF dest_mnu PROMPT " File"
DEFINE BAR 4 OF dest_mnu PROMPT " Screen"
DEFINE BAR 5 OF dest_mnu PROMPT " Exit to OPTION MENU"
DEFINE POPUP rpt_mnu FROM 11, 5 TO 17,38 MESSAGE mesg
DEFINE BAR 1 OF rpt_mnu PROMPT "===== REPORTS =====" SKIP
DEFINE BAR 2 OF rpt_mnu PROMPT " Database report: " + dbf
DEFINE BAR 3 OF rpt_mnu PROMPT " Mailing list: " + mlist ;
SKIP FOR mlist = "NOT AVAILABLE"
DEFINE BAR 4 OF rpt_mnu PROMPT " Custom programmed report: " + cust_rpt ;
SKIP FOR cust_rpt = "N/A"
DEFINE BAR 5 OF rpt_mnu PROMPT " Exit to OPTION MENU"
ON SELECTION POPUP main_mnu DO Barpop
ON SELECTION POPUP rpt_mnu DO Barpop_r
ON SELECTION POPUP dest_mnu DO Barpop_d
DEFINE WINDOW alert FROM 15, 3 TO 22,48 PANEL COLOR &c_alert.
DEFINE WINDOW duplicat FROM 15, 5 TO 21,70 PANEL COLOR &c_alert.
DEFINE WINDOW lister FROM 5, 5 TO 22,70 PANEL COLOR &c_list.
DEFINE WINDOW look FROM 6, 5 TO 16,65 PANEL COLOR &c_list.
DEFINE WINDOW memo_windo FROM 7, 4 TO 19,75 PANEL COLOR &c_list.
RETURN
```

PROCEDURE Barpop

```
DO CASE
* BAR() = 1 Is title of menu
CASE BAR() = 2
DO Add_new
CASE BAR() = 3
DO Edit
CASE BAR() = 4
DO Eraser
CASE BAR() = 6
DO Skip_rec WITH 1
CASE BAR() = 7
DO Skip_rec WITH -1
CASE BAR() = 8
GO TOP
CASE BAR() = 9
GO BOTTOM
```

APENDICE B.

```
CASE BAR() = 10
  DO Skip_rec WITH 0
CASE BAR() = 11
  DO Find_rec WITH key, key1, keyname1, key2, keyname2, key3, keyname3
CASE BAR() = 13
  DO List_rec
CASE BAR() = 14
  SAVE SCREEN TO Pre_rept
  ACTIVATE POPUP rpt_mnu
  RESTORE SCREEN FROM Pre_rept
  RELEASE SCREEN Pre_rept
CASE BAR() = 15
  DO Filter
CASE BAR() = 16
  *****
  IF NETWORK()
    * Turn off file lock to count
    SET LOCK off
    DO Kount
    SET LOCK on
    *****
  ELSE
    DO Kount
  ENDIF
CASE BAR() = 17
  *****
  IF NETWORK()
    old_tag = ORDER()
    USE (dbf) EXCLUSIVE
    IF net_choice <> 27
      DO Indexer
      SET EXCLUSIVE off
      USE (dbf) ORDER (old_tag)
    ENDIF
    *****
  ELSE
    DO Indexer
  ENDIF
CASE BAR() = 18      && Help
  SET COLOR TO &c_standard.
  DO Helper
CASE BAR() = 19
  DEACTIVATE POPUP
ENDCASE
```

APENDICE B.

```
DO Dstatus
DO Show_data
CLEAR GETS
SET COLOR TO &c_popup.
RETURN
```

PROCEDURE Barpop_d

```
SET COLOR TO &c_popup.
DO CASE
```

```
  CASE BAR() = 2
    DO Prt_menu
    SET PRINTER on
    SET CONSOLE off
    DO Printout
    SET PRINTER off
    SET CONSOLE on
  CASE BAR() = 3
    answer = SPACE(8)
    ACTIVATE WINDOW alert
    @ 0,0 SAY "----- SEND REPORT TO FILE -----"
    @ 2,1 SAY "Enter filename for report: " GET answer ;
    VALID "" <> TRIM(answer) ;
    MESSAGE "Enter a filename of up to eight characters"
    READ
    DEACTIVATE WINDOW alert
    SET ALTERNATE TO &answer.
    SET ALTERNATE on
    SET CONSOLE off
    GO TOP
    DO Printout
    SET ALTERNATE off
    SET CONSOLE on
  CASE BAR() = 4
    SET COLOR TO &c_standard.
    CLEAR
```

```
  plength = _plength
  rmargin = _rmargin
```

```
  _plength = 25
  _rmargin = 80
```

```
  DO Printout
  CLEAR
```

```
  * Reset page settings
  _plength = plength
  _rmargin = rmargin
  GO record_num
```

APENDICE B.

```
CASE BAR() = 5
  DEACTIVATE POPUP
ENDCASE
SET COLOR TO &c_standard.
DEACTIVATE POPUP
RETURN
```

PROCEDURE Barpop_r

```
SET COLOR TO &c_popup.
reportype = SPACE(6)
DO CASE
  CASE BAR() = 2
    reportype = "LISTING"
    ACTIVATE POPUP dest_mnu
  CASE BAR() = 3
    reportype = "LABELS"
    ACTIVATE POPUP dest_mnu
  CASE BAR() =
    reportype = "CUSTOM"
    ACTIVATE WINDOW alert
    ON READERROR
      SET BELL ON
      rpt_name = SPACE(8)
      @ 0,0 SAY "----- CUSTOM PROGRAMMED REPORT -----"
      @ 2,1 SAY "Enter report program name:" GET rpt_name ;
      VALID FILE(TRIM(rpt_name) + ".prg") ;
      MESSAGE "Enter a filename of up to eight " + ;
        "characters, e.g. Emp_rept " ;
      ERROR "invalid filename, please re-enter"
    READ
    ON READERROR
      SET BELL OFF
    DEACTIVATE WINDOW alert
    IF LASTKEY() <> 27
      SET COLOR TO &c_popup.
      ACTIVATE POPUP dest_mnu
    ENDIF
  ENDCASE
SET COLOR TO &c_popup.
DEACTIVATE POPUP
RETURN
```

APENDICE B.

```
PROCEDURE Sub_ret
IF erased

  IF NETWORK()
    USE (dbf) EXCLUSIVE
  ENDIF
  IF net_choice <> 27    && Skip if user pressed Esc

    ?? CHR(7)
    ACTIVATE WINDOW alert
    @ 0,0 SAY "----- PACKING DATABASE -----"
    @ 2,1 SAY "ERASING deleted records now....."
    @ 3,1 SAY "Please wait.....DO NOT TURN OFF"
    PACK
    DEACTIVATE WINDOW alert
  ENDIF
ENDIF

CLOSE DATABASES
CLEAR WINDOWS
RELEASE ALL
CLEAR
ON ERROR
ON KEY LABEL F9
ON KEY LABEL F10

DO Rest_env
CLEAR
RETURN TO MASTER

FUNCTION Duplicat
PARAMETERS key

  rec_is_dup = .F.
  IF RECCOUNT() = 0 .OR. "" = TRIM(key)
    * Do not check if database or key field(s) is empty
    RETURN rec_is_dup
  ENDIF
  record_num = RECNO()
  SEEK TRIM(key)

  DO CASE
    CASE PROMPT() = " Edit record"

      rec_is_dup = record_num <> RECNO() .AND. FOUND()
      CASE PROMPT() = " Add record"

        rec_is_dup = FOUND()
    ENDCASE
```

APENDICE B.

```
IF rec_is_dup
  ACTIVATE WINDOW duplicat
  CLEAR
  DO Wambell
  ? "----- DUPLICATE " + dbf + ;
  "RECORD -----"
  ? "      Duplicates not allowed"
  DO CASE
  CASE dbf = "CUST"
  ? "" + cust_id + "" + customer
  ? "This is the EXISTING record in the database; " + ;
  "re-enter Cust.ID."
  CASE dbf = "VENDORS"
  ? "" + vendor_id + "" + vendor
  ? "This is the EXISTING record in the database; " + ;
  "re-enter Vendor ID."
  CASE dbf = "GOODS"
  ? "" + part_id + "" + part_name
  ? "This is the EXISTING record in the database; " + ;
  "re-enter Part ID."
  CASE dbf = "ACCT_REC"
  ? "" + invoice_no + "" + cust_id + "" + DTOC(dat_of_bll)
  ? "This is the EXISTING record in the database; " + ;
  "re-enter Invoice ID."
  ENDCASE
  WAIT "  Press spacebar to continue..."
  DEACTIVATE WINDOW duplicat
  ENDIF
  GO record_num
  RETURN .NOT. rec_is_dup

PROCEDURE Dstatus

IF filters_on
  @ 0,51 SAY "Filter is ON" COLOR &c_blink.
ELSE
  SET COLOR TO &c_standard.

  @ 0,51
  ENDIF

  @ 0,66 SAY "Record #" + STR(RECNO(),5,0) COLOR &c_yellow.
  RETURN
```

APENDICE B.

PROCEDURE Edit

```
IF lookup_ok
  DO Sho_look WITH dbf
ENDIF
record_num = RECNO()

DO Load_fid
IF NETWORK()

  ready = .F.
  DO WHILE .NOT. ready
    IF CHANGE()

      DO Wambell
      GO RECNO()
      IF DELETED()
        DO Show_msg WITH "ALERT - Record has been deleted"
        SKIP
        DO Show_data
        RETURN
      ELSE
        DO Show_msg WITH ;
          "Data has been changed-screen shows revised data"
        DO Load_fid
      ENDIF
    ENDIF
  ENDIF
  DO Get_data
  READ
  ready = .NOT. CHANGE()  && DO loop will repeat if CHANGE() is .F.
ENDDO
ELSE
  DO Get_data
  READ
ENDIF
*****
@ 0,0 SAY SPACE(51)
IF "" = TRIM(&key.) .OR. READKEY() < 256

  RETURN
ELSE

  DO Sav_data
ENDIF
RETURN
```

APENDICE B.

PROCEDURE Eraser

```
answer = " "  
ACTIVATE WINDOW alert  
  @ 0,0 SAY "----- DELETE DATA RECORD -----"  
  @ 2,1 SAY "Erase this data record (Y=Yes,N=No)" GET answer PICTURE "Y"  
  READ  
DEACTIVATE WINDOW alert  
IF answer = "Y"  
  DELETE  
  
  SKIP  
  
DO CASE  
  CASE filters_on .AND. EOF()  
  
    SET FILTER TO  
    filters_on = .F.  
  
    GO TOP  
  CASE .NOT. filters_on .AND. EOF()  
  
    GO TOP  
  ENDCASE  
  
  erased = .T.  
ENDIF  
RETURN
```

PROCEDURE Filtr_ans

```
ACTIVATE WINDOW alert  
  @ 0,0 SAY "----- GROUP into SUBSET (Filter) -----"  
  IF filters_on  
    @ 2,0 SAY " Subset is currently selected."  
    @ 3,0 SAY " Turn Filter off ?"  
    @ 4,0 SAY " (T=Turn off, N=No) "GET choice PICTURE "I";  
    VALID choice $ "TN"  
  ELSE  
  
    @ 2,1 SAY "Select temporary subset of data "  
    @ 3,1 SAY "by entering filter condition(s) "  
    @ 4,1 SAY "  
    @ 5,1 SAY "Proceed ? (Y=Yes, N=No) "GET choice PICTURE "Y"  
  ENDIF  
  READ  
DEACTIVATE WINDOW alert
```

APENDICE B.

```
IF choice = "N"  
  RETURN TO Barpop  
ENDIF  
RETURN
```

```
PROCEDURE Findcode  
PARAMETERS acity
```

```
  I = INKEY()  
  acode = 0  
  ACTIVATE WINDOW alert  
  CLEAR  
  acode = LOOKUP(Codes->code, TRIM(acity), Codes->city)  
  ? "----- AREA CODE LOOKUP -----"  
  IF .NOT. FOUND("Codes") .OR. "" = TRIM(acity)  
    DO Wambell  
    ? "City: " + TRIM(acity) + " was" AT 2  
    ? "NOT FOUND in areacodes database." AT 2  
  ELSE  
    ?  
    ? "AREA CODE is: " + STR(acode,3) AT 2  
    ? "for " + TRIM(acity) AT 16  
  ENDIF  
  ?  
  I = INKEY(3)      && Display for 3 seconds  
  DEACTIVATE WINDOW alert  
RETURN
```

```
PROCEDURE Findcust  
PARAMETERS custid
```

```
  I = INKEY()  
  acust = ""  
  ACTIVATE WINDOW alert  
  CLEAR  
  acust = LOOKUP(Cust->customer, TRIM(custid), Cust->cust_id)  
  ? "----- CUSTOMER ID LOOKUP -----"  
  IF .NOT. FOUND("Cust") .OR. "" = TRIM(custid)  
    DO Wambell  
    ? "Customer ID: " + TRIM(custid) + " was" AT 2  
    ? "NOT FOUND in Cust database." AT 2  
  ELSE  
    ? "Customer: " + TRIM(acust) AT 2  
    ? "Phone: " + Cust->phone AT 2  
    ? "for ID: " + TRIM(custid) AT 12  
  ENDIF  
  WAIT " Press spacebar to continue..."  
  DEACTIVATE WINDOW alert  
RETURN
```

APENDICE B.

```
PROCEDURE Find_rec
PARAMETERS key, key1, keyname1, key2, keyname2, key3, keyname3

STORE "" TO target1, target2, target3
target1 = IIF(TYPE(key1) = "C", SPACE(LEN(&key1.)), { / / })

IF "NONE" <> key2
target2 = IIF(TYPE(key2) = "C", SPACE(LEN(&key2.)), { / / })

IF "NONE" <> key3
target3 = IIF(TYPE(key3) = "C", SPACE(LEN(&key3.)), { / / })
ENDIF
ENDIF
ACTIVATE WINDOW alert
@ 0,0 SAY "----- ENTER TARGET DATA TO FIND -----"
@ 2, 1 SAY keyname1
@ 2,15 GET target1 MESSAGE "Enter " + keyname1
IF "NONE" <> key2
@ 3, 1 SAY keyname2
@ 3,15 GET target2
IF "NONE" <> key3
@ 4, 1 SAY keyname3
@ 4,15 GET target3
ENDIF
ENDIF
@ 5,1 SAY "Enter partial or entire data"
READ
DEACTIVATE WINDOW alert
target = IIF(type(key1) = "C", target1, DTOC(target1))
IF "NONE" <> key2
target = target + IIF(type(key2) = "C", target2, DTOC(target2))
IF "NONE" <> key3
target = target + IIF(type(key3) = "C", target3, DTOC(target3))
ENDIF
ENDIF
target = TRIM(target)
IF RIGHT(target, 6) = " / "

target = LEFT(target, LEN(target) - 6)
ENDIF
IF "" = target
RETURN
ENDIF
record_num = RECNO()

IF .NOT. SEEK(target)

IF .NOT. SEEK(UPPER(target))
```

APENDICE B.

```
DO Wambell
DO Show_msg WITH "Record with target data was NOT found."
GO record_num
ENDIF
ENDIF
RETURN

PROCEDURE Findpart
PARAMETERS partid

I = INKEY()
p_name = SPACE(30)
ACTIVATE WINDOW alert
CLEAR
p_name = LOOKUP(Goods->part_name,TRIM(partid),Goods->part_id)
? "----- PART CODE LOOKUP -----"
IF .NOT. FOUND("Goods") .OR. "" = TRIM(partid)
DO Wambell
? "Part ID: " + TRIM(partid) AT 2
? "was NOT FOUND in Goods database." AT 2
ELSE
? "For ID: " + partid AT 2
? "Part name: " + TRIM(p_name) AT 2
? "Qty on hand: " + STR(Goods->qty_onhand,4) AT 2
? "Price: $ " AT 2, Goods->price PICTURE "99,999.99"
ENDIF
WAIT " .....Press spacebar to continue....."
DEACTIVATE WINDOW alert
RETURN

PROCEDURE Findvend
PARAMETERS vendr

I = INKEY()
v_name = SPACE(30)
ACTIVATE WINDOW alert
CLEAR
v_name = LOOKUP(Vendors->vendor,TRIM(vendr),Vendors->vendor_id)
? "----- VENDOR CODE LOOKUP -----"
IF .NOT. FOUND("Vendors")
DO Wambell
? "Vendor ID: " + TRIM(vendr) AT 2
? "was NOT FOUND in Vendors database." AT 2
ELSE
? "VENDOR is: " + TRIM(v_name) AT 2
? "Phone: " + Vendors->phone AT 2
? "for ID: " + vendr AT 16
ENDIF
WAIT " Press spacebar to continue..."
DEACTIVATE WINDOW alert
```

APENDICE B.

RETURN

PROCEDURE Kount

```
record_num = RECNO()
ACTIVATE WINDOW alert
@ 0,0 SAY "----- COUNT RECORDS -----"
@ 2,1 SAY "Counting, please wait..."
IF filters_on
    COUNT TO kount
ELSE
    kount = RECCOUNT()
ENDIF
@ 2,1 SAY "There are: " + STR(kount,6) + " records in "+ dbf
?
WAIT " Press any key to continue..."
DEACTIVATE WINDOW alert
GO record_num
RETURN
```

PROCEDURE List_rec

```
record_num = RECNO()
ACTIVATE WINDOW lster
answer = ""
CLEAR
@ 0,0 SAY "----- LIST RECORDS " + ;
"-----";
    COLOR &c_red.
SCAN WHILE .NOT. answer $ "r"
    LIST OFF NEXT 10 &list_flds.
    WAIT "Press spacebar to continue or R to return to " + ;
"OPTION MENU." TO answer
    CLEAR
ENDSCAN
DEACTIVATE WINDOW lster

GO record_num
RETURN
```

PROCEDURE Look_msg

```
DO CASE
CASE similar = .F.
    @ 1,1 SAY "Entered "+look_name+" ID does not exist in " + ;
    look_dbf+" database."
    ?
    WAIT "No " + look_name + " ID's are similar - " + ;
    "press R to return to screen." TO answer
CASE similar = .T. .AND. listcount > 0
    && Similar data found and listed
    WAIT "Press spacebar to continue list or " + ;
```

APENDICE B.

```
"R to return to screen." TO answer
CLEAR
ENDCASE
CLEAR
RETURN

FUNCTION Lookupid
PARAMETERS l_target, look_dbf, look_name, matchchars

IF .NOT. SEEK(l_target,(look_dbf))
ACTIVATE WINDOW look
DO Wambell
answer = ""
similar = .F.
SELECT (look_dbf)
GO TOP
DO WHILE .NOT. (EOF) .OR. answer $ "r"

    @ 0,0 SAY "----- DATA ENTRY ERROR: " + look_name + ;
        " ID WAS INVALID -----"
    @ 1,0 SAY "      This is a list of similar " + look_name + ;
        " ID's"
    ?
    listcount = 0
    DO CASE
    CASE dbf = "ORDERS"
    DO CASE
    CASE VARREAD() = "CUST_ID"
    SCAN FOR LIKE(SUBSTR(l_target,1,matchchars)+"*",cust_id) ;
    WHILE listcount <= 4
    ? cust_id, customer
    listcount = listcount + 1
    similar = .T.
    ENDSKAN
    CASE VARREAD() = "PART_ID"
    SCAN FOR LIKE(SUBSTR(l_target,1,matchchars)+"*",part_id) ;
    WHILE listcount <= 4
    ? part_id, SUBSTR(part_name,1,21) ;
    SUBSTR(descript,1,24)
    listcount = listcount + 1
    similar = .T.
    ENDSKAN
    CASE VARREAD() = "EMP_ID"
    SCAN FOR LIKE(SUBSTR(l_target,1,matchchars)+"*",emp_id) ;
    WHILE listcount <= 4
    ? emp_id, lastname, firstname
    listcount = listcount + 1
```

APENDICE B.

```
        similar = .T.
    ENDCASE
ENDSCAN
ENDCASE
CASE dbf = "GOODS"
    SCAN FOR LIKE(SUBSTR(l_target,1,matchchars)+"*",vendor_id) ;
        WHILE listcount <= 4
            ? vendor_id, vendor
            listcount = listcount + 1
            similar = .T.
        ENDCASE
    CASE dbf = "ACCT_REC"
        SCAN FOR LIKE(SUBSTR(l_target,1,matchchars)+"*",cust_id) ;
            WHILE listcount <= 4
                ? cust_id, customer
                listcount = listcount + 1
                similar = .T.
            ENDCASE
        ENDCASE
    DO Look_msg
ENDDO
DEACTIVATE WINDOW look
SELECT 1
ENDIF
RETURN not_valid = .NOT. FOUND((look_dbf))

PROCEDURE Net_err
PARAMETERS err_number

DO CASE
CASE err_number = 108
    IF "" <> TRIM(LKSYS(2))
        message = "" + dbf + " is in use by: " + LKSYS(2)
    ELSE
        message = "" + dbf + " is in use by someone"
    ENDIF
CASE err_number = 109
    message = " Record is locked by: " + LKSYS(2)
CASE err_number = 110
    message = "File should be USED EXCLUSIVE"
CASE err_number = 372 .OR. err_number = 373
    message = MESSAGE()
OTHERWISE
    message = " Unknown error: " + MESSAGE()
ENDCASE
DO Warnbell
ACTIVATE WINDOW alert
```

APENDICE B.

```
CLEAR
? "----- NETWORK ERROR -----"
?
? message AT 1
? "Press spacebar to try again" AT 1
? " - or press Esc to Quit" AT 1
net_choice = INKEY(0)
DEACTIVATE WINDOW alert
IF net_choice <> 27

RETRY
ENDIF
RETURN

PROCEDURE Printout

DO CASE
CASE reportype = "LISTING"
REPORT FORM &dbf.
CASE reportype = "LABELS"
LABEL FORM &dbf.
CASE reportype = "CUSTOM"
DO &rpt_name.
ENDCASE
GO record_num
RETURN

PROCEDURE Prt_menu

msg_num = "Enter a number"
msg_logic = "Enter a Y or N"
msg_enum = "Press spacebar for other options"

loffset = 0
lmargin = 0
rmargin = 80
lindent = 4
plength = 66
STORE 1 TO pspacing, pbpage, pcopies
pepage = 9999
peject = "NONE "
STORE .F. TO pwait, pquality
ppitch = "PICA "
ACTIVATE WINDOW lister
CLEAR
@ 0, 0 SAY "----- PRINT MENU " + ;
"-----" COLOR &c_red.

@ 2, 1 SAY "Page settings:"
@ 3, 1 SAY "===== "
@ 4, 1 SAY "Offset from left " GET loffset ;
```

APENDICE B.

```

    PICTURE "99" MESSAGE msg_num
@ 5, 1 SAY "Left margin " GET lmargin ;
    PICTURE "99" MESSAGE msg_num
@ 6, 1 SAY "Right margin " GET rmargin ;
    PICTURE "99" MESSAGE msg_num
@ 7, 1 SAY "indentation " GET indent ;
    PICTURE "99" MESSAGE msg_num
@ 8, 1 SAY "Page length " GET plength ;
    PICTURE "99" MESSAGE msg_num
@ 9, 1 SAY "Spacing " GET pspacing ;
    PICTURE "9" RANGE 1,3 MESSAGE msg_num
@ 2,26 SAY "Print settings:"
@ 3,26 SAY "======"
@ 4,26 SAY "Begin printing on page " GET pbpage ;
    PICTURE "999" MESSAGE msg_num
@ 5,26 SAY "End printing on page " GET pepage ;
    PICTURE "9999" MESSAGE msg_num
@ 6,26 SAY "Number of copies " ;
    GET pcopies PICTURE "999" MESSAGE msg_num
@ 7,26 SAY "Eject paper " GET peject ;
    PICTURE "@M BEFORE,AFTER,BOTH,NONE" MESSAGE msg_enum
@ 8,26 SAY "Wait between pages " GET pwait ;
    PICTURE "Y" MESSAGE msg_logic
@ 9,26 SAY "Pitch " GET ppitch ;
    PICTURE "@M DEFAULT,PICA,ELITE,CONDENSED" MESSAGE msg_enum
@ 10,26 SAY "Quality print " GET pquality ;
    PICTURE "Y" MESSAGE msg_logic
@ 12, 1 SAY "Please enter desired settings; press PgDn to save and exit"
READ
DEACTIVATE WINDOW lster

_ploffset = loffset
_lmargin = lmargin
_rmargin = rmargin
_indent = indent
_plength = plength
_pspacing = pspacing
_pbpage = pbpage
_pepage = pepage
_pcopies = pcopies
_peject = peject
_pwait = pwait
_ppitch = ppitch
_pquality = pquality
SET COLOR TO &c_standard.
RETURN

```

APENDICE B.

PROCEDURE Rest_env

```
SET COLOR TO &c_standard.  
SET SCOREBOARD &scor.  
SET DELIMITERS &deli.  
SET HELP &help.  
SET CLOCK &clock.  
SET ESCAPE &esca.  
SET DELETED &delee.  
SET HEADING &head.  
SET STATUS &stat.  
SET SAFETY &safe.  
SET EXACT &exac.  
SET BELL &bell.  
SET NEAR &near.
```

```
DO Colo_rese  
SET TALK &talk.  
RETURN
```

PROCEDURE Sav_data

```
choice = "Y"  
ACTIVATE WINDOW alert  
@ 0,1 SAY "----- SAVE DATA -----"  
@ 2,1 SAY "Save this data to disk ? (Y/N) " GET choice PICTURE "Y"  
READ  
DEACTIVATE WINDOW alert  
IF choice = "Y"  
    IF PROMPT() = " Add record"  
        APPEND BLANK  
        record_num = RECNO()  
    ENDIF  
  
    DO Repl_fid  
ELSE  
  
    GO record_num  
ENDIF  
RETURN
```

PROCEDURE Set_env

```
PUBLIC talk      && First set TALK OFF  
talk           = SET("TALK")  
SET TALK off  
  
PUBLIC c_standard, c_data, c_fields, c_popup, c_alert, c_list  
PUBLIC c_red, c_blue, c_yellow, c_yelowhit, c_green, c_blink
```

APENDICE B.

IF ISCOLOR()

c_standard = "WB,BG+/R,B"
c_data = "B/W,R/BG,B"
c_fields = "B/BG"
c_popup = "B/W,GR+/R"
c_alert = "GR+/R,B/W,R/G"
c_list = "W+/G,GR+/B,GR+/GR"
c_red = "R/W"
c_blue = "B/W"
c_yellow = "GR+/B"
c_yelowhit = "GR+/W"
c_green = "G/W"
c_blink = "GR+*/B"

ELSE

STORE "W+/N,N/W" TO c_standard, c_data, c_popup, c_alert, c_list
STORE "W" TO c_red, c_blue, c_yellow, c_yelowhit, c_green, c_fields
c_blink = "W+*/N,N/W"

ENDIF

SET COLOR OF MESSAGES TO &c_blue.

SET COLOR TO &c_standard.

PUBLIC scor, delli, help, clock, esca, delee, head, stat, safe

PUBLIC exac, bell, near

scor = SET("SCOREBOARD")

delli = SET("DELIMITERS")

help = SET("HELP")

clock = SET("CLOCK")

esca = SET("ESCAPE")

delee = SET("DELETED")

head = SET("HEADING")

stat = SET("STATUS")

safe = SET("SAFETY")

exac = SET("EXACT")

bell = SET("BELL")

near = SET("NEAR")

SET SCOREBOARD off

SET DELIMITERS off

SET HELP off

SET CLOCK off

SET ESCAPE off

SET DELETED on

SET HEADING on

SET STATUS off

SET SAFETY off

SET TALK off

SET EXACT off

xxxx

APENDICE B.

```
SET BELL off
SET NEAR off
PUBLIC erased, not_valid, rec_js_dup, filters_on, lookup_ok, choice
PUBLIC record_num, net_choice
PUBLIC target, look_dbf, matchchar, scanfield
```

```
STORE .F. TO erased, not_valid, rec_js_dup, filters_on
lookup_ok = .T.
* Other variables
STORE "" TO choice, subset
STORE 0 TO record_num, net_choice
```

```
IF NETWORK()
  * Network programming assumes databases have been CONVERTed
  SET EXCLUSIVE off
  ON ERROR DO Net_err WITH ERROR()
  * Retry a reasonable amount of time (depends on computer)
  SET REPROCESS TO 3
ENDIF
```

```
ON READERROR ??
RETURN
```

```
PROCEDURE Sho_look
PARAMETERS db
DO CASE
  CASE db = "EMPLOYEE" .OR. db = "CUST" .OR. db = "VENDORS"
    look_txt = "Press F9 to look up Area code"
  CASE db = "GOODS"
    look_txt = "Press F9 to look up Vendor name and phone"
  CASE db = "ORDERS"
    look_txt = "Press F9 to look up Cust data; F10 for Part ID data"
  CASE db = "ACCT_REC"
    look_txt = "Press F9 to look up Customer name and phone"
ENDCASE
@ 0,0 SAY look_txt COLOR &c_blink.
I = INKEY(1)
@ 0,0 SAY look_txt COLOR &c_yellow.
RETURN
```

APENDICE B.

```
PROCEDURE Show_msg
PARAMETERS u_message
_wrap = .T.
ACTIVATE WINDOW alert
@ 1,0 SAY u_message
?
WAIT " Press spacebar to continue..."
DEACTIVATE WINDOW alert
RETURN
```

```
PROCEDURE skip_rec
PARAMETERS skipno
```

```
DO CASE
CASE skipno = 1
```

```
IF .NOT. EOF()
SKIP
ENDIF
```

```
CASE skipno = -1
IF .NOT. BOF()
SKIP -1
ENDIF
```

```
CASE skipno = 0
```

```
numb_2skip = 0
```

```
ACTIVATE WINDOW alert
```

```
@ 0,0 SAY "----- SKIP NUMBER OF RECORDS -----"
```

```
@ 2,1 SAY "How many records do you want to skip?"
```

```
@ 3,0 SAY " (Example: 15 or -5) ? " ;
```

```
GET numb_2skip PICTURE "9999" ;
```

```
MESSAGE "Enter positive no. to go forward " + ;  
"or negative no. to go backward"
```

```
READ
```

```
DEACTIVATE WINDOW alert
```

```
IF .NOT. (BOF() .AND. numb_2skip < 0) .OR. (EOF() .AND. numb_2skip > 0)  
SKIP numb_2skip
```

```
ENDIF
```

```
ENDCASE
```

APENDICE B.

```
DO CASE
  CASE EOF()
    GO BOTTOM
    DO Show_msg WITH " Bottom record in " + dbf + " database"
  CASE BOF()
    DO Show_msg WITH " Top record in " + dbf + " database"
  ENDCASE
RETURN
```

```
PROCEDURE Wambell
PRIVATE mwrap
mwrap = _wrap      && Save _wrap value
_wrap = .F.

SET BELL TO 880,4
?? CHR(7)
SET BELL TO 1400,4
?? CHR(7)
SET BELL TO 880,4
?? CHR(7)
SET BELL TO
_wrap = mwrap
RETURN
```

BIBIOGRAFIA

BIBLIOGRAFIA

- Brabb, George J.** **Computadoras y Sistemas de Información en los negocios**
1a. Edición. Edit. Latinoamericana
México, 1983. Cap. 5 pp 59-97
- Kendall & Kendall** **Análisis y Diseño de Sistemas**
Editorial Prentice Hall
- Wiederhold, Gio** **Diseño de bases de datos**
Editorial Mc Graw Hill
- Burch G. John** **Diseño de Sistemas de Información**
Editorial Megabyte
- Murdick, G. Robert** **Sistemas de Información Administrativa**
Editorial Prentice Hall
- Leblond, T. Geoffrey** **Dbase IV Manual de Referencia**
Serie Osborne Mc Graw Hill
- Senn, James A.** **Análisis y Diseño de Sistemas de Información**
Editorial MC Graw Hill