

7
2ej



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

APLICACION DE LA INGENIERIA DE
PROGRAMACION A UN SISTEMA
DE CUENTAS POR COBRAR

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
C E L S O B R I T O A L D A Y

DIRECTOR DE TESIS:
ING. LUIS GONZAGA CORDERO



MEXICO, D. F.

1960

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 está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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

I N D I C E

	PAGINA
INTRODUCCION	1
CAPITULO I	
CONCEPTOS DE LA INGENIERIA DE PROGRAMACION	3
CAPITULO II	
PLANTEAMIENTO DEL SISTEMA	5
II.1 - Definición de necesidades	
II.2 - Estudio de factibilidad	
II.2.1 - Alternativas de solución	
II.2.2 - Análisis de beneficios	
II.2.3 - Estimación de costos	
CAPITULO III	
ANÁLISIS DEL SISTEMA	17
III.1 - Diagrama de flujo de datos	
III.2 - Diccionario de datos	
III.3 - Miniespecificaciones	
CAPITULO IV	
DISEÑO DEL SISTEMA	133
IV.1 - Diagrama de estructura	
IV.2 - Aspectos de los lenguajes de programación	
IV.3 - Diseño de la Base de Datos	
CAPITULO V	
DESARROLLO DEL SISTEMA	167
V.1 - Instrumentación	
V.2 - Documentación	
CAPITULO VI	
IMPLANTACION DEL SISTEMA	175
VI.1 - Pruebas de Integración	
VI.2 - Pruebas de aceptación	

CAPITULO VII	
MANTENIMIENTO	182
CAPITULO VIII	
DEMOSTRACION DE RESULTADOS	185
VIII.1 - Características del Sistema	
VIII.2 - Operación del Sistema (manual de operación)	
CONCLUSIONES	210
APENDICE A	
FORMATOS DE REPORTES	
BIBLIOGRAFIA	

Aplicación de la Ingeniería de programación
a un Sistema de Cuentas por Cobrar.

INTRODUCCION

Cuando en 1959 se comenzó la construcción del puente Verrazano Narrows en Nueva York, los oficiales estimaron que el costo de dicha obra sería de 335 millones de dólares y que quedaría concluida en el año de 1965. Este era el puente suspendido más largo que a la fecha se hubiera construido, sin embargo, fué terminado en 1964 y dentro del gasto presupuestado.

Los grandes proyectos de programación no observan mucha similitud con el caso anterior, ya que por lo general son entregados tarde y son tan extensos que no es posible darles un buen mantenimiento a un costo razonable.

Porqué es la ingeniería de puentes tan exacta, mientras que la ingeniería de programación parece tropezar tanto?

Precisamente, uno de los mayores problemas de la era de la computación ha sido la adolescencia de técnicas formales de desarrollo de sistemas de cómputo. Técnicas cuyo objetivo debería ser, brindar una respuesta adecuada a las necesidades de las grandes corporaciones, ya que estas requieren operar con volúmenes de datos excesivamente grandes.

En respuesta a esta problemática se han desarrollado procedimientos como el que en esta tesis se trata, orientados al manejo de altos volúmenes de información, pero con la tendencia de reducir el elevado costo que implica el desarrollar y mantener eficientemente un sistema.

Fué en la década de 1960 cuando se hizo patente la necesidad de implementar enfoques sistemáticos para el desarrollo y mantenimiento de sistemas de información, ya que durante la misma aparecieron las computadoras de la tercera generación y se desarrollaron técnicas de programación como la multiprogramación y el tiempo compartido. Estos nuevos desarrollos aportaron la tecnología necesaria para el establecimiento de sistemas computacionales interactivos, multiusuario, en línea y en tiempo real. De esta forma surgieron nuevas aplicaciones para la computación, como las reservaciones aéreas, bancos de información, tiempo compartido para diversas aplicaciones, control de procesos industriales, etc.

Aunque muchos de estos sistemas fueron desarrollados y producidos adecuadamente, gran parte de los sistemas intentados nunca se concluyeron, de los realizados muchos quedaron sujetos a sobrecosto, otros se entregaron tarde, otros más fueron poco confiables e ineficientes y muchos tuvieron poca aceptación por parte de los usuarios.

Por otra parte, en la mayoría de los centros de trabajo, la cantidad de empleados, los gastos, los débitos, y las actividades administrativas en general son tan extensas que no es posible

controladas en forma manual. Esto ha ocasionado que las aplicaciones administrativas para el manejo de información se conviertan en un renglón muy importante dentro del extenso ámbito de las aplicaciones computacionales.

Tomando en cuenta esta problemática casi generalizada en la pequeña y mediana industria es que se derivó el presente trabajo, mismo que trata de aliviar en cierta medida la situación antes descrita ilustrando las técnicas de la ingeniería de programación aplicándolas al diseño e implantación de un sistema administrativo de CUENTAS POR COBRAR.

Este sistema nació de la necesidad que tienen las empresas de automatizar el control de su cobranza. Con un sistema de esta naturaleza es posible llevar un control estricto sobre los saldos de los clientes, agilizar el cobro a los clientes morosos y tener a la mano un estado de cuenta pormenorizado de los movimientos efectuados por cada cliente, logrando con ello una mejor toma de decisiones en el momento más necesario.

Esta es quizá la función más importante del sistema. Sin embargo, existen otras funciones de no menor importancia que también realiza el sistema en cuestión. Una de ellas es la facturación, aspecto que representa las siguientes ventajas: se crea la factura haciendo el respectivo cargo al saldo del cliente; se acumula la venta en los archivos estadísticos; es posible hacer la descarga automática de los archivos de existencias; al facturar se tiene disponible el precio actualizado de los productos que se van a vender, evitando así posibles errores por manejo de listas de precios obsoletas, etc.

Por último, una función adicional del sistema es el manejo estadístico de las ventas realizadas, coadyuvando de esta manera a una planeación estratégica de las futuras actividades de la empresa.

CAPITULO I

CONCEPTOS DE LA INGENIERIA DE PROGRAMACION

Las actividades que se desarrollan para el proceso de datos dan como resultado información. La información es algo que requerimos las personas, los animales, las organizaciones, etc. y que sirve para guiar el comportamiento. Así, una organización procesa datos para obtener información que guie sus actividades hacia el logro de sus objetivos.

En las actividades del proceso de datos la participación de las personas es cada día mayor; en los EU se estima que cerca del 60% de la fuerza de trabajo está orientada hacia estas actividades. Para tener una idea de lo que este porcentaje significa, en los EU solo el 2% de la fuerza de trabajo está dedicada a la agricultura. Esto refleja la importancia del procesamiento de datos como una actividad fundamental en las organizaciones actuales y se debe fundamentalmente a:

- lo complejo de las relaciones actuales
- las nuevas técnicas de administración
- la tecnología de las computadoras

En la figura I.1 puede verse la cantidad de entidades con las que una organización tiene que relacionarse y por consiguiente demanda de información.

Frederick Taylor, Franck y Lilian Gilbreth desarrollaron técnicas que permitieran operar en forma eficiente a las organizaciones, estas constituyen la base de lo que se conoce como técnicas de administración científica.

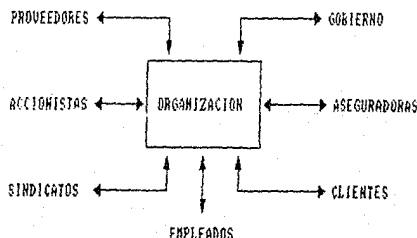


Figura I.1.- Interacción entre la organización y el medio ambiente

La disponibilidad de la tecnología de las computadoras se demanda a diario para su uso. Información que antes era demasiado costosa o prácticamente imposible de obtener ahora es posible a un costo razonable. Con esto ha crecido la demanda de este tipo de información.

Conforme las computadoras crecieron haciéndose más complejas, resultó obvio que la demanda por los sistemas de información superó la capacidad de producir y mantener dichos sistemas.

De esta forma, la utilización de las computadoras se ha diversificado y se ha hecho más compleja y crítica para la sociedad moderna; como resultado de esto, el campo de la ingeniería de programación ha evolucionado para convertirse en una disciplina tecnológica de considerable importancia.

Una de las mejores definiciones que se han dado de la ingeniería de programación, es la de Richard Fairley, la cual dice: 'La ingeniería de programación es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido'.

Así, las metas primordiales de esta nueva disciplina tecnológica son mejorar la calidad de los sistemas de información y aumentar la productividad de los mismos mediante la aplicación de técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los sistemas de información dentro del tiempo y el presupuesto establecidos para el proyecto.

Dado que el objetivo del presente trabajo gira en torno a un sistema de información, se van a definir algunos aspectos importantes sobre lo que son los sistemas y la técnica del enfoque de sistemas, misma que dará la pauta para el posterior desarrollo del proyecto.

Se ha definido el término sistema, como un "grupo organizado de componentes relacionados funcionalmente entre sí". Un sistema existe debido a que es diseñado para lograr un objetivo. Podemos hablar de un sistema productivo y de su organización como un sistema compuesto de departamentos interrelacionados llamados subsistemas. Ninguno de los subsistemas es independiente de los demás y cuando están propiamente coordinados podemos decir que el sistema funciona exitosamente y cumple con su cometido.

Se ha observado que un sistema posee las siguientes características primarias: es abierto (o cerrado) cuando interactúa con el medio ambiente, tiene dos ó más subsistemas, hay interdependencia entre los subsistemas, es autoajustable, es autorregulable, tiene un propósito, tiene estructura, tiene un comportamiento, tiene un ciclo de vida.

Los sistemas en general se clasifican por el tipo y grado de complejidad. Un sistema puede ser determinístico o probabilístico y en cada caso simple, complejo o excesivamente complejo.

Los sistemas se deben representar en base a un modelo. El objetivo del uso del modelo es señalar los elementos significantes y las interrelaciones de un sistema.

Un sistema de información es un componente de una organización. Su propósito es obtener información dentro y fuera de la organización y hacerla disponible a todos los otros componentes en la forma como estos la demanden y también presentar información a los que se encuentran fuera como la necesitan. La información para que sea útil debe presentarse en el tiempo adecuado y estar disponible a un costo razonable.

Posiblemente la técnica más utilizada para el estudio de los sistemas es el Enfoque de Sistemas. Dicho enfoque es un proceso de desarrollo ordenado y analítico que se puede utilizar continuamente para analizar, evaluar y diagnosticar la naturaleza de un sistema, así como los resultados de su desempeño para captar todo lo necesario a esos fines y proveer la continua autocorrección del funcionamiento del sistema con el propósito de alcanzar los objetivos propuestos.

El enfoque de sistemas se basa en los 5 pasos siguientes: análisis, diseño, desarrollo, instrumentación y evaluación. Las actividades que se desarrollan en cada paso se pueden resumir de la siguiente manera:

a) Análisis

Representación o caracterización del sistema.

- se describe toda la información acerca del sistema dando detalles concretos y útiles
- se definen y analizan las entradas
- se describen los procesos y sus características.
- las salidas se definen, identifican y cuantifican.
- se describe la estructura del sistema para aclarar las relaciones entre los componentes.
- se define el ambiente para aclarar su interacción con el sistema.

- se define el flujo de información y en especial los mecanismos de realimentación.
- finalmente se estudia la relación entre las salidas y los objetivos y metas.

b) Diseño

Con el análisis se han descubierto fallas en el sistema y el paso siguiente es diseñar nuevos métodos para el sistema. Según sea la naturaleza del problema el diseño puede comprender:

- un nuevo sistema
- cambio de componentes
- cambio en las entradas y/o salidas
- cambios en los procesos

Para llevar a cabo el diseño, se prepara un plan detallado con todas las especificaciones requeridas para modificar al sistema.

c) Desarrollo

Sobre la base de los diseños se procede a la construcción, edificación, explicación o lo que sea necesario para el desarrollo del sistema.

d) Instrumentación

Una vez que se ha desarrollado el nuevo componente, elemento, proceso o sistema, debe incorporarse a lo ya existente.

La instrumentación requiere del establecimiento de prioridades y secuencia de pasos para incorporar el nuevo elemento, ya que si se fracasa en la instrumentación los demás elementos pueden trastornar al sistema temporalmente al tratar de ajustarse a los requerimientos del nuevo. Otra falla podría ser que el nuevo componente no diera los resultados esperados.

e) Evaluación

Después de haber instrumentado el nuevo componente, este debe evaluarse, ya que no se sabe si realmente resolvió los problemas planteados o provocó nuevos problemas. La evaluación debe hacerse en términos de los objetivos que se establecieron en las etapas de análisis y diseño.

En cualquier plan o método para el desarrollo de sistemas de información encontramos que los pasos del Enfoque de sistemas gobiernan de forma general el proceso. Este plan o método también se conoce como 'Ciclo de vida del desarrollo de un sistema'.

Los estados del "Ciclo de vida de un sistema" se esquematizan en la figura 1.2

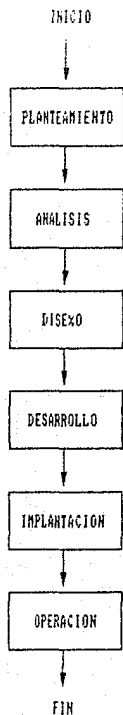


Figura 1.2 .- Ciclo de vida de un sistema.

Se pueden describir cada una de las etapas de la siguiente manera:

a) Planteamiento del sistema

Es en esta etapa donde nace el sistema, y es aquí mismo donde se establece claramente cual es el problema a resolver.

b) Análisis

Esta etapa tiene como objetivo definir clara y concisamente lo que se tiene que realizar para resolver el problema presentado.

c) Diseño del sistema

Aplicando técnicas de Diseño Estructurado en esta fase se debe definir 'COMO' es que se realizará la solución del problema.

d) Desarrollo del sistema

La etapa de desarrollo comprende la codificación y documentación del sistema.

e) Implantación o instrumentación

En esta fase se tratan de unificar los módulos implementados para que queden integrados en un solo sistema. Además, en esta etapa se deben preparar y realizar las pruebas necesarias para la evaluación y posterior liberación del sistema.

f) Operación

Esta etapa está orientada hacia el usuario final. Básicamente, lo que comprende esta etapa es la documentación necesaria para la operación del sistema.

De esta manera se ha pretendido dar una semblanza de lo que es la Ingeniería de programación y lo que esta ya tan importante rama de la ingeniería pretende. También se ha procurado exponer de forma breve la metodología que se utilizará para el desarrollo del proyecto así como los estadios o etapas que lo constituyen.

CAPÍTULO II

PLANTEAMIENTO DEL SISTEMA

Todo elemento desarrollado por el hombre primero es una idea en su mente. Los sistemas computacionales, como otros productos de la tecnología, se desarrollan en respuesta a requerimientos específicos. Las fuentes que producen las ideas de productos de programación incluyen las necesidades del cliente generadas externamente, las necesidades internas de la organización, planes de mercadotecnia, y los planes organizacionales. La mayor parte de las organizaciones que desarrollan productos de programación son muy selectivas al decidir qué productos desarrollarán; no explotan todas las oportunidades. La decisión de llevar a cabo el proyecto se basa, generalmente, en el resultado de un estudio de factibilidad. Es por esto que en esta segunda etapa se realizarán las siguientes actividades:

II.1) Definición de necesidades

II.2) Estudio de factibilidad

II.1) Definición de necesidades

El objetivo de esta subetapa es proporcionar, en la terminología del cliente, un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. Es decir, se definirá el problema y se establecerá el medio ambiente en el que se da.

En base a pláticas sostenidas con los usuarios, se logró determinar la naturaleza del problema y cuales son las necesidades involucradas en el mismo, necesidades que en forma genérica se dan a conocer a continuación.

El problema que aquí se examina, se origina básicamente por la interacción que existe entre los clientes de la compañía y el departamento de administración y cobranza de la misma. Las relaciones que existen entre la compañía y sus clientes se ilustran en la figura II.1.1

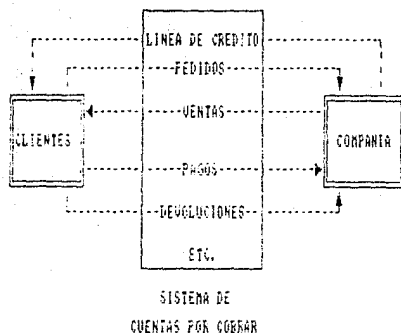


Figura II.1.1 Relaciones existentes entre la compañía y sus clientes.

A continuación se definen las necesidades que se consideró que debe satisfacer el sistema de cuentas por cobrar:

a) Crear y dar mantenimiento a diferentes catálogos, como son, el catálogos de clientes y de artículos.

b) Se deben facturar todas las compras hechas por los clientes, haciendo el respectivo cargo a su cuenta y acumulando las cantidades e importes de los artículos comprados en los archivos de naturaleza estadística.

c) Los archivos estadísticos deberán contemplar la necesidad de información a nivel cliente y a nivel artículo.

d) Los clientes pueden hacer sus compras por crédito concedido por la empresa. Por lo tanto, es necesario considerar el manejo de las diferentes condiciones de venta que la empresa quiera conceder a sus cliente según referencias crediticias de los mismos; esto se hace con la finalidad de poder obtener la información de la cobranza, es decir, los saldos vencidos o por vencer de los clientes.

e) Existen diversos conceptos por los cuales se le pueden hacer cargos a la cuenta del cliente, por ejemplo: la facturación, el cobro de fletes, cargos por cheques que no se pudieran cobrar, cargos por cobranza, cargos diversos, etc.

f) Existen algunos conceptos por los cuales se le pueden hacer abonos a los clientes, ejemplo de ello pueden ser los pagos, las notas de crédito, anticipos, abonos diversos, etc.

g) Es necesario poder hacer consultas en pantalla de los estados de cuenta de los clientes, así como de sus saldos vencidos.

h) Se requiere que el sistema proporcione diversos reportes que ayuden a conservar una memoria escrita de la situación de los clientes al momento de ser obtenidos.

Dado que la cartera de clientes puede constar desde unos cuantos, hasta varios miles de ellos, es deseable que el número máximo de clientes y de movimientos asociados a los mismos esté limitado únicamente por la capacidad que se tenga en el disco duro de la computadora en que se opere el sistema.

El presente problema se desenvuelve en un medio ambiente totalmente administrativo, donde las personas encargadas del proceso de la cobranza normalmente no están familiarizadas con el uso de la computadora, por lo que es necesario considerar una capacitación previa y darle mucha importancia a este aspecto en el manual de operación del sistema.

II.2) Estudio de factibilidad

El segundo paso en la planeación del sistema es determinar lo apropiado de una solución computacional. Además de ser eficaz en términos de costo, un sistema debe aceptarse social y políticamente. Para ser eficiente en costo, un nuevo producto de programación debe proporcionar los mismos servicios e información que el sistema antiguo, usando menos tiempo y personal, o proporcionar servicios e información que antes no se tenían.

Ya que se ha definido el problema en la subetapa anterior, se procede a dar alternativas de solución para contar con una base que permita realizar el análisis de beneficios, mediante los cuales sea posible comparar las alternativas propuestas y escoger la mejor de todas. Una vez elegida la mejor opción, se procederá a hacer la estimación de costos de la misma.

II.2.1) Alternativas de Solución

Las alternativas encontradas como solución al presente problema son:

a) Comprar un paquete comercial que se adapte lo mejor posible a las necesidades antes descritas de la empresa. b) Otra alternativa es implementar el sistema aplicando la metodología ya mencionada, haciendo el desarrollo en algún lenguaje de programación de alto nivel como Pascal, Obase III+, etc.

c) Una última alternativa, podría ser, optimizar los procedimientos administrativos actuales y hacer que el sistema funcione en forma semi-manual, es decir, programando únicamente aquellos procesos que sean más laboriosos y difíciles de operar manualmente.

II.2.2) Análisis de beneficios

Podemos distinguir dos tipos de beneficios, los que son tangibles como son el ahorro económico y aumento en la productividad, y los beneficios intangibles como pueden ser la imagen, la eficacia, la confiabilidad y la rapidez.

La solución que da la primer alternativa quizás sugiera los mayores beneficios tangibles, ya que es la que ofrece por una parte la obtención de resultados en un tiempo muy corto y por la otra, evita el tener que contratar a personal de programación con sus consiguientes inconveniencias. Sin embargo esta opción también carece de beneficios intangibles ya que debemos considerar que todo paquete comercial está diseñado en base a requerimientos muy generales y que esto puede ser una limitante muy fuerte en cuanto a versatilidad de proceso. Adicionalmente debemos tomar en cuenta otro aspecto muy importante que se persigue en el sistema, que es el de la confiabilidad en la información., aspecto que en ocasiones no es totalmente tomado en cuenta en el diseño de este tipo de paquetes.

La segunda opción es la que ofrece los mayores beneficios intangibles, ya que con ella se consigue la versatilidad de proceso que se desea, así como también, un alto grado de confiabilidad, eficacia y productividad que es precisamente lo que se persigue en el desarrollo del presente sistema. Además esta es la única alternativa que va a permitir dar al sistema un mantenimiento fácil y oportuno. La única desventaja que puede tener esta opción es que el tiempo empleado en el desarrollo puede ser muy grande, sin embargo esto no es una razón de peso como para desechar esta opción.

La última opción quizás sea la menos indicada pues es la que menos beneficios tangibles e intangibles ofrece ya que el único beneficio posible es el ahorro económico, contraponiéndose a la idea inicial de obtener una mayor productividad, eficiencia y confiabilidad en el sistema.

Del anterior análisis podemos concluir que lo más apropiado para la solución del problema planteado es desarrollar el sistema de cuentas por cobrar en un lenguaje de alto nivel, ya que esta opción ofrece el mayor número de beneficios.

II.2.3) Estimación de costos

La estimación de costos de un proyecto de programación es una de las más difíciles y erráticas tareas de la ingeniería de programación; es difícil hacer estimaciones exactas durante la fase de planeación de un desarrollo debido a la gran cantidad de factores desconocidos en ese momento. Sin embargo, la práctica normal en los contratos implica un firme compromiso monetario como parte del estudio de factibilidad. Lo anterior aunado a la naturaleza competitiva de este negocio, es un factor que contribuye a los retrasos de entrega y sobregiro en presupuesto tan comunes en proyectos de programación.

Existen muchos factores que influyen en el costo de un proyecto de programación. El efecto de estos factores es difícil de estimar y por ende también lo es el costo del esfuerzo en el desarrollo o en el mantenimiento. Entre los factores que afectan el costo, se observan en forma general, la capacidad individual del personal asignado al proyecto y su familiarización con el área de aplicación, la complejidad del producto, el tamaño de éste, el tiempo asignado, el nivel de confiabilidad y el nivel tecnológico utilizado.

Existen tres categorías para los proyectos de programación: Programas de aplicación, en los que se incluyen procesamiento de datos y programas científicos; Programas de apoyo, como compiladores y ligadores, y programas de sistemas, como sistemas de bases de datos, sistemas operativos y sistemas de tiempo real.

Boehm (BOE81) utiliza tres niveles de complejidad de un producto y proporciona ecuaciones para predecir el esfuerzo total en meses de programador (MP), considerando como variable independiente al número de millares de instrucciones de código fuente que se van a escribir (MICF). De esta forma los costos de programación de un proyecto pueden obtenerse con la multiplicación del esfuerzo requerido en términos de meses de programador por el costo unitario del mes de programador. Estas ecuaciones fueron obtenidas a través del análisis de datos históricos de un gran número de proyectos reales.

Programas de aplicación :	MP = 2.4 * (MICF ^ 1.05)
Programas de apoyo :	MP = 3.0 * (MICF ^ 1.12)
Programas de sistemas :	MP = 3.6 * (MICF ^ 1.20)

El tiempo de desarrollo para un programa según lo sugiere Boehm, es de:

Programas de aplicación :	TDD = 2.5 * (MP ^ 0.38)
Programas de apoyo :	TDD = 2.5 * (MP ^ 0.35)
Programas de sistemas :	TDD = 2.5 * (MP ^ 0.32)

Dado el número total de meses programador de un proyecto y el tiempo nominal de desarrollo requeridos, el nivel promedio de contratación puede obtenerse mediante una simple división.

$$\text{No. de programadores} = \text{MP} / \text{TDD}$$

Estos datos representan el promedio de contratación de programadores; para las fases de análisis y diseño solo se requieren algunos programadores, y tal vez durante la fase de instrumentación se necesiten del 125 al 130 % del promedio de contratación.

En los modelos de costo basados en algoritmos, los costos se estiman mediante la adición de los costos de cada uno de los módulos o subsistemas que conforman al sistema.

El Modelo constructivo de costos ó CCCCMMO (Constructive Cost Model) es un modelo de costos por algoritmos descrito por Boehm. Cuando se usa el CCCCMMO, las ecuaciones anteriormente presentadas se emplean para proporcionar los valores nominales de la estimación de meses de programador y del calendario de desarrollo para cada unidad de trabajo, basándose en el número de instrucciones de código fuente entregadas de cada unidad; después se utilizan factores multiplicadores para ajustar la estimación de acuerdo con los atributos del proyecto, del equipo y del personal. A continuación se presentan dichos multiplicadores así como sus intervalos de valores:

<u>FACTOR MULTIPLICADOR</u>	<u>INTERVALO DE VALORES</u>
-----------------------------	-----------------------------

Atributos del producto

Confiabilidad requerida.....	0.75 a 1.40
Tamaño de la Base de Datos.....	0.94 a 1.16
Complejidad del producto.....	0.70 a 1.65

Características del equipo

Limitantes en el tiempo de ejecución..	1.00 a 1.66
Limitaciones en memoria principal.....	1.00 a 1.56
Tiempo de entrega de programas.....	0.87 a 1.15

<u>FACTOR MULTIPLICADOR</u>	<u>INTERVALO DE VALORES</u>
-----------------------------	-----------------------------

Características del personal

Capacidad de los analistas.....	1.46 a 1.71
Capacidad de los programadores.....	1.42 a 0.70
Experiencia en programas de aplic. ...	1.29 a 0.82
Experiencia en lenguajes de prog. ...	1.14 a 0.95

Características del proyecto

Uso de técnicas modernas.....	1.24 a 0.82
Uso de herramientas de programación.	1.24 a 0.83
Tiempo requerido para el desarrollo...	1.23 a 1.10

Aplicando esta técnica al problema del presente trabajo tenemos que:

$$MP = 2.4 * (MICF ^ 1.05)$$

$$TDD = 2.5 * (MP ^ 0.38)$$

(En base a los requerimientos del usuario se tiene que MICF = 9)

$$MP = 2.4 * (9 ^ 1.05) = 24.1$$

$$TDD = 2.5 * (24.1 ^ 0.38) = 8.38 \text{ (meses)}$$

$$\text{No. de programadores} = 24.1 / 8.38 = 3$$

Con el fin de ajustar las estimaciones con respecto a desviaciones del valor inicial obtenido, se utilizarán los siguientes factores multiplicadores de esfuerzo.

<u>FACTOR MULTIPLICADOR</u>	<u>INTERVALO DE VALORES</u>
- Confiabilidad	0.90
- Tamaño Base de datos	0.95
- Complejidad	0.70
- Tiempo de CPU	1.20
- Memoria utilizada	1.50
- Tiempo de entrega de programas	1.00
- Calidad de los Analistas	1.10
- Calidad de los Programadores	1.10
- Experiencia en prog. de aplic.	0.90
- Experiencia en leng. de prog.	0.95
- Prácticas en técnicas modernas	0.93
- Herramientas de programación	0.98
- Tiempo requerido para el desarrollo	1.10
	<hr/>
	13.31

Por lo tanto el factor de ajuste del esfuerzo = 1.02 . Aplicando dicho factor a la estimación inicial, el ajuste produce una estimación final de 24.58 meses de programador y de 8.5 meses de desarrollo. Considerando que en México el costo promedio de analistas programadores es de \$1000 U.S. persona-mes, el costo total del personal asignado al proyecto será:

$$\text{COSTO TOTAL} : 24.58 * 1000 = \$ 24,580.00 \text{ U.S.}$$

CAPITULO III

ANALISIS DEL SISTEMA

Según el orden establecido en el ciclo de vida de un sistema, la siguiente etapa a desarrollar es la del Análisis. Análisis es el proceso que permite obtener una descripción ordenada y rigurosa del sistema que se va a construir a partir de la información concerniente a sus operaciones actuales y de los nuevos requerimientos.

Para llevar a cabo esta etapa, utilizaremos la metodología propuesta por Edward Yourdon, que es la del Análisis Estructurado.

El Análisis Estructurado se puede definir como una disciplina que intenta sistematizar el proceso de análisis de tal forma que permita obtener las especificaciones funcionales y requerimientos de un sistema.

Dichas especificaciones deben tener las siguientes características:

Deben ser...

- gráficas
- partitionadas
- mantenibles
- entendibles
- lógicas.

Según Ed. Yourdon, las principales fallas del análisis clásico las encontramos en el proceso de especificación del sistema. Algunas de estas son:

- 1) Las especificaciones funcionales en el análisis clásico son generalmente una narrativa en un documento solamente leible en forma serial del principio al fin.
- 2) Las especificaciones funcionales son imposibles de actualizar.
- 3) Debido a que las especificaciones funcionales no son entendibles por sí mismas, no se le puede mostrar nada al usuario sino hasta el final del análisis, esto bloquea la interacción que podría ayudar a refinar y perfeccionar el producto.

En contraparte, el análisis estructurado ayuda a resolver los problemas de la fase de análisis ya que:

- 1) Ataca el problema modularmente.
- 2) Ataca el problema de la comunicación en forma interactiva y con una inversión de los puntos de vista.
- 3) Resuelve el problema del mantenimiento de especificaciones por redundancia limitada.

En resumen, como resultado de la etapa de análisis se debe obtener un documento con las especificaciones funcionales del sistema. Para ello el análisis estructurado hace uso de dos herramientas, que son el Diagrama de flujo de datos y el Diccionario de datos.

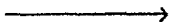
III.1) Diagrama de flujo de datos

El Diagrama de Flujo de Datos (DFD) es una de las principales herramientas del análisis estructurado y sirve para modelar un sistema particionándolo.

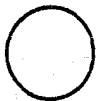
El DFD es una herramienta para modelar un sistema mediante una representación gráfica en forma de red, mismo que puede ser automatizado, manual o mixto. Un DFD retrata al sistema en término de sus componentes y las interfaces entre los componentes.

Los DFD no muestran un flujo de control, sino un flujo de datos y se construyen en base a cuatro elementos básicos.

- 1) Flujo de datos.- representados por vectores o flechas.



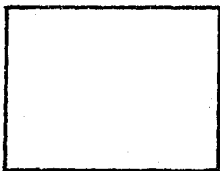
- 2) Procesos.- representados por círculos o rectángulos redondeados.



3) Archivos.- se representan por líneas rectas o rectángulos abiertos.



4) Entidades externas.- representadas por cajas o cuadrados.
(Datos fuente o destino)



FLUJO DE DATOS

El flujo de datos representa una interfase entre los componentes del diagrama de flujo de datos. Estos, se encuentran entre procesos y van o vienen de archivos o de fuentes y destinos.

Convenciones para asignar un nombre al flujo de datos:

- Los nombres están ligados con guiones y se usan letras mayúsculas. P.ej.: NUMERO-DE-CUENTA

- No hay dos flujos de datos con el mismo nombre.

- Los nombres no sólomente deben representar los datos que se muevan en el flujo de datos, si se sabe más de ellos pueden aparecer en el nombre.

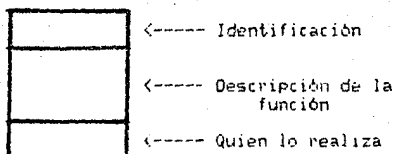
- La misma línea de flujo de datos puede llegar a dos procesos o dos salidas pueden unirse en una sola línea.

EL PROCESO

El proceso invariablemente muestra alguna cantidad de trabajo realizado sobre los datos.

Un proceso es una transformación del flujo de datos de entrada a un flujo de datos de salida. Cada proceso debe tener un nombre.

Convenciones para referenciar el proceso:



- Identificación: es un número cuya función es identificar al proceso.

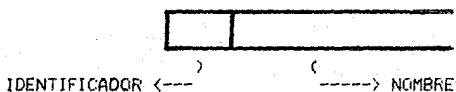
- Descripción de la función: es una oración corta que describe al proceso.

- Quién lo realiza: Es el nombre de un departamento, persona, programa, etc., que realiza la función.

EL ARCHIVO

Para los propósitos del DFD, el archivo es considerado como un depósito temporal de datos.

Convenciones:

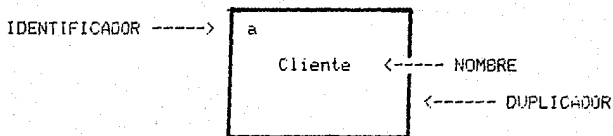


- Identificador: Para facilitar su referencia y se utiliza la letra D seguida de un número.

- Nombre: Un nombre lo más descriptivo que se pueda.

LA FUENTE ó DESTINO

La fuente ó destino es una persona ó una organización situado fuera del contexto del sistema, es el origen o el receptor de los datos del sistema.



- Identificador: Letras minúsculas asociadas a las entidades externas.

- Nombre: nombre de la entidad.

- Duplicador: indica que el símbolo se repite en el diagrama.

Cabe señalar que se pueden obtener varios DFD con diferente nivel pertenecientes a un mismo DFD de acuerdo a la complejidad del problema, de tal suerte que el diagrama cero englobará a todo el sistema y por lo tanto es el más general. No se puede hablar de un número fijo de niveles de detalle aunque se recomienda que el nivel más bajo obtenido sean primitivas funcionales.

Una primitiva funcional es un proceso que se puede relatar en una hoja de miniespecificación (pseudocódigo) y tiene una entrada y una salida.

Es recomendable que en un DFD se considere el aspecto del Balanceo, es decir que los flujos de entrada y salida que aparecen en un proceso Padre aparezcan en el DFD del proceso hijo y viceversa.

Pasos sugeridos para la elaboración del DFD.

1.- Identificar las entidades externas involucradas.

2.- Identificar las entradas y salidas, agrupándolas lógicamente y separando las que manejen errores y excepciones.

3.- Identificar las necesidades específicas de información que puedan presentarse.

4.- Realizar un diagrama tentativo el cual comience con las fuentes primarias de entrada y que contenga todas las transformaciones lógicas de los datos, excepto aquellos que manejen errores y excepciones. No hay que poner decisiones.

5.- Verificar el diagrama obtenido contra el procedimiento real asegurándose que se incluyeron todas las entradas y salidas requeridas, así como las transformaciones lógicas de los datos.

6.- Dibujar un nuevo diagrama en el que se incluyan los errores y excepciones y se minimice el número de cruces de flujo de datos.

7.- Validar el diagrama con alguno de los usuarios, mostrándole el diagrama "paseándolo" por él; mencionarle que es solamente una aproximación pidiéndole que haga todas las observaciones pertinentes.

8) Hacer una explosión de cada uno de los procesos. Es posible que esto pueda hacer que se modifique el diagrama del nivel anterior.

9) Conferir con el usuario la aceptación del proyecto para delimitar los objetivos a cumplir.

Especificación de requerimientos del usuario

1) Que exista una rutina donde se pueda crear y dar mantenimiento a Catálogo de Clientes y de Artículos. Es necesario que en el Catálogo de Clientes se tenga el saldo del cliente actualizado en todo momento.

2) Mediante los datos de los pedidos, se accederán al sistema todas las compras realizadas por los clientes, esto con la finalidad de crear su factura y su cargo correspondiente.

3) Se deben poder capturar al sistema los diferentes cargos y abonos asignados a los clientes. Dichos cargos y abonos, así como las facturas mencionadas en el inciso anterior se deben mantener almacenadas en archivos temporales con el objeto de que puedan modificarse antes de consolidarlas en archivos maestros.

4) Se deberá implementar un proceso que permita transferir los movimientos registrados en los archivos temporales hacia los archivos maestros.

5) A partir de la información contenida en archivos maestros, se requiere obtener diferentes consultas como son los estados de cuenta y la antigüedad de saldos de los clientes existentes en el catálogo.

6) Se pide que el sistema sea capaz de emitir los siguientes reportes:

- De ventas

- a) Informe de los pagos recibidos durante el día.
- b) Informe de las ventas logradas durante la jornada de trabajo.
- c) Informe de las ventas y comisiones a pagar por agente de ventas.
- d) Informe de las notas de crédito realizadas a los clientes.

- De cobranza

- a) Informe de los clientes que adeudan, del monto del adeudo y de los conceptos por los que adeudan (estados de cuenta).
- b) Informe de adeudos vencidos. (Antigüedad de saldos)
- c) Estados de cuenta por documentos. (análisis de documentos)

- Estadísticos

- a) ABC de clientes por importe de compras.
- b) ABC de clientes por volumen de compras.
- c) ABC de artículos por importe de compras.
- d) ABC de artículos por volumen de compras.

Identificación de entidades externas

Analizando la filosofía de operación del sistema, se deduce que solamente existirá una entidad externa involucrada, misma que proporcionará y extraerá la información al sistema. Para este caso dicha entidad externa es la misma empresa y por convención en lo sucesivo se le referenciará como "el usuario".

Identificación de entradas

Las primeras y fundamentales entradas a considerar son las facturas, ya que estas representan los cargos de mayor importancia que repercuten en el saldo del cliente, además de ser la fuente de información para los archivos estadísticos

Toda factura que aparezca en el sistema tendrá un NUMERO DE FACTURA único con el cual se le podrá hacer referencia. Dicho número de factura se conformará de una cadena de 6 caracteres y para su control se llevará un consecutivo que el mismo sistema actualizará en un archivo específico al momento de generar una nueva factura.

Los ingredientes de toda factura se extraerán del catálogo de artículos del cual únicamente se utilizará la descripción y la unidad. El número máximo de artículos permitidos será de 20 por factura.

Al momento de crear una factura se validarán diferentes aspectos como son el que no exista dos veces un mismo artículo en la misma factura, que no haya artículos sin cantidad ó precio unitario, que no haya renglones intermedios en blanco, que los

artículos existan en catálogo y que las cantidades no rebasen el rango de formato establecido. Para su captura se sugiere el formato de la figura III.1

Como segunda entrada al sistema se considerarán los cargos y los abonos. Para la captura de dichas entradas se utilizarán cifras de control en el número de documentos capturados y en el importe total de los mismos. Los diferentes cargos y abonos se clasificarán por un NUMERO DE CONCEPTO el cual se constituye de una cadena de tres caracteres y mediante el cual se dará la naturaleza de cargo ó abono al movimiento.

Ejemplo de cargos y abonos que se contemplan son:

- 110 : Facturación.
- 130 : Cargo por cheque devuelto.
- 150 : Cargo diverso.
- 170 : Cargo por flete.
- 211 : Abono por nota de crédito por devolución de mercancía.
- 212 : Abono por nota de crédito por descuentos o diferencia en precio.
- 213 : Abono por nota de crédito por flete no recuperado.
- 214 : Abono por nota de crédito por comisión de cheque devuelto no cobrada.
- 220 : Pagos a facturas.
- 230 : Pagos a cheques devueltos.
- 235 : Anticipos.
- 250 : Abonos diversos.

Para la captura de los diferentes cargos y abonos se recomienda utilizar el formato de la figura III.2

- Otras entradas al sistema son las altas en los catálogos de Clientes y de Artículos. Estas entradas se contemplan en el rubro de mantenimiento a catálogos y al igual que en las otras entradas, para estas también se realizan algunas validaciones como son que no exista ya en catálogo el código que se pretende dar de alta.

DATOS DEL CLIENTE		FECHA :	AGENTE :	
		CONDICIONES :	No. FED.	
		EMBARCAR POR :		
CANTIDAD	UNIDAD	CODIGO	DESCRIPCION	P. UNITARIO

FIGURA III.1 FORMATO PARA LA CAPTURA DE LOS PEDIDOS.

- g) Análisis de documentos.
- h) ABC de clientes por importe de compras.
- i) ABC de clientes por volumen de compras.
- j) ABC de artículos por importe de compras.
- k) ABC de artículos por volumen de compras.

En la figura III.3 se da el formato genérico que dichos reportes deben tener:

FECHA	NOMBRE DE LA EMPRESA NOMBRE DEL SISTEMA	PAGINA : #
HORA	NOMBRE DEL REPORTE	PROGRAMA
(LINEAS DE DETALLE)		
		----- SUBTOTAL POR ENTIDAD
		----- SUBTOTAL POR ENTIDAD
		----- TOTALES DEL REPORTE

FIGURA III.3 FORMATO PARA LA EMISION DE LOS REPORTES.

Descripción de los procesos que integran al sistema

En base a los requerimientos del usuario y una vez que se han establecido las entradas y salidas, se procede a determinar los procesos globales que deberán realizar las diferentes funciones del sistema.

1) FACTURACION

En este proceso se hará la creación de las facturas, así como las bajas, cambios y cancelaciones de las mismas. Existirán dos tipos de cancelaciones, una que es únicamente la cancelación de la factura y otra en donde además de cancelar se creará una nueva (con el número de factura que continúa según el consecutivo de facturas) que contendrá los mismos ingredientes de la factura que se cancela.

2) MOVIMIENTOS ADICIONALES

La función de este módulo es la de aceptar, validar y registrar todos los cargos y abonos que sean adjudicados a los clientes. En este proceso se determinará si existe o no la factura a la que se realiza el movimiento, y en caso de no localizarse se prenderá una bandera que indicará que ese movimiento debe tener un tratamiento especial en el cierre diario.

3) CIERRE DIARIO

Mediante este proceso se realizará la transferencia de los movimientos que se encuentran en los archivos de movimientos hacia los archivos maestros. En forma general, la finalidad de este proceso es afectar el saldo de los clientes y acumular en archivos estadísticos los importes y volúmenes correspondientes a sus compras. A los movimientos con "bandera prendida", mismos a los que se hace referencia en el proceso de MOVIMIENTOS ADICIONALES, se les almacenará en un archivo especialmente dedicado a ese fin.

4) CIERRE MENSUAL

En todo sistema es necesario contar con un procedimiento que nos permita deshacernos de los datos que ya no son útiles al mismo. Para este fin se implementará este proceso en el cual se borrarán todas las facturas que se encuentran saldadas o canceladas y todos los movimientos diversos que atañen a dichas facturas, los artículos o ingredientes que correspondan a las mismas y la información estadística ya obsoleta.

5) REALIZA CONSULTAS

En este módulo será posible realizar algunas consultas en pantalla de la información almacenada en archivos. Las consultas que se podrán hacer son, obtener un estado de cuenta de un cliente en particular y/o su antigüedad de saldos.

6) OBTIENE REPORTES

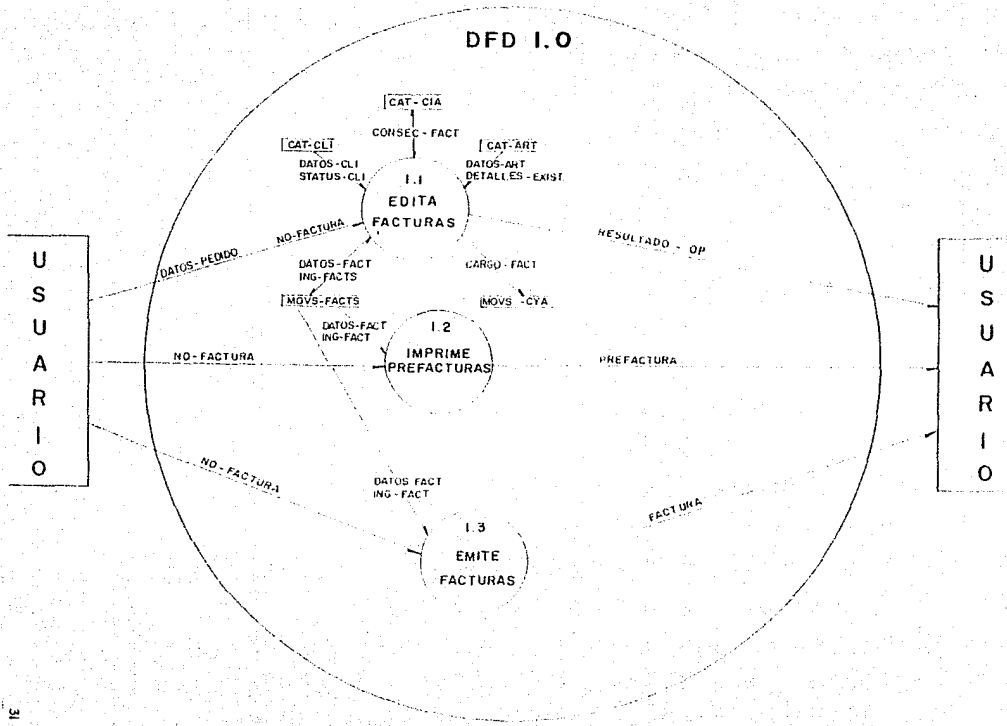
La importancia de un sistema radica en la información que produce, y es en este módulo en donde se obtienen todos los reportes requeridos por el usuario y que ya se mencionaron anteriormente.

7) MANTENIMIENTO A CATALOGOS

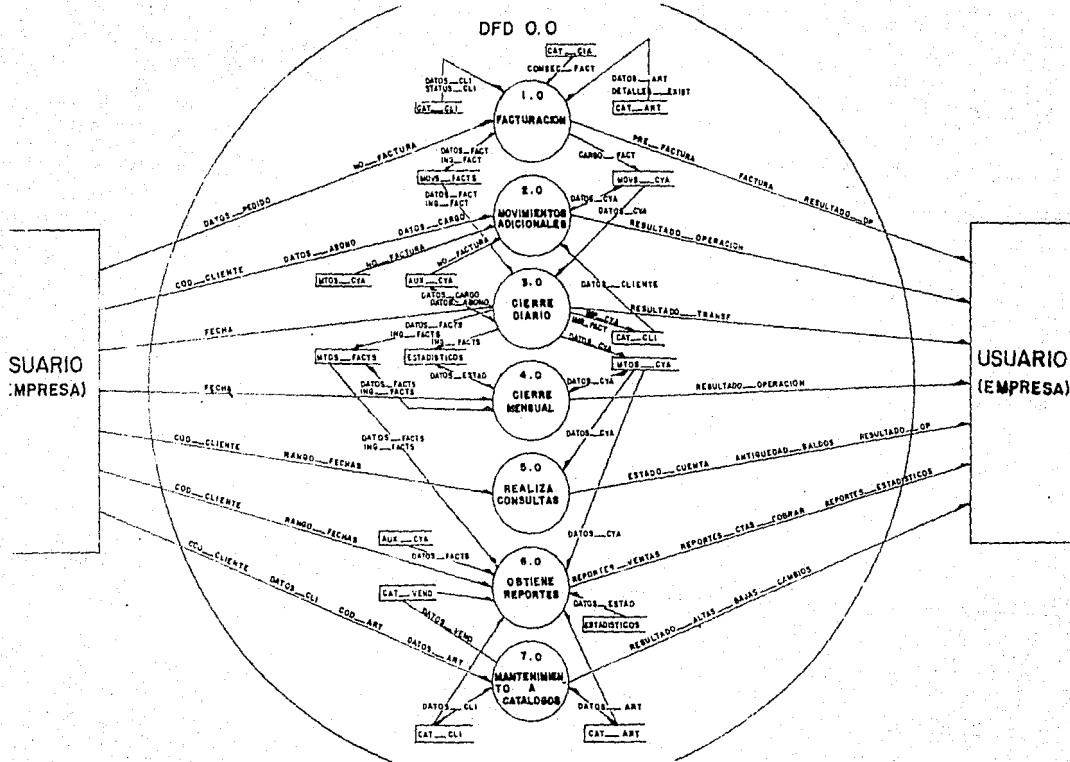
Este proceso le permitirá al usuario hacer las altas, bajas, cambios a los catálogos de Clientes y de Artículos.

Una vez que se han definido las entradas, salidas y procesos del sistema, se procede a crear el Diagrama de Flujo de Datos mismo que aparece en las páginas siguientes.

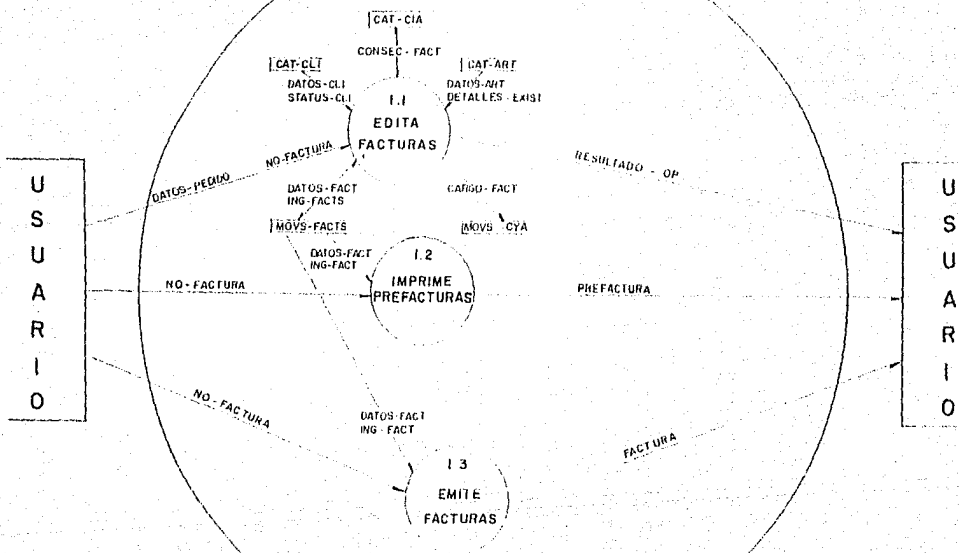
DFD I.O

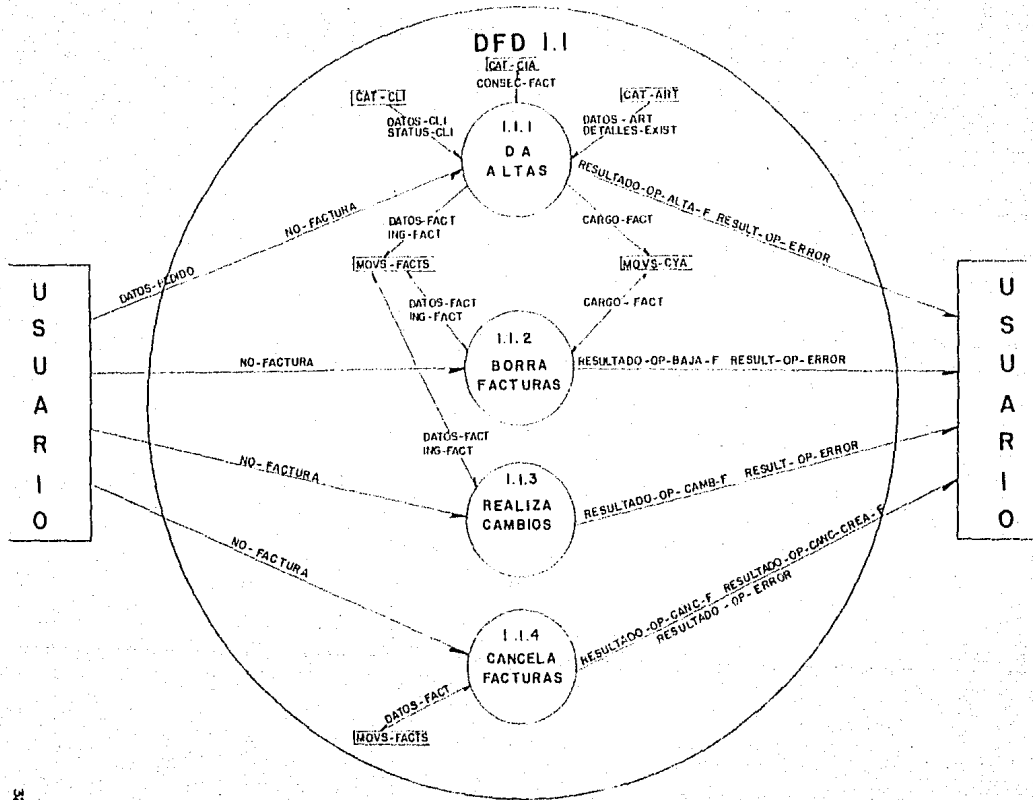


DFD 0.0

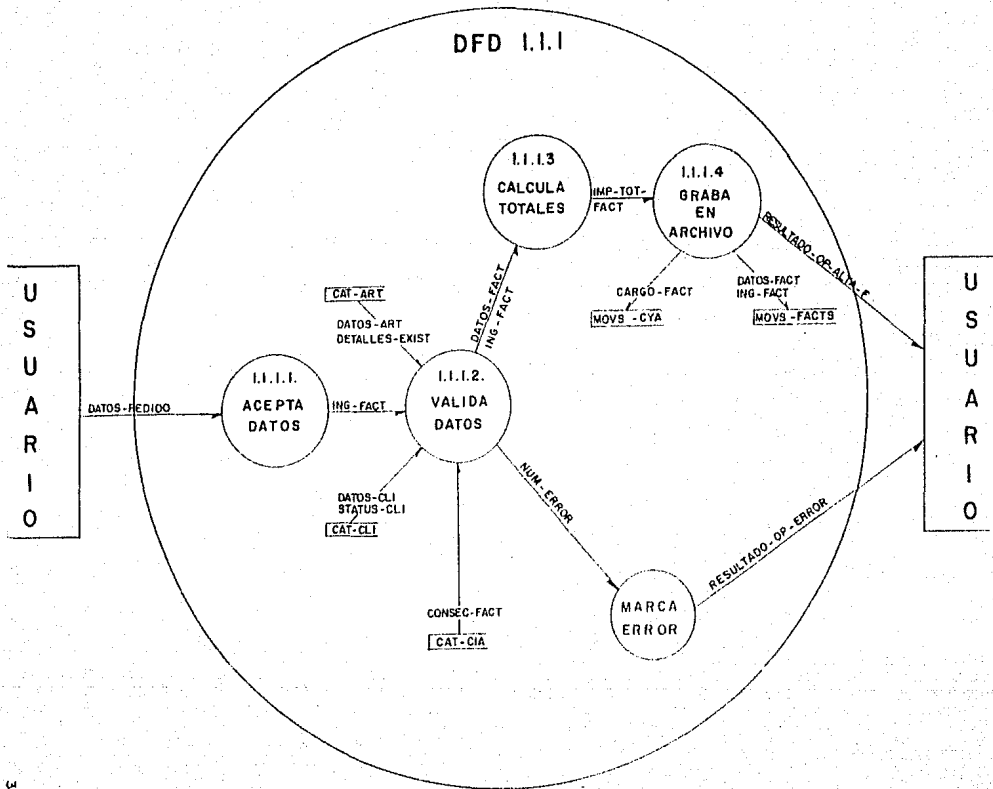


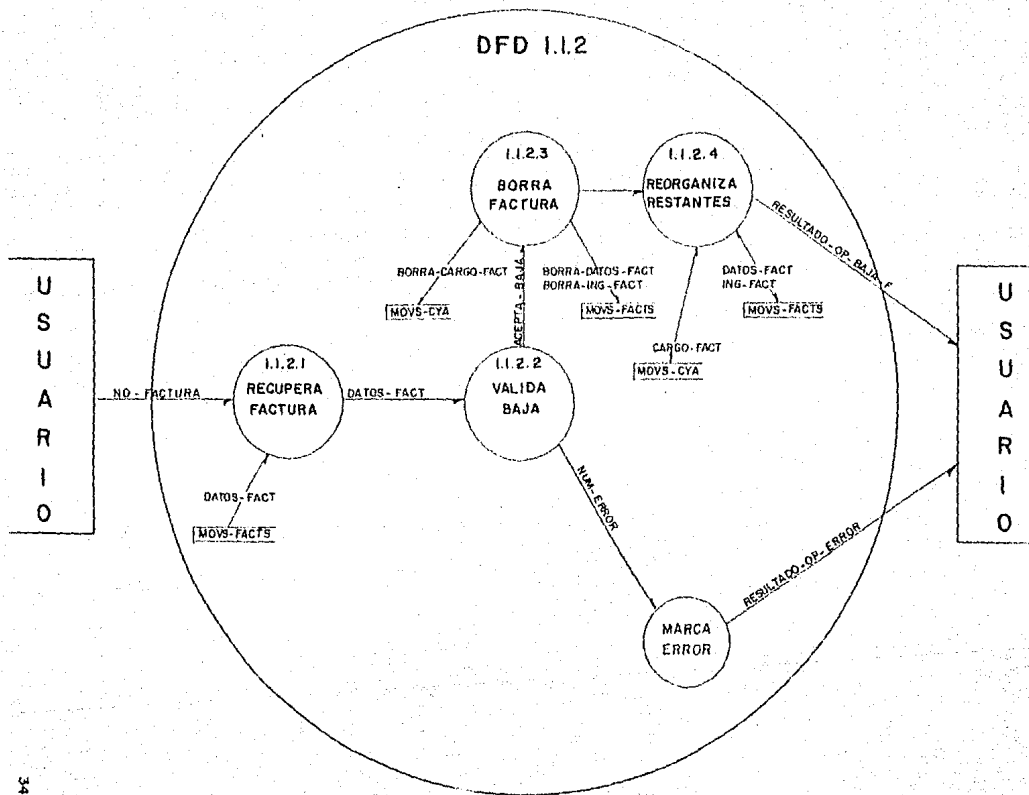
DFD 1.0



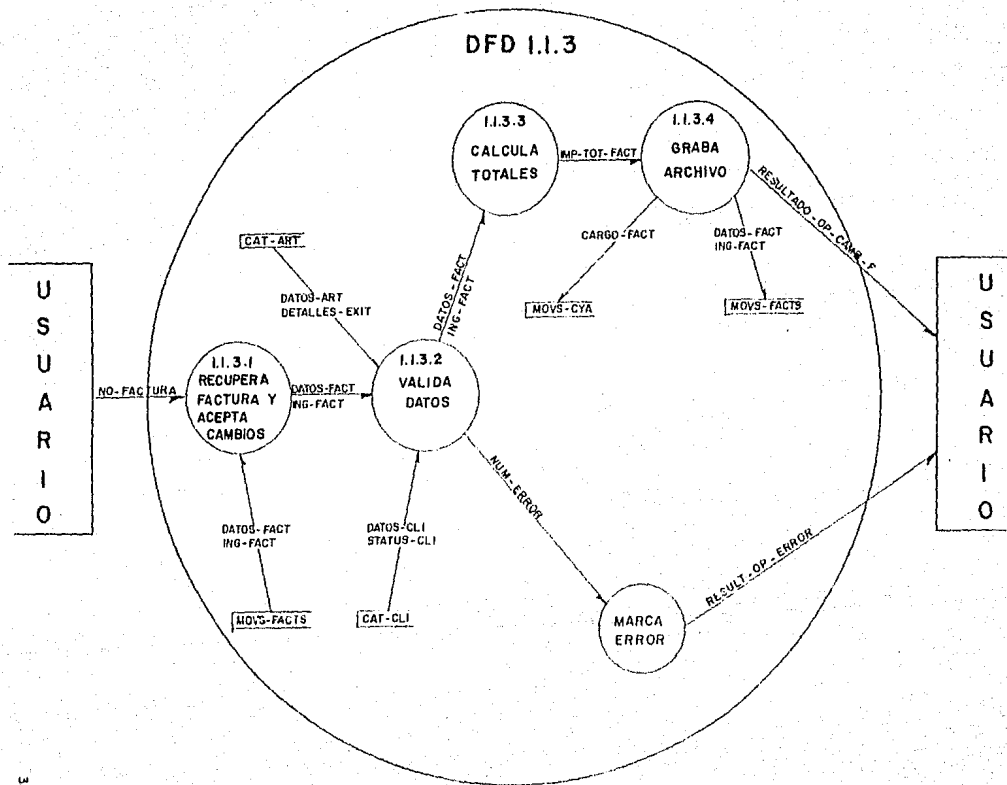


DFD 1.1.1

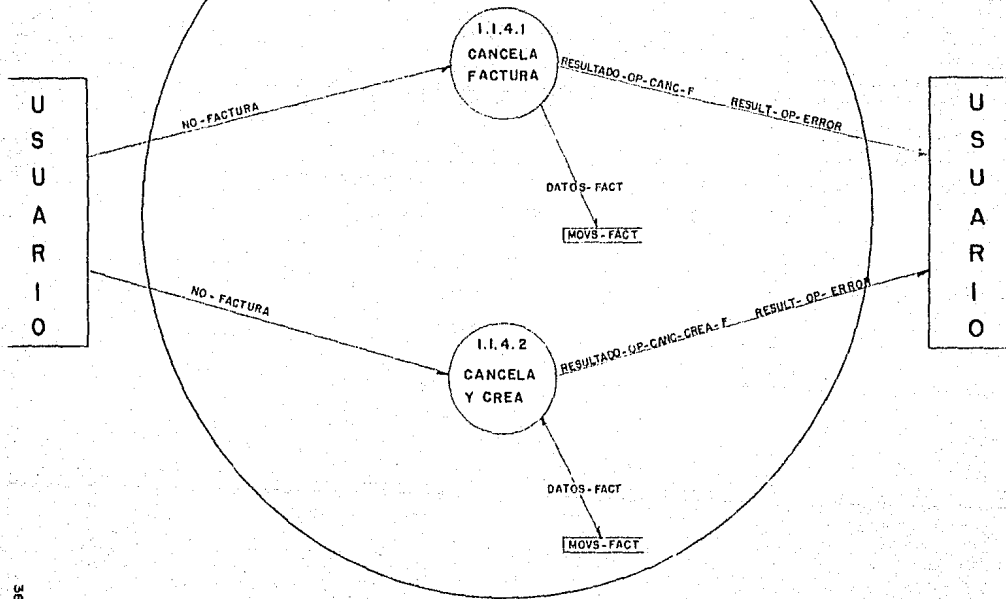




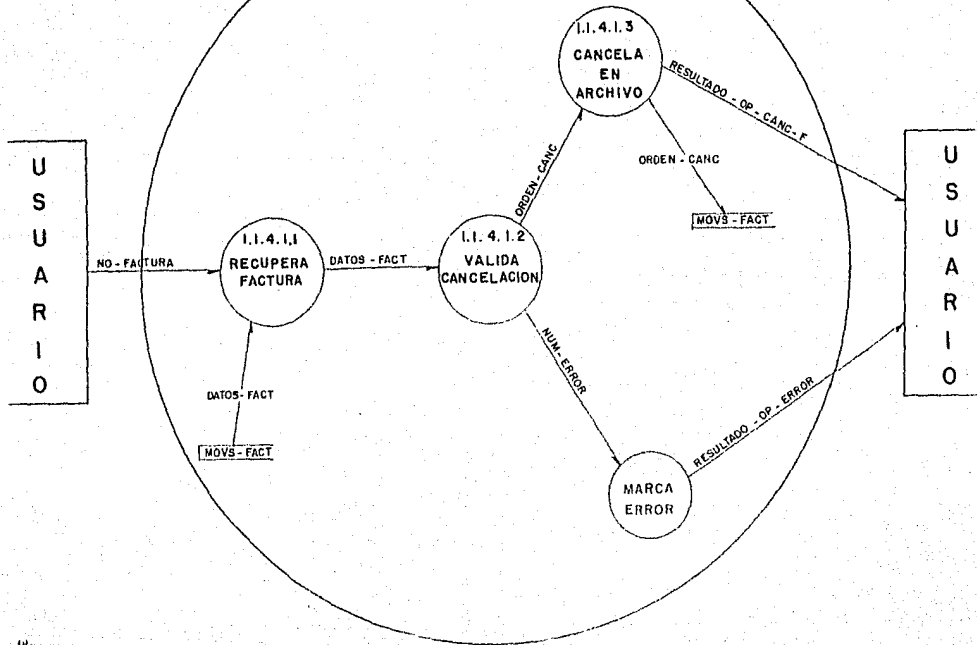
DFD 1.1.3



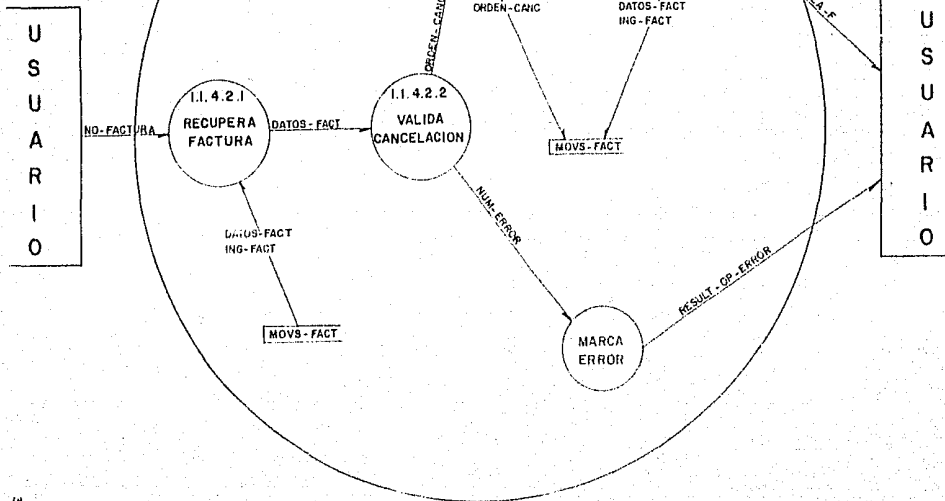
DFD 1.1.4



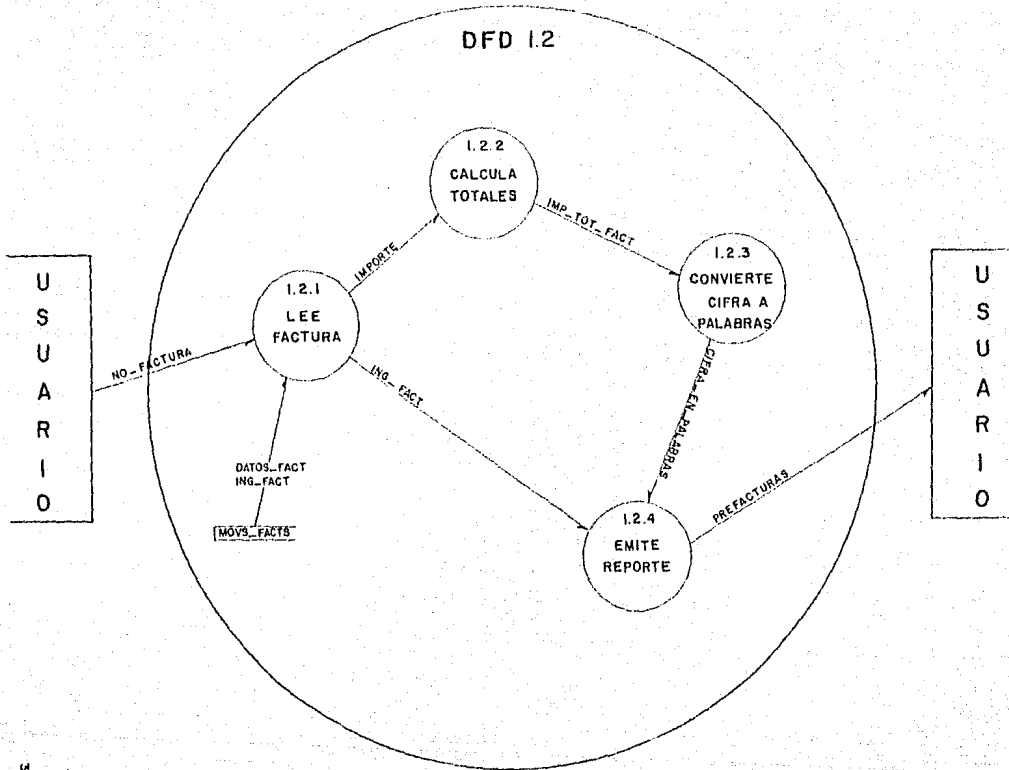
DFD 1.1.4.1



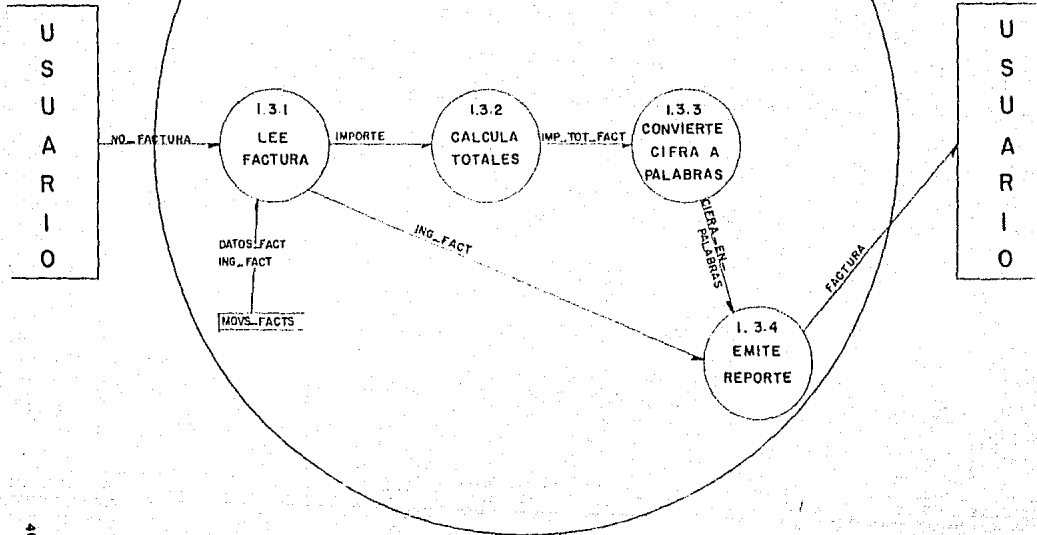
DFD 1.1.4.2



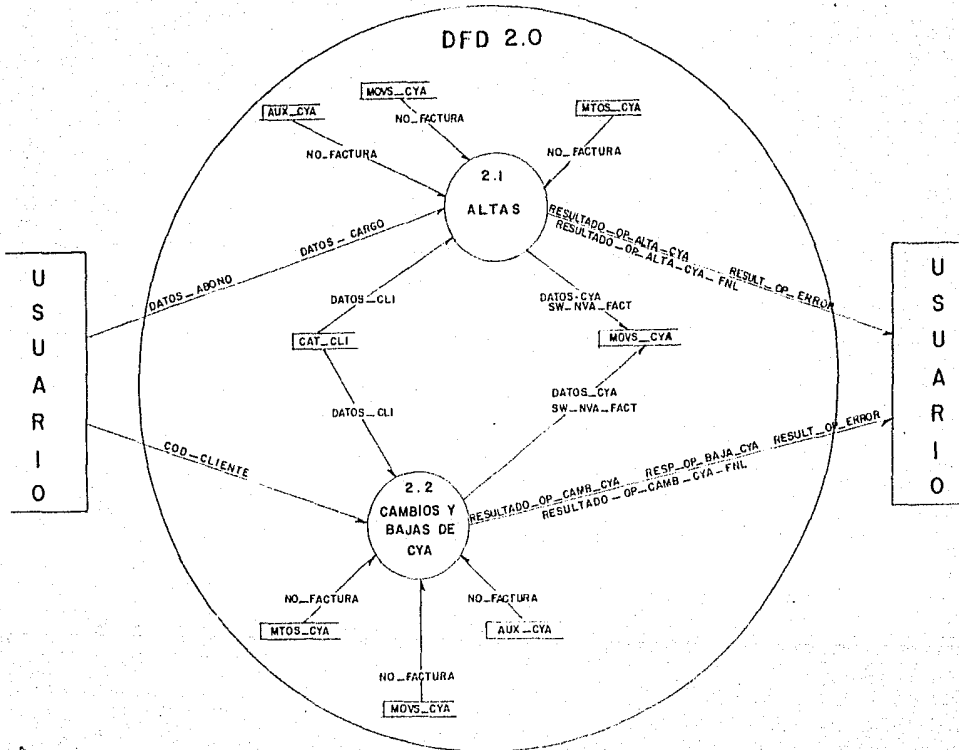
DFD 1.2



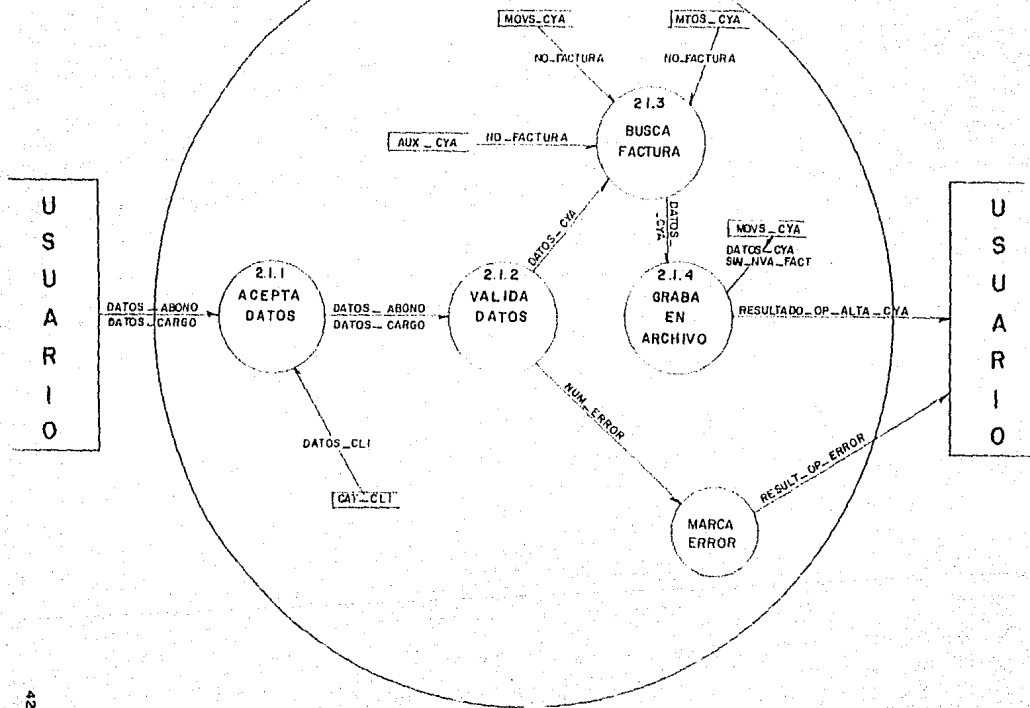
DFD 1.3



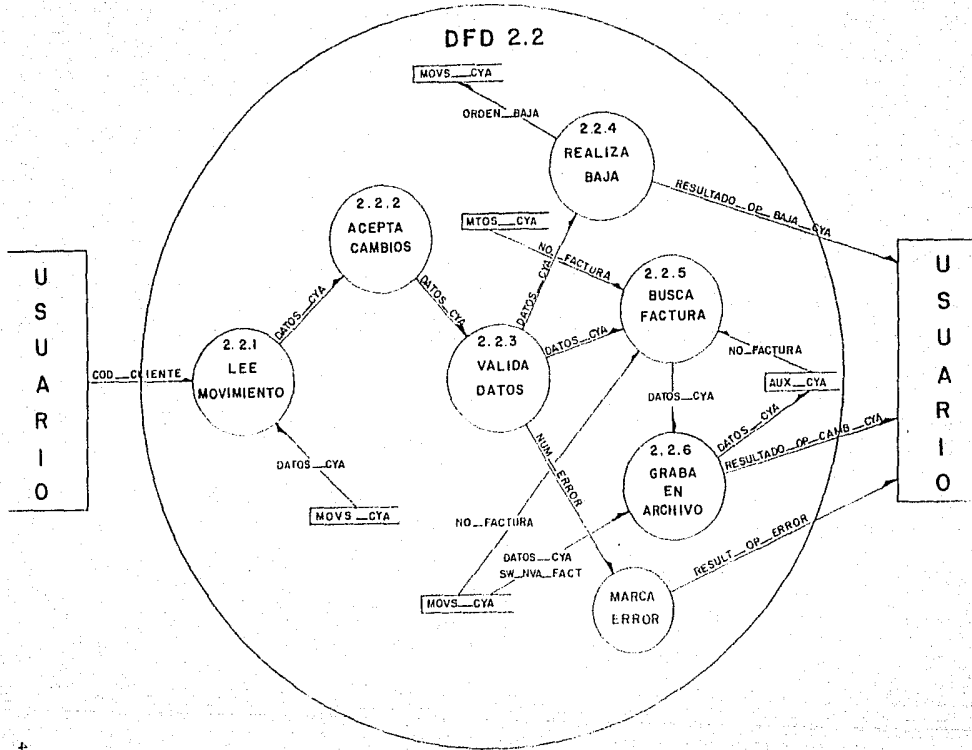
DFD 2.0

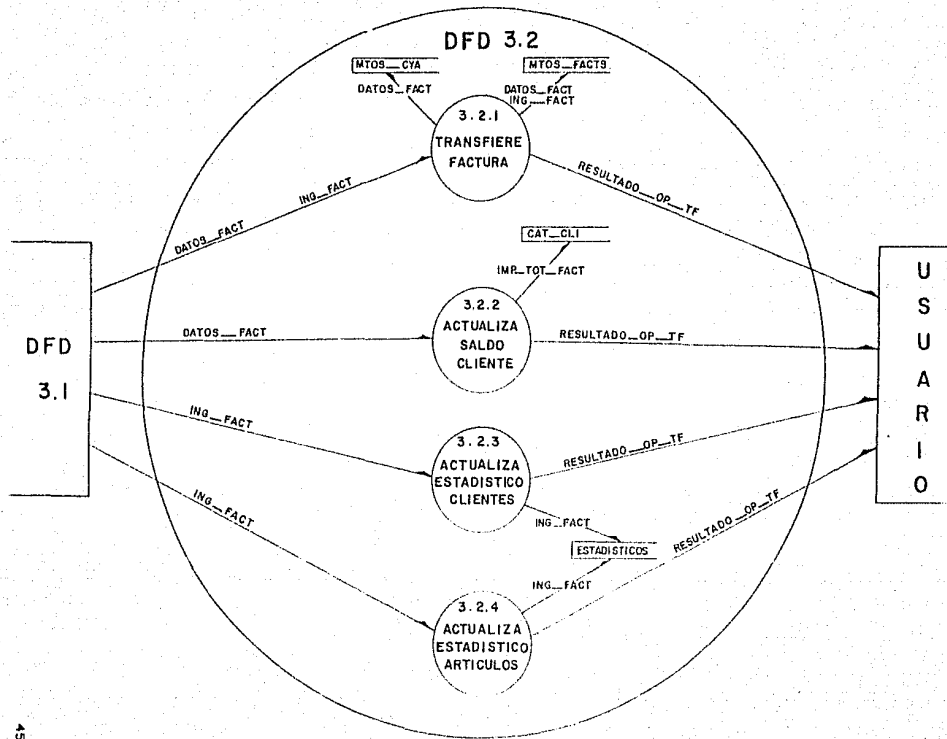


DFD 2.1



DFD 2.2





III.2) El diccionario de datos

El Diccionario de Datos (DD) es un conjunto de definiciones de las interfaces que aparecen en el DFD. Se puede decir que un DD logra que los DFD sean más rigurosos.

Un buen DD debe incluir los Flujos de datos y los almacenes de datos que se encuentran en el DFD. Para documentar dichos elementos del DD primero se debe establecer que son un Datos Elemental y una Estructura de Datos.

- Dato Elemental - es sinónimo de Data-item, campo y elemento. Cualquier dato elemental, toma valores atómicos, es decir aquellos a los que no es posible descomponer en otros elementos.

- Estructura de Datos - Es un agregado de estructuras más elementales o de otro tipo de datos (agregado de datos elementales, grupos de repetición, etc.) o de una mezcla de ambos.

Las características que debe tener un buen DD es que debe particionar la información, debe ser sencillo, no redundante y debe contar con una sección de sinónimos.

CONVENCIONES PARA ESPECIFICAR LAS ESTRUCTURAS DE DATOS

SIMBOLOGIA	SIGNIFICADO
=	es equivalente a
+	and
[]	or
(, n)	iteraciones, n = # de repetición
()	opcional

Construcción del diccionario de datos

Tomando como válidas las siguientes definiciones:

- DIGITO = 1,2,3,4,5,6,7,8,9,0
- CARACTER_NUMERICO = "1","2","3","4","5","6","7","8","9","0"
- LETRA = a,b,c,...,z,A,B,...,Z
- VAR_BOOLEANA = [T] [F]

Tenemos que:

DATOS ELEMENTALES

- CANTIDAD = (DIGITO) + "." + (DIGITO)
- CIFRA_EN_PALABRAS = ((LETRA) + " " + (LETRA))
- CODIGO = (CARACTER_NUMERICO,5)
- COD_CONCEPTO = (DIGITO,3)
- COD_VEND = (DIGITO,2)
- COLONIA = ((LETRA) + " " + (LETRA))
- COMENTARIO = ((LETRA) + " " + (LETRA))
- COMISION = (CARACTER_NUMERICO,2)
- COND_VENT = (DIGITO,2)
- DATOS_ART = COD_ART + DESCRIP + UNIDAD
- DATOS_FACT = NO_FACTURA + COD_CLIENTE + FECHA + CONDIC_VTA
+ COD_VEND
- DESCRIP = ((LETRA) + " " + (LETRA))
- DETALLES_EXIST = (DIGITO) + "." + (DIGITO)
- DET_SI_EXIST = VAR_BOOLEANA
- DOCUMENTO_FTE = (DIGITO,3)
- FECHA = dd + mm + aa
- IMPORTE = (DIGITO) + "." + (DIGITO)

- MENSAJE = ((LETRA) + " " + (LETRA))
- NO_FACTURA = (CARACTER_NUMERICO,6)
- NOMB_VEND = ((LETRA) + " " + (LETRA))
- NOMB_CALLE = ((LETRA) + " " + (LETRA))
- NUMERO = (DIGITO)
- NUM_ERROR = (DIGITO,2)
- PREC_UNIT = (DIGITO) + "." + (DIGITO)
- RAZON_SOC = ((LETRA) + " " + (LETRA))
- RUTA = (DIGITO,2)
- SW_CANCEL, SW_MES, SW_TRIM, SW_SEM, SW_ANU = VAR_BOOLEANA
- TELEFONO = (DIGITO,7)
- TIPO_PERIODO = [SW_MES] [SW_TRIM] [SW_SEM] [SW_ANU]
- UNIDAD = (LETRA,3)

ESTRUCTURAS DE DATOS

- ANTIGUEDAD_SALDOS = (FECHA+NO_FACTURA+[SALDO_CORRIENTE]
[VENC_30_DIAS] [VENC_MAS_90_DIAS]+SALDO)
- DATOS_ABONO = COD_CONCEPTO + COD_CLIENTE + DOCUMENTO_FTE +
COMENTARIO + NO_FACTURA + FECHA + IMP_ABONO
- DATOS_CLI = COD_CLIENTE + RAZON_SOC + DIRECCION + COLONIA
+ POBLACION + COD_VEND + TELEFONO
- DATOS_ESTAD = CODIGO + [CANT_MENS] [CANT_TRIM] [CANT_SEM]
[CANT_ANU]
- DATOS_NC = COD_CLIENTE + RAZON_SOC + DOCUMENTO_FTE +
NO_FACTURA + FECHA + IMPORTE
- DATOS_PAGOS = COD_CLIENTE + RAZON_SOC + DOCUMENTO_FTE +
NO_FACTURA + FECHA + IMPORTE_PAGO

- DATOS_PEDIDO = COD_CLIENTE + FECHA + COND_VENT + COD_VEND
+ (LINEA_PEDIDO)
- DATOS_VEND = COD_VEND + NOMB_VEND + COMISION
- DIRECCION = NOMB_CALLE + NUMERO
- ESTADO_CUENTA = (FECHA + NO_FACTURA + DOCUMENTO_FTE +
[IMP_CARGO] [IMP_ABONO] + SALDO)
- FACTURA = DATOS_FACT + ING_FACT + IMPORTE
- LINEA_PEDIDO = COD_ART + CANTIDAD + PREC_UNIT +
DESCRIP_ART
- RANGO_FECHAS = FECHA_INICIAL + FECHA_FINAL
- RELACION_COM_PAGAR = (DATOS_VEND + (FECHA + NO_FACTURA +
COD_CLIENTE + RAZON_SOC + IMP_TOT_FACT + COMISION +
IMPORTE_COM) + (FECHA + NO_FACTURA + NO_NOTA_CRED +
COD_CLIENTE + RAZON_SOC + IMP_TOT_FACT + COMISION +
DESCTO_COM))
- RELACION_NOTAS_CRED = (FECHA + DOCUMENTO_FTE +
COD_CLIENTE + RAZON_SOC + IMPORTE_NC + IVA +
IMPORTE_TOTAL_NC)
- RELACION_PAGOS = (COD_CLIENTE + RAZON_SOC + NO_FACTURA +
FEC_EMISION + FEC_VENCIMIENTO + IMPORTE_PAGO)
- RELACION_VENTAS = (NO_FACTURA + FECHA + COD_CLIENTE +
RAZON_SOC + COD_VEND + IMP_TOT_FACT)
- REP_ABC_ART = (COD_ART + DESCRIP_ART + [CANT_MENS]
[CANT_TRIM] [CANT_SEM] [CANT_ANU]) + TOTAL_REP
- REP_ABC_CLIENTES = (COD_CLIENTE + RAZON_SOC + COD_VEND +
[CANT_MENS] [CANT_TRIM] [CANT_SEM] [CANT_ANU]) + TOTAL_REP
- REP_ANAL_DOCTOS = (COD_CLIENTE + RAZON_SOC + (NO_FACTURA

- + SALDO_FACT + (FECHA + COMENTARIO + FEC_VENCIMIENTO + (IMP_CARGO) [IMP_ABONO] + SALDO))
- REP_ANT_SDDS = (COD_CLIENTE + RAZON_SOC + COD_VEND + (NO_FACTURA + FECHA + [SALDO_CORRIENTE] [VENC_30_DIAS] [VENC_60_DIAS] [VENC_MAS_90_DIAS] + SALDO))
- REP_CTES_VEND = (DATOS_VEND + (COD_CLIENTE + RAZON_SOC + VENTA_PROM_AÑO_ANT + VENTA_PROM_MENS + VENTA_MES + VENTA_ACUM_ANUAL) + TOTAL_VEND) + TOTAL_REP
- REP_ESTADO_CTA = (COD_CLIENTE + RAZON_SOC + (FECHA + COMENTARIO + NO_FACTURA + [IMP_CARGO][IMP_ABONO] + SALDO))
- STATUS_CLI = COND_VENT + LIM_CRED

ALMACENES DE DATOS

- MTOS_FACTS = NO_FACTURA + COD_VEND + COMISION + COND_VENT + DESCTO1 + DESCTO2 + SUBTOTAL + IVA + SW_EMIT + SW_CANCEL + FEC_EMISION
- MTOS_CYA = COD_CLIENTE + COD_CONCEPTO + COMENTARIO + IMPORTE + FECHA + DOCUMENTO_FTE + NO_FACTURA + SW_NVA_FACT
- MTOS_FACTS = NO_FACTURA + COD_VEND + COMISION + COND_VENT + FEC_EMISION + FEC_VENCIMIENTO + DESCTO1 + DESCTO2 + SUBTOTAL + IVA + SALDO + SW_EMIT + SW_CANCEL
- AUX_CYA = COD_CLIENTE + NO_FACTURA + SALDO
- CAT_ART = COD_ART + DESCRIP + DETALLES_EXIST + UNIDAD + COSTO
- CAT_CLI = COD_CLIENTE + RAZON_SOC + TELEFONO + DIRECCION + POBLACION + COD_POSTAL + COD_VEND + COND_VENT + LIM_CRED + SALDO + RUTA + FECHA
- CAT_VEND = COD_VEND + NOMB_VEND + COMISION

- CAT_CIA = NOMBRE + RFC + DIRECCION + COLONIA + POBLACION +
REG_IMSS + NO_OF_HDA + REP_LEGAL + RFC_REF + FACT_EDIT
FACT_EMIT
- CAT_CDV = COND_VENT + DESCRIP
- CAT_CON = COD_CONCEPTO + DESCRIP
- MTO_IFA = NO_FACTURA + COD_ART
- ESTAD_CLI = COD_CLIENTE + IMPORTE
- ESTAD_ART = COD_ART + IMPORTE

SINONIMOS

- 1) CODIGO = COD_ART = COD_CLIENTE
- 2) COLONIA = POBLACION
- 3) DATOS_ABONO = DATOS_CARGO = DATOS_CYA
- 4) DATOS_ESTAD = ESTAD_CTES = ESTAD_ART
- 5) DESCRIP = DESCRIP_ART
- 6) DOCUMENTO_FTE = NO_NOTA_CRED
- 7) FACTURA = PREFACTURA
- 8) FECHA = FECHA_INICIAL = FECHA_FINAL = FEC_EMISION =
FEC_VENCIMIENTO
- 9) IMPORTE = CANTIDAD = SALDO = IMP_TOT_FACT = IMP_CARGO =
IMP_ABONO = SALDO_CERO = IMPORTE_NC = LIM_CRED = CARGO_FACT =
SALDO_FACT = IVA = SALDO_CORRIENTE = IMPORTE_PAGO = VENC_30_DIAS
= VENC_60_DIAS = VENC_MAS_90_DIAS = IMPORTE_COM = DESCTO_COM =
IMPORTE_TOT_NC = TOTAL_REF = CANT_MENS = CANT_TRIM = CANT_SEM =
CANT_ANU = TOTAL_VEND = DESCTO1 = DESCTO2 = SUBTOTAL = COSTO =

VENTA_FROM_AMO_ANT = VENTA_PROM_MENS = VENTA_MES =
VENTA_ACUM_ANUAL
- 10) LINEA_PEDIDO = ING_FACT
- 11) MENSAJE = RESULTADO_OP_ALTA_F = RESULTADO_OP_BAJA_F =
RESULTADO_OP_CAMB_F=RESULTADO_OP_CANC_F=RESULTADO_OP_CANC_CREA_F =
RESULTADO_OP_ALTA_CYA=RESULTADO_OP_CAMB_CYA=RESULTADO_OP_BAJA_CYA=
RESULTADO_OP_TC = RESULTADO_OP_TA = RESULTADO_OP_BCYA =
RESULTADO_OP_BFC = RESULTADO_OP_BFS = RESULTADO_OP_BCYAX =
RESULTADO_OP_BFSX = RESULTADO_OP_MTTTO_CLI_CB =
RESULTADO_OP_MTTTO_CLI_ALTA = RESULTADO_OP_MTTTO_ART_CB =

RESULTADO_OP_MTTD_ART_ALTA = RESULT_OP_ERROR

12) MOVIS_CYA = MTOS_CYA

13) NO_FACTURA = NO_FACTURA_CANC = NO_FACTURA_SALDADA =
NO_FACTURA_VENCIDA = CONSEC_FACT = FACT_EDIT = FACT_EMIT

(11.3) MINIESPECIFICACIONES

Una miniespecificación es un documento en el que se da la descripción de la política que gobierna la transformación de las entradas a sus correspondientes salidas dentro de un proceso. Su objetivo es proporcionar la información concisa de lo que se pretende realizar. Las herramientas que nos ofrece el análisis estructurado para el desarrollo de dichas miniespecificaciones, son el Español estructurado, las Tablas de decisión y los Árboles de decisión. El español estructurado es un lenguaje de especificación considerado como un subconjunto del idioma español con un vocabulario limitado, sintaxis de status limitada y un conjunto limitado de maneras de combinar oraciones.

Una miniespecificación debe ser entendible para toda la gente y no debe tener expresiones algebraicas matemáticas, en su lugar estas deben narrarse.

Las miniespecificaciones que constituyen al sistema de cuentas por cobrar se realizaron en español estructurado y a continuación se presentan:

DEFINICION DE MINIESPECIFICACIONES
DEL SISTEMA DE CUENTAS POR COBRAR

PROCESO : MENU PRINCIPAL DEL SISTEMA (0.0)

OBJETIVO : Mostrar las opciones del menú principal del sistema y permitir seleccionar alguna de ellas para su ejecución.

Proceso : SISTEMA DE CUENTAS POR COBRAR

REPITE

Despliega el menú principal

lee opción

CASO de opción :

- 1) Realizar el proceso de facturación
- 2) Capturar los movimientos adicionales
- 3) Realizar el cierre diario
- 4) Realizar el cierre mensual
- 5) Realiza consultas
- 6) Obtiene reportes
- 7) Realiza mantenimiento a catálogos
- *) Salir del sistema

De otro modo:

Despliega mensaje de error

Hasta opción = *

FIN DEL PROCESO

PROCESO : REALIZA EL PROCESO DE FACTURACION (1.0)

OBJETIVO : Permitir la captura de los pedidos de los
clientes y crear la factura correspondiente.

Proceso : FACTURACION

REPITE

Despliega el menú de facturación

Lee opción

CASO de opción :

- 1) Realizar la edición de las facturas
- 2) Imprimir la prefacturación
- 3) Realizar la emisión de las facturas
- *) Regresar al menú anterior

De otro modo:

Despliega mensaje de error

Hasta opción = *

FIN DEL PROCESO

EBOCESO : REALIZA LA EDICION DE LAS FACTURAS (1.1)

OBJETIVO : Realizar la captura de la factura (altas),
así como las bajas, cambios y cancelaciones

Proceso : EDITA FACTURAS

REPITE

Despliega pantalla de captura

Recupera el consecutivo de facturas

Acepta ó modifica el número de factura

Captura datos generales de la factura

Captura los ingredientes de la factura

Lee tecla de función

CASO de tecla de función:

- 1) Realiza la alta
- 2) Borra facturas
- 3) Realiza cambios
- 4) Cancela facturas

FIN_CASO

Hasta número factura = '

FIN DEL PROCESO

PROCESO : REALIZA LA ALTA DE LA FACTURA (1.1.1)

OBJETIVO : Validar los datos de la factura y grabarla
en los archivos correspondientes.

Proceso : DA ALTAS

Verifica que no existan datos no válidos

Acepta descuentos especiales

Calcula totales de la factura

SI es nueva factura

Graba la factura en archivos

DE OTRO MODO

Borra factura anterior

Graba la factura en archivos

FIN_SI

Incrementa consecutivo de facturas

FIN DEL PROCESO

EBDCESD : BORRA FACTURAS (1.1.2)

OBJETIVO : Borra de archivos de movimientos las facturas
que el usuario considere pertinentes.

Proceso : BORRA FACTURAS

Verifica que la factura no esté cancelada

SI factura cancelada

Despliega mensaje de error

DE OTRO MODO

Verifica que la factura no se haya emitido

SI factura emitida

Despliega mensaje de error

DE OTRO MODO

Borra la factura de archivos

Decrementa el consecutivo de facturas

FIN_SI

FIN_SI

FIN DEL PROCESO

PROCESO : REALIZA CAMBIOS EN LAS FACTURAS (1.1.3)

OBJETIVO : Permite realizar las modificaciones necesarias
en las facturas no emitidas ni canceladas

Proceso : REALIZA CAMBIOS

Verifica que la factura no esté cancelada

SI factura cancelada

Despliega mensaje de error

DE OTRO MODO

Verifica que la factura no se haya emitido

SI factura emitida

Despliega mensaje de error

DE OTRO MODO

Acepta cambios

Verifica que no existan datos no válidos

Acepta descuentos especiales

Calcula nuevos totales de la factura

Borra factura anterior

Graba la factura en archivos

Incrementa consecutivo de facturas

FIN_SI

FIN_SI

FIN DEL PROCESO

PROCESO : REALIZA CANCELACIONES DE FACTURAS (1.1.4)

OBJETIVO : Cancelar las facturas que por alguna razón ya no sean útiles al usuario. La cancelación puede ser con ó sin creación de factura.

Proceso : CANCELA FACTURAS

Verifica que la factura no este cancelada

SI factura cancelada

Despliega mensaje de error

DE OTRO MODO

Verifica que la factura no se haya emitido

SI factura emitida

Despliega mensaje de error

DE OTRO MODO

SI cancelación con creación de factura

Recupera datos de la factura anterior

Crea nueva factura

Graba nueva factura en archivos

Cancela factura anterior

Despliega mensaje al usuario

DE OTRO MODO

Cancela factura

Despliega mensaje al usuario

FIN_SI

FIN_SI

FIN_SI

FIN DEL PROCESO

PROCESO : REALIZAR LA IMPRESION DE LAS PREFACTURAS (1.2)
OBJETIVO : Obtener un reporte de las prefacturas del dia
con la finalidad de revisarlas y localizar
posibles errores.

Proceso : IMPRIME PREFACTURAS

Pide el rango de prefacturas a imprimir

REPITE

Despliega datos de la prefactura

SI datos correctos

Calcula totales

Imprime prefactura

DE OTRO MODO

Despliega mensaje de error

Suspende proceso

FIN_SI

Incrementa factura inicial

Hasta factura inicial = factura final

FIN DEL PROCESO

EBULESO : REALIZAR LA IMPRESION DE LAS FACTURAS (1.3)

OBJETIVO : Obtener la impresión de las facturas que
amparan los pedidos surtidos a los clientes para
su posterior envío.

Proceso : EMITE FACTURAS

Selecciona el tipo de impresora a utilizar
Pide el rango de facturas a imprimir
Valida que el rango seleccionado sea apropiado

REPITE

Despliega datos de la factura
Si datos correctos
Calcula totales
Convierte cifra total a palabras
Imprime factura

DE OTRO MODO

Despliega mensaje de error
Suspende proceso

FIN_SI

Incrementa factura inicial
Hasta factura inicial = factura final

FIN DEL PROCESO

PROCESO : CAPTURAR LOS MOVIMIENTOS ADICIONALES (2.0)

OBJETIVO : Realizar la captura de los restantes cargos
y abonos asignados al cliente.

Proceso : MOVIMIENTOS ADICIONALES

REPITE

Despliega menú de opciones

Lee opción rango seleccionado sea apropiado

CASO de opción :

- 1) Realiza altas
- 2) Realiza cambios y bajas
- *) Regresa a menú anterior

De otro modo

Despliega mensaje de error

Hasta opción = *

FIN DEL PROCESO

PROCESO : ALTAS DE CARGOS Y ABONOS ADICIONALES (2.1)
OBJETIVO : Dar de alta los cargos y abonos adicionales
de los clientes.

Proceso : ALTAS

Pide cifras de control

REPITE

Despliega pantalla de captura

Lee la clave del cargo o abono

Valida el concepto proporcionado

SI concepto valido

Pide datos del movimiento

Si datos correctos

Graba nuevo registro e incrementa totales

DE OTRO MODO

Despliega mensaje de error

FIN_SI

DE OTRO MODO

Despliega mensaje de error

FIN_SI

Hasta concepto = '000'

Compara cifras de control VS. totales

SI cantidades son diferentes

Despliega mensaje de error

FIN_SI

FIN DEL PROCESO

PROCESO : CAMBIOS Y BAJAS DE CARGOS Y ABONOS (2.2)

OBJETIVO : Editar los cargos y abonos adicionales
de los clientes.

Proceso : CAMBIOS Y BAJAS

REPITE

Despliega pantalla de edición

Pide el código del cliente

Valida el código proporcionado

SI código valido

REPITE

Despliega datos del cargo ó abono

Acepta cambios al movimiento

SI datos correctos e importe no es cero

Borra registro anterior

Graba nuevo registro

DE OTRO MODO

SI importe es cero

Borra registro

DE OTRO MODO

Despliega mensaje de error

FIN_SI

FIN_SI

Hasta fin de movimientos del cliente

DE OTRO MODO

Despliega mensaje de error

FIN_SI

Hasta código = '

FIN DEL PROCESO

PROCESO : REALIZA EL CIERRE DIARIO (3.0)

OBJETIVO : Transferir los cargos y abonos de archivos de movimientos hacia archivos maestros, además de afectar saldos de clientes y archivos estadísticos.

Proceso : REALIZA CIERRE DIARIO

Respalda archivos de datos

Transfiere facturas

Transfiere cargos

Transfiere abonos

FIN DEL PROCESO

PROCESO : TRANSFIERE FACTURAS (3.2)

OBJETIVO : Realizar la transferencia de las facturas
hacia los archivos maestros afectando el
saldo del cliente así como los acumulados de
los archivos estadísticos.

Proceso : TRANSFIERE FACTURAS

Valida la transferencia

SI hay facturas con problemas

Despliega mensaje de error

Suspende el proceso

DE OTRO MODO

REPITE

Determina la fecha de vencimiento

Transfiere la factura a maestros

Crea el cargo por facturación

Afecta el saldo del cliente

Actualiza el estadístico de clientes

Actualiza el estadístico de artículos

Hasta transferir todas las facturas

FIN_SI

FIN DEL PROCESO

PROCESO : TRANSFERENCIA DE CARGOS (3.3)

OBJETIVO : Transferir los diversos cargos que afectan
el saldo del cliente en archivos maestros.

Proceso : TRANSFIERE CARGOS

REPITE

Valida el concepto

SI no es concepto válido de cargo

Almacena mensaje de error

DE OTRO MODO

REPITE

Transfiere el cargo a maestros

SI switch de nueva factura

Usa el archivo especial para esas facturas

Busca la factura

SI factura localizada

Afecta saldo de la factura

DE OTRO MODO

Crea nuevo registro con el saldo de la factura

FIN_SI

DE OTRO MODO

Afecta el saldo de la factura

FIN_SI

Afecta el saldo del cliente

Hasta transferir todos los cargos del mismo concepto

FIN_SI

Hasta último concepto de cargo

FIN DEL PROCESO

PROCESO : TRANSFERENCIA DE ABONOS (3.4)

OBJETIVO : Transferir los diversos abonos que afectan
el saldo del cliente en archivos maestros.

Proceso : TRANSFIERE ABONOS

REPITE

Valida el concepto

SI no es concepto válido de abono

Almacena mensaje de error

DE OTRO MODO

REPITE

Transfiere el abono a maestros

SI switch de nueva factura

Usa el archivo especial para esas facturas

Busca la factura

SI factura localizada

Afecta saldo de la factura

DE OTRO MODO

Crea nuevo registro con el saldo de la factura

FIN_SI

DE OTRO MODO

Afecta el saldo de la factura

FIN_SI

Afecta el saldo del cliente

Hasta transferir todos los abonos del mismo concepto

FIN_SI

Hasta último concepto de abono

FIN DEL PROCESO

PROCESO : REALIZA EL CIERRE MENSUAL (4.0)

OBJETIVO : Depurar los archivos maestros borrando la información que se encuentre obsoleta.

Proceso : REALIZA CIERRE MENSUAL

Respalda archivos de datos

Verifica reportes emitidos

Borra facturas de archivos maestros

Borra facturas de AUXCYA

Borra estadísticos

FIN DEL PROCESO

PROCESO : VERIFICA LA EMISION DE REPORTES MENSUALES (4.1)

OBJETIVO : Verificar que se hayan emitido todos los reportes necesarios antes de ejecutar el proceso del cierre mensual.

Proceso : VERIFICA REPORTES EMITIDOS

Determina la naturaleza de los reportes.

CASO de tipo de reporte :

 mensual : prende switch de mes

 trimestral : prende switch trimestral

 semestral : prende switch semestral

 anual : prende switch anual

FIN_CASO

SI switch de mes y hay reportes mensuales no emitidos
 despliega mensaje de error

FIN_SI

SI switch trimestral y hay reportes trimest. no emitidos
 despliega mensaje de error

FIN_SI

SI switch semestral y hay reportes semest. no emitidos
 despliega mensaje de error

FIN_SI

SI switch anual y hay reportes anuales no emitidos
 despliega mensaje de error

FIN_SI

FIN DEL PROCESO

PROCESO : BORRADO DE FACTURAS DE ARCHIVOS MAESTROS (4.2)
OBJETIVO : Borrar las facturas con saldo cero o canceladas
y todos los movimientos asociados a ellas.

Proceso : BORRA FACTURAS DE ARCHIVOS MAESTROS

USA archivo maestro de facturas

REPITE

SI factura cancelada

USA archivo de movimientos de facturas

REPITE

SI concepto pertenece a la factura

Borra movimiento

FIN_SI

Hasta borrar todos los movimientos de la factura

FIN_SI

SI factura con saldo cero

USA archivo de movimientos de facturas

REPITE

SI concepto pertenece a la factura

Borra movimiento

FIN_SI

Hasta borrar todos los movimientos de la factura

FIN_SI

Hasta revisar todas las facturas

FIN DEL PROCESO

PROCESO : DEPURA EL ARCHIVO AUXILIAR DE FACTURAS (4.3)

OBJETIVO : Borrar las facturas que no se hayan localizado en archivos al capturar un movimiento y que tengan saldo cero ó se hayan cancelado y todos los movimientos asociados a ellas.

Proceso : BORRA FACTURAS DE AUXCYA

USA archivo de facturas no localizadas (AUXCYA)

REPITE

SI factura cancelada

USA archivo de movimientos de facturas

REPITE

SI concepto pertenece a la factura

Borra movimiento

FIN_SI

Hasta borrar todos los movimientos de la factura

FIN_SI

SI factura con saldo cero

USA archivo de movimientos de facturas

REPITE

SI concepto pertenece a la factura

Borra movimiento

FIN_SI

Hasta borrar todos los movimientos de la factura

FIN_SI

Hasta revisar todas las facturas

FIN DEL PROCESO

PROCESO : DEPURACION DE ARCHIVOS ESTADISTICOS (4.4)

OBJETIVO : Borrar de los archivos estadísticos todos los importes y volúmenes que se hayan acumulado en el periodo que se pretende borrar.

Proceso : BORRA ESTADISTICOS

Valida la fecha VS. el tipo de switch prendido

CASO de switch prendido :

mensual : inicia los campos mensuales de clientes

 inicia los campos mensuales de artículos

trimestral : inicia los campos trim. de clientes

 inicia los campos trim. de artículos

semestral : inicia los campos semest. de clientes

 inicia los campos semest. de artículos

anual : inicia los campos anuales de clientes

 inicia los campos anuales de artículos

FIN_CASO

FIN DEL PROCESO

PROCESO : REALIZA CONSULTAS EN PANTALLA (5.0)

OBJETIVO : Consultar en pantalla los estados de cuenta
y la antigüedad de saldos de los clientes que se
encuentran en catálogo.

Proceso : REALIZA CONSULTAS

REPITE

Despliega menú de opciones

Lee opción :

CASO de opción :

- 1) Obtiene estados de cuenta
- 2) Obtiene antigüedad de saldos
- * Regresa a menú anterior

FIN_CASO

Hasta opción = *

FIN DEL PROCESO

PROCESO . CONSULTA DE ESTADOS DE CUENTA (5.1)

OBJETIVO : Consultar en pantalla los estados de cuenta
de los clientes que se encuentran en catálogo.

Proceso : OBTIENE ESTADOS DE CUENTA

Despliega títulos de la consulta

Pide el código del cliente

Busca el cliente en el catálogo

SI encuentra el cliente

REPITE

Busca la factura del movimiento

SI encuentra la factura

SI el concepto es un cargo

Presenta cargo en pantalla

Suma el cargo al saldo acumulado del cliente

EN OTRO CASO

Presenta abono en pantalla

Resta el abono al saldo acumulado del cliente

FIN_SI

EN OTRO CASO

Despliega mensaje de error

FIN_SI

Presenta el saldo acumulado del cliente

Hasta recorrer todos los movimientos del cliente

EN OTRO CASO

Despliega mensaje de error

Suspende el proceso

FIN_SI

FIN DEL PROCESO

```

EBUCESO : CONSULTA LA ANTIGUEDAD DE SALDOS (S.2)
OBJETIVO : Consultar en pantalla la antigüedad de saldos
           de los clientes que se encuentran en catálogo.
Proceso : OBTIENE ANTIGUEDAD DE SALDOS
           Despliega títulos de la consulta
           Pide el código del cliente
           Busca el cliente en el catálogo
           SI encuentra el cliente
               REPITE
                   SI movimiento es una factura
                       Busca la factura en maestros
                       SI localiza la factura
                           Verifica el vencimiento de la factura
                           SI no esta vencida la factura
                               Suma el saldo de la factura al saldo
                               corriente del cliente
                           EN OTRO CASO
                               Calcula los días de vencimiento de la
                               factura
                               CASO de días vencidos sean :
                                   0-30 : Suma el saldo de la factura
                                       a saldos_vencidos_a_30_dias
                                   31-60 : Suma el saldo de la factura
                                       a saldos_vencidos_a_60_dias
                                   61--> : Suma el saldo de la factura
                                       a saldos_venc_a_mas_de_60_dias
                               FIN_CASO
                           FIN_SI
                       EN OTRO CASO
                           Despliega mensaje de error
                           Suspende el proceso

```

```

FIN_SI
EN OTRO CASO
    SI es un movimiento sin asignación a factura
        Verifica el vencimiento del movimiento
        SI no esta vencido el cargo ó abono
            Suma el saldo del movimiento al saldo
            corriente del cliente
        EN OTRO CASO
            Calcula los dias de vencimiento del
            movimiento
            CASO de dias vencidos sean :
                0-30 : Suma el saldo del movimiento
                a saldos_vencidos_a_30_dias
                31-60 : Suma el saldo del movimiento
                a saldos_vencidos_a_60_dias
                61--> : Suma el saldo del movimiento
                a saldos_venc_a_mas_de_60_dias
            FIN_CASO
        FIN_SI
    EN OTRO CASO
        Despliega mensaje de error
        Suspende el proceso
    FIN_SI
FIN_SI
    Presenta el saldo acumulado del cliente
    Hasta recorrer todos los movimientos del cliente
EN OTRO CASO
    Despliega mensaje de error
    Suspende el proceso
FIN_SI
FIN DEL PROCESO

```

EBDCESD : IMPRIME LOS REPORTES DEL SISTEMA (E.O)

OBJETIVO : Emitir los diferentes reportes que el sistema tiene contemplados y que pueden ser de ventas, estadísticos ode cuentas por cobrar.

Proceso : OBTIENE REPORTES

REPITE

Despliega menú de opciones

Lee opción :

CASO de opción :

- 1) Obtiene reportes de ventas
- 2) Obtiene reportes de cuentas por cobrar
- 3) Obtiene reportes estadísticos
- * Regresa a menú anterior

FIN_CASO

Hasta opción = *

FIN DEL PROCESO

PROCESO : IMPRIME LOS REPORTES DE VENTAS (6.1)
OBJETIVO : Emitir los diferentes reportes de ventas.

Proceso : OBTIENE REPORTES DE VENTAS

REPITE

Despliega menú de opciones

Lee opción :

CASO de opción :

- 1) Relación de ventas
- 2) Relación de pagos
- 3) Comisiones a pagar
- 4) Relación de notas de crédito
- *) Regresa a menú anterior

FIN_CASO

Hasta opción = *

FIN DEL PROCESO

PROCESO : IMPRIME LA RELACION DE VENTAS (6.1.1)

OBJETIVO : Imprimir el reporte de las ventas del mes.

Proceso : RELACION DE VENTAS

Pide fechas

Valida rango de fechas

SI existen ventas dentro del rango de fechas

REPITE

Almacena fecha del día en que hay ventas

REPITE

SI el movimiento es una factura

SI la factura no está cancelada

Imprime datos de la factura

Acumula totales por día

Acumula totales por reporte

EN OTRO CASO

Imprime mensaje de factura cancelada

FIN_SI

FIN_SI

Hasta reportar las ventas del día

Imprime total de ventas por día

Hasta la fecha final del rango

Imprime total de ventas del rango de fechas

FIN DEL PROCESO

PROCESO : IMPRIME LA RELACION DE PAGOS (6.1.2)

OBJETIVO : Imprimir el reporte de los pagos recibidos en un rango de fechas deseado.

Proceso : RELACION DE PAGOS

Pide fechas

Valida rango de fechas

SI existen pagos dentro del rango de fechas

REPITE

Almacena fecha del día en que hay pagos

REPITE

SI el movimiento es un pago

SI es un pago con asignación a factura

Busca la factura

SI existe la factura

Imprime datos del pago

Acumula totales por día

Acumula totales por reporte

EN OTRO CASO

Imprime mensaje de error

FIN_SI

EN OTRO CASO

Imprime datos del pago

Acumula totales por día

Acumula totales por reporte

FIN_SI

FIN_SI

Hasta reportar los pagos del día

Imprime total de pagos por día

Hasta la fecha final del rango

Imprime total de pagos del rango de fechas

FIN DEL PROCESO

PROCESO : IMPRIME LAS COMISIONES A PAGAR (6.1.3)

OBJETIVO : Imprimir el reporte de las comisiones que se deben pagar a los agentes de ventas por las ventas logradas en el rango de fechas dado.

Proceso : COMISIONES A PAGAR

Valida fechas

USA el catálogo de vendedores

REPITE

Valida rango de fechas

SI hubo ventas dentro del rango de fechas

USA el archivo maestro de facturas

REPITE

SI la factura pertenece al rango de fechas

Imprime datos de la factura y la comisión

Acumula total de facturas por vendedor

Acumula total de facturas por reporte

FIN_SI

Hasta reportar todas las facturas del agente

FIN_SI

USA el archivo de movimientos

REPITE

SI concepto es nota de crédito y pertenece al vendedor

SI la nota de crédito pertenece al rango de fechas

Imprime datos de la nota de crédito

Acumula total de NC por vendedor

Acumula total de NC por reporte

FIN_SI

FIN_SI

Imprime total de facturas por vendedor

Imprime total de notas de crédito por vendedor

Hasta recorrer los movimientos del vendedor

Hasta recorrer todo el catálogo de vendedores

Imprime total de facturas por reporte

Imprime total de notas de crédito por reporte

FIN DEL PROCESO

PROCESO : REPORTA LA NOTAS DE CREDITO (6.1.4)
OBJETIVO : Imprimir el reporte de las notas de crédito
que se les hicieron a los clientes.

Proceso : RELACION DE NOTAS DE CREDITO

Pide fechas

Valida rango de fechas

SI hubo notas de crédito dentro del rango de fechas

REPITE

SI la nota de crédito pertenece al rango de fechas

SI es nota de crédito cancelada

Imprime mensaje de cancelada

EN OTRO CASO

Imprime datos de la nota de crédito

Acumula total de notas de crédito por
reporte

FIN_SI

FIN_SI

Hasta reportar todas las notas de crédito

FIN_SI

FIN DEL PROCESO

PROCESO : IMPRIME LOS REPORTES DE CUENTAS POR COBRAR (6.2)

OBJETIVO : Emitir los diferentes reportes de cuentas por cobrar.

Proceso . OBTIENE REPORTES DE CUENTAS POR COBRAR

REPITE

Despliega menú de opciones

Lee opción :

CASO de opción :

- 1) Estados de cuenta
- 2) Antigüedad de saldos
- 3) Análisis de documentos
- * Regresa a menú anterior

FIN_CASO

Hasta opción = *

FIN DEL PROCESO

PROCESO : REPORTA LOS ESTADOS DE CUENTA (6.2.1)
OBJETIVO : Obtener un reporte de los estados de cuenta
de los clientes que se encuentran en catálogo.

Proceso : ESTADOS DE CUENTA

Pide el rango de clientes

REPITE

Busca el cliente en el catálogo

SI encuentra el cliente

REPITE

Busca la factura del movimiento

SI encuentra la factura

SI el concepto es un cargo

Imprime el cargo

Suma el cargo al total de cargos

Suma el cargo al saldo acumulado del cliente

EN OTRO CASO

Imprime el abono

Suma el abono al total de abonos

Resta el abono al saldo acumulado del cliente

FIN_SI

EN OTRO CASO

Imprime mensaje de error

FIN_SI

Hasta recorrer todos los movimientos del cliente

Imprime el total de abonos, cargos y saldo del
cliente

Inicia el total de abonos, cargos y saldo del


```
cliente
EN OTRO CASO
  Despliega mensaje de error
FIN_SI
Hasta recorrer todo el rango de clientes
FIN DEL PROCESO
```

```

PROCESO : REPORTA LA ANTIGUEDAD DE SALDOS (6.2.2)
OBJETIVO : Obtener un reporte con la antigüedad de saldos
           de los clientes que se encuentran en catálogo.
Proceso : OBTIENE ANTIGUEDAD DE SALDOS

Pide el rango de clientes
REPITE
  Busca el cliente en el catálogo
  SI encuentra el cliente
    REPITE
      SI movimiento es una factura
        Busca la factura en maestros
        SI localiza la factura
          Verifica el vencimiento de la factura
          SI no esta vencida la factura
            Suma el saldo de la factura al saldo
            corriente del cliente
          EN OTRO CASO
            Calcula los días de vencimiento de la
            factura
            CASO de días vencidos sean :
              0-30 : Suma el saldo de la factura
                    a saldos_vencidos_a_30_días
              31-60 : Suma el saldo de la factura
                    a saldos_vencidos_a_60_días
              61--> : Suma el saldo de la factura
                    a saldos_venc_a_mas_de_60_días
          FIN_CASO
        FIN_SI
      EN OTRO CASO
        Imprime mensaje de error
      FIN_SI

```

EN OTRO CASO

SI es un movimiento sin asignación a factura

Verifica el vencimiento del movimiento

SI no esta vencido el cargo abono

Suma el saldo del movimiento al saldo
corriente del cliente

EN OTRO CASO

Calcula los dias de vencimiento del
movimiento

CASO de dias vencidos sean :

0-30 : Suma el saldo del movimiento
a saldos_vencidos_a_30_dias

31-60 : Suma el saldo del movimiento
a saldos_vencidos_a_60_dias

61--> : Suma el saldo del movimiento
a saldos_venc_a_mas_de_60_dias

FIN_CASO

FIN_SI

FIN_SI

FIN_SI

Imprime los saldos_vencidos_a_30_dias

Imprime los saldos_vencidos_a_60_dias

Imprime los saldos_vencidos_a_mas_de_60_dias

Imprime el saldo acumulado del cliente

Hasta recorrer todos los movimientos del cliente

EN OTRO CASO

Despliega mensaje de error

FIN_SI

Hasta reportar todos los clientes del rango

FIN DEL PROCESO

PROCESO : REPORTA EL ANALISIS DE DOCUMENTOS (6.2.3)
OBJETIVO : Reporte que obtiene un estado de cuenta del
cliente pero haciendo un análisis por
factura.

Proceso : ANALISIS DE DOCUMENTOS

Pide el rango de clientes

REPITE

Busca el cliente en el catálogo

SI encuentra el cliente

REPITE

SI movimiento es una factura

Busca la factura en maestros

SI localiza la factura

SI la factura no está cancelada

REPITE

SI concepto es cargo

Imprime datos del cargo

Suma el cargo al saldo
acumulado de la factura.

EN OTRO CASO

Imprime datos del abono

Resta el abono al saldo
acumulado de la factura.

FIN_SI

Imprime saldo de la factura

HASTA reportar todos los movimientos

EN OTRO CASO

Imprime mensaje de error

```
FIN_SI
EN OTRO CASO
    Imprime mensaje de error
FIN_SI
FIN_SI
    Hasta reportar todos los movimientos del cliente
EN OTRO CASO
    Despliega mensaje de error
FIN_SI
    Hasta reportar todos los clientes del rango
FIN DEL PROCESO
```

PROCESO : IMPRIME LOS REPORTES ESTADISTICOS (6.3)

OBJETIVO : Emitir los diferentes reportes estadísticos.

Proceso : OBTIENE REPORTES ESTADISTICOS

REPITE

Despliega menú de opciones

Lee opción :

CASO de opción :

- 1) ABC de clientes
- 2) ABC de artículos
- 3) Clientes por vendedor
- *) Regresa a menú anterior

FIN_CASO

Hasta opción = *

FIN DEL PROCESO

PROCESO : REPORTA LAS COMPRAS DE LOS CLIENTES (6.3.1)

OBJETIVO : Emitir diferentes reportes de las compras que realizaron los clientes en los periodos mensual, trimestral, semestral y anual.

Proceso : ASC DE CLIENTES

Selecciona tipo del periodo del reporte

Pide fecha del reporte

Valida la fecha proporcionada

USA el archivo estadístico de clientes

SI hubo compras en el periodo

REPITE

Busca el cliente en el catálogo

SI encuentra al cliente

Imprime datos del cliente

Imprime datos de la compra

Suma la compra al total por reporte

EN OTRO CASO

Imprime mensaje de error

FIN_SI

Hasta reportar todas compras del periodo

EN OTRO CASO

Imprime mensaje al usuario

FIN_SI

FIN DEL PROCESO

PROCESO : REPORTA LAS VENTAS POR ARTICULO (6.3.2)

OBJETIVO : Emitir diferentes reportes de los artículos
que se vendieron en los periodos mensual,
trimestral, semestral y anual.

Proceso : ABC DE ARTICULOS

Selecciona tipo del periodo del reporte

Pide fecha del reporte

Valida la fecha proporcionada

USA el archivo estadístico de artículos

SI hubo ventas en el período

REPITE

Busca el artículo en el catálogo

SI encuentra el artículo

Imprime datos del producto

Imprime datos de la venta

Suma la venta al total por reporte

EN OTRO CASO

Imprime mensaje de error

FIN_SI

Hasta reportar todas ventas del periodo

EN OTRO CASO

Imprime mensaje al usuario

FIN_SI

FIN DEL PROCESO

PROCESO : REPORTA LAS VENTAS DE LOS AGENTES (6.3.5)

OBJETIVO : Reportar las ventas del mes agrupándolas por agente de ventas.

Proceso : CLIENTES POR VENDEDOR

Fine fecha del reporte

Valida la fecha proporcionada

USA el catálogo de vendedores

REPITE

USA el catálogo de clientes

REPITE

SI el cliente realizó compras

Almacena la compra en un archivo auxiliar

FIN_SI

Hasta recorrer todos los clientes del vendedor

USA el archivo auxiliar

Ordena las compras en orden ascendente

REPITE

Imprime los datos de la compra

Acumula la compra al total por agente

Acumula la compra al total por reporte

Hasta reportar todas las compras de los clientes

Imprime totales por agente

Hasta recorrer el catálogo de vendedores

Imprime totales del reporte

FIN DEL PROCESO

PROCESO : MENU DE MANTENIMIENTO A CATALOGOS (7.0)

OBJETIVO : Permitir realizar las altas, bajas y cambios en los catálogos de clientes y de artículos, validando diferentes aspectos.

Proceso : MANTENIMIENTO A CATALOGOS

REPITE

Despliega menú de opciones

Lee opción :

CASO de opción :

- 1) Realiza mantenimiento a clientes
- 2) Realiza mantenimiento a artículos
- *) Regresa a menú anterior

FIN_CASO

Hasta opción = *

FIN DEL PROCESO

PROCESO : MANTENIMIENTO AL CATALOGO DE CLIENTES (7.1)

OBJETIVO : Permitir realizar las altas, bajas y cambios en el catálogo de clientes validando diferentes aspectos.

Proceso : REALIZO MANTENIMIENTO A CLIENTES

Pide código del cliente

Busca el cliente en el catálogo

SI encuentra al cliente

REPITE

Despliega pantalla de captura

Pide datos del cliente

Valida datos del cliente

SI datos del cliente son correctos

Graba en archivo

EN OTRO CASO

Despliega mensaje al usuario

FIN_CASO

Hasta que los datos sean correctos

SI razón social esta en blanco (dar de baja)

Pide la clave para dar la baja

Si clave correcta

Valida la baja

SI la baja es posible

Borra los movimientos del cliente

Borra el registro del cliente

EN OTRO CASO

Despliega mensaje al usuario

```
FIN_SI
EN OTRO CASO
    Despliega mensaje al usuario
FIN_SI
FIN_SI
EN OTRO CASO
    Despliega mensaje de error
FIN_SI
FIN DEL PROCESO
```

PROCESO : MANTENIMIENTO AL CATALOGO DE ARTICULOS (7.2)

OBJETIVO : Permitir realizar las altas, bajas y cambios en el catálogo de artículos validando diferentes aspectos.

Proceso : REALIZA MANTENIMIENTO A ARTICULOS

Pide código del artículo

Busca el artículo en el catálogo

SI encuentra al producto

REPITE

Despliega pantalla de captura

Pide datos del artículo

Valida datos del artículo

SI datos del artículo son correctos

Graba en archivo

EN OTRO CASO

Despliega mensaje al usuario

FIN_CASO

Hasta que los datos sean correctos

SI la descripción esta en blanco (dar de baja)

Pide la clave para dar la baja

Si clave correcta

Valida la baja

SI la baja es posible

Borra el registro del artículo

EN OTRO CASO

Despliega mensaje al usuario

FIN_SI

EN OTRO CASO

Despliega mensaje al usuario

FIN_SI

FIN_SI

EN OTRO CASO

Despliega mensaje de error

FIN_SI

FIN DEL PROCESO

CAPITULO IV

DISEÑO DEL SISTEMA

A diferencia de otras áreas de la Ingeniería, el desarrollo de sistemas es una disciplina a la que se le ha dado poca importancia en lo que a métodos formales de desarrollo se refiere, frecuentemente los sistemas terminados dejan poco satisfechos a usuarios y diseñadores. Lo anterior provoca la generación de sistemas con problemas de administración (muy tardados), poco satisfactorios, poco confiables, inflexibles, difíciles de mantener e ineficientes, es decir, sistemas con muy poca calidad.

Con el Diseño Estructurado se pretende lograr la producción de sistemas que no incurran en los problemas antes mencionados.

El diseño es la fase en donde se ilustra el "como" resolver el problema. Esto se logra mediante la generación de lo que se conoce como "carta de estructura". Para poder obtener dicho documento, el Diseño estructurado se basa en dos elementos que se obtuvieron en la fase anterior, los cuales son el Diagrama de flujo de Datos y el Diccionario de Datos.

Se puede decir que un buen diseño se logra cuando por sistema el problema es subdividido en pequeños subsistemas y además todas sus partes se organizan de manera jerárquica. Tal es la filosofía del Diseño Estructurado.

La idea de dividir un problema en partes más pequeñas no es nueva y siempre ha demostrado su gran utilidad. En el Diseño Estructurado, así como en otros aspectos, se utiliza con la finalidad de poder controlar la complejidad del problema en cuestión.

La herramienta más utilizada para el control de complejidad en el diseño de sistemas es la Caja Negra. Las características de esta herramienta son que de antemano se conocen sus entradas y las salidas que producirá, se conoce la función que realiza y no es necesario saber cómo la realiza.

La principal ventaja del uso de la Caja Negra para controlar la complejidad del sistema, radica en que subdivide al sistema en Cajas Negras más pequeñas que se conectan de tal forma que cada una de ellas representa una parte bien definida del problema, son fáciles de entender, sus conexiones corresponden a flujos de datos del problema y son lo más simples posibles, de tal manera que pueden llegar a ser independientes entre sí.

De esta manera, la principal herramienta del Diseño Estructurado es el Diagrama de estructura, ya que este ilustra la modularización del sistema (cajas negras) mostrando además su jerarquía, organización y comunicación.

Las ventajas de emplear un Diagrama de Estructura para el diseño es que se cuenta con una herramienta de tipo gráfico, particionable, riguroso, pero a la vez flexible que además sirve como documentación del sistema y representa una gran ayuda para el mantenimiento y modificación del mismo.

IV.1) Diagrama de estructura

El diagrama de estructura es una representación gráfica del sistema que se utiliza como herramienta para el diseño, implementación, documentación, modificación y mantenimiento del sistema.

El Diagrama de estructura es un modelo independiente del tiempo de la ejecución de las relaciones jerárquicas de los módulos de un programa o sistema; es por esto que no se puede inferir de un Diagrama de Estructura, cual es el orden en que se ejecutan los módulos.

Los elementos que forman el Diagrama de Estructura son los módulos, las conexiones y las interfases.

Los Módulos son secuencias de instrucciones continuas de programa y se referencian mediante un identificador. Todo módulo se caracteriza por tener los siguientes atributos:

- Entradas (parámetros del módulo que lo llama)
- Salidas (respuesta al módulo que lo llamó)
- Función (transformación de entradas en salidas)
- Mecánica (como hace su función)
- Datos internos (datos que únicamente el mismo utiliza)

Nótese que no existe diferencia entre Módulo, programa y sistema.

Un Módulo se representa gráficamente como una caja rectangular con su nombre en el interior, siendo el nombre un enunciado breve que describe la función que realiza el módulo.

Las conexiones son las llamadas que se hacen a una subrutina y son las que permiten conectar un módulo con otro. Gráficamente se representan por una flecha y mediante el sentido de la misma es que se puede saber cual es el módulo que invoca al otro.

Los Datos o interfases se representan por flechas más pequeñas y mediante el sentido de estas es posible saber hacia donde se realiza la transferencia. Si la flecha tiene en el extremo anterior un círculo vacío significa que se transfieren datos. Si por el contrario, posee un círculo relleno, entonces se

trata de la transferencia de una bandera. Un rombo en el módulo padre, significa que la invocación de los módulos hijos es opcional. En la figura IV I se puede ver la representación gráfica de un Diagrama de Estructura.

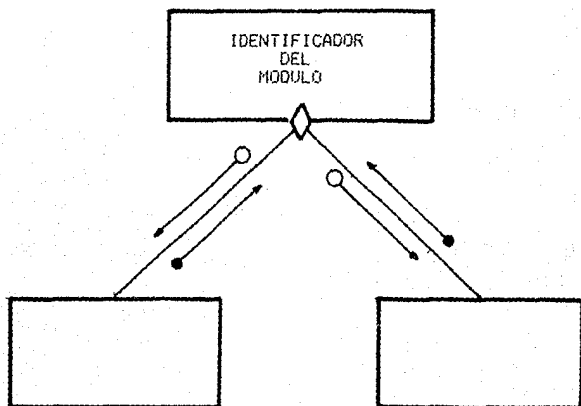


FIG. IV.1 REPRESENTACION GRAFICA DE UN DIAGRAMA DE ESTRUCTURA

Durante el desarrollo del diagrama de estructura se deben cuidar dos aspectos muy importantes que son los que le darán al sistema la característica de baja o alta dependencia entre sus subrutinas. Dichos aspectos son el acoplamiento y la cohesión.

ACOPLAMIENTO

Si un módulo para su funcionamiento no necesita en lo absoluto de otro con el que se encuentre conectado, entonces se puede decir que dichos módulos son totalmente independientes entre sí.

El acoplamiento es el grado de dependencia que existe entre un módulo con respecto a otro. De esta forma se pueden tener módulos altamente acoplados, es decir que están unidos por conexiones muy rígidas, y módulos escasamente acoplados que poseen interconexiones muy débiles.

El acoplamiento es el criterio más importante para juzgar las bondades de un diseño.

Existen 5 tipos de acoplamiento:

- Acoplamiento de datos
- Acoplamiento de estampado
- Acoplamiento de control
- Acoplamiento de área común
- Acoplamiento de contenido

Acoplamiento de datos

El acoplamiento de datos se tiene cuando solo los datos necesarios son comunicados entre módulos.

Este tipo de acoplamiento es el más deseable, y de hecho cualquier sistema puede construirse de tal manera que el único acoplamiento sea de datos.

Acoplamiento de estampado

Dos módulos presentan acoplamiento de estampado si hacen referencia a la misma estructura de datos (no global).

Se debe entender una estructura de datos como un compuesto de elementos.

El problema de este tipo de acoplamiento es que si se hace un cambio a la estructura de datos, este afectará a todos los módulos que están "estampados" con la estructura. Sin embargo, usando una buena y natural estructura de datos, este tipo de acoplamiento se acerca al acoplamiento de datos.

Acoplamiento de control

Dos módulos presentan acoplamiento de control si se comunican usando al menos un elemento de control, es decir que uno de ellos pasa al otro información que intente controlar la lógica interna del otro.

Este acoplamiento es indeseable porque uno o ambos módulos dejan de funcionar como una caja negra.

Acoplamiento de área común

Un grupo de módulos presentan acoplamiento de área común si comparten una misma área global de datos.

La problemática que presenta este tipo de acoplamiento es que es más difícil reutilizar los módulos que utilizan áreas comunes ya que usualmente estos módulos se ejecutan como subrutinas. Otro problema es que se hace más difícil el mantenimiento, sobre todo cuando se pasan diferentes tipos de datos a través del área común, y por último, otro inconveniente es que el uso de áreas globales de datos dificulta la legibilidad de los programas además de que es difícil saber que módulos usan qué datos.

Acoplamiento de contenido

Este es el peor de los casos y ocurre cuando:

- Un módulo altera instrucciones en otro módulo
- Un módulo referencia o cambia datos contenidos en otro módulo.
- Un módulo brinca a otro.
- Dos módulos comparten las mismas literales.

COHESION

Cohesión es una medida de la consistencia o fuerza de la asociación de los elementos dentro de un módulo, entendiéndose como elemento una instrucción, un grupo de instrucciones o una llamada a otro módulo. Es deseable tener módulos altamente cohesivos, es decir, módulos cuyos elementos que estén alta y genuinamente relacionados. Además se debe evitar que los elementos de un módulo se relacionen con los de otro para evitar el acoplamiento. Es obvio que cohesión y acoplamiento están íntimamente relacionados, pues a mayor cohesión de los módulos del sistema, existirá un menor acoplamiento.

Existen los siguientes tipos de cohesión :

- Funcional
- Secuencial
- Comunicacional
- De procedimiento
- Temporal
- Lógica
- Coincidental

Siendo mejor la cohesión funcional y peor la coincidental.

Cohesión funcional

Un módulo con cohesión funcional es aquel en el que todos sus elementos contribuyen a una y solo una tarea completa; cada elemento es parte integral y es esencial para la ejecución de la función del módulo.

Cohesión secuencial

Un módulo con cohesión secuencial es aquel en el que sus elementos están involucrados en tareas en las que algunos datos que salen de un elemento sirven de entrada a otro elemento.

Generalmente los módulos con cohesión secuencial son fuertes y tienen buen acoplamiento. No son los mejores porque contienen varias funciones que no forman una función completa.

Cohesión comunicacional

Un módulo con cohesión comunicacional es aquel en el que sus elementos contribuyen a diferentes actividades que hacen referencia a los mismos parámetros de entrada y/o salida, es decir que todas las tareas están en función de un parámetro.

Por ejemplo, si se tienen las siguientes tareas:

- 1 busca número de parte
- 2 lee precio de la parte
- 3 localiza proveedor de la parte

Como se puede observar, en todas las actividades anteriores existe una relación que está dada por el mismo dato de entrada que es "la parte", lo que ocasiona que exista una cohesión comunicacional.

Este tipo de cohesión todavía es fuerte ya que el agrupamiento de funciones dentro de un módulo tienen alguna relación con el problema.

Al dividir estos módulos en módulos separados, se simplifica el acoplamiento y se incrementa la cohesión.

Cohesión de procedimiento

Un módulo con cohesión de procedimiento es aquel en el que el control fluye de un elemento al siguiente, pero, los datos no necesariamente fluyen de la misma manera, es decir que todos los elementos están envueltos en diferentes tareas que probablemente no están relacionadas. Es típico que en un módulo que presenta este tipo de cohesión, los datos que llegan a él y los datos que regresa tienen alguna relación.

El mayor problema con estos módulos es que manipulan resultados parciales, variables internas, banderas, switches, etc. y manejan partes de varias funciones.

Cohesión Temporal

Los módulos con cohesión temporal son aquellos cuyos elementos están relacionados en tiempo, es decir tienden a estar compuestos por funciones parciales en donde la única interacción entre estas, es que todas producen su salida en un tiempo definido. Usualmente sus elementos realmente pertenecen a diferentes funciones.

Cohesión Lógica

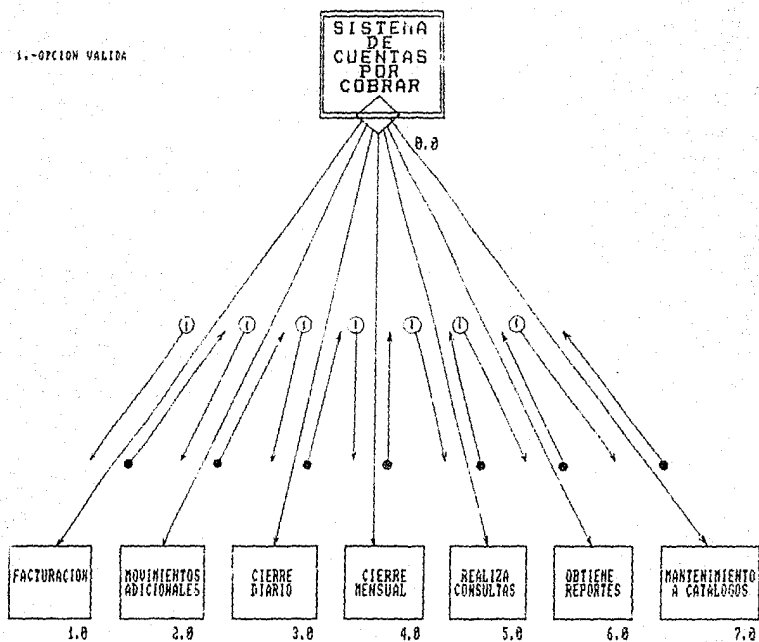
Los módulos con cohesión lógica son aquellos en que sus elementos aparentan estar relacionados a tareas de la misma categoría general, en donde las tareas a ejecutarse se seleccionan desde el exterior del módulo. Sin embargo al ejecutarse el módulo solo se ejecutarán algunas de las tareas.

Cohesión coincidental

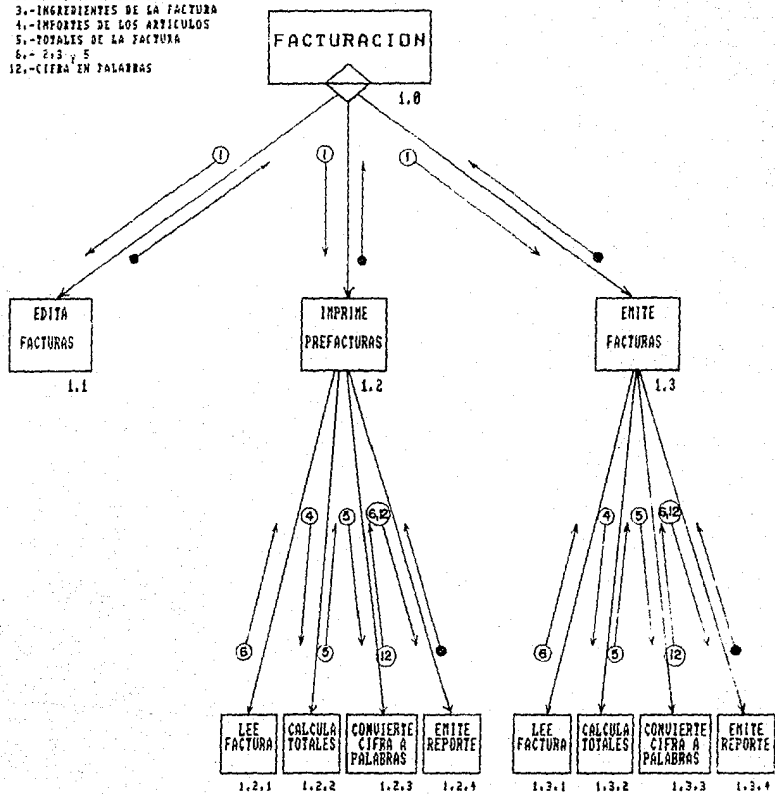
Este tipo de módulos se identifican por tener elementos sin relación significativa entre ellos. Usualmente ejecutan tareas diferentes y sin relación para diferentes módulos "padres".

En las páginas siguientes se presenta el diagrama de estructura que se obtuvo en el diseño del sistema de cuentas por cobrar:

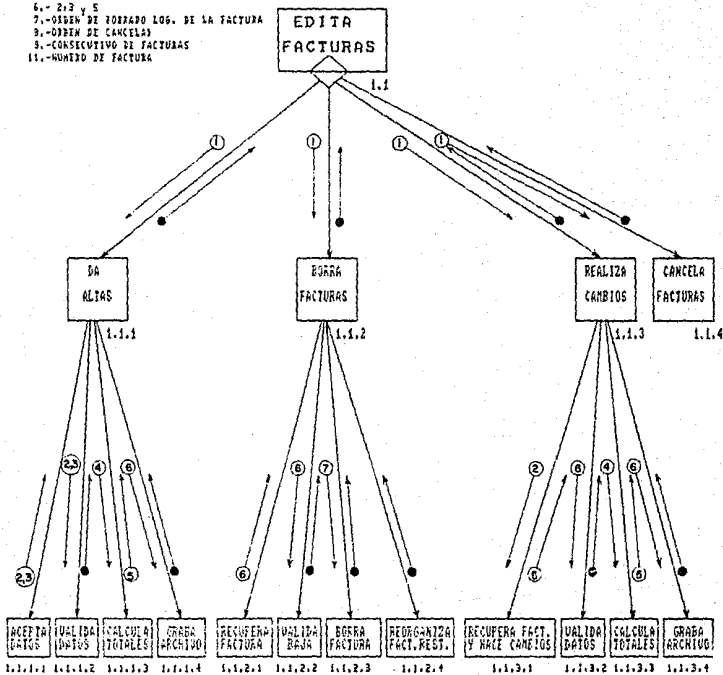
1.-OPCION VALIDA



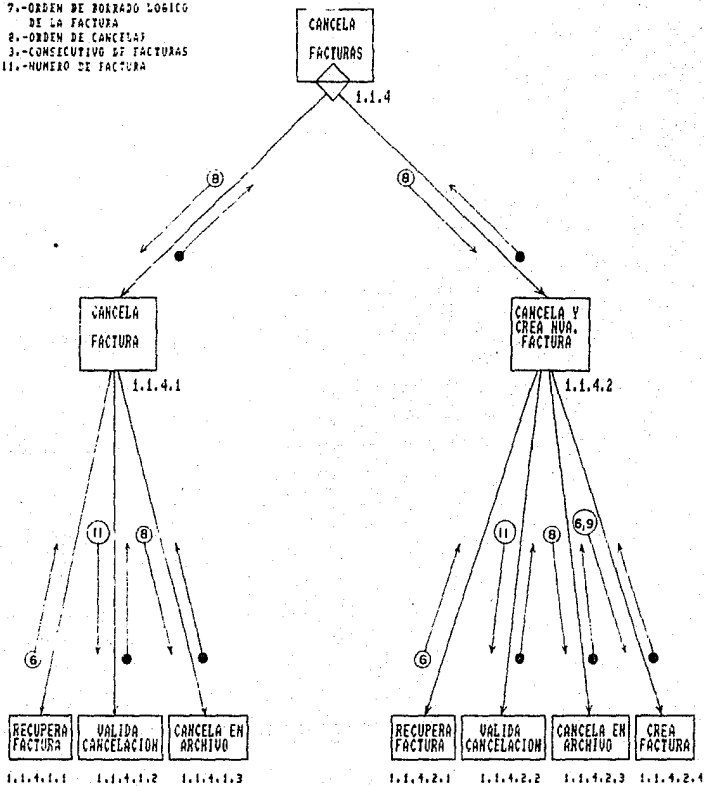
- 2.-DATOS GENERALES DE LA FACTURA
- 3.-INGREDIENTES DE LA FACTURA
- 4.-IMPORTE DE LOS ARTICULOS
- 5.-TOTALES DE LA FACTURA
- 6.- 2+3 y 5
- 12.-CIFRA EN PALABRAS



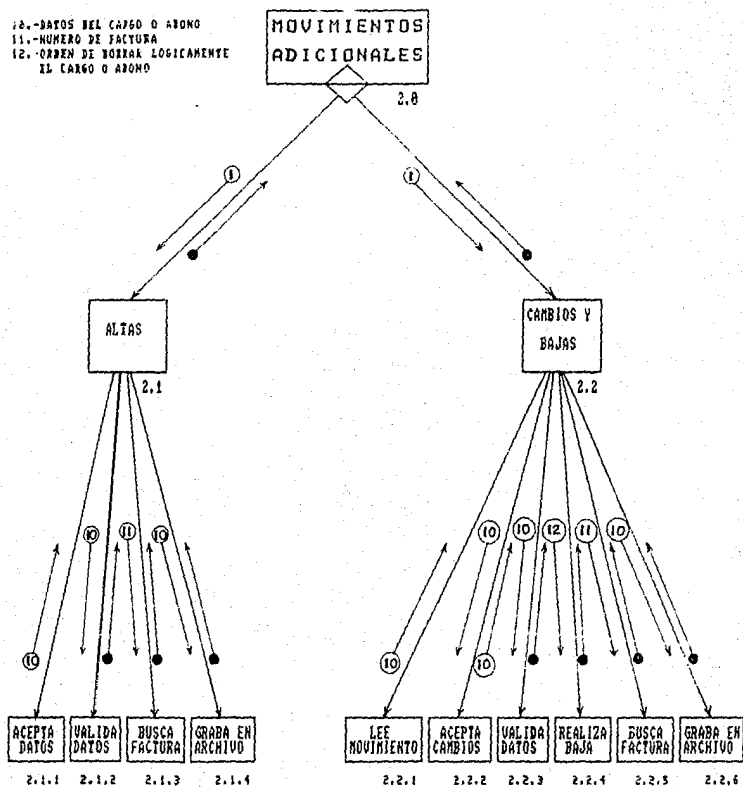
- 2.-DATOS GENERALES DE LA FACTURA
- 3.-INGREDIENTES DE LA FACTURA
- 4.-IMPORTE DE LOS ARTICULOS
- 5.-TOTALES DE LA FACTURA
- 6.- 2:3 y 5
- 7.-ORDEN DE ZORRADO LOG. DE LA FACTURA
- 8.-ORDEN DE CANCELAS
- 9.-CONSECUTIVO DE FACTURAS
- 11.-NUMERO DE FACTURA



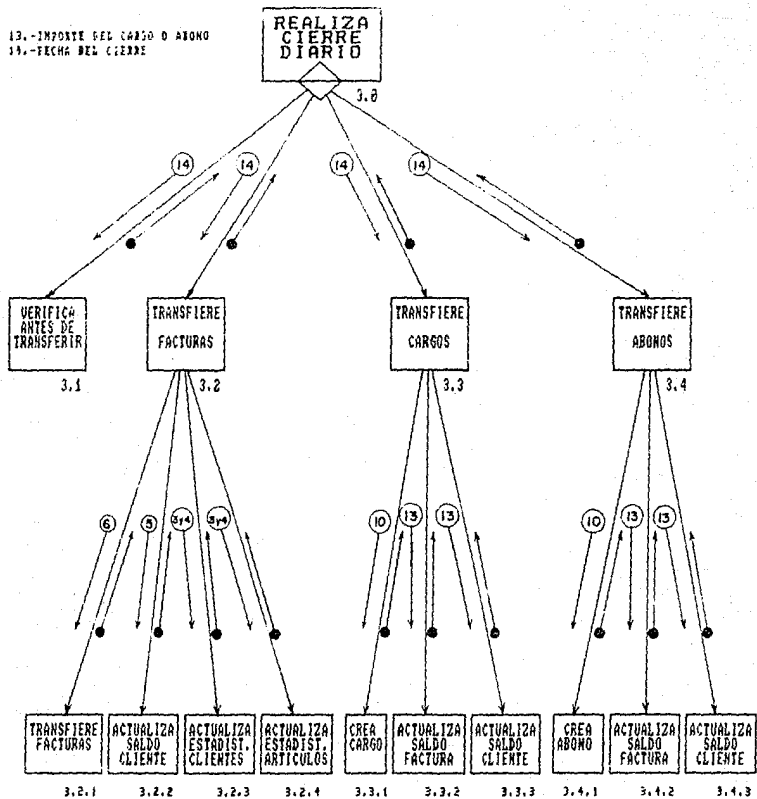
- 2.-DATOS GENERALES DE LA FACTURA
- 3.-INGREDIENTES DE LA FACTURA
- 4.-IMPORTES DE LOS ARTICULOS
- 5.-TOTALES DE LA FACTURA
- 6.-2,3 Y 5
- 7.-ORDEN DE BORRADO LOGICO DE LA FACTURA
- 8.-ORDEN DE CANCELAS
- 9.-CONSECUTIVO DE FACTURAS
- 11.-NUMERO DE FACTURA



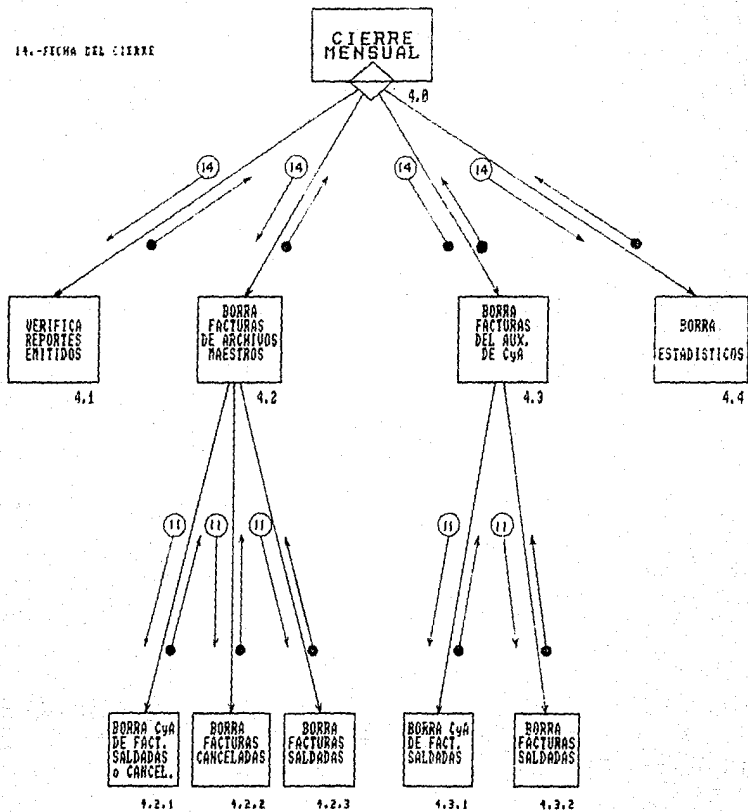
10.- DATOS DEL CARGO O ABONO
 11.- NUMERO DE FACTURA
 12.- ORDEN DE BOZAS LOGICAMENTE
 EL CARGO O ABONO



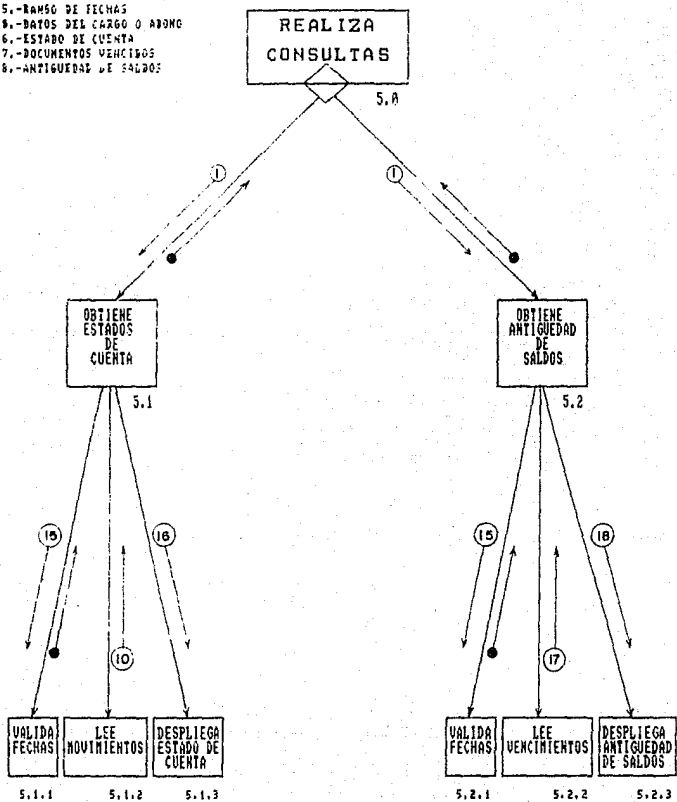
13.-IMPORTE DEL CARGO O ABOVO
14.-FECHA DEL CIERRE



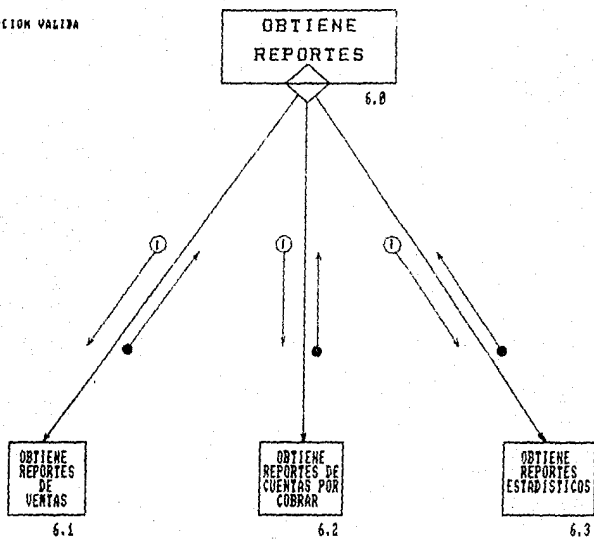
14.-FECHA DEL CIERRE



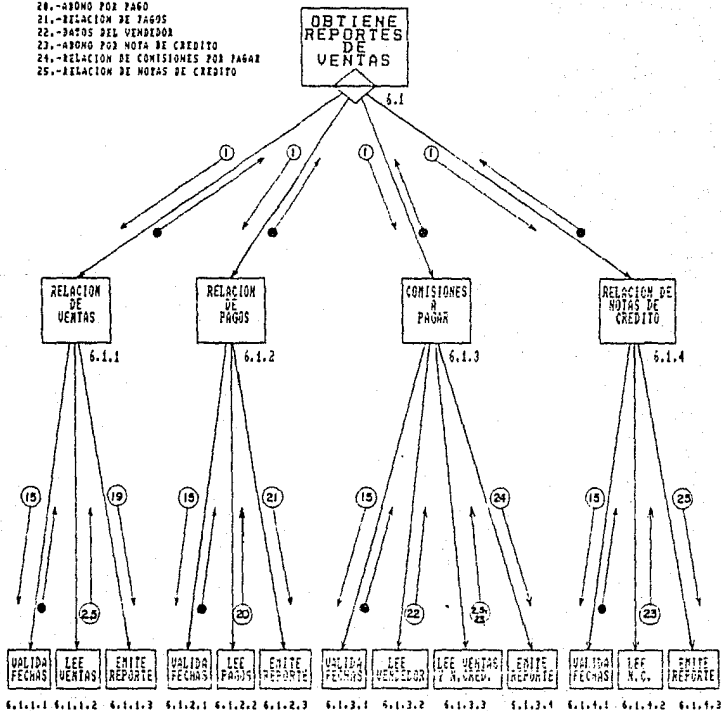
- 15.-RANGO DE FECHAS
- 16.-DATOS DEL CARGO O ABONO
- 17.-ESTADO DE CUENTA
- 18.-DOCUMENTOS VENCIDOS
- 19.-ANTIGUEDAD DE SALDOS



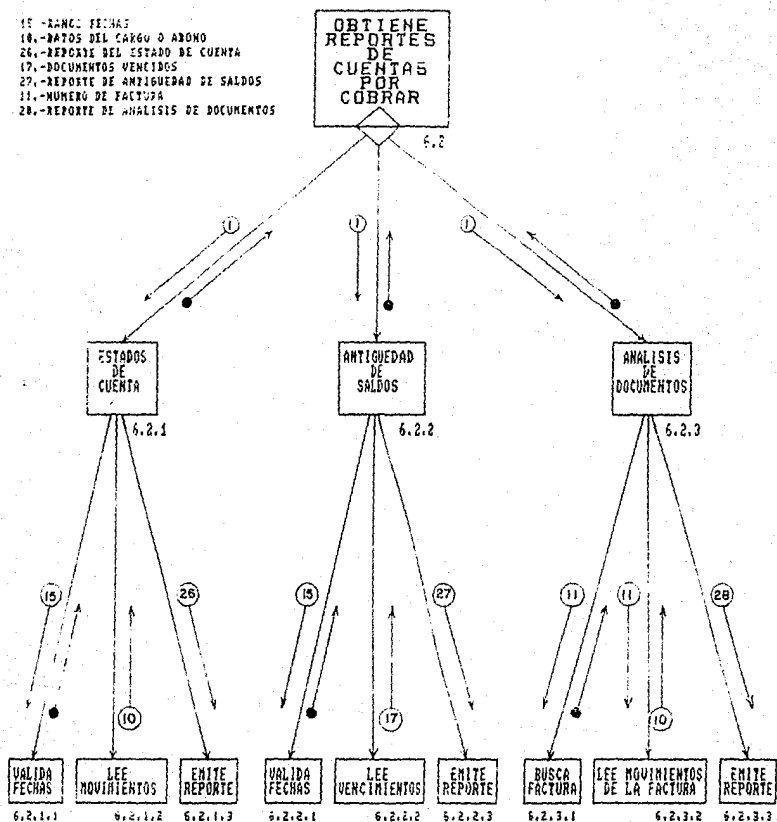
01.-OPCION VALIDA



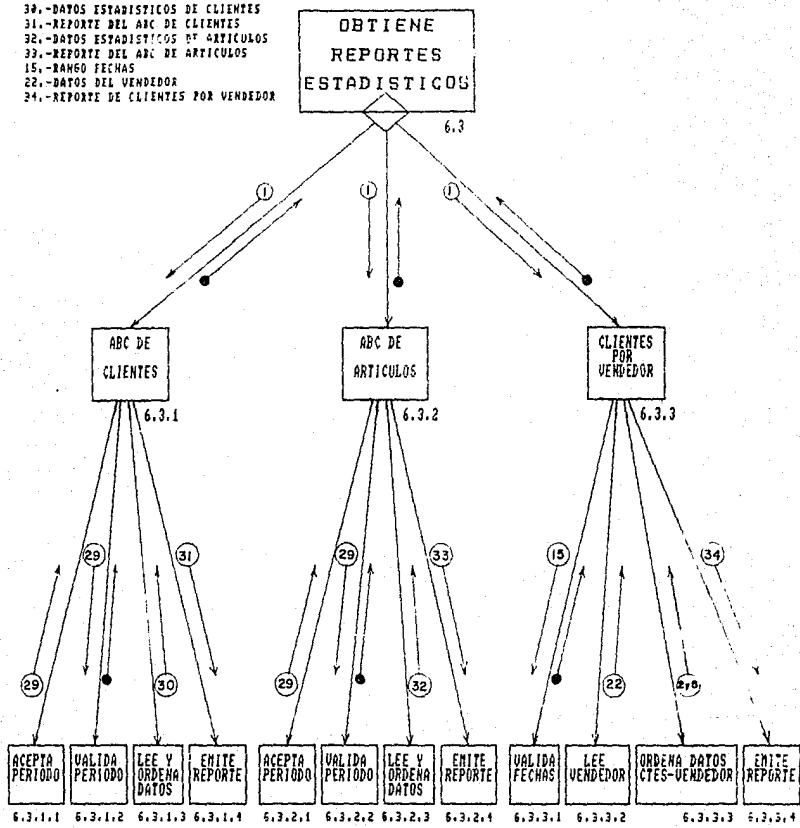
- 15.-RANGO DE FECHAS
- 2.-DATOS GENERALES DE LA FACTURA
- 5.-TOTALIS DE LA FACTURA
- 19.-RELACION DE VENTAS
- 20.-ABONO POR PAGO
- 21.-RELACION DE PAGOS
- 22.-DATOS DEL VENDEDOR
- 23.-ABONO POR NOTA DE CREDITO
- 24.-RELACION DE COMISIONES POR PAGAR
- 25.-RELACION DE MORAS DE CREDITO



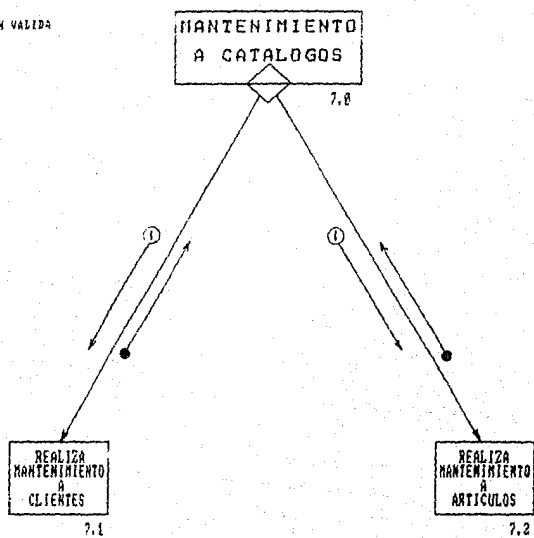
- 15.-RANGO FECHAS
- 16.-DATOS DEL CARGO O ABOGO
- 26.-REPORTE DEL ESTADO DE CUENTA
- 17.-DOCUMENTOS VENCIDOS
- 27.-REPORTE DE ANTIGUEDAD DE SALDOS
- 11.-NUMERO DE FACTURA
- 28.-REPORTE DE ANALISIS DE DOCUMENTOS



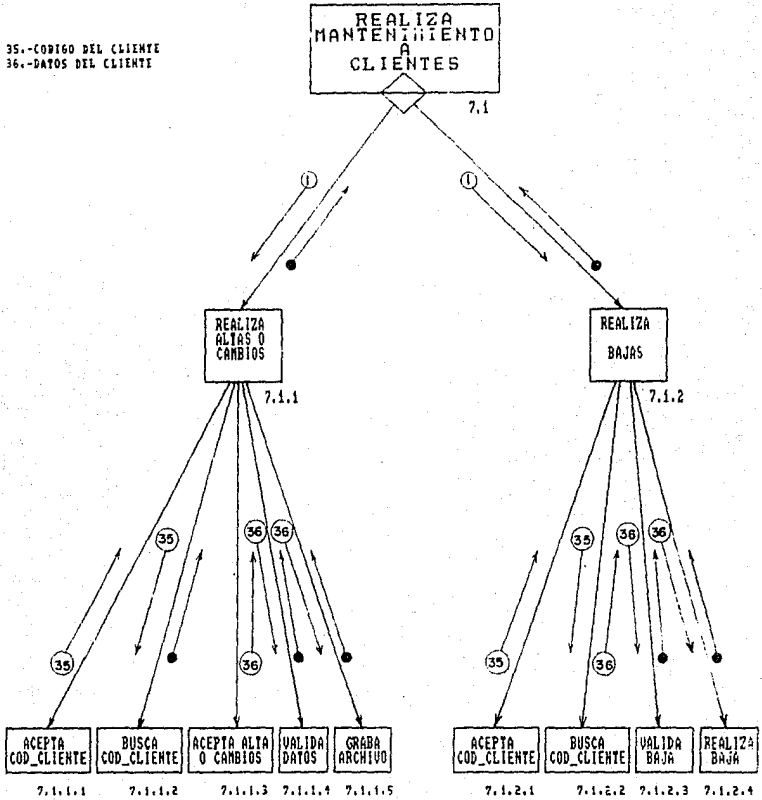
- 29.-PERIODO DEL REPORTE
- 30.-DATOS ESTADISTICOS DE CLIENTES
- 31.-REPORTE DEL ABC DE CLIENTES
- 32.-DATOS ESTADISTICOS DE ARTICULOS
- 33.-REPORTE DEL ABC DE ARTICULOS
- 15.-RANGO FECHAS
- 22.-DATOS DEL VENDEDOR
- 24.-REPORTE DE CLIENTES POR VENDEDOR



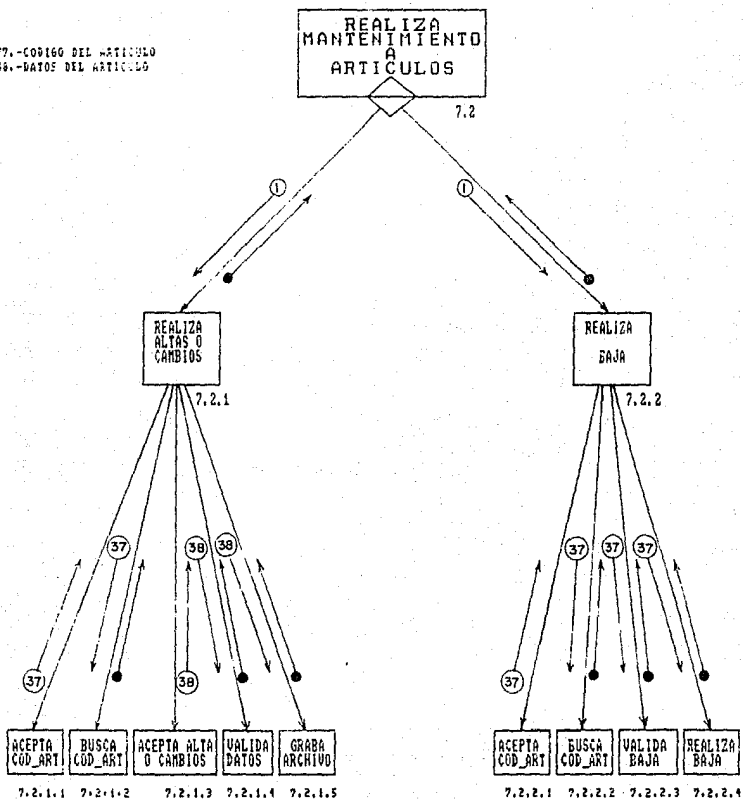
01.-OPCION VALZDA



35.-CODIGO DEL CLIENTE
36.-DATOS DEL CLIENTE



37.-CODIGO DEL ARTICULO
38.-DATOS DEL ARTICULO



IV.2) Aspectos de los lenguajes de programación

Los lenguajes de programación son mecanismos notacionales que se usan para instrumentar sistemas de cómputo. Las características particulares del lenguaje de programación seleccionado para el desarrollo del sistema ejercen una fuerte influencia sobre la estructura arquitectónica y los detalles algorítmicos del mismo. Por ejemplo un sistema programado en Lisp se diseñará e instrumentará naturalmente utilizando estructuras de datos de listas y funciones recursivas, mientras que uno desarrollado en Fortran utilizará arreglos, iteraciones, bloques de datos COMMON y subrutinas.

Los lenguajes de programación modernos proporcionan una variedad de características para apoyar el desarrollo y mantenimiento de los sistemas. Estas características incluyen la verificación de los tipos de datos, compilación por separado, tipos de datos definidos por el usuario, encapsulado de datos, abstracción de datos, reglas de alcance genéricas, manejo de excepciones y mecanismos de concurrencia.

Verificación de los tipos de datos

Un tipo de datos especifica un conjunto de objetos de datos y un conjunto de operaciones permitidas sobre los objetos de ese tipo. Así, los objetos del tipo "entero" comprenden un rango de valores enteros dependientes de la instrumentación y un conjunto de operadores relacionales y aritméticos sobre literales y variables del tipo entero.

Los propósitos de especificar tipos de datos son permitir la clasificación de los objetos de acuerdo a un uso deseado; permitir al traductor del lenguaje seleccionar las representaciones de memoria para los objetos de tipos diferentes y en el caso de los lenguajes con verificación "fuerte" de los tipos de datos detectar y prevenir operaciones entre objetos de tipos distintos.

La verificación de tipos de datos se refiere a las restricciones y limitaciones impuestas a los modos en que los datos elementales pueden ser manejados por el programa. Cada lenguaje impone restricciones diferentes ya que estos pueden ir desde los que no usan tipos, hasta los que realizan una verificación muy fuerte.

Compilación separada

Algunos lenguajes de programación poseen un poderoso mecanismo que es el de la compilación separada, es decir que poseen la capacidad de desarrollar módulos y subsistemas que se van almacenando en bibliotecas, posteriormente el cargador accederá automáticamente esas unidades y las ligará dentro del código objeto del sistema. Los módulos precompilados pueden utilizarse como componentes funcionales de un sistema y la

compilación por separado permite que diferentes programadores desarrollen o modifiquen simultáneamente diferentes módulos de un sistema.

Definición de tipos por el usuario

El conjunto de tipos de datos proporcionado por un lenguaje de programación determina grandemente las aplicaciones para las que está hecho el lenguaje, p. ej. FORTRAN IV provee tipos entero real, lógico, complejo, doble precisión, hollerit y arreglo, mientras que LISP puro provee únicamente listas de átomos (clases y números).

Los lenguajes de programación modernos (PASCAL p.ej.) proporcionan una variedad de tipos de datos predefinidos incluyendo enteros, punto flotante, booleano, carácter, cadena de caracteres, apuntador, registro, arreglo y archivo. No todos los lenguajes proveen de todos los tipos. En lugar de eso, se pueden predefinir unos cuantos tipos y proporcionar los mecanismos para que el usuario pueda utilizar otros tipos en términos de los ya existentes. Por ejemplo las cadenas de caracteres pueden ser definidas por el usuario como arreglos de longitud variable de caracteres.

Hay dos razones fundamentales para darle al usuario la posibilidad de definir sus propios tipos de datos. La primera es permitir la especificación de tipos que se necesiten constantemente en términos de tipos predefinidos y segundo, permitir el mapeo de conceptos del dominio del problema al lenguaje de instrumentación.

En lenguajes que proporcionan tipos de datos definidos por el usuario, los objetos que tienen características diferentes en el dominio del problema y con representaciones idénticas pueden declararse para que tengan distintos tipos. Los tipos de datos definidos por el usuario pueden así usarse para segmentar y modelar el dominio del problema. Esto puede mejorar enormemente la claridad y la seguridad de un programa de computadora. El mejoramiento en la claridad y seguridad facilita las tareas de depuración, prueba, documentación y modificación de un programa. El uso de los tipos de datos definidos por el usuario es un fuerte indicador de la buena calidad de un programa.

Abstracción de datos

El concepto de abstracción de datos incluye al "encapsulado" de datos y a los tipos de datos abstractos. Ambos mecanismos definen objetos de datos compuestos en términos de las operaciones que se pueden efectuar sobre ellos, y los mecanismos suprimen los detalles de la representación y manipulación de los datos. De esta manera una pila se puede caracterizar por operaciones tales como PUSH, POP, EMPTY_TEST, FULL_TEST. Similarmente una cola se puede caracterizar por operaciones tales como APPEND, REMOVE, SEARCH e INSERT. Así, los programas que hagan uso de una pila, cola o lista no necesitan preocuparse de

como están instrumentados la lista, la pila o la cola y no necesitan saber de los detalles de las funciones de su manipulación.

El "encapsulado de datos" difiere de los "tipos de datos abstractos" en que proporciona solamente un ejemplar de una entidad, mientras que un tipo de datos abstractos es una plantilla a partir de la cual se pueden crear múltiples ejemplares. P.ej. el tipo de punto flotante es una plantilla para la creación de múltiples encapsulados de punto flotante.

La abstracción de datos proporciona un poderoso mecanismo para escribir programas bien estructurados y fáciles de modificar. Los detalles internos de la representación y manipulación de datos puede modificarse a voluntad, suponiendo que las interfaces de los procedimientos de manipulación permanezcan iguales, los otros componentes del programa no serán afectados por la modificación, excepto tal vez por las modificaciones en las características del funcionamiento y los límites de capacidad.

Reglas de alcance

Las reglas de alcance de un lenguaje de programación dictan la manera en la cual el programador puede definir y usar los identificadores.

Las reglas de alcance utilizadas en diversos lenguajes de programación incluyen el alcance global, alcance anidado y alcance restringido. El alcance global se da en lenguajes como BASIC y COBOL. En ellos todos los identificadores son conocidos en todas las regiones de un programa. En FORTRAN los identificadores son conocidos en toda una unidad de programa, pero no son conocidos fuera de la unidad si no son declarados en una proposición COMMON o como parámetros reales en una invocación a un subprograma.

En la mayoría de los lenguajes de programación modernos se utilizan variaciones del alcance anidado. En un lenguaje de programación con alcance anidado, las unidades de programa se pueden anidar dentro de otras unidades. Los nombres declarados en niveles exteriores se encuentran disponibles para su uso dentro de las unidades anidadas, a menos que esos nombres sean declarados en una unidad anidada. El alcance anidado promueve la modularidad proporcionando agrupamientos naturales de entidades relacionadas y permitiendo la distinción entre identificadores globales y locales.

Manejo de excepciones

Una excepción es un acontecimiento que suspende la ejecución normal de un programa. Los acontecimientos de excepción incluyen, por ejemplo, valores fuera de rango, violación de límites de capacidad, aplicación de operadores a valores de datos ilegales e

intento de procesar datos que no están disponibles, se ejecuta una condición de excepción cuando ocurre un acontecimiento de excepción. Un manejador de excepciones constituye las acciones que se ejecutan en respuesta a una excepción levantada. El control se transfiere a un manejador de excepciones cuando se levanta la condición de excepción correspondiente. En la mayoría de los lenguajes de alto nivel, los manejadores de excepciones son parte del sistema del apoyo al tiempo de ejecución y los usuarios no tienen mecanismos para controlar el manejo de algunas excepciones, razón por la cual se recomienda utilizar subrutinas escritas en lenguaje ensamblador.

IV.3) Diseño de la base de datos

Sin duda, una de las actividades más importantes en el campo de la informática es el desarrollo de las bases de datos. Todavía en la década de los 70's no se conocía la gran utilidad que dicha herramienta podía ofrecer.

Las bases de datos aparecieron como una respuesta a las limitaciones que se tenía con el manejo convencional de archivos, ya que un archivo de datos podía tener más de un uso, pero sólo podía satisfacer un "enfoque" de datos almacenados, por lo que los diferentes enfoques de un archivo sólo se podían satisfacer después de hacer una reclasificación de los datos. En 1971, el grupo de tareas de bases de datos CODASYL, en un intento por mejorar la independencia de los datos y la estructura del proceso de diseño de las bases de datos, propuso una arquitectura de base de datos consistente de un doble esquema (DSG71), la cual fué seguida por la del triple esquema propuesta por el grupo de estudio ANSI/X3/SPARC en 1975.

Dado que el esquema propuesto por la ANSI/X3/SPARC es el más utilizado, hablaremos un poco más de este. Dicha arquitectura perseguía la independencia total de los datos y para ello se debían considerar tres subesquemas: el esquema externo, el cual representa los puntos de vista de los datos por parte de los usuarios; el esquema conceptual, el cual es una vista total e imparcial de la empresa total, misma que es independiente de los puntos de vista de los diferentes usuarios así como de la estructura del almacenamiento físico; y por último, el esquema interno, el cual es una vista más definida de la estructura del almacenamiento físico.

La parte innovadora de esta propuesta con respecto a la de CODASYL es precisamente el esquema conceptual, el cual aísla el almacenamiento físico y los mecanismos de acceso, de las vistas de los usuarios, con lo que ya proporciona una ligera independencia de los datos. Se puede definir al modelo conceptual como un modelo inherente de las entidades, basado en los campos de datos que las representan así como en las relaciones que las interconectan.

Conceptos básicos de bases de datos

EMPRESA. - Una empresa es cualquier clase de organización, como un banco, una universidad, una planta industrial o un hospital

ENTIDAD. - Una entidad es una persona, un lugar, una cosa, un evento o un concepto acerca del cual se registra información.

ATRIBUTOS. - Toda entidad tiene algunos atributos básicos que la caracterizan. Al atributo frecuentemente se le llama elemento de datos, campo de datos o dato elemental.

VALOR DE LOS DATOS. - Es la información o los datos mismos contenidos en cada campo de datos. Al conjunto de valores tomados por los campos de datos de una entidad se le llama "ocurrencia de la entidad".

CAMPOS LLAVE. - son aquellos campos de datos a partir de los cuales se pueden inferir otros campos.

REGISTRO DE DATOS. - es una colección de valores tomados por campos de datos relacionados. Los registros de datos se almacenan en algún medio como puede ser un hoja de papel, la memoria magnética de una computadora, una cinta, un disco, etc.

ARCHIVOS DE DATOS. - Los archivos de datos son una colección de registros de datos organizados lógicamente. Al archivo que se conforma de registros similares se le conoce como homogéneo, aunque también puede haber archivos con registros de diferentes tipos a los que se les llama heterogéneos.

Para definir la estructura fundamental de los datos, se utiliza un modelo de los mismos que debe expresar la relación entre las entidades. Actualmente la mayoría de las implantaciones de bases de datos usan modelos de datos jerárquicos, reticulares o relacionales.

Una de las principales tareas del administrador de la base de datos es el desarrollar el modelo conceptual, también llamado modelo de la empresa o de la organización. Un modelo conceptual debe expresar las entidades y las relaciones que lo integran y se considera como la herramienta ideal para representar la organización conceptual de los datos. También es una herramienta de comunicación entre los diversos usuarios de los datos, y como tal se desarrolla sin tomar en cuenta la representación física de los mismos. En resumen, se puede decir que el modelo conceptual se utiliza para organizar, visualizar, planear y comunicar ideas, y debe ser independiente del sistema de manejo de base de datos.

Una vez que el modelo conceptual ha quedado completamente definido, se procederá a transformarlo a un modelo lógico y este a su vez se transcribirá a un modelo físico. El modelo lógico puede ser de tipo relacional, jerárquico o reticular.

El DBMS (sistema de manejo de la base de datos) no es un factor para el diseño del modelo conceptual, pero el diseño del modelo lógico sí depende del DBMS que se tenga instalado ó que se vaya a utilizar, por lo que se deberá tener cuidado en la elección del modelo lógico.

El sistema de cuentas por cobrar se desarrollo utilizando el modelo relacional, ya que dBase es un DBMS de este tipo.

Modelo relacional

Un modelo relacional de datos se construye en base a tablas bidimensionales a las que se les llama tablas de relación (ver la figura 3.1). Cada columna de la tabla de relación es un atributo y a los renglones se les llama ocurrencias (figura IV.3.2). Un atributo o conjunto de atributos se llaman llave cuando sus valores identifican de manera única las ocurrencias de la tabla. También es posible que una tabla de relación tenga más de una llave, en cuyo caso se debe designar a una de ellas como llave primaria.

TABLA DE RELACION			
ATRIBUTO 1	ATRIBUTO 2	ATRIBUTO 3	...

Fig. IV.3.1 REPRESENTACION DE LAS TABLAS DE RELACION

Existen algunas propiedades de las tablas de relación, que son:

- No debe haber ocurrencias duplicadas.
- El orden de las ocurrencias no es significativo.
- El orden de las columnas no es significativo.
- Todos los valores de los atributos son indivisibles.

Una base de datos está compuesta por varias tablas y como es lógico suponer, estas deben estar interrelacionadas de alguna forma. Para poder establecer esta conexión entre las tablas, algunas de éstas deben tener atributos comunes.

Se pueden mencionar las siguientes ventajas del modelo relacional:

Simplicidad.- Un modelo relacional es lo que el usuario vé y no necesariamente lo que se implantará físicamente.

Consultas no planeadas.- Debido a que no hay una dependencia de posición entre las relaciones, las consultas no tienen que reflejar ninguna estructura preferida y por lo tanto pueden hacerse sin el procedimiento estandar de consultas.

Independencia de los datos.- Esto debe ser uno de los principales objetivos de cualquier sistema de manejo de base de datos. Este modelo elimina los detalles relativos a la estructura del almacenamiento y la estrategia del acceso de la interface con el usuario. El modelo relacional proporciona un grado de independencia de los datos relativamente más alta que los modelos jerárquico y reticular. Sin embargo, para poder usar esta propiedad del modelo relacional, el diseño de las relaciones debe ser completo y exacto.

Fundamentos teóricos.- El modelo relacional está fundamentado en la bien desarrollada teoría matemática de las relaciones. El riguroso método del diseño de una base de datos (normalización) le da a este modelo un fundamento sólido. Esta clase de fundamentación no existe para los otros dos modelos.

TABLA DE PACIENTES			
NUMERO DE PACIENTE	NOMBRE	DIRECCION	TELEFONO

Fig. IV.3.2 Representación de datos mediante el modelo relacional.

Modelo jerárquico

Un modelo jerárquico es un modelo que organiza a los datos en una estructura jerárquica de árbol (ver fig. IV.3.3).

La estructura jerárquica de árbol se construye con nodos y ramas. Un nodo es una colección de atributos de datos que describen a la entidad en ese nodo. El nodo más alto de una estructura jerárquica de árbol se conoce como RAIZ. Los nodos dependientes se encuentran en niveles más bajos en la estructura.

Cada vez que aparece el nodo raíz, da principio un registro lógico de base de datos, esto es, una base de datos jerárquica está compuesta de varios árboles. En este tipo de estructura no es posible la existencia de nodos hijos que compartan diferentes nodos padres.

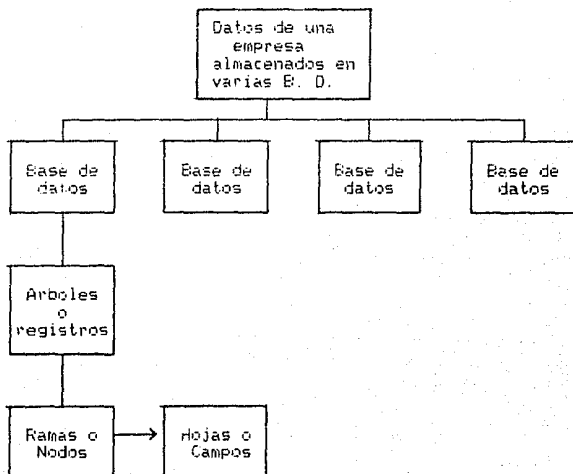


Fig. IV.3.3 Esquematzación del modelo jerárquico

Modelo reticular

En el modelo reticular las entidades están interconectadas en forma de red. Este modelo se caracteriza porque todas sus relaciones son de uno a varios (ver figura IV 3.4).

En un modelo que representa una relación de uno a varios, un tipo de registro padre tiene de cero a "n" ocurrencias de un tipo de registro hijo y este a su vez puede ser padre de cero a "n" ocurrencias en una "ocurrencia de conjunto", entendiéndose por esto último a aquella relación que existe entre una ocurrencia de registro padre con una de registro hijo.

Se puede decir que el modelo jerárquico en un caso especial del modelo reticular, ya que este también se compone de una estructura arborea.

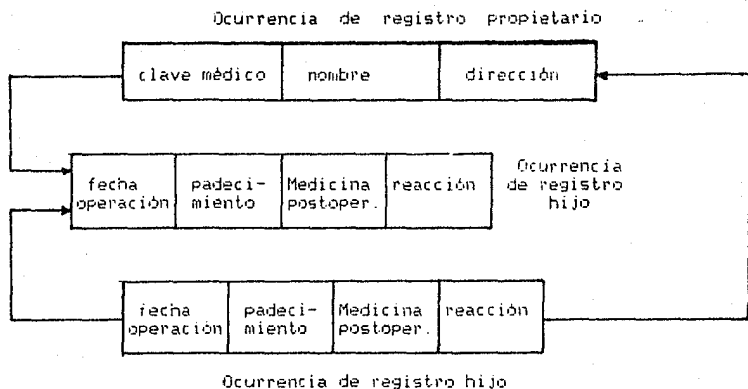


Fig. IV.3.4 Representación de datos mediante el modelo reticular

Normalización

La normalización es una técnica que se aplica en el desarrollo del modelo conceptual con la finalidad de obtener un modelo de datos óptimo que nos permita a partir de él, desarrollar un modelo lógico que pueda adecuarse a un sistema de manejo de base de datos relacional, jerárquico o reticular.

La normalización es el proceso de agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones. La teoría de la normalización está basada en la observación de que un cierto conjunto de relaciones tiene mejores propiedades en un medio donde existe inserción, actualización y supresión, que las que tendrían otros conjuntos de relaciones conteniendo los mismos datos.

La razón de usar el procedimiento de normalización es asegurar que el modelo conceptual de la base de datos funcionará.

Esto no significa que una estructura no normalizada no funcionará, pero sí que puede causar algunos problemas cuando los usuarios tratan de modificar la base de datos.

El primer paso de la normalización (1a.FN) consiste en transformar los campos de los registros de datos no normalizados, en tablas de dos dimensiones. Lo que se busca en este paso es la eliminación de ocurrencias repetidas de los campos de datos. El segundo paso de la normalización (2a.FN) persigue establecer los posibles campos llave y relacionarlas con los campos restantes.

En la 1a.FN el registro completo depende de todos los campos llave. En la 2a.FN se hace un intento por identificar los campos de datos que están relacionados con alguna parte de la llave completa. Si los campos de datos sólo dependen de una parte de la llave, esta y los campos conectados a la llave parcial son susceptibles de separarse en registros independientes. La separación de la primera tabla normalizada, en una serie de tablas en las que cada campo sólo depende de la llave completa da como resultado la segunda forma normalizada. La tercera forma normalizada consiste en separar los campos de las segundas relaciones normales que, aunque dependan de una llave única, deben tener una existencia independiente en la base de datos.

Esto se hace de forma tal que la información sobre estos campos pueda introducirse separadamente a partir de las relaciones en las que se encuentra implicada. A continuación se da un ejemplo de la normalización de una base de datos:

1a. Forma Normal

"Remover campos repetidos de un archivo de bases de datos"

PARTES	VENDEDORES	PARTES
Cod_vend	Cod_vend <-	Cod_parte
Nomb_vend	Nomb_vend ;	Nomb_parte
Cod_parte	Dirección ;	Fecha_orden
Nomb_parte	teléfono ->	Cod_vend

Estructura impropia

Estructura apropiada

2a. Forma Normal

"Todos los campos dependientes de un registro deben depender de la expresión llave completa, y no únicamente de una parte de ella"

ORD_CTES	CLIENTES	ORDENES
Cod_cte	(llave) Cod_cte <---	Cod_cte (llave)
No_ord_pte	Nomb_cte	Cant_pedido
Nomb_cte	Dirección1	Cod_parte
Dirección1	Dirección2	Fecha_venta
Dirección2	Teléfono	
Teléfono		
Fecha_venta		

Estructura impropia

Estructura apropiada

Si suponemos que la llave en la estructura impropia es Cod_cte + No_parte entonces existe una "dependencia parcial" de algunos campos respecto a la llave, ya que los mismos datos del cliente como son dirección, teléfono, etc. pueden existir en diferentes ordenes de compra, lo que ocasiona la existencia de redundancia.

3a. Forma Normal

"Todos los campos dependientes, deben depender únicamente de la expresión llave , y nunca de campos no llave"

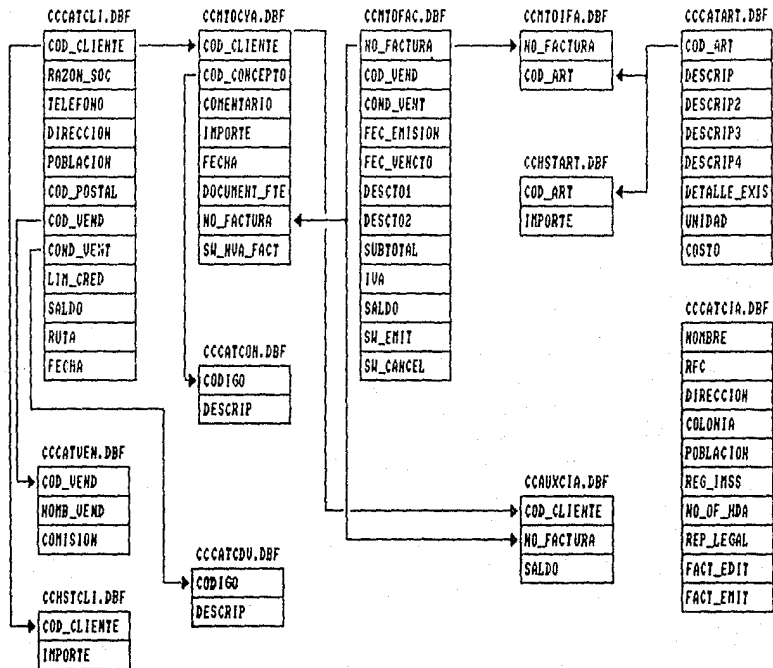
ORD_PARTE	PARTES	ORDENES
Cod_cte	(llave) Cod_parte<-	Cod_cte
No_ord_pte	Nomb_parte	No_ord_pte
Cod_parte	Fecha_venta -->	Cod_parte (llave)
Nomb_parte	Cod_vendedor	Cant_pedido
Descripción	Descripción	Fecha_venta
Costo		
Estructura impropia		Estructura apropiada

Los campos Nomb_parte y Costo dependen del campo Cod_parte, siendo que la llave del archivo es Cod_cte + No_ord_pte, lo que ocasiona la posible repetición de los datos de las partes. Esta condición se conoce como dependencia transitiva.

Como se pudo observar, el modelo de datos que más ventajas ofrece y menos desventajas tiene, además de ser de los más comerciales en este momento, es el modelo relacional, por lo que se buscó un DBMS que cumpliera con esta característica. Esto nos condujo a seleccionar al manejador de bases de datos líder en el mercado actual que es Dbase III+ para la implementación de nuestra base de datos.

Dbase III+ además de ser un DBMS de tipo relacional cuenta con otra serie de características que lo hacen muy atractivo, características que se mencionarán en capítulos posteriores.

Tomando como base el DFD y el DD desarrollados en el capítulo III, como resultado del diseño lógico se obtuvo el esquema relacional de archivos que se muestra en la figura IV.3.5



NOTA LOS ARCHIVOS DE MOVIMIENTOS NO SE ILUSTRARON POR TENER LA MISMA ESTRUCTURA QUE LOS ARCHIVOS MAESTROS.

FIG. IV.3.5 ESQUEMA RELACIONAL DE ARCHIVOS.

CAPITULO V

DESARROLLO DEL SISTEMA

Durante la etapa del desarrollo del sistema las tareas que se deben realizar son las de codificación y documentación de los programas especificados por el diseño del sistema, es decir se debe instrumentar el desarrollo del sistema.

V.1) Aspectos de la instrumentación

Con la instrumentación se realiza la traducción de las especificaciones de diseño a código fuente. El objetivo principal es la escritura del código fuente y la documentación interna, de modo que la concordancia del código con sus especificaciones sea fácil de verificar y que se faciliten la depuración, pruebas y modificaciones. Este objetivo puede alcanzarse haciendo el código fuente tan claro y sencillo como sea posible. Sencillez, claridad y elegancia son los sellos de los buenos programas; obscuridad, ingeniosidad y complejidad son indicaciones de un diseño inadecuado y un pensamiento mal orientado.

La claridad del código fuente se mejora mediante técnicas de codificación estructurada, buen estilo de codificación, documentos adecuados de apoyo, buenos comentarios internos y el aprovechamiento de las características de los lenguajes de programación modernos.

Programación estructurada

La programación estructurada se puede definir como una disciplina que nos permite resolver problemas complejos en donde la calidad de los programas se evalúa por la facilidad con que se les puede dar mantenimiento. Con la programación estructurada es posible reducir el costo de programación y aumentar la productividad ya que se reducen considerablemente los errores y por ende el costo y el tiempo que estos representan.

En la programación estructurada el conjunto de estructuras para la construcción de programas está restringido a unas cuantas estructuras de control que poseen la característica de única entrada, única salida. Dichas estructuras de control básicas se presentan en la figura V.1.

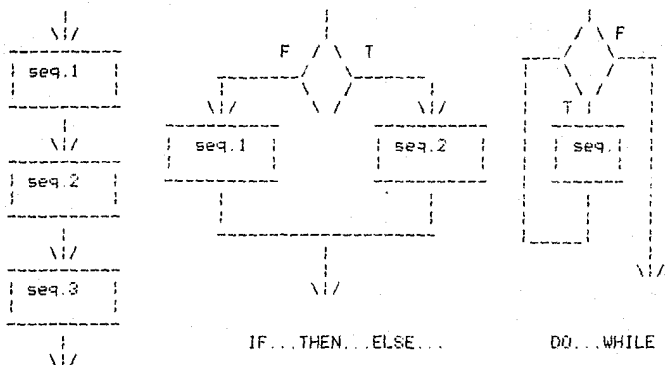


FIG. V.1. ESTRUCTURAS DE CONTROL BASICAS.

La propiedad de una entrada y una salida permite el anidamiento de construcciones dentro de otra en cualquier forma que se desee.

Toda persona posee un estilo propio de programar, sin embargo existen algunas reglas generales que se sugieren para lograr un producto de calidad, a continuación se listan dichas reglas :

- Emplear unas cuantas construcciones estandar de control.
- Utilizar el GOTO de manera disciplinada (o no usarlo).
- Introducir tipos de datos definidos por el usuario en la medida en que el lenguaje lo permita.
- Utilizar un número moderado de proposiciones ejecutables por subrutina.
- Emplear sangrias, paréntesis, espacios y líneas de comentario para mayor legibilidad de los programas.
- No se deben elaborar rutinas demasiado complejas.
- No utilizar identificadores para propósitos múltiples.

En el desarrollo del sistema de cuentas por cobrar se aplicó la técnica de programación estructurada así como la de TOP-DOWN, por lo que se fueron implementando primero los módulos más generales y posteriormente los módulos subordinados a estos, llegando finalmente a los niveles más finos de detalle.

Estándares de programación

A continuación se mencionan los estándares de programación que se siguieron para el desarrollo del presente sistema.

1) Los identificadores de programas o de archivo de datos se construirán aplicando las siguientes reglas:

1.1) Un identificador constará de 8 caracteres para el nombre y de 3 para la extensión, como se presenta a continuación:

```
  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] . [ ] [ ] [ ]  
  \-----/ \-----/  
      NOMBRE      EXTENSION
```

1.2) Se tomará como convención la nomenclatura de las extensiones que proporciona dBase para la identificación de los archivos, la cual es:

DBF.- Archivo de datos.
LBL.- Archivo de etiquetas.
PRG.- Programa fuente.
NDX.- Archivo de índices.
BAK.- Archivo de respaldo.
FRM.- Archivo de reporte.
MEM.- Archivo de memoria.

1.3) Los primeros 2 caracteres (CC) son genéricos para todos los identificadores y sirven para identificar al sistema de cuentas por cobrar.

1.4) Los siguientes 2 caracteres servirán para identificar la función que realiza cada programa y se da de la siguiente manera:

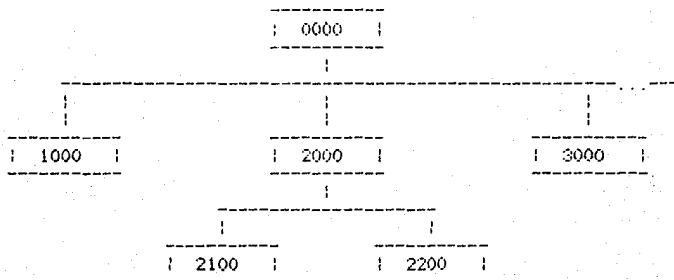
CL .- si es de cálculo.
RP .- si es de reporte.
MN .- si es un menú.
MT .- si es de mantenimiento.
MV .- si es de movimientos.

CO .- si es de consulta.

CA .- si es de captura.

1.5) Para el caso de los programas de uso auxiliar, en lugar de 2 caracteres, serán 3 (AUX). La función de dichos programas básicamente es de mensajes al usuario, encabezados de reportes, encabezados de consultas, etc.

1.6) Para los últimos 4 caracteres se asignará un número según el nivel jerárquico que ocupe cada programa o subrutina, de la manera que se ejemplifica a continuación:



1.7) La tendencia general será de usar el menor número posible de programas auxiliares y de consultas no planeadas.

1.8) Para el caso de los archivos de datos, los caracteres 3, 4 y 5 se utilizarán como se presenta en la siguiente tabla, dejando a criterio del programador los últimos 3 caracteres.

MTD .- Archivo maestro.

CAT .- Catálogo.

MOV .- Archivo de movimientos.

HST .- Archivo histórico.

AUX .- Archivo auxiliar.

TMP .- Archivo temporal.

2) La longitud de las subrutinas no deberá exceder las 100 líneas de código ni deberá ser inferior a 30.

3) Antes de abandonar una subrutina para regresar a un menú, se deberán borrar de memoria las variables locales y cerrar los archivos de datos utilizados.

Cabe aclarar que la codificación de los programas se realizó apegándose a las miniespecificaciones que se diseñaron en el capítulo anterior, mismas que como puede apreciarse, están diseñadas utilizando la técnica de la programación estructurada.

Esto también fue posible ya que el lenguaje de programación dBase III+ ofrece dentro de su conjunto de instrucciones aquellas que permiten instrumentar las estructuras de control básicas anteriormente mencionadas.

Por considerarse pertinente, a continuación se muestra como quedó estructurado modularmente el sistema:

CCMNO.PRG (MENU PRINCIPAL DEL SISTEMA)

CCMN1000.PRG (FACTURACION)

CCCA1100.PRG (EDICION DE FACTURAS)

CCRP1200.PRG (PREFACTURACION)

CCRP1300.PRG (EMISION DE FACTURAS)

CCMN2000.PRG (MOVIMIENTOS ADICIONALES)

CCCA2100.PRG (CAPT. MOVS. ADIC.)

CCCA2200.PRG (BAJAS Y CAMB. DE MOVS.)

CCCL3000.PRG (CIERRE DIARIO)

CCCL4000.PRG (CIERRE MENSUAL)

CCMN5000.PRG (MENU DE CONSULTAS)

CCCO5100.PRG (CONSULTA EDDS. DE CUENTA)

CCCO5200.PRG (CONSULTA ANTIG. DE SALDOS)

CCMN6000.PRG (MENU DE REPORTES)

CCMN6100.PRG (MENU DE REP. DE VENTAS)

CCRP6110.PRG (RELACION DE VENTAS)

CCRP6120.PRG (RELACION DE PAGOS)

CCRP6130.PRG (COMISIONES A PAGAR)

CCRP6140.PRG (REL. NOTAS DE CREDITO)

CCMN6200.PRG (MENU DE REP. DE CTAS. POR COB.)

CCRP6210.PRG (ESTADOS DE CUENTA)

CCRP6220.PRG (ANTIGUEDAD DE SALDOS)

CCRP6230.PRG (ANALISIS DE DOCTOS.)

CCMN6300.PRG (MENU DE REP. ESTADISTICOS)

CCRP6310.PRG (ABC DE CLIENTES)

CCRP6320.PRG (ABC DE ARTICULOS)

CCRP6330.PRG (CLIENTES POR VENDEDOR)

CCMN7000.PRG (MENU DE MANTENIMIENTO A CATALOGOS)

CCMT7100.PRG (MANTENIMIENTO A CLIENTES)

CCMT7200.PRG (MANTENIMIENTO A ARTICULOS)

V.2) Documentación

La documentación de un sistema debe incluir todos los documentos de apoyo generados durante las fases de análisis, diseño, instrumentación, pruebas y mantenimiento del sistema, así como el código fuente del mismo.

La necesidad de documentar un sistema se presenta porque en la mayoría de los casos unas personas son las que codifican los programas y otras son las que les dan mantenimiento.

La documentación puede darse a dos niveles, la documentación interna y la documentación externa.

La documentación interna consiste en los comentarios que existen inmersos en el mismo código ejecutable, mientras que la documentación externa se refiere a toda la información que se asienta en un manual de programación. Esta información incluye lo siguiente:

- a) Propósito del programa.
- b) Parámetros de entrada y salida.
- c) Una explicación a grandes rasgos de la mecánica del prog.
- d) Listado del programa.
- e) Advertir sobre posibles errores fatales.
- f) Programas o subprogramas que llamen al programa que se documenta.
- g) Programas o subprogramas que llama el programa que se está documentando.

En el caso del sistema de cuentas por cobrar, como documentación interna, en todos los programas se introdujeron los siguientes datos:

- NOMBRE DEL PROGRAMA.
- OBJETIVO DEL MISMO.
- AUTOR.
- FECHA DE CREACION.
- FECHA DE LA ULTIMA ACTUALIZACION.

También como parte de esta documentación se introdujeron suficientes comentarios dentro del código fuente de cada programa para hacer lo más claro posible el objetivo de los mismos.

La documentación externa del sistema se integra con todos los documentos generados en las fases de análisis y diseño, como son el Diagrama de flujo de datos, el Diccionario de datos, las miniespecificaciones, la Carta de estructura, etc.

CAPITULO VI

IMPLANTACION DEL SISTEMA

A la fase del desarrollo del sistema le sigue otra de no menor importancia que es la de implantación del mismo.

El objetivo que se persigue en esta parte es el de someter el desarrollo a un control de calidad, es decir, se pretende valorar y mejorar en la medida de lo posible la calidad del producto obtenido durante las etapas de diseño y desarrollo.

El control de calidad debe ser un modelo planeado y sistemático de todas las acciones necesarias que proporcionen la confianza de que el producto obtenido se ajusta a los requisitos técnicos obtenidos. A menudo, el personal encargado del control de calidad es diferente al grupo de desarrollo, aunque generalmente deban trabajar en conjunto para el establecimiento del Plan de pruebas.

La situación de que los dos grupos se integren por personas diferentes otorga un grado de imparcialidad a las actividades del control de calidad, y permite que el personal encargado de tal control sea especialista en su disciplina.

Un plan de pruebas, debe especificar los objetivos de las pruebas, los criterios para la terminación de las mismas, el plan de integración del sistema, los métodos que se usarán en módulos particulares, así como entradas de pruebas particulares y resultados esperados.

La mejor forma de valorar la calidad de un producto, es sometiéndolo a una serie de pruebas que permitan determinar su apego a las especificaciones y a las expectativas originales de diseño.

Las pruebas al sistema implican dos clases de actividades: pruebas de integración y pruebas de aceptación. Las diferentes estrategias de integración tienen como objetivo principal la integración funcional de los módulos ajustándolos a las particularidades del sistema.

VI.1) Pruebas de integración

Se pueden definir los siguientes tipos de integración:

Integración no incremental

Este tipo de integración se caracteriza por seguir la siguiente secuencia:

- 1.- Escribir y probar cada módulo separadamente (unit-test).

- 2.- reunir y probar todas las unidades agrupadas en subsistemas (subsystem-test).
- 3.- combinar todos los subsistemas de tal forma que se integre todo el sistema.
- 4.- hacer una prueba global del sistema.

Esta forma de integración presenta el problema de que se puede encontrar una fase errónea y crucial para el sistema en la fase final de las pruebas quedando ya muy poco tiempo para corregir los errores, lo cual puede redundar en un retraso en la entrega de resultados y en un mayor costo.

Integración Incremental

Se conocen tres formas de integración incremental, las cuales son, integración descendente (Top-down), integración ascendente (Bottom-up) e integración combinada o de Sandwich.

Integración incremental descendente

La integración descendente consiste en probar primero el módulo de mayor nivel jerárquico simulando cada uno de sus módulos subordinados con módulos falsos, es decir, módulos que no hacen nada (solo despliegan letreros). Posteriormente cada módulo falso se reemplaza por su módulo real, el cual será probado de la misma forma hasta llegar a la parte inferior del sistema.

Integración incremental ascendente

Este tipo de integración se realiza de manera inversa a la anterior, ya que primero se prueban los módulos de los niveles inferiores, integrándose así los módulos en forma piramidal hasta llegar al nivel más alto.

La ventaja de utilizar este tipo de integración es que primero se probarán los módulos que realizan las entradas y salidas físicas, logrando con ello obtener resultados inmediatos del sistema, aunque tiene la desventaja de que es muy difícil implementarse.

Integración incremental combinada

Como su nombre lo dice, esta técnica combina las características de los dos métodos anteriores. En este tipo de integración, primero se integra la parte superior del sistema siguiendo el método descendente (Top-down), integrándose posteriormente la otra mitad por el otro método (Bottom-up). Esta técnica presenta la desventaja de no poder traslapar las etapas de diseño y pruebas, (no se ahorra tiempo), aspecto que de ninguna manera es fundamental para el desarrollo del sistema.

VI.2) Pruebas de aceptación

Las pruebas de aceptación implican la planeación y la ejecución de pruebas funcionales, de desempeño y de tensión para demostrar que el sistema satisface sus requisitos. Las pruebas de tensión se llevan a cabo con el fin de establecer las limitaciones del sistema. Por ejemplo, un compilador puede probarse para determinar el efecto de sobreflujo en la tabla de símbolos, o un sistema de tiempo real se puede probar para conocer el efecto de la llegada simultánea de varias interrupciones de alta prioridad.

Las pruebas de aceptación generalmente requieren de la presencia del usuario final, ya que como es de suponer, es el que conoce perfectamente el problema y sus posibles casos cruciales.

Existen diferentes técnicas para ejecutar las pruebas de aceptación:

Pruebas de caja negra

Como el nombre lo sugiere, esta técnica implica la utilización del concepto de cajas negras para probar el sistema o los subsistemas que lo componen, es decir, se aplican ciertas entradas al sistema o a los subsistemas, analizando si las salidas corresponden a los resultados esperados.

Dado que las posibles entradas a un sistema por lo general son infinitas, se debe seleccionar un grupo de datos prueba que cumpla con características similares, esto es, datos que cubran entradas típicas, casos frontera y datos inválidos. La selección de un buen grupo de datos prueba supondrá un gran ahorro de tiempo y costo. Esta actividad normalmente es un proceso muy minucioso y depende de la habilidad del analista el que la agrupación de las entradas sea lo más acertada posible, ya que se parte de la idea de que si una de las entradas del grupo produce una salida errónea, entonces todas las entradas de dicho grupo generarán salidas erráticas.

Análisis estático

El análisis estático es una técnica que se usa para valorar las características estructurales del código fuente. Con esta prueba se examina la estructura del código, pero no en estado de ejecución. Puesto que el análisis estático se relaciona con la estructura de los programas, es especialmente útil para descubrir prácticas de codificación cuestionable y el alejamiento de los estándares de codificación, además de detectar errores estructurales tales como variables no inicializadas y discrepancias entre parámetros reales y formales. El análisis estático puede realizarse a mano, aunque existen herramientas automatizadas que pueden ayudarnos. Un analizador estático por lo general construye una tabla de símbolos y una gráfica de flujo de control para cada subprograma, además de una gráfica de llamados

al programa completo. La tabla de símbolos contiene información acerca de cada variable, p. ej. los atributos de sus tipos, la proposición donde se declararon, las proposiciones donde se les asignan nuevos valores, y las proposiciones donde se usan para dar valores a otras.

Dada una gráfica de flujo de control y una tabla de símbolos donde para cada variable dentro de un subprograma se conozcan los números de proposición en donde las variables se declaran, se les asignan valores y se usan, entonces un analizador estático puede determinar información del flujo de los datos tal como variables no inicializadas, variables declaradas pero no utilizadas y variables con valores asignados pero que no se usan nunca.

Las principales limitaciones teóricas de este análisis las impone la teoría de la decidibilidad. Esta teoría establece que no es posible escribir un analizador estático que pueda examinar un programa arbitrario y unos datos de entrada arbitrarios para determinar si tal programa ejecutará o no una proposición o llamará una subrutina particular, o ejecutará una instrucción particular de entrada/salida, o saltará a una etiqueta dada. Por lo que es imposible efectuar un análisis estático sobre un programa arbitrario y obtener automáticamente un conjunto de datos de entrada que guiarán al programa a lo largo de una ruta de control particular. Sin embargo, se pueden imponer restricciones a los programas de tal forma que los aspectos indecidibles se conviertan en decidibles.

Análisis dinámico

Los casos de pruebas dinámicas se suelen utilizar para obtener información que es difícil o imposible de obtener con el análisis estático. Existen limitaciones tanto prácticas como teóricas del análisis estático. Una limitación práctica es la evaluación dinámica de las referencias a la memoria al momento de ejecución. P. ej. en algunos lenguajes de programación de alto nivel, los subíndices de arreglos y las variables de tipo apuntador proporcionan referencias dinámicas a la memoria y un analizador estático no puede evaluar valores de esta naturaleza, por lo que para estos casos, es mucha ayuda un analizador dinámico.

El plan de pruebas que se utilizó para la evaluación del sistema quedó establecido de la siguiente forma:

- Objetivos de las pruebas.
- Acotación de las pruebas.
- Integración del sistema.
- Grupos de datos de prueba seleccionados.

Objetivos de las pruebas.

- a) Determinar si los datos que se almacenan en los archivos de movimientos corresponden a los capturados en la facturación y en la captura de los movimientos adicionales.
- b) Hacer pruebas de tensión sobre los campos de las rutinas de captura y de los archivos que integran la base de datos.
- c) Verificar que el conjunto de cargos y abonos asignados a la cuenta del cliente corresponda a su saldo correcto.
- d) comprobar que los datos transferidos durante el cierre diario afectan correctamente a los archivos estadísticos y al saldo de los clientes.
- e) Determinar la no duplicidad de los movimientos capturados tanto en archivos de movimientos como en maestros.
- f) Revisar que el proceso del cierre mensual, realmente depure en su totalidad las facturas y los movimientos que ya no se encuentren vigentes en archivos.

Acotación de las pruebas

Las diferentes pruebas que se realizaron se acotaron de la siguiente manera:

- a) Se deberá hacer un seguimiento de la información desde su captura hasta su almacenamiento calculando las operaciones matemáticas de los totales con hasta dos decimales.
- b) Solamente se someterán a esta prueba las rutinas de captura en los campos de cantidad, de importe, de subtotales y de totales. Lo mismo se realizará para los archivos de datos.
- c) En esta prueba se incluirán todos los tipos de movimientos tanto de cargos como de abonos.
- d) Esta prueba se realizará con una transferencia de carga ligera, otra con media carga y por último con una carga pesada de datos, examinando además si las facturas canceladas o borradas no afectan a los archivos maestros y al saldo del cliente.
- e) Esta prueba requiere realizarse de una forma totalmente completa, es decir que deben examinarse mediante un programa todas y cada una de las facturas existentes en archivos y todos los movimientos que se le hubieran aplicado a la misma.

Integración del sistema

Para la integración del sistema de cuentas por cobrar se utilizó la técnica incremental descendente ya que primero se probaron los módulos del nivel más alto en el diagrama de estructura (por ser los más generales) integrando posteriormente los módulos subordinados. Con esto se logró un ahorro considerable de tiempo ya que para probar un módulo de cualquier nivel se tenían los módulos superiores ya desarrollados y se podía contar con los parámetros que estos mismos debían proporcionar.

También cabe mencionar que se llevaron a cabo pruebas de caja negra para determinar la correcta funcionalidad, desempeño e integración de los diferentes módulos que constituyen al sistema.

Grupos de datos de prueba seleccionados

A continuación se listan los grupos de datos de prueba que se utilizaron para llevar a cabo las mismas.

a) Datos de una factura:

- Datos generales de la misma:
 - Código del cliente
 - Fecha de elaboración.
 - Número de pedido.
 - Condiciones de venta.
 - Código del agente que los atiende.
 - Descuentos especiales (si los hay).
- Datos de la mercancía solicitada.
 - Código del artículo.
 - Cantidad solicitada.
 - Precio unitario.

b) Datos de un cargo o abono:

- Código del movimiento.
- Código del cliente.
- Número del documento que ampara el movimiento.
- Factura a la que se asigna el movimiento.
- Fecha en que se realiza.
- Importe.

CAPITULO VII

MANTENIMIENTO

Un aspecto de fundamental importancia en lo que a operación de un sistema de información se refiere, es el del mantenimiento. Una vez que un sistema ha sido liberado, comienza una nueva etapa del mismo, ya que a partir de ese momento deberá proporcionar los resultados para los que ha sido diseñado con el menor número de fallas posible. Esto se logra solo si el producto obtenido ha sido desarrollado con calidad. Se ha definido como mantenimiento a la facilidad con que el software pueda ser expandido o contraído para satisfacer nuevos requerimientos o pueda ser corregido cuando se detecten errores o deficiencias.

Se puede afirmar que el mantenimiento de un sistema es el proceso más caro realizado durante su ciclo de vida. La figura VII.1 ilustra los costos asociados a cada etapa del ciclo de vida de un sistema (Perspectives on Software Engineering).

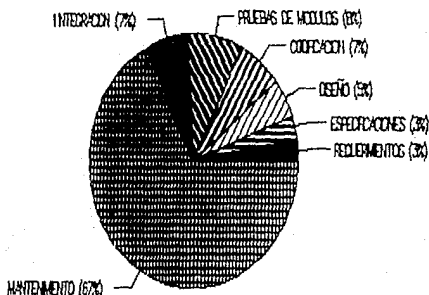


Fig. VII.1 Costos asociados a las etapas del ciclo de vida.

El mantenimiento del software tiene básicamente los siguientes propósitos: Corrección, Adaptación, Engrandecimiento y Reestructuración.

Corrección .- Es la modificación que se realiza con la idea de solucionar un problema previamente localizado.

Adaptación .- Es la modificación debida a un cambio en el hardware o software en el que reside el sistema.

Engrandecimiento .- Son las modificaciones que permitirán al sistema realizar nuevas funciones debidas a requerimientos de los usuarios. Este tipo de modificación es en la que más incurren la mayoría de las instalaciones.

Reestructuración - Generalmente son cambios que se efectúan con el objeto de mejorar la estructura interna del sistema, conservando su comportamiento externo.

Gran parte de los problemas a los que se enfrenta al encargado de dar mantenimiento a un sistema se deben a que el grupo de desarrollo no se preocupó por preveer la fase del mantenimiento. Con el objeto de lograr un producto mantenible, se deben cumplir los siguientes estándares:

1.- Requerimientos.

Los requerimientos deben ser escritos, priorizados y definidos de tal forma que se puedan probar facilmente. Los requerimientos opcionales y los futuros deben ser claramente diferenciados.

2.- Especificaciones.

Las especificaciones deben estar escritas en términos que se puedan probar y deben estar diferenciadas en requeridas, opcionales y futuras.

3.- Diseño.

Las facilidades del diseño para poder extenderlo, contraerlo y adaptarlo deben explicarse con ejemplos y ejercicios de los cambios esperados.

4.- Código fuente.

Se deben emplear lenguajes de alto nivel siempre que sea posible. Únicamente versiones estandar y características estandar del lenguaje de programación deberán permitirse.

Todo el código deberá estar bien estructurado y deberá documentarse explicando el propósito de cada módulo en sus entradas, salidas y variables, para facilitar la prueba del módulo.

5.- Información del sistema.

Todos los documentos del sistema se deberán entregar al encargado del mantenimiento del sistema inmediatamente después de la definición de los mismos.

De igual forma que para el desarrollo del sistema, se requieren los tres ingredientes básicos para darle el mantenimiento adecuado.

1) Herramientas técnicas.

Deberá asignarse tiempo de máquina libre para las tareas de mantenimiento aun en las horas pico.

Se procurará un medio ambiente de pruebas capaz de simular un medio ambiente operacional.

Se utilizarán como herramientas generadores de pruebas y verificadores.

Se deberá disponer de librerías.

Deberán existir auditores de código para checar la estructura y la complejidad.

2) Experiencia técnica.

La labor de mantenimiento es muy compleja y de gran responsabilidad, por lo que se debe buscar que la lleve a cabo personal con la experiencia debida. No es recomendable dedicar al mantenimiento a los programadores novatos, ya que esto puede ser muy costoso si se llegara a destruir la integridad del producto.

3) Técnicos de administración.

Con el fin de mantener la calidad y confiabilidad del producto se deben implantar un conjunto de técnicas que garanticen la adecuada adhesión a los estándares. Siendo fundamental para tal efecto el incorporar las siguientes cuatro técnicas:

- Establecer las prioridades y metas buscadas.
- Seguir exigiendo los mismos estándares que fueron usados para controlar la calidad durante el desarrollo.
- Documentar el proceso de mantenimiento así como las modificaciones a los programas.
- Establecer auditorías periódicas de control de calidad así como revisiones de aceptación.

CAPITULO VIII

DEMOSTRACION DE RESULTADOS

En el presente capítulo se hablará acerca de las características que posee el sistema de cuentas por cobrar y de los requerimientos de equipo para el mismo. Por otro lado, también se dará una guía para la correcta operación del sistema, el cual puede ser utilizado como el manual del usuario VII.1

Características del sistema

El sistema de cuentas por cobrar fué desarrollado en el lenguaje de programación que acompaña al DEMS dBase III+. Se seleccionó este lenguaje porque permite programar estructuralmente y por su gran facilidad en el manejo de archivos y de creación de consultas no planeadas. Algunas de las características más relevantes de dBase III+ es que puede manejar hasta 1000 millones de registros con hasta 128 campos en cada uno de ellos conteniendo hasta 4000 caracteres por registro; puede operar varios campos a la vez y trabajar hasta con 10 archivos simultáneamente. Otra característica que vale la pena mencionar es que dBase III+ permite la exportación e importación de archivos con otros sistemas de manejo de datos con la única limitante de que los datos en los dos sistemas deben tener un formato ASCII.

Con la idea de proporcionar una mayor velocidad de ejecución, el sistema fue compilado utilizando el compilador Clipper (summer 87'). Se eligió este compilador por diversas razones, una de ellas es que es totalmente compatible con dBase III+, otra razón es que Clipper soporta una serie de "mejoras" que a continuación se mencionan:

- Incluye la función INKEY, con lo que es posible modificar la función de algunas tecla para poder efectuar rutinas de uso especial.

- Se pueden usar hasta 64000 variables de memoria activas.
- Permite usar hasta 1024 campos por registro.
- Se pueden crear funciones definidas por el usuario.
- Usa macros en declaraciones DO WHILE.
- Permite el uso de macros recursivas.
- Se pueden llamar un número ilimitado de programas externos.
- Se puede utilizar el commando FOR...NEXT

Independientemente de las mejoras antes enumeradas, la razón más importante de utilizar Clipper sigue siendo la rapidez en el tiempo de ejecución, lo cual representa una gran ventaja sobre dBase. Como se puede apreciar fácilmente en la etapa del análisis, el sistema se desarrolló bajo la filosofía del diseño estructurado, por lo que para su programación, se utilizaron todas las herramientas que para tal fin ofrecen dBase y Clipper.

El sistema quedó en un único módulo ejecutable, esto gracias a la característica de Clipper de que no necesita de un RUNTIME para la ejecución de un proceso.

El sistema está compuesto por archivos de tipo: ejecutables (.EXE), de datos (.DBF) y de índices (.NTX). A continuación se da una lista de dichos archivos:

CCMNO.EXE .- Es el archivo que permite la ejecución del sistema.

INICIA.EXE .- Este archivo sólo se necesita correr una vez al momento de instalar el sistema y su función es crear los archivos de índices que se utilizarán durante la operación del sistema.

CCMOVFAC .- En este archivo se almacenan temporalmente las facturas que se van creando.

CCMOVCYA .- Archivo que se utiliza para el almacenamiento temporal de los cargos y abonos.

CCMOVIFA .- En este archivo se almacenan temporalmente los ingredientes de las facturas.

CCMTOFAC .- Es el archivo maestro en el que se almacenan los datos generales de las facturas.

CCMTOCYA .- Es el archivo maestro en el que se almacenan los datos de los cargos y abonos.

CCMTGIFA .- Este archivo maestro contiene los ingredientes de las facturas almacenadas en el archivo CCMTOFAC.

CCAUXCYA .- En este archivo se almacenan los cargos y abonos de facturas no localizadas en el archivo CCMTOFAC.

CCCATCLI .- Archivo del Catálogo de clientes.

CCCATART .- Archivo del Catálogo de artículos.

CCHSTCLI .- Archivo histórico de clientes.

CCHSTART .- Archivo histórico de artículos.

CCCATCON - Archivo del Catálogo de conceptos.

CCCATVEN .- Archivo del Catálogo de vendedores.

CCCATCDV .- Archivo del Catálogo de condiciones de venta.

Requerimientos del sistema

El sistema puede ejecutarse en sistemas personales IBM ó compatibles y los requerimientos mínimos que necesita para ello son:

- 512 Kbytes de memoria RAM
- Disco duro de 10 Mbytes
- Sistema operativo MS-DOS versión 2.0 o posterior.
- Impresora con carro de 10" ó 15" Epson o compatible.

VII.2 Operación del sistema

Para lograr una correcta operación del sistema, se debe proporcionar al usuario un manual en el cual se contemple en forma clara y concisa cada uno de los pasos que deben seguirse para lograr ese objetivo. A continuación se presenta dicho material.

Instalación del sistema

El sistema se proporciona en dos diskettes de 5 1/4", cuyo contenido es el siguiente:

En el disco uno se encuentra el archivo CCMNO que como se mencionó anteriormente es el que nos permite la ejecución del sistema. También en este disco se encuentra el archivo INICIA.

En el disco dos se localizan las estructuras de los archivos de datos que van a ser utilizados por el sistema.

Para comenzar la instalación, se recomienda al usuario crear un subdirectorio en el disco duro que sea de uso exclusivo para el sistema, mismo en el que se deberá copiar el contenido de ambos discos. A continuación, se deberán crear los archivos de índices ejecutando el programa INICIA. Mientras se lleva a cabo la creación de estos archivos se verá que el foco indicador (led) del disco duro se prenderá y apagará en forma repetitiva.

Para ejecutar el sistema de Cuentas por cobrar se deberá teclear la palabra CCMNO, con lo que se desplegará el siguiente menú, que es el menú principal del sistema.

- 1.- FACTURACION
- 2.- MOVIMIENTOS ADICIONALES
- 3.- CIERRE DIARIO.
- 4.- CIERRE MENSUAL
- 5.- REALIZA CONSULTAS
- 6.- OBTIENE REPORTES
- 7.- MANTENIMIENTO A CATALOGOS
- 8.- SALIR DEL SISTEMA

Como se sugiere en el menú anterior, el usuario debe seleccionar una opción de las ocho que se presentan (Si se está trabajando por primera vez en el sistema, se sugiere seleccionar en primer lugar la opción de mantenimiento a catálogos, ya que con esto se podrán dar de alta los catálogos de clientes y de artículos).

De acuerdo a la opción que se haya seleccionado, se ejecutará el proceso respectivo de la siguiente manera:

1.0 .- Proceso de FACTURACION:

Este menú permite al usuario realizar diferentes funciones, tales como la edición de facturas, la prefacturación y la emisión de facturas.

- 1.- FACTURACION
 - 2.- MOVIMIENTOS ADICIONALES
 - 3.- CIERRE DIARIO.
 - 4.- CIERRE MENSUAL
 - 5.- REALIZA CONSULTAS
 - 6.- OBTIENE REP
 - 7.- MANTENIMEN
 - 8.- SALIR DEL S
- 1.- EDICION DE FACTURAS
 - 2.- PREFACTURACION
 - 3.- EMISION DE FACTURAS
 - 4.- REGRESA MENU ANTERIOR

1.1 .- EDICION DE FACTURAS

Al entrar a este proceso, automáticamente se presentará en pantalla el número de factura siguiente a elaborarse, ya que el sistema lleva un registro de dicho número consecutivo. Si lo que se desea es corregir o aumentar una factura previamente elaborada que no haya sido emitida, entonces se deberá introducir el número de dicha factura. En cualquiera de los dos casos, se presentará en pantalla el siguiente formato de captura:

```

CCCA1100.FRG                                NOMBRE DE LA EMPRESA
S. CUENTAS POR COBRAR                       FECHA: 90.11.26
MEMO DE FACTURACION                         EDICION DE FACTURAS  HORA: 20:05:35
    
```

```

FACTURA :138641  CLIENTE: 24363 FEDERICO LOPEZ MENDOZA
FECHA :90.11.05  PEDIDO : 3281  CONDICIONES: 30 30 DIAS
AGENTE : 05 OSCAR EDUARDO SURTIGA RDEZ.  COMISION: 8 %
    
```

ARTICULO	CANTIDAD	UNIDAD	PRECIO UNITARIO	IMPORTE
50075	28.000	KGS	18000.00	504,000.00
CONC. FRESA #428				
51010	28.000	KGS	5000.00	140,000.00
CONC. LIMON 6				0.00
				0.00
				0.00

TECLAS: <INS>=ACEPTAR <HOME>=DAR DE BAJA <END>=CANCELAR

La anterior pantalla de captura trabaja en forma similar a un editor de textos, de tal manera que aunque da la impresión de únicamente poder capturar cinco artículos, realmente se pueden introducir hasta un máximo de veinte. Existen algunas funciones inherentes que se pueden realizar en este proceso, tales como las que a continuación se presentan:

- Con la tecla DEL se puede borrar un renglón de la factura, con la única condición de estar posicionado en él.
- Con las teclas PgUp y PgDn se puede seleccionar cualquiera de las cuatro descripciones que existen para un mismo artículo.

- Si se introduce un código de artículo que ya existe en la factura pero en diferente renglón se reportará como un error.

- La tecla INS permite aceptar la factura tal como esté.

- Es posible cancelar una factura ya emitida oprimiendo la tecla END. Hay dos diferentes formas de hacer esto; la primera es cancelando únicamente y la segunda es cancelando y creando una

nueva factura con el mismo contenido de la que se cancela. Para lograr esto último, el sistema preguntará si se desea la creación de una nueva factura.

Una vez capturados los ingredientes de una factura, el sistema pasará a la siguiente pantalla para pedir los datos complementarios, como pueden ser los diferentes descuentos y el IVA a cobrar.

Para el sistema de Cuentas por Cobrar se considera que las facturas del usuario vienen foliadas, por lo que el sistema lleva un control especial sobre dicho folio.

Esto se hace con la idea de que exista una seguridad sobre la factura que el usuario va emitiendo. El folio anteriormente referido es inalterable por lo que una factura que no se desee, no se debe destruir, solamente se debe cancelar en el sistema.

1.2 .- PREFACTURACION

Como su nombre lo indica, en este proceso es posible imprimir un reporte de las facturas elaboradas con la finalidad de revisar si su contenido es el correcto, para el usuario lo que se pretende que al imprimir las facturas definitivas, no existan errores. Para ello se utiliza la siguiente pantalla:

CCRP1200.PRG
S.CUENTAS POR COBRAR
MENU DE REPORTES

REPORTE DIARIO DE PREFACTURACION

NOMBRE DE LA EMPRESA
FECHA: 90.11.25
HORA: 20:06:42

FACTURA INICIAL: 138640

FACTURA FINAL: 138641

COMENTARIO:

<<PREPARE SU IMPRESORA>>

PARA CONTINUAR TECLEE (S/N)

1.3 .- EMISION DE FACTURAS

De manera similar a la anterior se lleva a cabo el proceso de emision de facturas, donde en lugar de imprimir un reporte, lo que se imprimen son las facturas definitivas ya revisadas y depuradas (previa seleccion de la impresora a utilizar), para lo cual se utiliza la siguiente pantalla:

CCRP1300.PRG
S. CUENTAS POR COBRAR
MENU DE FACTURACION

EMISION DE FACTURAS

NUMERE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:07:30

FACTURA INICIAL: 138542 FACTURA FINAL: 138553 FECHA EMISION: 90.11.26

<<PREPARE SU IMPRESORA>>

PARA CONTINUAR TECLEE <S/N>

2.0 .- MOVIMIENTOS ADICIONALES

El menú que a continuación se presenta permite al usuario capturar y editar los cargos y abonos asignados a los clientes.

CONG2000.PRG
S. CUENTAS POR COBRAR
MENU PRINCIPAL

MENU DE MOVIMIENTOS ADICIONALES

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:08:06

1.- FACTURACION
2.- MOVIMIENTOS ADICIONALES
3.- CIERRE DIARIO.
4.- CIERRE MENSUAL
5.- REALIZA CONSULTAS
6.- OBTIENE REP
7.- MANTENIMIEN
8.- SALIR DEL S

1.- CAPTURA MOVIMIENTOS
2.- EDICION DE MOVIMIENTOS
3.- REGRESA A MENU PRINCIPAL

2.1 .- CAPTURA MOVIMIENTOS

Se denomina con el nombre de "movimientos adicionales" a todos aquellos movimientos (cargos o abonos) que se le tienen que hacer a los clientes sobre alguna factura que tengan vigente. Dicho movimiento deberá respetar la siguiente codificación de los diferentes conceptos:

CARGOS	ABONOS
110 FACTURACION	211 NOTA DE CREDITO
130 CHEQUES DEVUELTOS	220 PAGOS
150 CARGOS DIVERSOS	230 PAGOS A CHEQUES DEVUELTOS
170 FLETES	250 ABONOS DIVERSOS

CCCA2100.FRG
S. CUENTAS POR COBRAR
MOVS. AL CLIENTE

CAPTURA DE DOCUMENTOS ADICIONALES

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:08:26

C.C.TOT.DOC. :	10	C.C.IMP.DOC. :	150,000.00			
CPTO. CLIENTE	DOCTO.	COMENTARIO	FACTURA	FECHA	IMPORTE	
220	01050	11523	PAGO DEL DIA 16.	137480	90.11.06	150,600.00

CLIENTE: JOSE ALBERTO ALVAREZ ALVAREZ

Para terminar teclee <000> en concepto

2.1 .- EDICION DE MOVIMIENTOS

En esta opción es posible hacer correcciones a los movimientos adicionales capturados en el punto anterior. Si se quiere borrar totalmente un movimiento, se pide como condición que el campo "importe" se deje en ceros.

CCCA2200.PRG
S. CUENTAS POR COBRAR
MOVS. AL CLIENTE

EDICION DE MOVIMIENTOS AL CLIENTE

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:09:50

CLIENTE: 01050 JOSE ALBERTO ALVAREZ ALVAREZ

CONCEPTO: 220

DOCUMENTO: 11523

COMENTARIO:

FACTURA: 137480

FECHA: 90.11.06

IMPORTE: 15,000,000.00

Para borrar este movimiento el importe debe ser cero

3.0 .- CIERRE DIARIO

Este proceso lleva a cabo una labor muy importante, en la cual se realiza la transferencia de toda la información existente en archivos de movimientos hacia los archivos maestros, con lo que esta información queda consolidada, por lo mismo, antes de proceder a esta transferencia se realizan dos cosas de mucha importancia; la primera es que se da la oportunidad al usuario de respaldar los archivos de datos previniendo así cualquier contingencia que pudiera presentarse, y la segunda es que se realiza una serie de validaciones sobre la información a transferir de tal manera que no existan inconsistencias en la misma.

CCCL3000.PRG
S. CUENTAS POR COBRAR
MENU PRINCIPAL

CIERRE DIARIO

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:11:26

DESEA CONTINUAR CON EL CIERRE DIARIO?

4.0 .- CIERRE MENSUAL

Otro proceso de suma importancia es el del cierre mensual. La finalidad de este proceso es la de hacer la depuración de los archivos maestros borrando los movimientos que ya no tengan vigencia para el sistema. Antes de llevarse a cabo este proceso, el sistema da la oportunidad al usuario de respaldar los archivos de datos de forma similar al cierre diario.

CCCL4000.PRG
S. CUENTAS POR CERRAR
CIERRE DEL PERIODO

CIERRE MENSUAL

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:12:03

FECHA DEL CIERRE : 90.11.06
DESEA CONTINUAR CON EL CIERRE MENSUAL?

5.0 .- REALIZA CONSULTAS

Como se puede observar, en este menú es posible hacer consultas inmediatas del estado de cuenta de algún cliente o de la antigüedad de sus débitos.

COMNS000.PRG
S.CUENTAS POR COBRAR
MENU PRINCIPAL

MENU DE CONSULTAS EN PANTALLA

NOMBRE DE LA EMPRESA
FECHA: 50.11.26
HORA: 20:12:20

- | | |
|-----------------------------|----------------------------------|
| 1.- FACTURACION | |
| 2.- MOVIMIENTOS ADICIONALES | |
| 3.- CIERRE DIARIO | |
| 4.- CIERRE MENSUAL | |
| 5.- REALIZA CONSULTAS | |
| 6.- OBTIENE REPORTE | |
| 7.- MANTENIMIENTO A | 1.- CONSULTA ESTADOS DE CUENTA |
| 8.- SALIR DEL SISTE | 2.- OBTIENE ANTIGÜEDAD DE SALDOS |
| | 3.- REGRESA A MENU PRINCIPAL |

5.1 .- CONSULTA ESTADOS DE CUENTA

En la pantalla siguiente se presenta un ejemplo de un estado de cuenta. Aquí se puede apreciar como se desglosan los movimientos del cliente ordenados por fecha (que es el orden normal de llegada). También se puede ver que el saldo que arroja el total de sus movimientos es el saldo de su cuenta (es el que aparece en la parte superior de la pantalla). Para entrar a esta opción, se debe proporcionar el código del cliente, el cual es validado por el sistema.

```

CCCC05100.PRG                                NOMBRE DE LA EMPRESA
S.CUENTAS POR COBRAR                          FECHA: 90.11.26
CONSULTAS DIVERSAS                            ESTADO DE CUENTA      HORA: 20:12:45
    
```

```

-----
CLIENTE: 24363 FEDERICO LOPEZ MENDOZA        FECHA:
LIMITE CREDITO: 0.00                          SALDO: 740,600.00  INIC.: 01.01.01
                                                    FINAL: 99.12.31
    
```

FECHA	CONCEPTO	FACTURA	DOCUMENTO	CARGOS	ABONOS	SALDO
90.05.07	110	137381		796,950.00		796,950.00
90.11.01	220	137381	878		796,950.00	0.00
90.11.05	110	138641		740,600.00		740,600.00

Fin del estado de cuenta. Para continuar presione cualquier tecla

5.2 .- Consulta antigüedad de saldos

Al igual que en los estados de cuenta, para consultar la antigüedad de saldos de un cliente se debe proporcionar su código. La antigüedad de saldos, como puede observarse, consta de las facturas que el cliente no ha liquidado y cuya fecha de pago ya se encuentra "vencida", es decir, que la fecha actual es posterior a la de pago. Esta consulta es de especial importancia, ya que permite conocer los saldos que pueden ser enviados a cobro.

CCC05200.PRG
S. CUENTAS POR COBRAR
CONSULTAS DIVERSAS

ANTIGÜEDAD DE SALDOS

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:13:05

CLIENTE: 24363 FEDERICO LOPEZ MENDOZA
AGENTE: 05 OSCAR EDUARDO SURTIGA HDEZ.

FECHA FACTURA VENCIDO	SALDO CORRIENTE	SALDOS VENCIDOS		
		A 30 DIAS	A 60 DIAS	90 DIAS O MAS
138641 90.12.05	740,600.00			
T O T A L E S :	740,600.00	0.00	0.00	0.00
TOTAL DE SALDOS VENCIDOS :		0.00		

Fin de la antigüedad de saldos. Presione cualquier tecla para cont.

6.0 .- OBTIENE REPORTES

El siguiente menú nos presenta los tipos de reportes que pueden obtenerse:

CCMS000.PRG
S.CUENTAS POR COBRAR
MENU PRINCIPAL

MENU DE REPORTES

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:13:23

1.- FACTURACION
2.- MOVIMIENTOS ADICIONALES
3.- CIERRE DIARIO.
4.- CIERRE MENSUAL
5.- REALIZA CONSULTAS
6.- OBTIENE REP
7.- MANTENIMIENT

8.- SALIR DEL S
1.- OBTIENE REPORTES DE VENTAS
2.- OBTIENE REPORTES DE COBRANZA
3.- OBTIENE REPORTES ESTADISTICOS
4.- REGRESA A MENU PRINCIPAL.

6.1 .- OBTIENE REPORTES DE VENTAS

Este menú permite al usuario elegir de entre los diferentes reportes de ventas de la compañía:

CCMNS100.PRG
S. CUENTAS POR COBRAR
MENU DE REPORTES

MENU DE REPORTES DE VENTAS

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:13:36

- 1.- FACTURACION
 - 2.- MOVIMIENTOS ADICIONALES
 - 3.- CIERRE DIARIO
 - 4.- CIERRE MENSUAL
 - 5.- REALIZA CONSULTAS
 - 6.- OBTIENE REP
 - 7.- MANTENIMIEN
 - 8.- SALIR DEL S
- 1.- OBTIENE REPORTES DE VENTAS
 - 2.- OBTIENE REPORTES DE COBRANZA
 - 3.- OBTIENE
 - 4.- REGRE
- 1.- RELACION DE VENTAS
 - 2.- RELACION DE PAGOS
 - 3.- COMISIONES A PAGAR
 - 4.- RELACION DE NOTAS DE CREDITO
 - 5.- REGRESA A MENU DE REPORTES

Para poder obtener cualesquiera de los reportes que aparecen en el menú de ventas, es necesario proporcionar la fecha hasta la cual se desea el reporte, es decir, que el sistema asume que se desea obtener el reporte de los movimientos realizados desde el primer día hábil del mes hasta el día de la fecha proporcionada.

CCRP6130.PRG
S. CUENTAS POR COBRAR
MENÚ DE REPORTES

REPORTE DE COMISIONES A PAGAR

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:16:02

LA FECHA DEL REPORTE ES: 6 DE NOVIEMBRE DE 1990

COMENTARIO:

<<PREPARE SU IMPRESORA>>

PARA CONTINUAR TECLEE <S/N>

6.2 .- OBTIENE REPORTES DE CUENTAS POR COBRAR

Con este menú el usuario puede elegir de entre los diferentes reportes de cuentas por cobrar; en este rubro se contempla a los estados de cuenta de los clientes, su antigüedad de saldos y un reporte de análisis de sus movimientos.

COMING200.PRG
S.CUENTAS POR COBRAR
REPORTES DIVERSOS

REPORTES DE CUENTAS POR COBRAR

NOMBRE DE LA EMPRESA:
FECHA: 90.11.26
HORA: 20:16:51

1.- FACTURACION
2.- MOVIMIENTOS ADICIONALES
3.- CIERRE DIARIO.
4.- CIERRE MENSUAL
5.- REALIZA CONSULTAS
6.- OBTIENE REP

7.- MANTENIMIEN
8.- SALIR DEL S
1.- OBTIENE REPORTES DE VENTAS
2.- OBTIENE REPORTES DE COBRANZA
3.- OBTIE

4.- REGRE
1.- ESTADOS DE CUENTA
2.- ANTIGÜEDAD DE SALDOS
3.- ANALISIS DE DOCUMENTOS
4.- REGRESA A MENU DE REPORTES

La condición para obtener cualesquiera de los reportes que aparecen en el menú de cuentas por cobrar, solo es necesario proporcionar el código o rango de códigos del o los clientes de los que se desea obtener el reporte. El sistema validará si el o los códigos proporcionados son códigos válidos.

CCRP6230.PRG
S. CUENTAS POR COBRAR
MENÚ DE REPORTES

ANÁLISIS DE DOCUMENTOS

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:18:21

CLIENTE INICIAL : 01029 ARCOOTMA,S.A.

CLIENTE FINAL : 24363 FEDERICO LOPEZ MENDOZA

COMENTARIO :

<<PREPARE SU IMPRESORA>>

PARA CONTINUAR TECLÉE <S/N>

6.3 .- OBTIENE REPORTES ESTADISTICOS

El último tipo de reportes que el sistema permite obtener son los reportes estadísticos. En este tipo de reportes se contempla un ABC de clientes, un ABC de artículos y un reporte de clientes por agente de ventas. Se denomina como reporte ABC a todo aquel reporte que presenta a las entidades objeto del mismo en orden descendente en importancia, dentro de la actividad a que se hace referencia. Por ejemplo, en el ABC de clientes, aparecerá en primer lugar el cliente que más compras haya realizado durante el rango de fechas especificado.

CCMNS00.FRG
S. CUENTAS POR COBRAR
REPORTES DIVERSOS

MENU DE REPORTES ESTADISTICOS

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:18:48

- 1.- FACTURACION
 - 2.- MOVIMIENTOS ADICIONALES
 - 3.- CIERRE DIARIO
 - 4.- CIERRE MENSUAL
 - 5.- REALIZA CONSULTAS
 - 6.- OBTIENE REP
 - 7.- MANTENIMIE
 - 8.- SALIR DEL S
- 1.- OBTIENE REPORTES DE VENTAS
 - 2.- OBTIENE REPORTES DE COBRANZA
 - 3.- OBTIE
 - 4.- REGRE
- 1.- ABC DE CLIENTES
 - 2.- ABC DE ARTICULOS
 - 3.- CLIENTES POR VENDEDOR
 - 4.- REGRESA A MENU DE REPORTES

De forma similar al menú de reportes de ventas, para poder obtener alguno de los reportes estadísticos es necesario proporcionar la fecha hasta la cual se desea obtener el reporte.

El sistema supondrá que la fecha de inicio es el primer día del mes. En el apéndice 'A' se presenta el formato de todos los reportes que el sistema permite generar.

CCRF6300
S. CUENTAS POR COBRAR
MENÚ DE REPORTES

REPORTES DE CLIENTES POR AGENTE

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:19:58

AGENTE INICIAL : 05 OSCAR EDUARDO SURIGA ROEZ.

AGENTE FINAL : 07 FERNANDO MARTINEZ REYES

MES DEL REPORTE : 11

COMENTARIO :

<<PREPARE SU IMPRESORA>>

— PARA CONTINUAR TECLÉE <S/N> —

7.0 .- MANTENIMIENTO A CATALOGOS

Mediante este proceso es factible realizar las ALTAS, BAJAS y CAMBIOS a los catálogos de clientes y de artículos, para lo cual se utiliza el siguiente menú:

CCM7000.PRG
S.CUENTAS POR COBRAR
MENU PRINCIPAL

MANTENIMIENTO DE CATALOGOS

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:20:39

1.- FACTURACION
2.- MOVIMIENTOS ADICIONALES
3.- CIERRE DIARIO
4.- CIERRE MENSUAL
5.- REALIZA CONSULTAS
6.- OBTIENE REP
7.- MANTENIMIENTOS
8.- SALIR DEL S

1.- CATALOGO DE CLIENTES
2.- CATALOGO DE ARTICULOS
3.- REGRESA A MENU PRINCIPAL

Si se selecciona la opción del mantenimiento a clientes, a continuación el sistema preguntará por el código del cliente con el que se desea trabajar, el cual deberá ser un número de cinco dígitos. Una vez introducido el código, el sistema procederá a buscar si ya existe el cliente en el catálogo, de ser así, presentará en pantalla los datos del cliente para que puedan ser actualizados, como puede observarse.

Una vez capturados o corregidos los datos y solo en el caso de clientes que ya existían en catálogo, el sistema preguntará al usuario si desea dar de baja al cliente en cuestión, para lo cual deberá proporcionarse una clave que sólo un usuario autorizado conocerá.

CCNT2100.PRG
S. CUENTAS POR COBRAR
MTO. DE CATALOGOS

CATALOGO DE CLIENTES

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:20:58

CODIGO : 24363
RAZON SOCIAL : FEDERICO LOPEZ MENDOZA
NOMBRE :
DIRECCION : H COLEGIO MILITAR # 2031
COLUMIA : LA ESCOMBIDA
POBLACION : CUALIACAN, SIN. CODIGO POSTAL : 43620
TELEFONO1 : TELEFONO2 :
AGENTE : OS OSCAR EDUARDO SURIGA HOEZ. RUTA : 01
CONDICIONES : 30 30 DIAS
LIM. CREDITO : 0.00
TRANSPORTE :
FECHA ALTA :

Datos correctos. Para continuar oprima cualquier tecla

El mantenimiento a artículos, se realiza de forma totalmente similar al mantenimiento del catálogo de clientes, por lo que únicamente se presenta la pantalla que se utiliza para ese fin.

CCHT2400.PRG
S.CUENTAS POR COBRAR
MTO. DE CATALOGOS

CATALOGO DE ARTICULOS

NOMBRE DE LA EMPRESA
FECHA: 90.11.26
HORA: 20:21:44

CÓDIGO : 01001

DESCRIPCION 1 : AC. ES. ALCARAVEA

DESCRIPCION 2 :

DESCRIPCION 3 :

DESCRIPCION 4 :

UNIDAD : KGS

CONCLUSIONES

En base al material examinado para la elaboración de esta tesis, puedo concluir que hoy en día no es justificable el fracaso de ningún proyecto de software, ya que como se expuso en la misma, actualmente existe la metodología necesaria para desarrollar productos de programación con un alto grado de disciplina, obteniendo por ende, una gran formalidad y confiabilidad en los productos obtenidos.

También se pudo constatar, que parte de la problemática actual del desarrollo de sistemas se deriva de la práctica común de intentar extrapolar las experiencias personales con pequeños programas, hacia grandes proyectos de programación de sistemas, ya que como es obvio, la complejidad de un sistema de software muy grande sobrepasa la comprensión de un único individuo.

Por otra parte podemos concluir que un proyecto de programación correctamente elaborado, dará por resultado un producto que requiera un mínimo de mantenimiento, proporcionando con ello un mayor rendimiento y un gran beneficio económico.

Actualmente se habla de herramientas como CASE (Ingeniería de Software auxiliada por computadora) y OOP (Programación orientada a objetos). Estas herramientas tienden a elevar la productividad del grupo de desarrollo, y con ellas, el concepto de programador se cambia por el de analista experto. Sin embargo, no hay que perder de vista que de ninguna manera vienen a suplantar al desarrollador, ya que finalmente su objetivo es el de automatizar la metodología que se ha descrito en este trabajo, es decir solamente están ayudando a aplicar correctamente la metodología en todas sus fases, comenzando desde el modelaje de los datos, llegando incluso al diseño y la instrumentación.

Es recomendable utilizar las herramientas antes mencionadas, ya que con ellas se aminora el costo y el tiempo de desarrollo, y como se mencionó al principio de la tesis, son factores que pueden llegar a determinar incluso la realización ó no de un proyecto.

En lo personal, puedo afirmar que desarrollar esta tesis, si me ayudo, ya que pude aplicar la metodología analizada a un problema real en el que, como es lógico suponer, se van a encontrar muchas problemas de diferente índole, problemas que van dando experiencia.

Como todo sistema de información, el sistema de cuentas por cobrar presenta algunas desventajas, como es el hecho de que el usuario no puede elaborar reportes a necesidades diferentes a las que ya contempla el sistema, sin embargo, si es posible realizar consultas no planeadas dentro del intérprete del dBase, con lo que este aspecto quedaría parcialmente solucionado. Una sugerencia sería elaborar un generador de reportes para solucionar totalmente esta deficiencia.

Cabe mencionar que el problema más frecuente que se presentó en el desarrollo del sistema fue el entendimiento con el usuario, ya que algunos aspectos se habían considerado de forma diferente a sus necesidades originales, con la consecuente pérdida de tiempo.

Actualmente el sistema de cuentas por cobrar se utiliza en algunas empresas, en las que se puede afirmar que ha dado buenos resultados, ya que no se han presentado problemas de gravedad, como podrían ser problemas de análisis o de inconsistencia en la información.

APENDICE 'A'

FORMATO DE LOS REPORTES DEL SISTEMA

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

19:50:27

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 1

REPORTE DIARIO DE VENTAS:

CORP6110.FRG

FACTURA	FECHA	CLIENTE	NOMBRE	AGENTE	IMPORTE	I.V.A.	TOTAL
136621	90.11.05	01053	BEBIDAS S.A. DE C.V.	10	672,000.00	100,800.00	772,800.00
136622	90.11.05	01189	ESENCIAS DE MEX. S.A. DE C.V.	10	75,900.00	11,385.00	87,285.00
136623	90.11.05	01322	INDUSTRIAL DEL VALLE, S.A.	10	200,000.00	30,000.00	230,000.00
136624	90.11.05	01813	DULCES, S.A. DE C.V.	07	244,830.00	36,724.00	281,554.00
136625	90.11.05	26274	FELIFE FALCON	05	500,000.00	75,000.00	575,000.00
136626	90.11.05	18258	MAURICIO GONZALES	10	139,000.00	20,850.00	159,850.00
136627	90.11.05	25019	ROBERTO GARCIA	10	996,260.00	149,439.00	1,145,699.00
136628	90.11.05	29648	JESUS V. BENITEZ	04	609,000.00	91,200.00	699,200.00
136629	90.11.05	01195	CHOCOLATES, S.A.	07	5,814,000.00	872,100.00	6,686,100.00
136630	90.11.05	01782	JAIIME ESCOBERO AMORES	07	30,000.00	4,500.00	34,500.00
136631	90.11.05	01517	MANUFACTURERA ERAVO, S.A.	07	1,476,500.00	221,475.00	1,697,975.00
136632	90.11.05	01150	EMPACADORA, S.A. DE C.V.	07	3,153,440.00	473,016.00	3,626,456.00
136633	90.11.05	21406	TEL'S DE MEXICO, S.A.	07	4,105,100.00	616,265.00	4,721,365.00
136634	90.11.05	01462	MACHURAS, S.A.	08	1,440,000.00	216,000.00	1,656,000.00
136635	90.11.05	29352	INDUSTRIAS UNIDAS, S.A. DE C.V.	06	3,445,000.00	516,750.00	3,961,750.00
136636	90.11.05	01922	EXCLUSIVAS BUENO, S.A. DE C.V.	07	650,000.00	97,500.00	747,500.00
136637	90.11.05	18257	BUAU, S.A. DE C.V.	10	1,375,000.00	206,250.00	1,581,250.00
136638	90.11.05	30715	JUGOS Y CONCENTRADOS, SA. DE CV.	10	155,300.00	23,295.00	178,595.00
136639	90.11.05	30715	JUGOS Y CONCENTRADOS, SA. DE CV.	10	114,000.00	17,100.00	131,100.00
136640	90.11.05	30715	JUGOS Y CONCENTRADOS, SA. DE CV.	10	19,400.00	2,910.00	22,310.00
136641	90.11.05	24363	FEDERICO LOPEZ MEXCOCA	05	644,000.00	96,600.00	740,600.00
					25,661,730.00	3,678,259.00	29,740,989.00
T O T A L E S :					25,661,730.00	3,678,259.00	29,740,989.00
TOTAL DE VENTAS CON TASA 15% I.V.A. : \$					25,661,730.00		
TOTAL DE VENTAS CON TASA 0% I.V.A. : \$						0.00	

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

19:51:09

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 1

REPORTE DIARIO DE PAGOS DEL DIA 1 DE NOVIEMBRE DE 1990

CORF6120.PRG

000160 CLIENTE	NOMBRE	CONCEPTO	FACTURA	FECHA EMISION	FECHA VENCIMIENTO	IMPORTE
24363	FEDERICO LOPEZ MENDOZA	220	137381	90.05.07	90.05.06	796,550.00
24641	QUIMICA ARDE, S.A. DE C.V.	220	136984	90.08.14	90.05.13	18,027,400.00
24631	MA. TERESA DEL CASTILLO LOPEZ	220	138169	90.03.15	90.09.14	1,716,950.00
24370	JOSE LUIS JARAS	220	136109	90.06.15	90.09.14	1,500,000.00
24706	ROSENDO SUARE, S.A. DE C.V.	220	138127	90.06.17	90.09.16	3,393,950.00
30180	SIGRA, S.A.	220	138224	90.05.04	90.10.04	1,322,500.00
01650	F.H. SOLUCIONES S.A.	220	138509	90.05.13	90.09.28	1,195,425.00
07754	SUPER MERCADO S.A. DE C.V.	220				3,187,800.00
86698	VENTAS MOSTRADOR	235	32581			138,000.00
86563	VENTAS MOSTRADOR	235	32584			51,750.00
86883	VENTAS MOSTRADOR	235	32586			18,170.00
86388	VENTAS MOSTRADOR	235	32587			130,000.00
86688	VENTAS MOSTRADOR	235	32589			1,920,500.00
86988	VENTAS MOSTRADOR	235	32589			37,376.00

33,046,771.00

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

19:52:20

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 3

REPORTE DE COMISIONES A PAGAR

CORP6130.PRG

FECHA	FACTURA	NOTA DE CREDITO	CLIENTE	NOMBRE	IMPORTE DOCUMENTO	COMISION (%)	IMPORTE COMISION
AGENTE: FERNANDO MARTINEZ REYES							
30.11.05	138624		01813	DULCES, S.A. DE C.V.	244,850.00	3	7,345.00
30.11.05	138629		01196	CHOCOLATES, S.A.	5,814,000.00	3	174,420.00
30.11.05	138630		01782	JAIME ESCOBERO AMOFES	30,000.00	3	900.00
30.11.05	138631		01317	MANUFACTURERA BRAVO, S.A.	1,476,500.00	3	44,295.00
30.11.05	138632		01180	EMPAQUETA-1, S.A. DE C.V.	3,155,440.00	3	94,663.00
30.11.05	138633		21493	KEL'S DE MEXICO, S.A.	4,199,100.00	3	123,273.00
30.11.05	138638		01322	EXCLUSIVAS SIENGO, S.A. DE C.V.	650,600.00	3	19,500.00
					<u>15,477,870.00</u>		<u>464,396.00</u>

TOTAL DE COMISIONES A PAGAR : 464,396.00

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

19:54:10

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 1

REPORTE DE ANTIQUEDAD DE SALDOS DEL CLIENTE 24363 AL 24363

CCRP6220.PRG

CLIENTE POBLACION	NOMBRE	FACTURA	FECHA EMISION	SALDO CORRIENTE	SALDOS A 30 DIAS	VENCIDOS A 60 DIAS	SALDOS A 90 DIAS
24363	FEDERICO LOPEZ MENDOZA CULTACAN, SIN.						
			AGENTE: 05 OSCAR EDUARDO SURIGA HDEZ. 198641 90.11.05	740,600.00			
TOTALES POR CLIENTE:				740,600.00	0.00	0.00	0.00
SALDO TOTAL :					TOTAL DE SALDOS VENCIDOS :		0.00
T O T A L E S :				740,600.00	0.00	0.00	0.00
SALDO TOTAL :				740,600.00	TOTAL DE SALDOS VENCIDOS :		0.00

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

19:54:37

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 1

ANALISIS DE DOCUMENTOS

CCRP6230.PRG

FECHA	CONCEPTO	FECHA VENCIMIENTO	CARGOS	ABONOS	SALDO
CLIENTE : 24363	FEDERICO LOPEZ MENDOZA		FACTURA : 137361	SALDO : \$	0.00
90.05.97	110 FACTURACION	90.06.06	796,350.00		796,350.00
90.11.01	220 PAGO A DOCUMENTO			796,350.00	0.00
CLIENTE : 24363	FEDERICO LOPEZ MENDOZA		FACTURA : 136641	SALDO : \$	740,600.00
90.11.05	110 FACTURACION	90.12.05	740,600.00		740,600.00

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

20:58:38

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 1

REC MENSUAL DE CLIENTES

CORP6310

CLIENTE	NOMBRE	AGTE.	FACTURADO	(%)
01198	CHOCOLATES, S.A.	07	\$ 5,814,000.00	22.48
21468	KEL'S DE MEXICO,S.A.	07	\$ 4,109,100.00	15.99
29382	INDUSTRIAS UNIDAS, S.A. DE C.V.	08	\$ 3,445,000.00	13.32
01180	EMPACADORA, S.A. DE C.V.	07	\$ 3,153,440.00	12.19
01317	MAQUINARIA ERAND, S.A.	07	\$ 1,478,500.00	5.71
01482	MACHUANG, S.A.	02	\$ 1,440,000.00	5.57
18287	SUAV, S.A. DE C.V.	10	\$ 1,375,000.00	5.32
29019	ROGELIO CORONA	10	\$ 996,260.00	3.85
01083	BEBIDAS S.A. DE C.V.	10	\$ 872,000.00	3.30
01922	EXCLUSIVAS BUENO, S.A. DE C.V.	07	\$ 850,000.00	3.21
24563	FEDERICO LOPEZ HERRERA	05	\$ 844,000.00	3.24
28548	JESUS M. BENITEZ	04	\$ 608,000.00	2.35
28274	FELIPE PALCOA	05	\$ 500,000.00	1.93
30715	AGROS Y CONCENTRADOS SA DE CV	09	\$ 288,700.00	1.12
01815	VALCE, S.A. DE C.V.	07	\$ 244,800.00	0.95
01322	INDUSTRIAL DEL VALLE, S.A.	10	\$ 200,000.00	0.77
10258	MARCELINO GONZALEZ	10	\$ 138,000.00	0.54
01189	ESENCIAS DE MEX, S.A. DE C.V.	07	\$ 75,500.00	0.29
01782	DAINE ESCOBEDO AMARIS	07	\$ 30,000.00	0.12

.....

TOTALES : \$ 25,861,720.00 100

.....

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

19:56:25

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 1

REC MENSUAL DE ARTICULOS

CORRESPON

CODIGO	NOMBRE	FACTURADO	(%)	ACUM. ANUAL
10018	AC. ES. CAJELA CJ	\$ 5,014,000.00	22.49	33,932,350.00
05000	COLOR AMARILLO #5	\$ 5,256,100.00	20.14	51,672,576.00
73049	COLOR CHOCOLATE #149-A	\$ 2,550,900.00	9.86	20,399,547.00
73049	COLOR ROJO PURPURA #4263 (BUFFALO)	\$ 1,575,440.00	6.10	5,155,340.00
73070	COLOR ROJO P/CATSUP #196 (BUFFALO)	\$ 1,575,000.00	6.09	17,273,799.00
10009	AC. ES. LIMON FRIO PAIS	\$ 1,443,000.00	5.57	43,431,057.00
50017	SABOR VAINILLA POLVO GL	\$ 1,375,000.00	5.12	13,102,500.00
22014	ES. MENTA E	\$ 1,250,000.00	4.85	3,100,000.00
21006	ES. NARANJA EXTRA FUERTE S	\$ 850,000.00	3.51	1,575,200.00
50075	CONC. FRESA #426	\$ 804,000.00	3.15	1,205,000.00
51001	CONC. COCO G	\$ 500,000.00	1.91	23,233,245.00
05001	COLOR AMARILLO #6	\$ 290,000.00	1.51	22,667,223.00
09056	COLOR CARAMELO	\$ 332,000.00	1.29	23,370,518.00
06910	ACIDO SALICILICO	\$ 310,260.00	1.20	11,754,522.00
25035	ES. CERVEZA DE HIERRO SR	\$ 276,000.00	1.07	1,585,505.00
70024	COLOR VIOLETA #1456	\$ 245,000.00	0.95	1,236,873.00
25127	ES. TAMARINDO # 212-M	\$ 244,950.00	0.95	1,417,300.00
20000	ES. ANELLAMA GL	\$ 200,000.00	0.77	200,000.00
50017	CONC. LIMON #4 C/DOLOR	\$ 178,500.00	0.69	3,245,400.00
70022	COLOR VERDE EMERALDA #1643	\$ 158,500.00	0.61	17,402,141.00
05002	COLOR AZUL #1	\$ 155,000.00	0.60	3,325,350.00
51010	CONC. LIMON G	\$ 140,000.00	0.54	1,147,150.00
02085	EUCALIFTOLO	\$ 133,000.00	0.51	2,795,000.00
01031	AC. ES. LIMON MEX. CENTRAF	\$ 114,000.00	0.44	10,355,230.00
55025	CONC. IVA #12	\$ 112,000.00	0.43	6,272,000.00
50029	CONC. PINA #12	\$ 84,000.00	0.32	1,218,000.00
05000	AMARILLO AL ACEITE G-2 100% USO IND	\$ 75,000.00	0.29	3,185,400.00
50018	CONC. NARANJINA #4	\$ 66,500.00	0.26	4,227,500.00
21003	ES. LIMON EXTRA FUERTE	\$ 60,000.00	0.23	2,260,000.00
09051	JUGO DE MANZANA	\$ 50,000.00	0.19	30,000.00
02149	TERPINEOL EXTRA	\$ 19,400.00	0.08	74,200.00

=====

TOTALES : \$ 25,861,730.00 100 466,670,739.00

=====

26 DE NOVIEMBRE DE 1990

NOMBRE DE LA EMPRESA

19:58:20

SISTEMA DE CUENTAS POR COBRAR

PAGINA: 2

REPORTE DE CLIENTES POR AGENTE

00P5390

000100	CLIENTE		VENTAS DEL MES	VENTAS ANUALES
AGENTE : 07 FERNANDO MARTINEZ REYES				
01529	NACIONAL DE JARASES, S.A.	\$	0.00	\$ 4,500,000.00
01484	SILVIA MARRIENDA VALLE	\$	0.00	\$ 15,450,000.00
01461	LABORATORIOS, S.A.	\$	0.00	\$ 26,050,070.00
01455	SISTEMAS, S.A. DE C.V.	\$	0.00	\$ 2,071,500.00
01447	OTIMA ESPECIAL, S.A.	\$	0.00	\$ 1,024,400.00
01441	TRATTO S.SAL. C.V.	\$	0.00	\$ 3,000,000.00
01350	TALACHEO, S.A.	\$	0.00	\$ 1,720,000.00
01304	FABRICANTES ESP. SA DE C.V.	\$	0.00	\$ 56,000,000.00
01255	HERNANDEZ VAJBE S.A.	\$	0.00	\$ 11,447,000.00
01246	SELATONAS SA DE CV	\$	0.00	\$ 25,125,000.00
01210	PARMITA, S.A.	\$	0.00	\$ 13,000,000.00
01150	COSMOS, S.A.	\$	0.00	\$ 217,000.00
01140	DISTRIBUIDORA, S.A.	\$	0.00	\$ 12,353,000.00
01101	DAN MEXICO, S.A.	\$	0.00	\$ 7,711,000.00
01105	EDMOS, S.A.	\$	0.00	\$ 276,000.00
01010	ERLEN S.A.	\$	0.00	\$ 2,440,000.00
01105	COLORES Y SABORES, S.A.	\$	0.00	\$ 1,007,000.00
01102	CIGARROS, S.A. DE C.V.	\$	0.00	\$ 11,020,470.00
01099	DISTILADORA, S.A.	\$	0.00	\$ 240,000.00
01055	CENTRAL DE MEXICO SA CV	\$	0.00	\$ 7,000,000.00
01040	INDUSTRIAS ALIMENTICIAS ARTS, S.A. DE C.V.	\$	0.00	\$ 10,054,400.00
01002	ALIM.MEXICANOS, S.A. DE C.V.	\$	0.00	\$ 3,761,150.00
01000	APLICACIONES, S.A.	\$	0.00	\$ 1,250,000.00
01005	ACEITES, S.A. DE C.V.	\$	0.00	\$ 1,320,000.00
TOTALES POR AGENTE :		\$	10,560,770.00	\$ 1,205,574,070.00

.....

TOTALES GLOBALES : \$ 10,560,770.00 \$ 1,205,574,070.00

.....

BIBLIOGRAFIA

- Award M. Elias System Analysis and Design
Richard D. Irwing Inc. 1979.
- Boehm, B. (80551) Software Engineering Economics
Prentice-Hall 1981.
- Dijkstra, E W Notes on Structured Programming
1972.
- Edward Jones Aplicando dBase III
Ed. McGraw Hill 1987.
- Marvin V. Zelkowitz Perspectives on Software Engineering
R.W Taylor & R.L.Frank
- R.W Taylor & R.L.Frank CODASYL, data base management systems
WCM Comput. 1976.
- Richard Fairley Ingenieria de Software
Ed. McGraw Hill 1988.
- Shakuntala, Atre Técnicas de Bases de Datos
John Wiley & Sons, Inc. - Trillas 1988.
- Study Group on Data ANSI/X3, SPARC : Interim Report
Base Management System ANSI February 1975.
- T. DeMarco Structured Systems and Systems Specification
Yourdon Press, 1978.

Yourdon N. Edward

Structured Analysis/Design Workshop

Edition 7.2 1981.

Yourdon N. Edward

The second Structure Revolution

Yourdon Press 1979.