



UNIVERSIDAD NACIONAL
AUTÓNOMA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DESARROLLO DE UN SISTEMA DE CONTABILIDAD
PARA PC
USANDO DBASEIII

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACION
PRESENTA:
INGRID MORGAN SOTOMAYOR

DIRECTOR DE TESIS
ING. SEBASTIAN POBLANO ORDONEZ

MEXICO, D.F.

1987.



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

INTRODUCCION	1
CAPITULO I	
INGENIERIA DE PROGRAMACION	5
CAPITULO II	
PLANEACION DEL SISTEMA	12
- Análisis de Necesidades	12
- Análisis de Factibilidad	13
- Descripción del problema	14
- Alternativas de Solución	18
- Beneficios	18
CAPITULO III	
ANALISIS DE REQUERIMIENTOS	23
- Requerimientos	24
- Diagrama de Flujo de datos	25
- Diccionario de Datos	56
- Miniespecificaciones	58
- Tablas de decisión	84
CAPITULO IV	
DISEÑO DEL SISTEMA	86
- Cualidades de un buen diseño	89
- Diagrama de Estructura	101
- Selección del lenguaje de Programación	102
- Diseño de la Base de Datos	108

CAPITULO V

IMPLEMENTACION	125
- Codificación	125
- Documentación	132

CAPITULO VI

INTEGRACION Y PRUEBAS	132
- Integración	132
- Pruebas	135

CAPITULO VII

PUESTA EN OPERACION	139
- Manual de Usuario	140

CAPITULO VIII

MANTENIMIENTO	167
---------------------	-----

APENDICE A :Descripción del equipo	175
--	-----

BIBLIOGRAFIA	185
--------------------	-----

INDICE DE DIAGRAMAS

DIAGRAMAS DE FLUJO DE DATOS

DFD 0. "SISTEMA DE CONTABILIDAD"	43
DFD 1. MANTENIMIENTO	44
DFD 1.1 ALTAS	45
DFD 1.2 BAJAS	45
DFD 1.3 CAMBIOS	46
DFD 2. CAPTURA Y/O DIAGNOSTICO	47
DFD 3. CORRECCION DE POLIZAS	48
DFD 3.1 INSERTA REGISTROS	49
DFD 3.2 MODIFICA REGISTRO	50
DFD 3.3 ELIMINA REGISTRO	50
DFD 3.4 ELIMINA POLIZA	50
DFD 4. REPORTES	51
DFD 4.1 REPORTE NUMERICO	52
DFD 4.2 REPORTE ALFABETICO	52
DFD 5. CIERRE DE PERIODO	53
DFD 6. ESTADOS DE RESULTADOS	54
DFD 6.1 BALANCE GENERAL	54
DFD 6.2 ANEXO AL BALANCE (AUXILIAR TRES)	54
DFD 7. CONSULTA	55
DIAGRAMA DE ESTRUCTURA	101
DIAGRAMA DE ENTER-RELACION	123

I N T R O D U C C I O N

La era actual se caracteriza por el alto volumen de población y por ende se han desarrollado muchos procedimientos para el manejo de altos volúmenes de información, cobrando gran auge las técnicas computacionales que se han ido desarrollando a través del tiempo a una velocidad vertiginosa, obteniéndose excelentes resultados en cuanto a eficiencia, costo, ahorro de tiempo, oportunidad y confiabilidad de la información.

Dentro de las aplicaciones más importantes de las técnicas computacionales para el manejo de información, se encuentran las enfocadas a la administración, rama que ha cobrado gran importancia en centros de trabajo donde la cantidad de empleados, gastos, y en general actividades diarias es excesiva para controlarlas de manera manual.

La Universidad Nacional Autónoma de México de ninguna manera ha quedado al margen del avance de la tecnología y atendiendo al gran crecimiento que ha sufrido nuestra máxima casa de estudios, se ha tenido la necesidad de automatizar todos los procesos administrativos que ahí se realizan, es así como a través de la Dirección General de Servicios de Cómputo para la Administración, se han desarrollado diferentes sistemas computacionales, con los cuales se pretende que todas y cada una de las dependencias universitarias realicen sus procesos administrativos usando la computadora, con este fin la universidad ha adquirido una gran cantidad de microcomputadoras tipo PC, para la realización de estos procesos.

En general el propósito de los sistemas administrativos

reunidos en lo que se llama "Paquete Administrativo" ,es que cada dependencia realice los procesos contables ,presupuestales,de almacén y de control de personal de manera independiente ,pero siempre bajo los lineamientos que establezca la UNAM para la realización de dichos procesos.

La tesis que se presenta bajo el nombre de DESARROLLO DE UN SISTEMA DE CONTABILIDAD PARA MICROCOMPUTADORA PC USANDO DBASEIII,fue realizada en el departamento de Asesoría e Implantación de Sistemas de la Dirección General de Servicios de Cómputo para la Admon. como un proyecto requerido por varias dependencias universitarias que les permita contar con un sistema para el control de sus procesos contables,haciendo uso de una microcomputadora PC de reciente adquisición.

La obra presentada esta formada por ocho capítulos y un apéndice.

A lo largo de todos los capitulos se presenta la forma en que se fué desarrollando el sistema de Contabilidad PC,desde su nacimiento hasta su liberación,todo esto bajo la metodología del ciclo de vida de un sistema presentada por la compañía Yourdon Inc.

En el primer capítulo se presentan algunos aspectos sobre lo que es la Ingeniería de Software y la razón de la necesidad de usar técnicas especializadas para la realización de sistemas ,además se realiza una explicación de la técnica utilizada para el desarrollo de nuestro problema.

En el segundo capítulo se presenta el desarrollo de la etapa de Planeación del Sistema,la cual nos permitirá decidir si

debemos realizar el proyecto o abandonarlo por incosteable.

La siguiente fase del desarrollo del sistema de Contabilidad se presenta en el capítulo tres ,esta etapa es la conocida como Análisis Estructurado en el cual se hace un estudio minucioso del problema a resolver para especificar "Que" hacer para encontrar la mejor solución al problema.

En el capítulo cuatro se realiza la presentación de la etapa correspondiente al Diseño del Sistema,empezando por definir todas las herramientas a considerar para obtener un buen diseño.

Después de haber definido las bases teóricas ,se aplican éstas al problema real.Además como parte complementaria al diseño se especifica la selección del lenguaje de programación a utilizar,y como en este caso se optó por el uso del administrador de bases de datos DBASEIII,se especifican los conceptos más importantes de Bases de datos para proceder al diseño de ésta.

El capítulo cinco es el correspondiente a la Implementación del sistema donde se especifica la codificación realizada para el sistema de Contabilidad PC.

En el capítulo seis se presenta la etapa del sistema correspondiente a la Integración y a las Pruebas,definiéndose las técnicas de integración y la manera en que fue aplicada la técnica elegida al problema real,asi también se especifican los tipos de prueba que existen y técnicas para la preparación de éstas.

El capítulo siete nos muestra la forma de Operación del sistema a través del manual del usuario integrado en este mismo

capítulo.

Para finalizar con el desarrollo del sistema, el capítulo ocho indica que tipos de mantenimiento existen y como es que se debe de mantener el sistema de contabilidad PC ya liberado.

Además se anexa un Apéndice ,donde se describe las características del equipo utilizado.

Para finalizar con esta obra se presenta la bibliografía que fue requerida a lo largo del trabajo que ahora se presenta como tesis.

Espero que el desarrollo de la obra sea satisfactorio al lector y, sobre todo pueda representar una guía para la realización de sistemas de cómputo con el fin de que estos cumplan su función de manera óptima.

Por otra parte el desarrollo de esta tesis se presenta como un esfuerzo para servir a mi Universidad , a la que debo mi formación profesional y con la que siempre me sentiré comprometida a servir y apoyar para el logro de su superación .

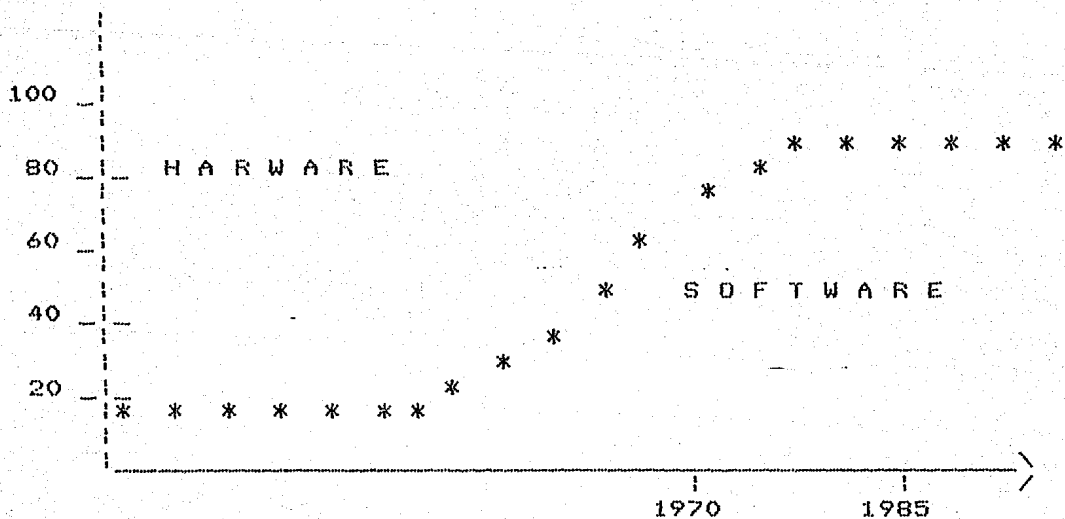
C A P I T U L O I

I N G E N I E R I A - D E P R O G R A M A C I O N

La ingeniería de Programación es una disciplina que había permanecido oculta hasta finales de los sesentas y se desarrolló como una respuesta a la evolución de los sistemas y por la necesidad de disminuir el costo del software, el cual se ha ido incrementado dando como resultado la ya conocida crisis del software .

Esta crisis es producto de la mala realización de los sistemas computacionales , los cuales son difíciles de mantener.

La primera parte de la crisis del software la constituye la comparación de costos entre el hardware y el software , que como se ilustra en el siguiente esquema, el software a través del tiempo ha incrementado su costo considerablemente , mientras que el hardware lo ha disminuido . .



La experiencia en la rama de software de los últimos 10 a 15

años presenta un avance muy rápido en el desarrollo de tecnología para hardware, y una demanda de requerimientos de sistemas de software más sofisticado, dando como resultado exigencias en el desarrollo de metodologías y en una educación más adecuada para los ingenieros de software.

Es gracias a las metodologías desarrolladas que las exigencias de software pudieron ser atendidas, aunque no totalmente, fue hasta principios de los setentas que la tecnología en el desarrollo de software prosperó grandemente.

En la actualidad se puede afirmar que todavía existen muchos retos en ésta área y aunque en realidad existe un progreso considerable, aún se exigen más profesionales en la rama de ingeniería de software, más herramientas y métodos más efectivos.

Se han establecido diversidad de definiciones de la Ingeniería de Software y una de las más acertadas es la que surge en 1972 por F L Bauer de la Universidad de Munich Alemania, el cual define: "la ingeniería de software es el establecimiento y uso de métodos de ingeniería con el fin de obtener software rentable y funcional".

En realidad el hecho de obtener software rentable y funcional implica muchas otras características que el software de alta calidad debe de cumplir y estas son las siguientes:

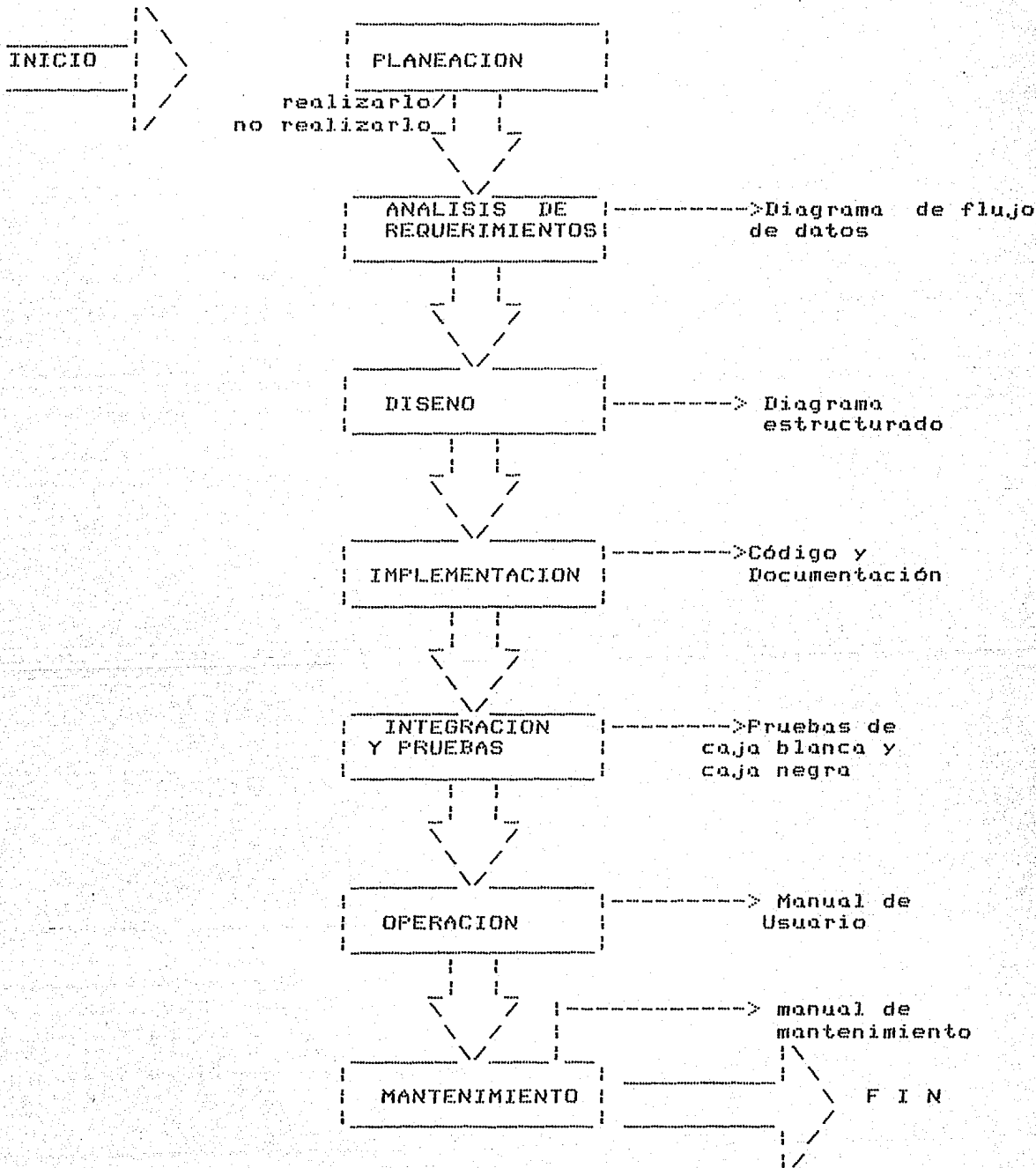
Software:

- Confiable : Que funcione de la manera en que fue conceptualizado.
- Eficiente : Se refiere a la utilización de los recursos de cómputo en forma adecuada.
- Eficaz : Es decir que satisfaga los

- requerimientos para el que fue realizado.
- Mantenable : Que al realizar la compactación y expansión del sistema no se generen problemas.
- Transportable : Realizar productos con un ambiente de cómputo común.

Es con el fin de obtener las características arriba mencionadas ,que se han ido desarrollando numerosas metodologías para el desarrollo de software, las cuales están basadas en principios de análisis y diseño estructurado .De todas estas metodologías (que en realidad son muy similares) se ha elegido para el desarrollo del sistema de Contabilidad Fc el que se basa en el "Ciclo de Vida del Sistema" y presenta siete estados o etapas las cuales deben de formar parte durante la vida de un sistema ,éstas fases engloban todo lo que es la vida de un sistema, desde su nacimiento hasta después de su liberación ya que se incluye su mantenimiento .

Los estados del "Ciclo de Vida de un Sistema" son esquematizados en el siguiente diagrama.



Es conveniente enfatizar que todas las etapas son importantes y que dependiendo de su buen desarrollo se podrá obtener un producto de alta calidad.

Si seguimos el esquema presentado podremos describir cada una de las etapas de la siguiente manera:

1.Planeación: es en esta etapa donde nace el sistema desde el momento en que se reconoce la existencia de éste, y es en esta etapa donde se establece claramente cual es el problema a resolver.

2.Análisis Estructurado: La etapa de Análisis Estructurado nos muestra una clara definición de "QUE" es lo que se tiene que realizar para resolver el problema presentado.

3.Diseño: Es en la etapa de diseño donde se define "COMO" es que vamos a realizar para resolver el problema ,lo que se logra a través de técnicas de Diseño Estructurado.

4.Implantación: La etapa de implantación comprende la codificación y documentación del sistema.

5.Integración:En la parte de integración y pruebas se deberán de integrar los módulos implementados para que queden todos integrados en un solo sistema ,además de realizarse las pruebas de éste,y pueda quedar listo para su liberación.

6.Operación: La etapa de operación es una fase del ciclo de vida de un sistema orientada hacia el usuario final,y comprende

la documentación necesaria para la operación del sistema.

7. Mantenimiento: Para finalizar la etapa de mantenimiento se desarrolla durante la vida de un sistema, siendo ésta la más costosa de todas las etapas del ciclo de vida, y dependiendo este costo del buen desarrollo y metodología utilizada en las etapas anteriores a ésta.

Entropía en el ciclo de vida de un sistema.

La Entropía es un fenómeno que juega un papel central en el desarrollo de un proceso .

El concepto de entropía es una de las contribuciones más elegantes a la ingeniería y se presenta en todos los procesos naturales y también es un fenómeno que sufre un sistema durante su ciclo de vida. Este fenómeno se define como la pérdida de estructuración que sufre un sistema durante su ciclo de vida , es por eso que se debe procurar que el porcentaje de entropía existente sea mínimo. En general un sistema se mantiene hasta un límite de un 60 % de entropía por lo que se debe poner mucha atención a la etapa de mantenimiento ya que los sistemas siguen las siguientes leyes de evolución :

Leyes de la Evolución dinámica de los Sistemas:

1) "Ley del cambio continuo "

Los sistemas que sufren muchos cambios son inmantenibles y tienen que rehacerse.

2) "Ley del incremento de entropía"

La entropía aumenta con el tiempo.

3) " Ley del crecimiento estáticamente plano"

En un período de tiempo disminuye la cantidad de

errores.

Por lo tanto el objetivo consistirá en minimizar el total de entropía integral durante el ciclo de vida, lo cual requerirá que cada fase este bien planeada y realizada.

Las causas mas generales de la aparición de entropía en un sistema son la mala comunicación entre las personas que desarrollan el sistema y el usuario, uso de una mala metodología para el desarrollo de cada etapa, mala planeación de tareas, documentos incompletos etc.

De esta manera se ha establecido un panorama muy general de lo que es la ingeniería de software ,lo que pretende esta rama de la ingeniería, y la metodología elegida para la aplicación de técnicas de ingeniería en el desarrollo del sistema de Contabilidad PC ,con el fin de obtener un sistema de alta calidad y alto rendimiento.

C A P I T U L O I I

P L A N E A C I O N D E L S I S T E M A

Dentro de lo que se conoce como "ciclo de vida " de un sistema, se encuentra como primera fase para el desarrollo , la etapa denominada "planeación del sistema", como resultado de esta fase se obtiene la decisión sobre si se continúa con la siguiente fase del proyecto o se abandona la realización por incosteable.

En la etapa de planeación se deben realizar las siguientes actividades:

- Análisis de Necesidades
- Análisis de Factibilidad

ANALISIS DE NECESIDADES.

Dentro de esta subetapa, se debe realizar un estudio del medio ambiente en el que se da el problema, al igual que los escollos o limitaciones que se presentan para su solución.

El problema en estudio esta referido al proceso contable para dependencias de la UNAM, por lo que a continuación se hace referencia al medio ambiente de dicho problema.

El medio en que se desenvuelve el problema, es un ambiente puramente administrativo, donde las personas que se hacen cargo del proceso de contabilidad son por lo general contadores, licenciados o administradores, que frecuentemente no estan familiarizados con el uso de la computadora. Por otra parte la gente que se dedica al trabajo contable, por lo general no solo trabaja para estas funciones, sino que tiene que realizar muchas

otras por lo que la premura de tiempo ocasiona muchos errores a través del proceso.

Por otra parte, es común en dependencias de la UNAM, que la gente cambie muy frecuentemente de actividades, por lo que se necesita una capacitación continua para la realización del proceso contable tal y como se realiza dentro de la universidad.

Por lo anterior se considera que sería muy necesario que este proceso se realizara de manera rápida, pero que además no sea muy complicado de aprender.

ANALISIS DE FACTIBILIDAD.

Como principio a esta subetapa se debe describir el problema y las limitaciones del mismo con el fin de poder definir claramente los requerimientos que se expresan dentro del planteamiento. Una vez definido el problema, se procede a dar alternativas para su solución, en base a estas alternativas se realiza el análisis de beneficios para contar con una base, de tal forma que sea posible comparar las alternativas propuestas y escoger la mejor de todas ellas.

Dentro del desarrollo de esta subetapa también es importante considerar lo que es el estudio de los recursos humanos y recursos de cómputo, al igual que lo concerniente al aspecto legal.

Descrito a grandes rasgos el procedimiento de análisis de factibilidad se presenta a continuación la aplicación de éste al problema real.

Descripción del problema.

El problema planteado surge a raíz de la importancia que tiene para cada dependencia de la UNAM llevar un control interno de sus procesos contables, este proceso se desenvuelve en la actualidad de manera manual en la mayoría de las dependencias, esto en algunos casos es complicado y consume tiempo de las personas destinadas a la actividad contable, es por esto que se busca que este proceso se realice de manera mas eficiente.

A continuación se describirá lo que se persigue a través de un proceso contable ,como es que se realiza y las particularidades para el caso de dependencias de la UNAM.

Para conocer la situación financiera de una institución es necesario formular los Estados Financieros constituidos por el Balance General y por el Estado de Pérdidas y Ganancias o Estado de Resultados .

Ahora bien para poder conocer los importes con los cuales se formulan dichos Estados es necesario llevar registros especiales donde se anoten todos los aumentos y disminuciones que sufren los diferentes valores o conceptos de Activo, Pasivo y Patrimonio como consecuencia de las operaciones practicadas por la dependencia. Cada uno de estos registros recibe el nombre de cuenta ,en el caso de la UNAM éstas son afectadas por el procedimiento de pólizas .

Las pólizas son documentos de carácter interno en los cuales se anotan las operaciones, detallada y ordenadamente.

Los datos para hacer los asientos en el formato de las

pólizas se toman de los documentos fuente, concluida la anotación en los registros auxiliares y en los mayores, las pólizas se deben archivar por numeración progresiva con objeto de poderlas localizar con facilidad.

Todos los movimientos realizados se registran en un libro de "diario" y con los datos de éste, cada fin de mes por lo regular se suman los cargos y abonos que afectaron a cada una de las cuentas, con los totales se forma un solo asiento de concentración que se registran en otro libro llamado Libro de Mayor y por último de éste se toman los datos para la formación de los Estados Financieros .

El rayado y la forma de las pólizas puede ser tan variado como se desee siempre que contenga por lo menos los siguientes datos :

1. Nombre de la dependencia
2. Número de póliza
3. Fecha de la operación
4. Cuentas y subcuentas de cargo y abono
5. Cantidades de cargo y abono
6. Redacción de la operación
8. Firmas

Este es a grandes rasgos el procedimiento de pólizas que siguen las dependencias de la UNAM y el cual es uno de los que cuenta con mayor aceptación en la actualidad.

Balance General.

El balance general es el documento contable que presenta la situación financiera, en este caso, de la dependencia en una fecha determinada.

El balance general debe contener los siguientes datos :

- | | |
|------------|--|
| Encabezado | <ol style="list-style-type: none">1. Nombre de la dependencia2. Indicación de que se trata de un balance general3. Fecha de formulación |
| Cuerpo | <ol style="list-style-type: none">1. Nombre y valor detallado de cada una de las cuentas que forma el activo2. Nombre y valor detallado de cada una de las cuentas que forman el pasivo3. Nombre y valor detallado de cada una de las cuentas que forman el patrimonio4. Importe del capital contable |

El balance general, para las dependencias universitarias se debe presentar en forma de reporte. Esta forma consiste en anotar clasificadamente el activo y el pasivo de tal manera que a la suma del activo se le debe restar verticalmente la suma del pasivo para determinar el capital contable.

Estado de resultados.

El Estado de resultados es un documento contable que muestra detallada y ordenadamente la forma en que se ha obtenido la utilidad o pérdida del ejercicio. Resulta un elemento de apoyo muy estimable para normar la futura política económica de la dependencia.

El estado de resultados principia anotando el encabezado, el cual debe contener los siguientes datos:

1. Nombre de la Institución
2. La indicación de ser Estado de Resultados
3. El período al que se refiere dicho estado

Después del encabezado se deben analizar todos los conceptos que comprende la compraventa de mercancías hasta determinar la utilidad bruta en la siguiente forma:

1. Se anotan los ingresos totales y, si hay devoluciones o rebajas sobre ventas, su importe se resta a dichas ventas para obtener así la utilidad neta.

2. A la utilidad neta se le resta el costo de ventas obteniéndose de esa forma la utilidad bruta.

3. Se obtiene la utilidad de operación restando a la bruta los gastos de operación y los gastos generales.

4. Obtenida la utilidad de operación se restan los gastos financieros y se suman los productos financieros dando por resultado la utilidad financiera.

5. Por último a la utilidad financiera obtenida se le debe restar o sumar, según sea el caso, el resultado neto entre otros

gastos y productos para obtener la utilidad líquida del ejercicio.

Es conveniente aclarar que el procedimiento descrito anteriormente es particular para dependencias de la UNAM, dado de que en este caso no se trata de instituciones lucrativas, sino de servicio.

Alternativas de Solución.

Se plantean las siguientes alternativas:

- a) Usar un sistema ya existente de contabilidad en un computador central Burroughs.
- b) Realizar un desarrollo para un sistema contable en microcomputadora.
- c) Comprar un paquete comercial.
- d) Dejar que el sistema siga funcionando en forma manual como se realiza en la actualidad.

Beneficios.

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

Beneficios tangibles.

En cuanto a la primera y segunda alternativa de las tres propuestas para la solución del problema no podemos hablar de un ahorro específicamente económico ya que las dependencias de la UNAM no son instituciones de carácter lucrativo, sin embargo si se puede observar un aumento en la productividad dado la

facilidad de la realización del proceso contable, ya que el automatizar éste, significa que la computadora será la que realice todo proceso, obteniendo las salidas deseadas dadas las entradas requeridas para el proceso, el esfuerzo de las personas encargadas del proceso es menor, que si lo llevan en forma manual, realizando mayor número de funciones por unidad de tiempo, lo que repercute en un aumento de productividad.

Por lo que se refiere a la alternativa del inciso c, la inversión económica a realizar sería mayor que la que se tendría si se eligiera cualquiera de las demás opciones. Este se debe a que, para cada dependencia que requiriera del paquete de contabilidad, se tendría que comprar un original. Por otra parte, si consideramos que actualmente el precio de un paquete de este tipo fluctúa entre cien mil y doscientos mil pesos, y que además el número de paquetes a adquirir sería considerable, el costo total es extremadamente elevado.

Además debemos considerar que un paquete comercial está diseñado en base a requerimientos muy generales.

Por todas las razones antes expuestas de antemano desechamos la alternativa del inciso c por incosteable.

Si comparamos la primera alternativa con la segunda, se puede afirmar que en el caso de inclinarnos por la primera opción, habría un aumento de productividad, mas sin embargo éste sería menor que de inclinarnos por la segunda opción, ya que el sistema del computador central no es interactivo con el usuario, por lo que su manejo es mas complicado que el que se da usando un sistema conversacional, en forma local.

Beneficios Intangibles.

Por lo que se refiere a este tipo de beneficios en el caso de inclinarnos por la primera alternativa es claro que se reportarían varios beneficios, por principio el hecho de que un proceso automatizado, siempre es más rápido que uno manual, incluyendo el tiempo que se necesitaría para la capacitación del personal que este destinada a utilizar el sistema y que de hecho serían la mismas personas encargadas anteriormente del proceso manual. Por otra parte es más difícil que se reporte un error en el proceso por parte de un sistema automatizado, que en los que se incurre cuando se realiza un proceso manualmente, además de que las entradas al sistema automatizado, son validadas para evitar entradas erróneas cosa que no sucede con el proceso manual.

El sistema automatizado reporta confiabilidad en la información obtenida y disponibilidad de información en cualquier momento, mientras que en el manual no se pueden asegurar dichas características, ya que depende del personal encargado del proceso.

Si nos inclinamos por la primera alternativa es claro que no es necesario invertir tiempo ni esfuerzo en un nuevo desarrollo.

Las características del sistema de contabilidad del sistema central Burroughs son las siguientes: Este sistema es recomendado para ser usado por dependencias cuyo volumen de información es muy elevado ya que por principio maneja cuatro niveles para codificar las cuentas. El proceso para capturar y analizar las pólizas se realiza por lote, es decir que no se trata de un

sistema interactivo con el usuario, por lo que para la ejecución de los módulos del sistema es necesario conocer CANDE y WFL que son herramientas para la utilización de equipos de la marca mencionada.

La validación de la información se realiza solo hasta finalizar todo el lote.

Tomando en cuenta que el computador central se encuentra localizado en la DGSCA, los reportes de correcciones, diagnósticos etc, tienen que ser recogidos en esta dependencia por el usuario del sistema. El sistema de contabilidad del computador central es muy completo por lo que requiere de información adicional además de la básica .

En realidad este sistema fue diseñado para ser usado mediante tarjetas perforadas .

Otro punto a tomar en cuenta es la posible saturación del equipo central A9, así como el problema que se tiene con las líneas de comunicación.

El planteamiento de realizar un nuevo desarrollo trae consigo la desventaja de la inversión de tiempo y recursos en el mismo, aunque un sistema conversacional es más fácil de utilizar que uno que no lo es. Se puede considerar que si el volumen de la información que se va a manejar no es muy alto, es más eficiente un sistema como el que se plantea a desarrollar, que el que ya está desarrollado por las siguientes razones:

- La capacitación del usuario es más rápida.
- La disponibilidad de la información (reportes) es mejor
- El tiempo de respuesta del sistema es mejor, ya que el

sistema no es multiusuario

Un problema que se podría presentar es la falta de recursos de cómputo para la utilización del sistema para microcomputadora ,aunque este problema tiende a ser resuelto porque el deseo de la administración de la UNAM es que se automaticen los procesos y que estos sean manejados en la misma dependencia,por lo que se han destinado para este fin microcomputadoras tipo PC adquiridas por la institución.

Todas las ventajas presentadas para el sistema automatizado,se presentan contra la desventaja de tener que realizar una inversión inicial en cuanto a recursos técnicos y humanos,aunque esta es relativa ya el equipo fue dado por la UNAM para su uso,el personal encargado del sistema puede ser el mismo que realizaba el sistema manual y los recursos de software son proporcionados por la D.G.S.C.A.

Del estudio de beneficios planteado,la decisión a tomar es que lo más conveniente para la solución del problema planteado,es automatizar el proceso de contabilidad y que es necesario contar con un sistema para microcomputadora PC para ser usado por las dependencias cuyo volumen de información no sea mucho muy elevado.

C A P I T U L O I I I

A N A L I S I S D E R E Q U E R I M I E N T O S

La etapa a desarrollar a continuación de la planeación del sistema es la de análisis de requerimientos, el cual nos permitirá estudiar más a fondo el problema para darle una solución óptima, usando lo que se conoce como Análisis Estructurado.

En realidad esta etapa del ciclo de vida de un sistema puede llegar a ser la más tediosa ya que se pueden presentar problemas de comunicación, o requerimientos de naturaleza cambiante, problemas con la asignación de recursos, problemas políticos etc.

Es en esta fase donde se entabla un enlace entre usuarios y personal encargado del desarrollo del sistema.

El análisis estructurado se puede definir como una metodología para especificar funcionalmente un sistema, y aunque esta metodología se ha tomado con escepticismo en el mercado, afortunadamente se aplica en la empresa privada, bancos y universidades.

Como entrada a esta etapa se tiene un problema claramente definido en la fase anterior y como salida deberá obtener un documento de especificación, el cual debe cumplir con las siguientes características: debe ser gráfico, conciso, mantenible, particionado, entendible y lógico. Para obtener este documento el análisis estructurado hace uso de tres herramientas que son el diagrama de flujo de datos (DFD's), el diccionario de datos, y

procedimientos lógicos como son árboles o tablas de decisión .

Para empezar a aplicar la metodología correspondiente a esta etapa se hará una lista de los requerimientos del sistema de acuerdo al planteamiento del problema realizado en la primera fase.

Requerimientos.

1. Se necesita poder crear y mantener un archivo de cuentas que servirá como libro de mayor, y donde se reflejan todos los movimientos realizados durante todos los periodos contables.

2. Es necesario obtener las entradas necesarias al sistema y además poder validar estas entradas para evitar errores en el proceso, estas entradas que son las pólizas de contabilidad siempre deben de cumplir con la partida doble.

3. Toda la información deberá quedar almacenada aunque no sea correcta , con el fin de evitar doble captura de información.

4. Si algún dato cumple con los requerimientos de entrada pero no esta correcto, debe de existir la manera de corregirlo.

5. Con toda la información correcta se deberán emitir los siguientes reportes donde se describen los movimientos realizados en el periodo:

- a). Balanza de Comprobación
- b). Diario General
- c). Auxiliar Uno
- d). Auxiliar Dos

Además se tiene que realizar lo que se conoce como cierre de

periodo con el fin de obtener los estados financieros es decir el Estado de resultados ,el Balance General y un tercer reporte auxiliar.

6. Como procedimientos alternos del sistema se debe contar con un proceso que emita reportes de el directorio de las cuentas que se utilizan en el periodo.

7. Se requiere que el sistema permita al usuario consultar la información capturada en cualquier momento por medio de la terminal.

DIAGRAMA DE FLUJO DE DATOS.

El Diagrama de Flujo de datos(DFD) es un elemento usado con el fin de particionar un sistema ,y es una de las principales herramientas del Análisis Estructurado.

Un DFD es una red que representa un sistema ,el que puede ser manual y/o computarizado .Un DFD retrata un sistema en término de sus componentes y las interfases entre los componentes. Las características con las que debe cumplir un DFD es que debe ser gráfico,particionado y multidimensional.

En algunas ocasiones es común que se confundan los términos de diagrama de flujo de datos y diagrama de flujo de un programa ,por lo que cabe aclarar que son términos muy diferentes,para empezar un DFD particiona un sistema en partes lo suficientemente pequeñas para ser especificadas concisamente y muestra las interfases de los flujos de datos entre estas partes. El contenido del flujo de datos esta precisamente especificado en el

Diccionario de datos. Un DFD no muestra secuencias de control como en el caso de un diagrama de flujo común. Un diagrama de flujo es una visión del mundo desde el punto de vista del CPU, es una secuencia de "primero realiza esto, después lo siguiente etc." en un orden bien definido.

El mundo real no trabaja como se comporta un diagrama de flujo, mas bien se trata de un proceso de cooperación donde todos los elementos trabajan mediante la comunicación de datos de unos a otros.

Un DFD es una red donde se muestra un flujo de datos no un flujo de control y consta de los siguientes cuatro elementos :

- Un arco o vector llamado flujo de datos, el cual es una trayectoria o canal a través del cual fluyen grupos de datos con una composición conocida.

- Un nodo o burbuja que indica transformación de información (proceso) de flujo de datos que llegan a flujo de datos que salen.

- Dos líneas horizontales y paralelas que indican un archivo o una base de datos.

- Un rectángulo que representa una fuente o un destino de datos, el cual es una persona u organización que cae fuera del contexto del sistema y que genera o recibe datos de éste.

Cabe señalar que se pueden desarrollar varios niveles de un DFD de acuerdo a la complejidad del problema del tal forma que el "diagrama cero" engloba a todo el sistema y es el más general. En realidad no se puede hablar de un número fijo de niveles de detalle aunque se recomienda que se llegue a un nivel de detalle

tal en que se obtengan burbujas o "primitivas funcionales".

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

Es recomendable que en un DFD se considere el 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 Para Realizar un DFD

1. Identificar fuentes externas.
2. Identificar las diferentes entradas y salidas que se pueden esperar del proceso agrupándolas lógicamente y separando las entradas y salidas que manejen errores y excepciones.
3. 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.
4. Checar este diagrama con el proceso real asegurándose que se incluyeron todas las entradas y salidas requeridas así como las transformaciones lógicas de los datos.
5. Dibujar un segundo diagrama en el que se incluya los errores y excepciones y se minimice el número de cruces de flujo de datos.
6. Realizar una "caminata" con una persona involucrada en el proceso para su revisión.
7. Mandarlo a dibujo.

Para el problema a resolver tenemos :

Fuentes externas.

Solo existe una fuente externa por ser el sistema de tipo conversacional y se distinguirá como "usuario".

Entradas.

Como primera entrada al sistema de acuerdo a la descripción de los requerimientos, se debe de tener un directorio de cuentas, este directorio vendría siendo el libro de mayor para un sistema manual, y en el se reflejarán los movimientos realizados en el periodo contable.

Las cuentas, para mayor simplicidad serán identificadas por un número de 10 dígitos (aunque también deben de tener un nombre) donde el último dígito es un dígito verificador que nos permitirá validar la captura de los números de cuenta.

Podemos distinguir dentro de la contabilidad varios tipos de cuenta, para el caso específico del sistema a desarrollar se consideraran los siguientes tipos:

Cuentas de Mayor

Cuentas de Nivel

Cuentas de Auxiliar

Los primeros nueve dígitos que conforman cada una de las cuentas se agrupan en tres niveles:

0000	00	000	0
Mayor	Nivel	Auxiliar	IC

Entonces para codificar la información se deberán ocupar los diferentes niveles de acuerdo al tipo de cuenta que manejen. Es decir, una cuenta de mayor solo ocupará el nivel de mayor y los

otros niveles tendra zeros.

ejemplo: 1301 00 000

Una cuenta de nivel ocupará el nivel de mayor y de nivel.

ejemplo: 1301 01 000

Mientras que una cuenta de auxiliar ocupará todos los niveles

ejemplo: 1301 01 001

Para obtener los primeros nueve digitos que conforman las cuentas la información se debera codificar de acuerdo a un criterio establecido por los contadores de la UNAM.

El último dígito que conforma la cuenta es el dígito verificador el cual es un número de control que se calcula de la siguiente manera : considerando los digitos de la cuenta de izquierda a derecha se suman los que se encuentran en posiciones nones y se multiplican por siete ;se suman los digitos de las posiciones pares y se multiplican por tres;los resultados se suman y el dígito menos significativo (el último de dicha suma) será el dígito de control.

Ejemplo 2101 00 003

$$2+0+0+0+3 =5; 5 \times 7=35$$

$$1+1+0+0 = 2; 2 \times 3= 6$$

$$\begin{array}{r} 41 \\ \hline \end{array}$$

El dígito verificador de la cuenta 210100003 es 1, finalmente el número de cuenta queda como 2101000031.

La cuenta también podrá ser referenciada para mayor claridad por un nombre de cuenta.

La primera vez que se corra el sistema se debe proceder a cargar el directorio de cuentas previamente codificadas de la manera descrita anteriormente además de que a través del uso del sistema se podrá incluir o borrar cuentas, así como cambiar el nombre de éstas, para esto se sugiere la siguiente forma de captura de cuentas:

La segunda entrada al sistema es la que nos permite realizar movimientos dentro de las cuentas, y siguiendo el método contable descrito en la fase de planeación, estas entradas son las pólizas contables cuyas características ya fueron descritas, y en base a esas características se sugiere que en la captura se consideren los siguientes datos:

Número de Póliza :Es un número consecutivo que mantendrá una secuencia creciente hasta el último día del año.

Fecha de la Póliza :Es la fecha de la elaboración de la póliza con el formato ddmmaa.

Clave :La clave indicará el tipo de movimiento que afectará a la cuenta asociada. Existen seis tipos de movimientos:

MD Mayor Debe (cargo a una cta de mayor)

PD Parcial Debe (cargo a una cta de auxiliar)

MH Mayor Haber (abono a una cta de mayor)

PH Parcial Haber(abono a una cta
de auxiliar)
RD Resumen Debe (total de cargos)
RH Resumen Haber(total de abonos)

Número de Cuenta : Es la cuenta sobre la que se
realizará el movimiento
contable.

Importe : Cantidad de dinero que se abona
o carga a cada cuenta.

Suma de cargos : Importe de la suma de los
cargos.

Suma de abonos : Importe de la suma de los
abonos.

Concepto : Breve descripción del contenido
de la póliza.

Para obtener los datos antes descritos se sugiere la
siguiente forma de captura:

POLIZA DE CONTABILIDAD

		NO. POLIZA FECHA		
CLAVE	NO. CTA	IMPORTE	IMPORTE	IMPORTE
		PARCIAL	MAYOR DEL DEBE	MAYOR DEL HABER
.
.
		SUMAS	CARGOS	ABONOS
CONCEPTO				
ELABORO				

Salidas del sistema.

Como salidas se tienen los siguientes reportes:

- Directorio de Cuentas.

El directorio de cuentas es un reporte de las cuentas que

existen para el periodo.

El directorio de Cuentas tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA				
DIRECTORIO DE CUENTAS				
MAYOR	SUBCTA	AUX	DC	NOMBRE
.....	CUENTA DE MAYOR
....	CUENTA DE NIVEL
....	CUENTA DE AUXILIAR
1401	00	000	2	Pagos anticipados
1401	01	000	5	Gastos de Instalacion
1401	01	001	2	Gastos x

- Balanza de Comprobación y Diario General.

Los cuales son reportes donde aparecen la totalidad de cuentas de Mayor sin importar si tuvieron o no movimientos durante el periodo.

Estos reportes deberán cumplir con los siguientes formatos:

NOMBRE DE LA DEPENDENCIA

BALANZA DE COMPROBACION

NUMERO DE CTA.	NOMBRE CTA.	SALDO ANTERIOR	SALDO DEBE	SALDO HABER	SALDO ACTUAL
-------------------	----------------	-------------------	---------------	----------------	-----------------

.
.

TOTALES \$

NOMBRE DE LA DEPENDENCIA

DIARIO GENERAL

NUMERO DE CTA.	NOMBRE CTA.	SALDO DEBE	SALDO HABER
-------------------	----------------	---------------	----------------

.	.	.	.
.	.	.	.

Totales \$

- Auxiliar Uno .

El auxiliar uno es un reporte que contiene para cada cuenta de mayor, una especificación de las cuentas de auxiliar que hicieron referencia a él. Cada cuenta de auxiliar es un resumen que representa el saldo total de las referencias que hace una misma cuenta.

El reporte debe cumplir con el siguiente formato:

NOMBRE DE LA DEPENDENCIA				
AUXILIAR UNO				
NOMBRE DE LA CUENTA:				
CUENTA	SALDO ANTERIOR	TOTAL DE CARGOS	TOTAL DE ABONOS	SALDO ACTUAL
.	\$	\$	\$	\$
NO. CTA	CARGOS DEL MES	ABONOS DEL MES	SALDO	
.	\$	\$	\$	

- Auxiliar Dos .

Este reporte contiene para cada cuenta de auxiliar una especificación de los movimientos que afectaron a las cuentas durante el periodo.

El Auxiliar Dos tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA					
AUXILIAR DOS					
NOMBRE DE LA CUENTA:					
CUENTA	SALDO ANTERIOR	TOTAL DE CARGOS	TOTAL DE ABONOS	SALDO ACTUAL	
.	\$	\$	\$	\$	
FECHA	DCTO	CONCEPTO	DEBE	HABER	SALDO
.	\$	\$	\$

El Estado de Resultados y el Balance general se generan de acuerdo a las cuentas que hayan sido definidas por el usuario ,de acuerdo al procedimiento descrito en la fase de definición del problema.

Estos reportes cumplen con el siguiente formato:

NOMBRE DE LA DEPENDENCIA		
ESTADO DE RESULTADOS		
CUENTA 1.1	NOMBRE	\$
	MENOS	
CUENTA 2.1	NOMBRE	\$
	UTILIDAD NETA	\$
CUENTA 3.1	NOMBRE	\$
	UTILIDAD BRUTA	\$
	MENOS	
CUENTA 4.1	NOMBRE	\$
	UTILIDAD DE OPERACION	\$
CUENTA 5.1	NOMBRE	\$
	UTILIDAD FINANCIERA	\$
	MENOS	
CUENTA 6.1	NOMBRE	\$
	UTILIDAD O PERDIDA	\$

NOMBRE DE LA DEPENDENCIA

BALANCE GENERAL

ACTIVO
CIRCULANTE

CUENTA 1.1.1 NOMBRE \$

SUMA EL CIRCULANTE \$

FIJO

CUENTA 1.2.1 NOMBRE \$

SUMA EL FIJO \$

DIFERIDO

CUENTA 1.3.1 NOMBRE \$

SUMA EL DIFERIDO \$

SUMA EL ACTIVO \$

PASIVO
CIRCULANTE

CUENTA 2.1.1 NOMBRE \$

SUMA EL CIRCULANTE \$

FIJO

CUENTA 2.2.1 NOMBRE \$

SUMA EL FIJO \$

PATRIMONIO

CUENTA 3.1.1 NOMBRE \$

SUMA DE PATRIMONIO \$

SUMA DE PATRIMONIO Y PASIVO \$

Ya una vez descritas las entradas y las salidas al sistema se analizarán los procesos globales para cumplir con los requerimientos ya especificados:

Podemos establecer la necesidad de contar con siete funciones específicas englobadas en un módulo cada una de ellas ;dichas funciones son las siguientes:

Módulo 1 Mantenimiento .

Nos debe permitir obtener la información de las cuentas a utilizar durante todo el proceso contable de la dependencia y además debemos poder mantener esta información dando altas,bajas y cambios en cuentas.

Módulo 2 Captura y/o Diagnóstico.

Este módulo es el que se encarga de la captura de la información contenida en las pólizas contables antes descritas, además se debe de validar la información de entrada para evitar errores de captura. En la parte de diagnóstico se debe checar que la póliza capturada cumpla con la partida doble, de no ser así se guarda la póliza como errónea de tal manera que pueda ser corregida .

Módulo 3. Correcciones.

Este proceso permite al usuario corregir pólizas capturadas que por algún error en saldo o clave se hayan diagnosticado incorrectas ,es decir que no cumplan con la partida doble.

Módulo 4. Reportes

El módulo de reportes emite reportes de directorio de cuentas, es decir cuentas que existen actualmente en el directorio ya sea clasificadas por número de cuenta o por nombre según especifique el usuario.

Módulo 5. Cierre de período.

El módulo de cierre de período realizó el resumen de toda la información que haya sido diagnosticada como correcta para emitir los siguientes reportes:

- Balanza de Comprobación y Diario General
- Auxiliar Uno
- Auxiliar Dos

Además al finalizar la emisión de los reportes se ejecuta el Cierre de período, el cual actualiza los saldos de las cuentas de directorio dejándolas listas para iniciar con un nuevo período.

Módulo 6. Estados Financieros.

Este proceso emite los reportes correspondientes a los Estados Financieros que son El Estado de Resultados, El Balance General y el Auxiliar Tres o Anexo al balance.

Módulo 7. Consulta.

El módulo de consulta nos permite realizar consultas de pólizas capturadas ya sean correctas o incorrectas, además de poder consultar saldos de cuentas específicas.

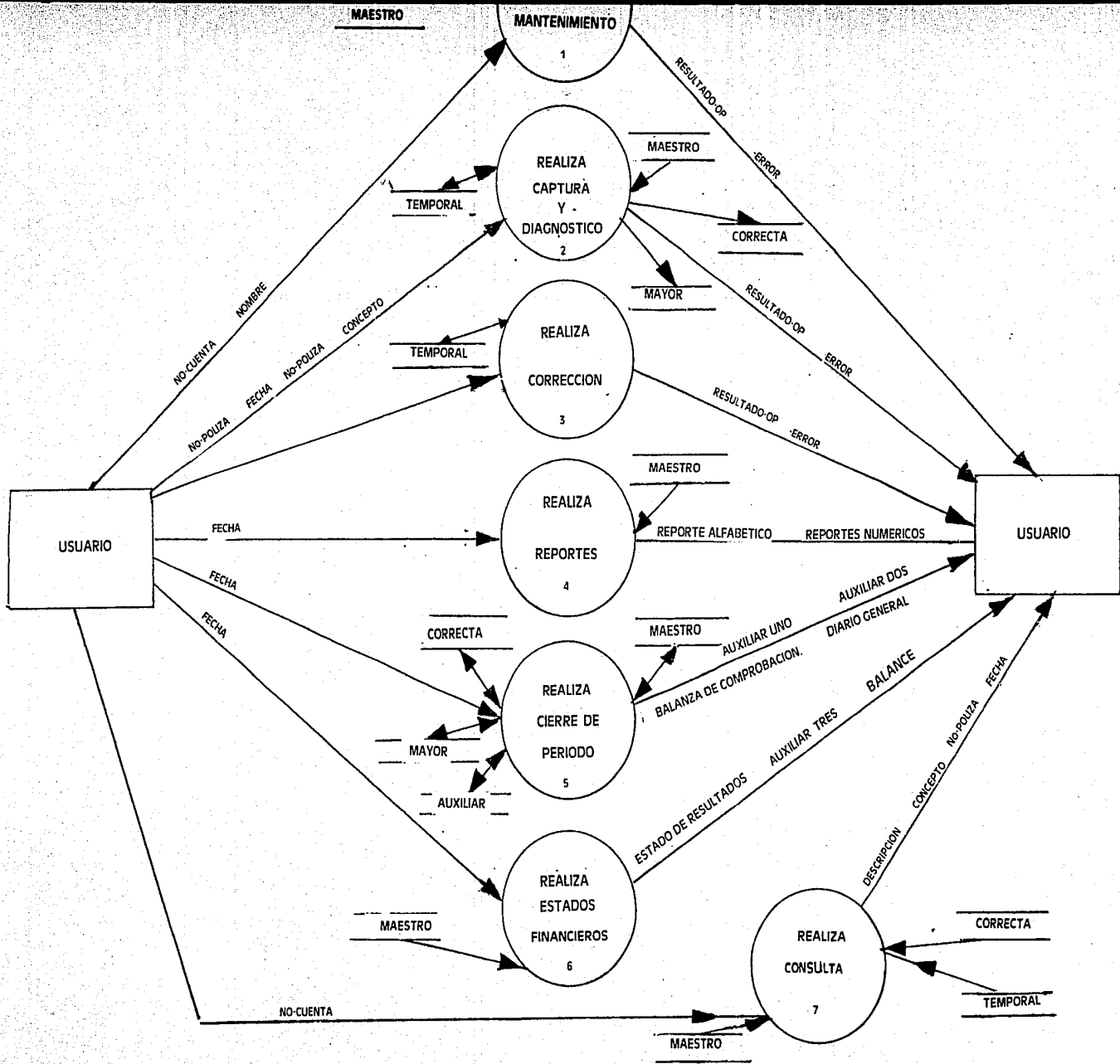
Además podemos establecer la necesidad de contar con programas auxiliares que realicen funciones para uso general como son:

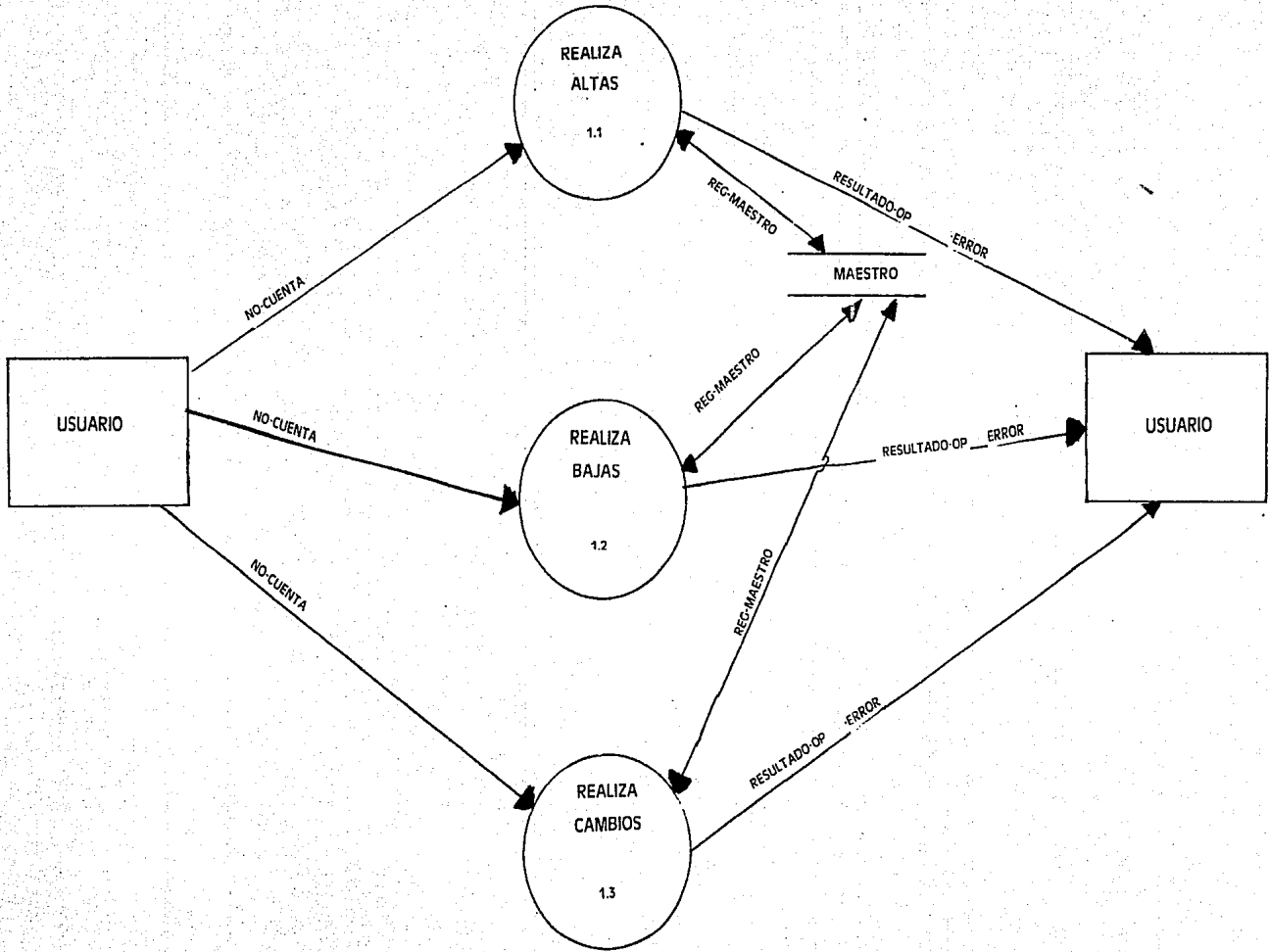
Auxiliar Uno -Desplegar encabezados y fecha.
Auxiliar Dos - Desplegar Mensajes de Error.
Auxiliar Tres - Calcula digito verificador.
Auxiliar Cuatro - Valida tipo de Movimiento.
Auxiliar Cinco - Valida tipo y cuenta.

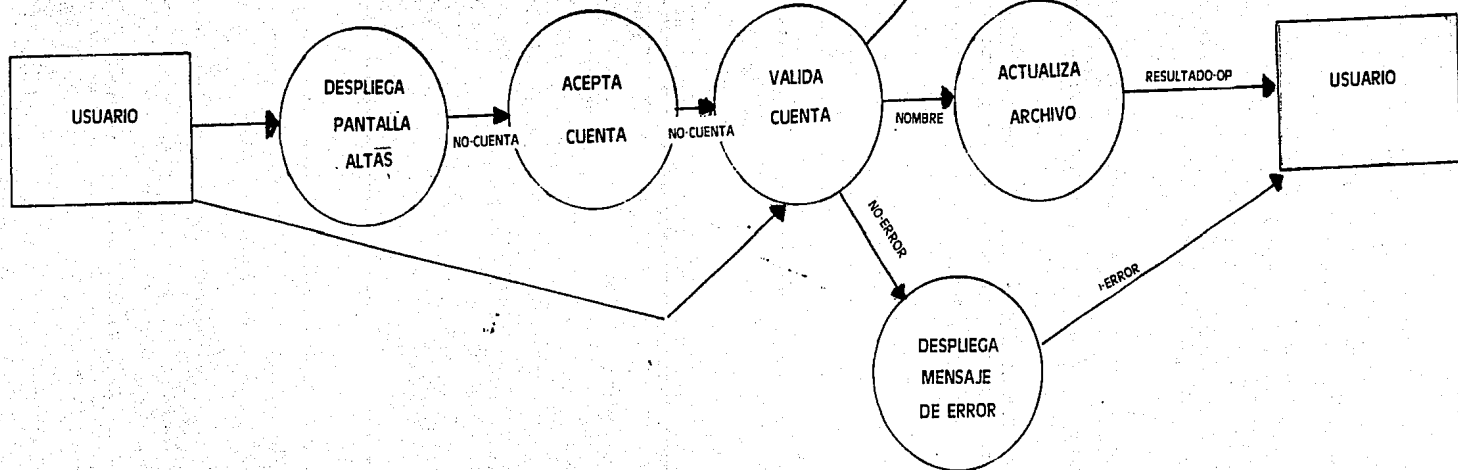
Por lo que se refiere a la validación de la información ,las mas generales son las referentes a la captura de cuentas,estas validaciones se contemplan en los programas auxiliares uno,dos y tres .El resto de validaciones que realizará el sistema son más generales y simples entre éstas están las siguientes:

- No se debe dar de alta un registro ya existente.
- No se debe dar de baja un registro no existente.
- No se deben capturar conceptos de pólizas en blanco.
- Se deben validar fechas y campos numéricos.
- Siempre que se captura una póliza se deben checar los importes,de tal forma que la suma del Debe sea igual a la del haber.

Después de describir entradas, salidas y procesos se presentan los diagramas de flujo de datos que corresponden a los procesos del sistema de contabilidad.

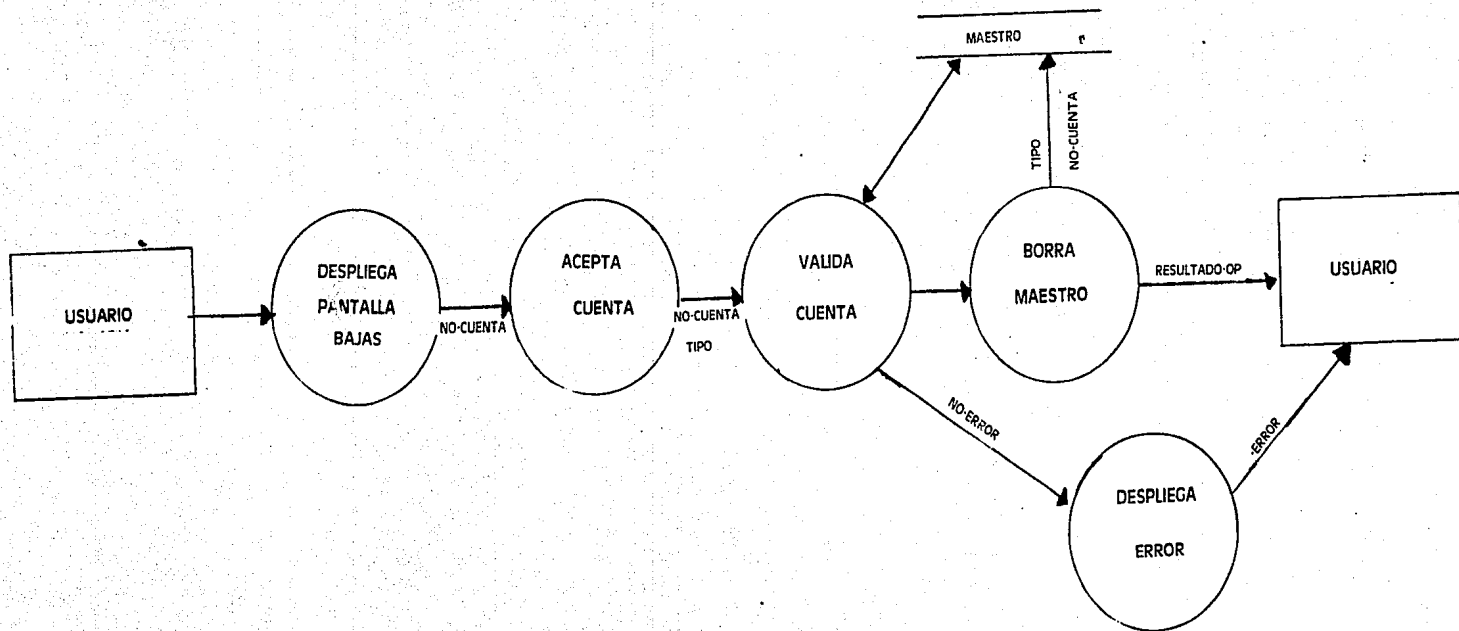


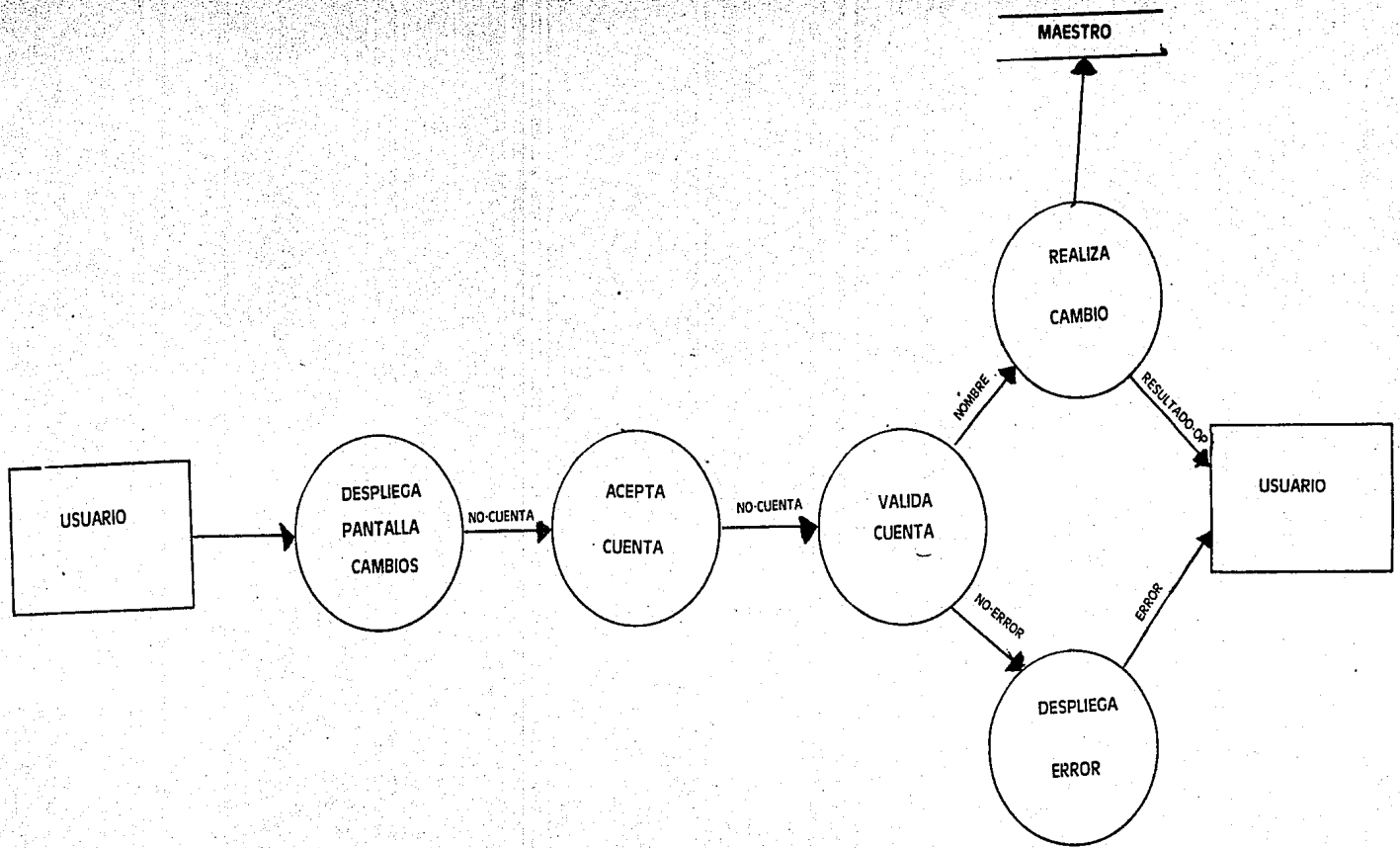


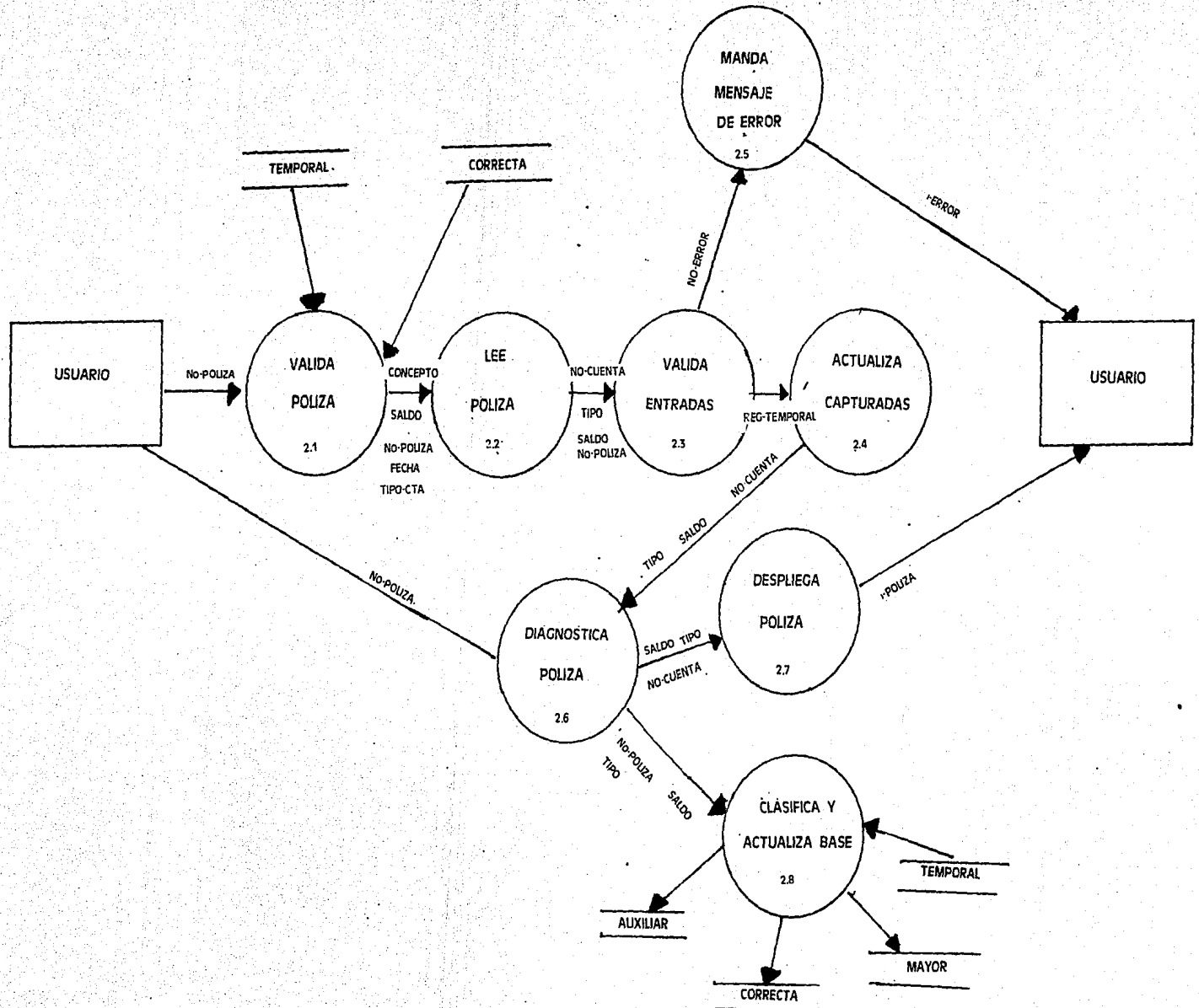


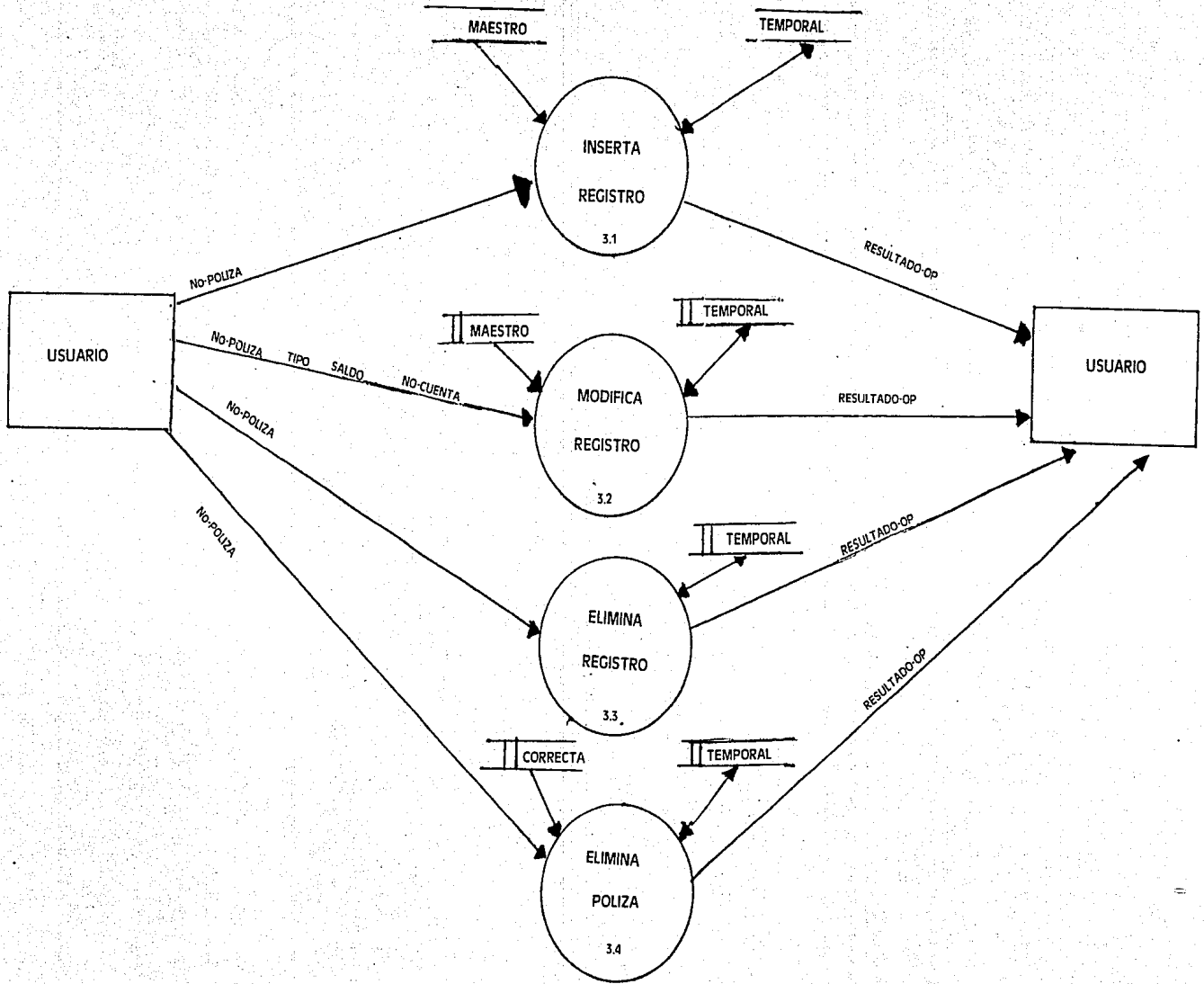
1.2

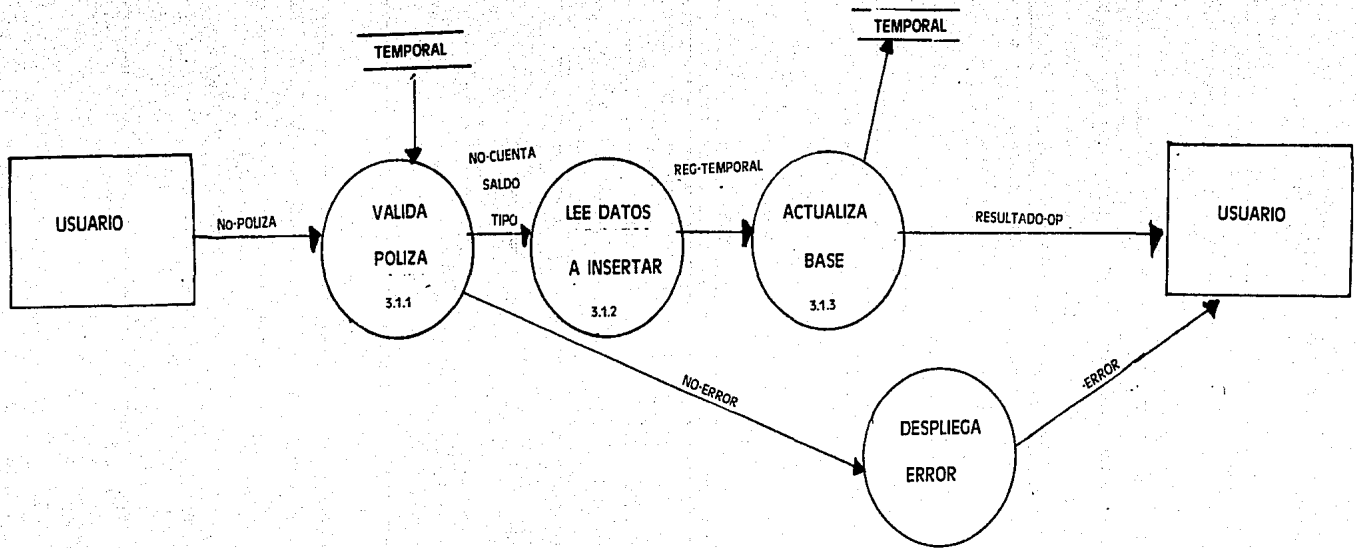
45



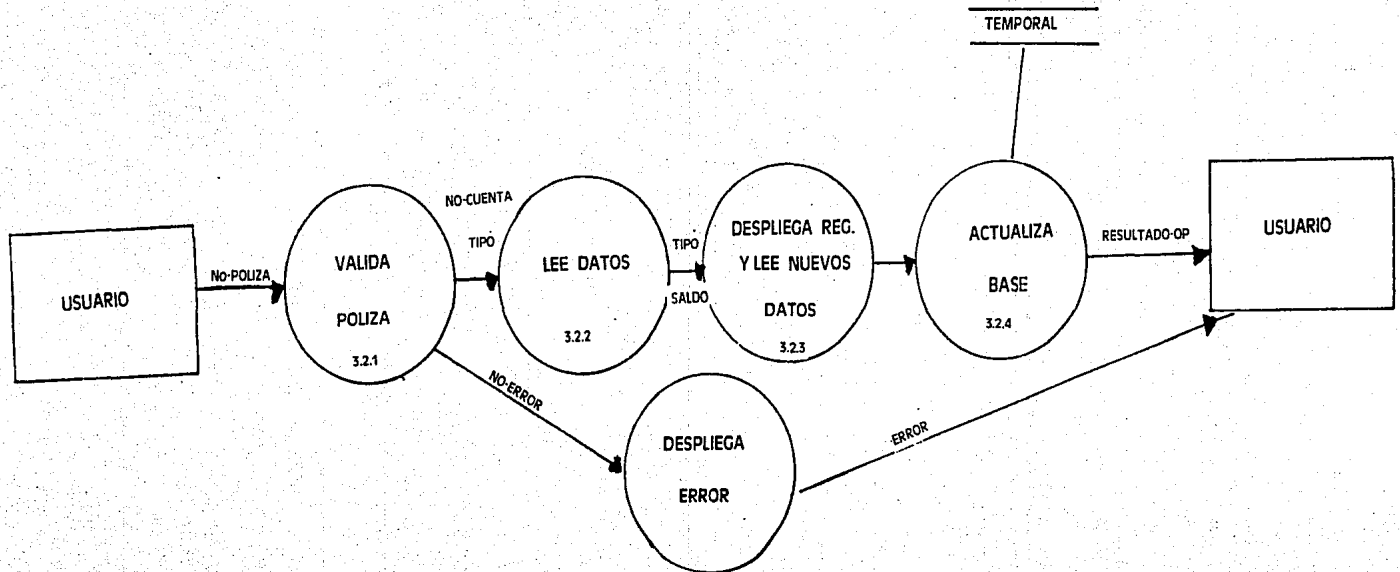


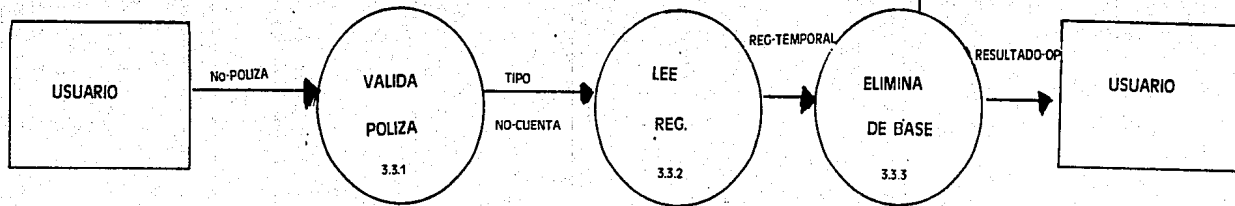




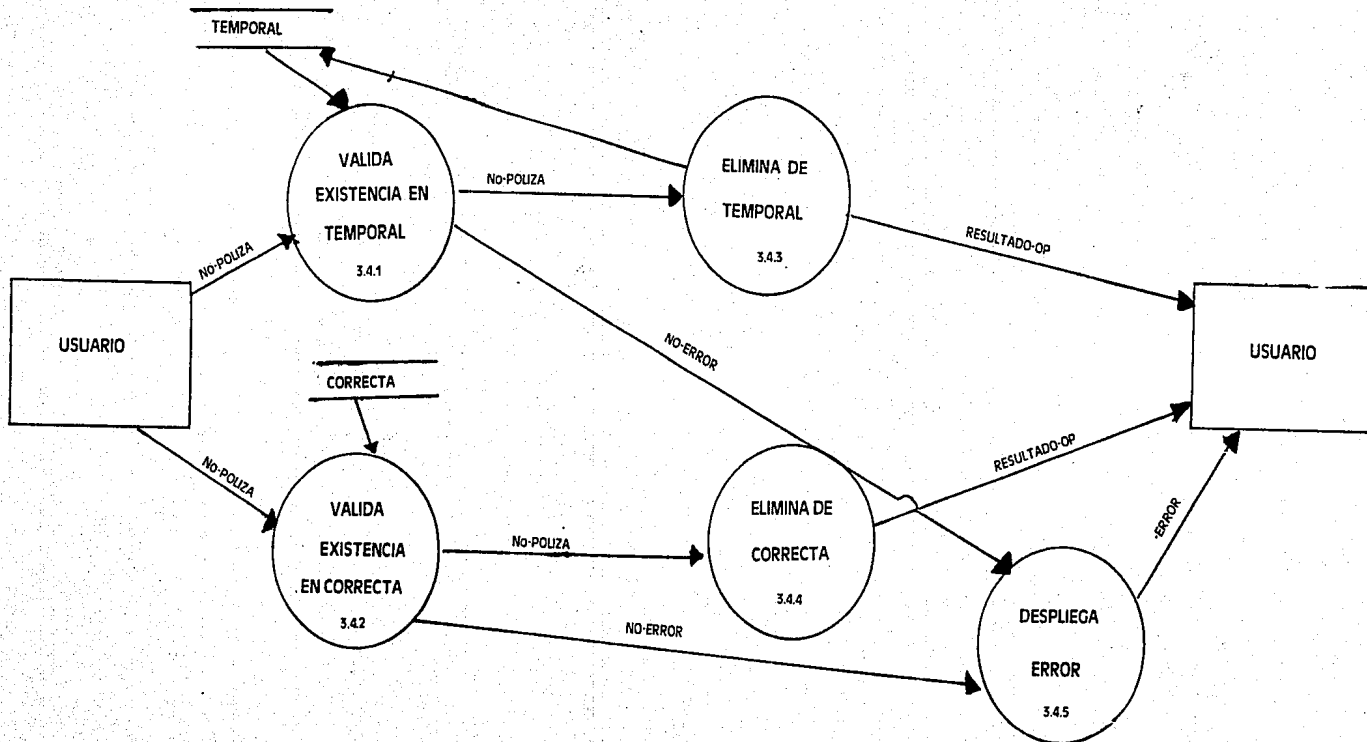


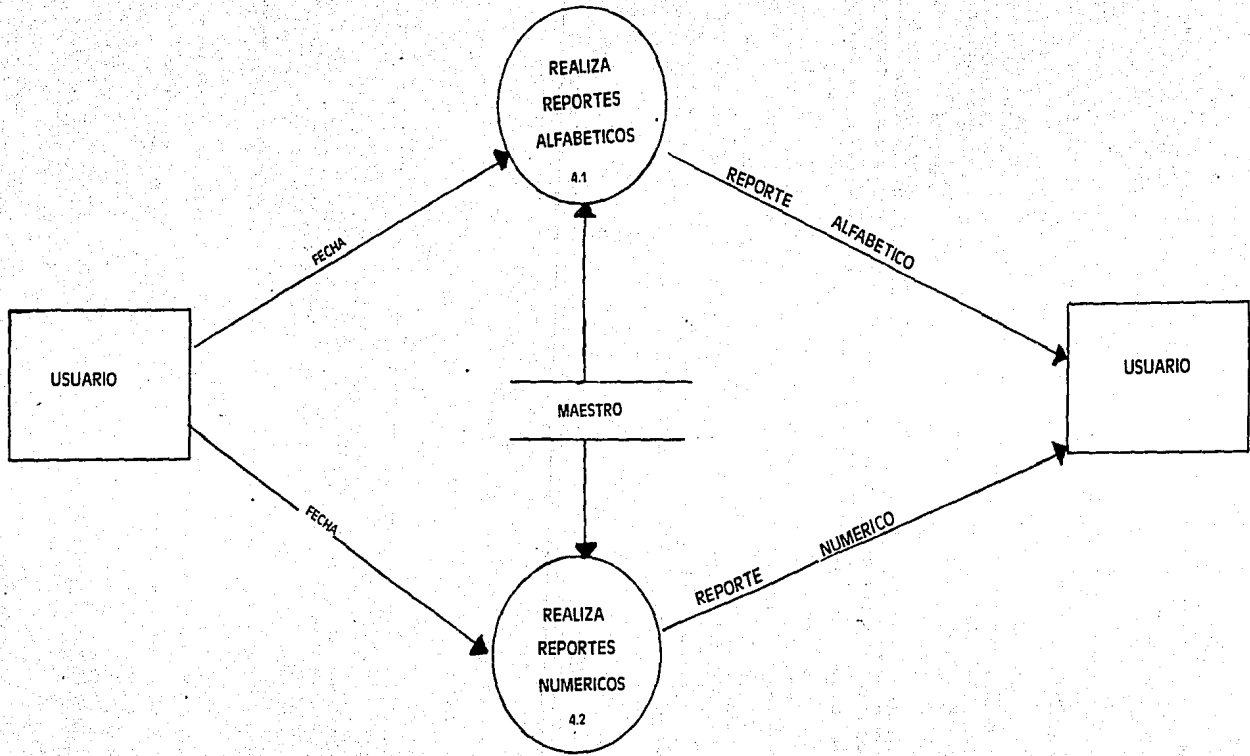
D.F.D. 3.2

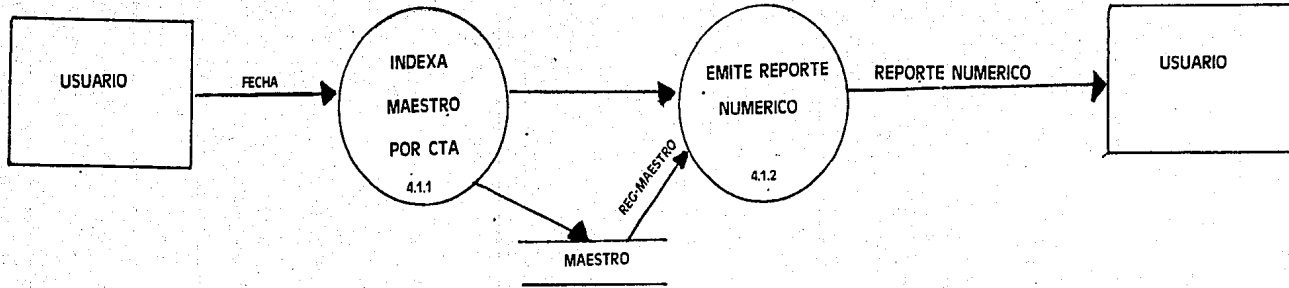




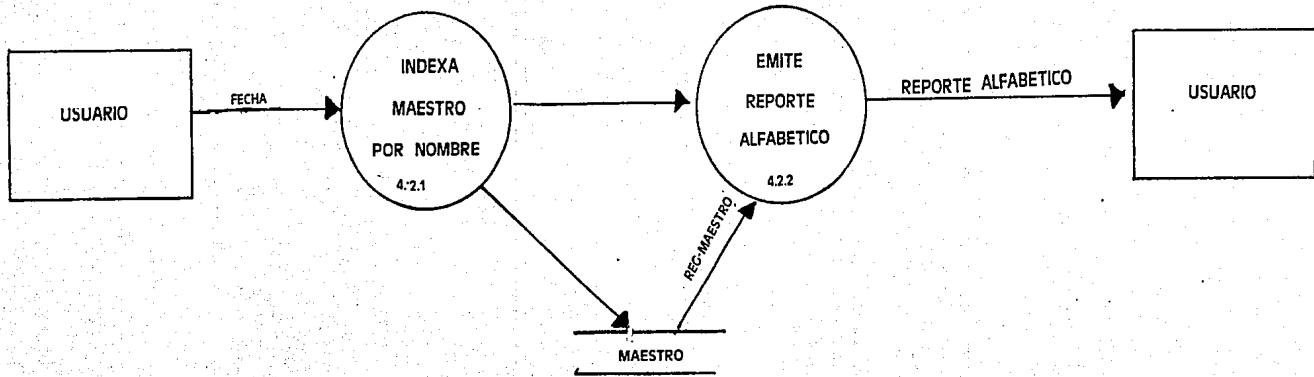
D.F.D. 3.4



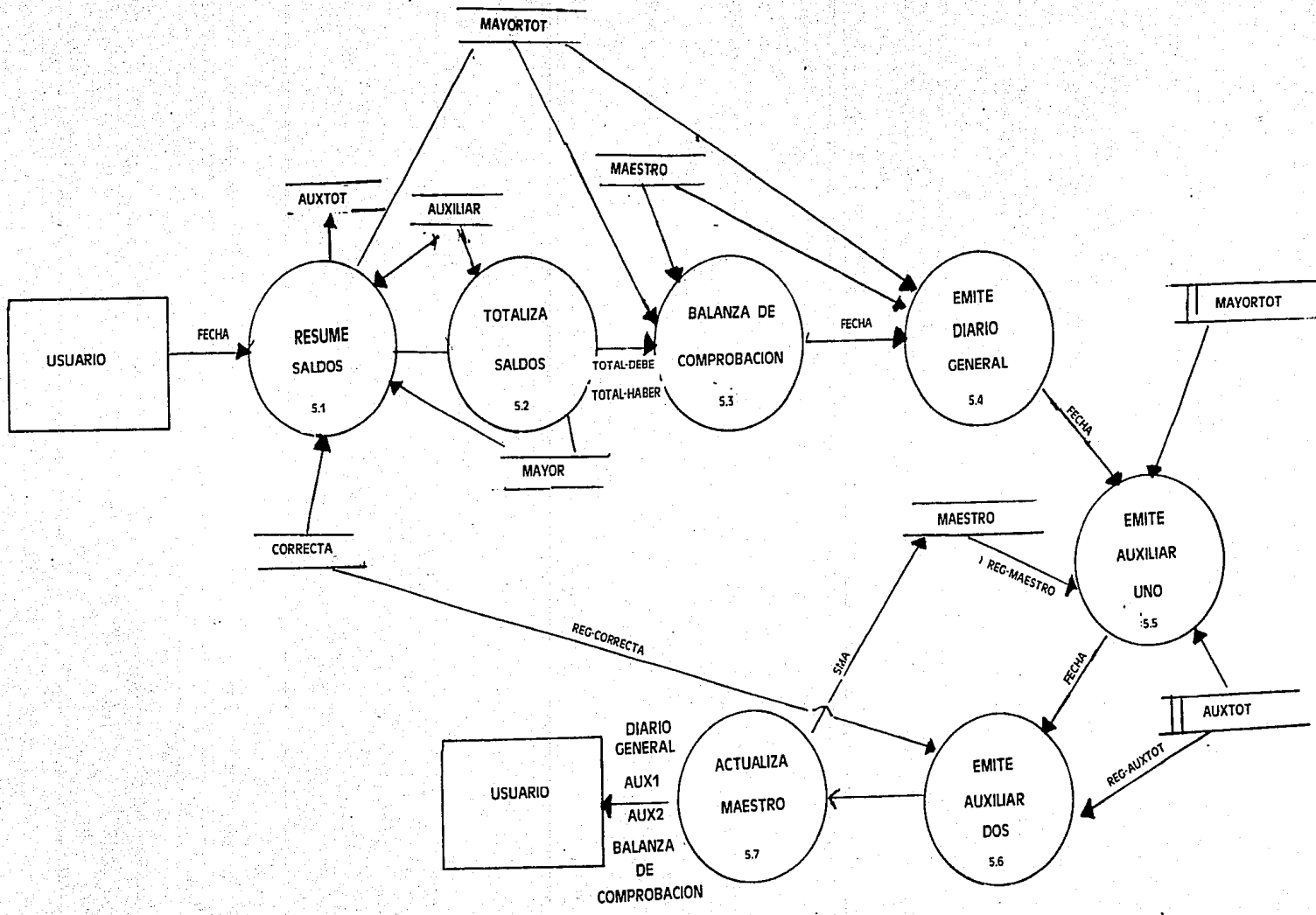


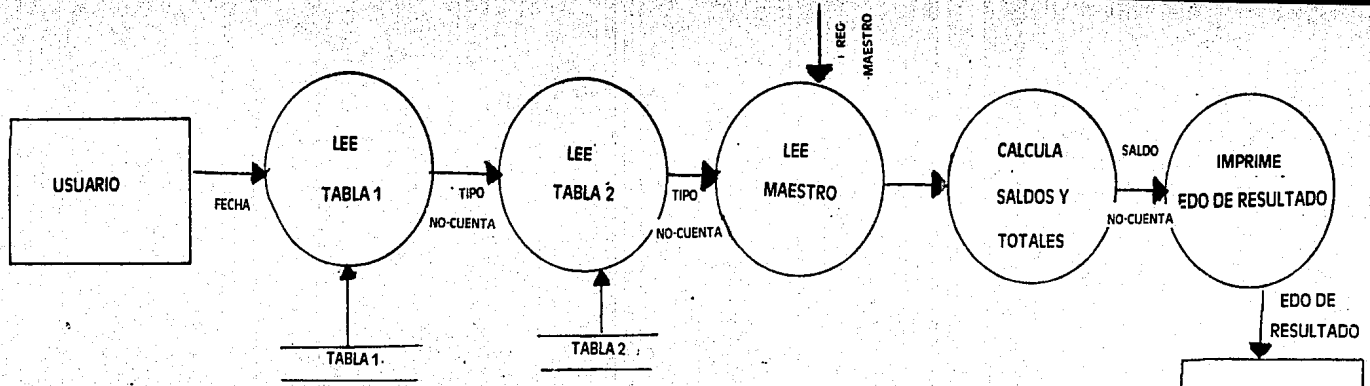


D.F.D. 4.2

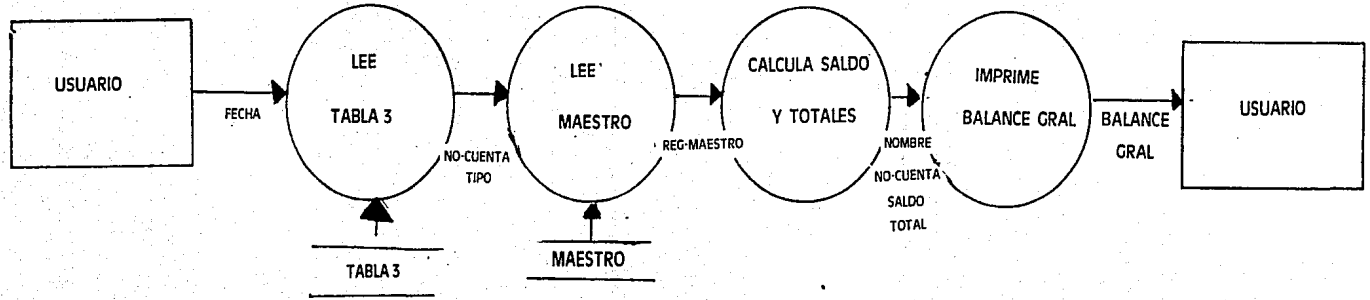


53

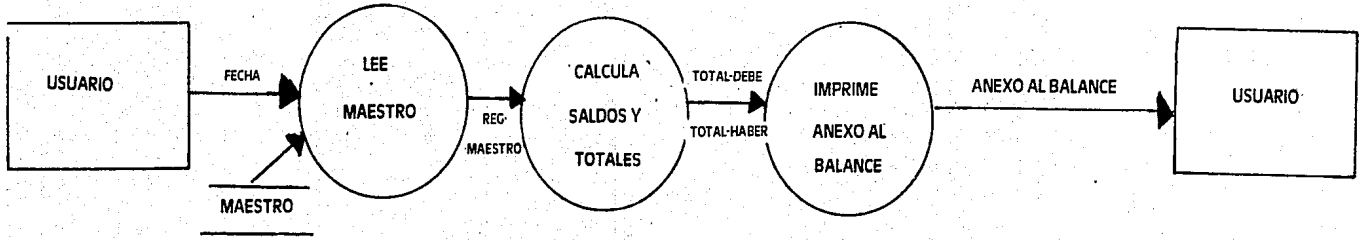


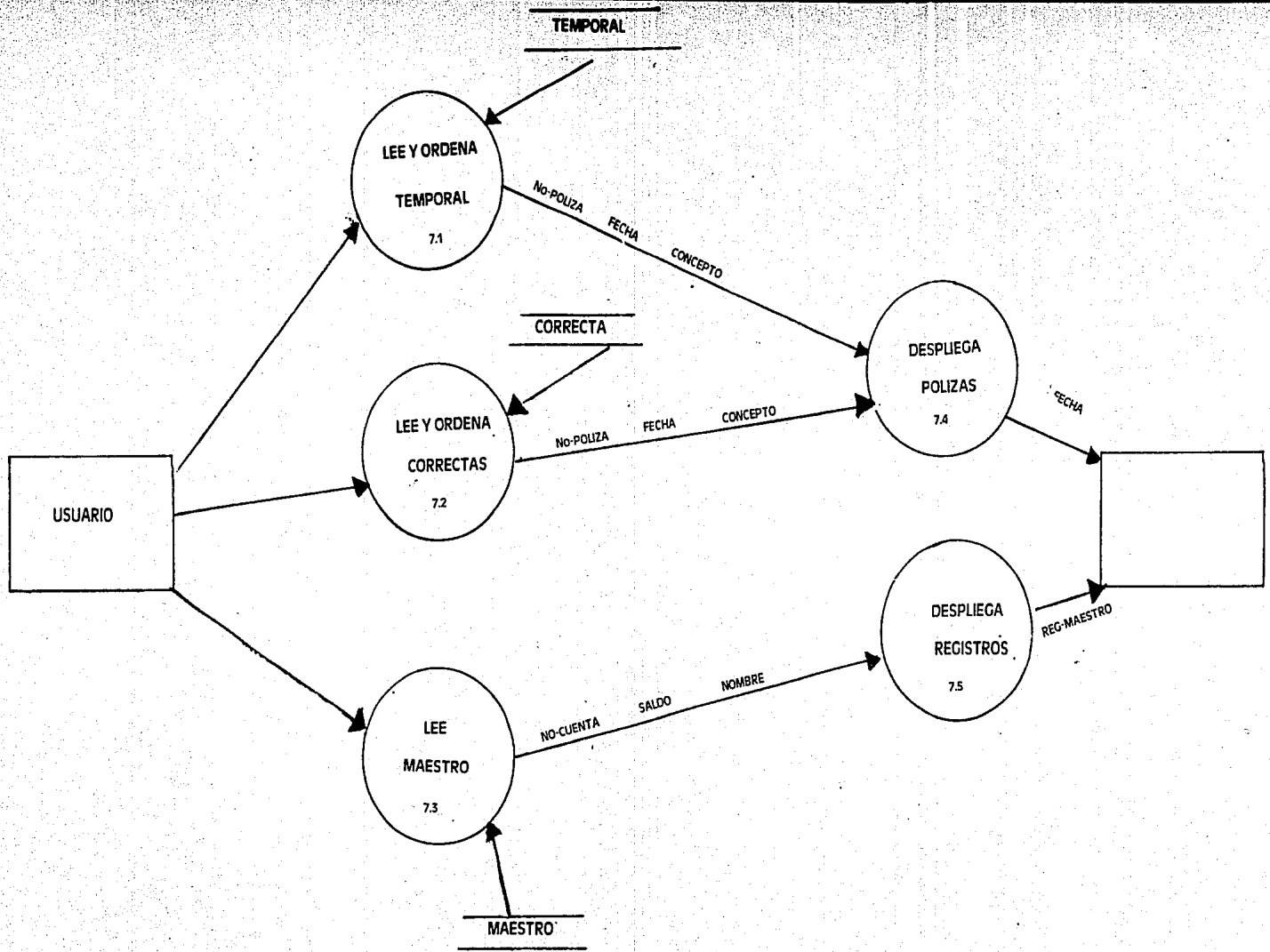


6.1



6.2





Diccionario de Datos.

Un diccionario de datos (DD) es un conjunto de definiciones de términos que aparecen en los DFD's. Un DD logra que los DFD sean rigurosos.

Los términos que se manejan dentro de un diccionario de datos son: datos, archivos o bases de datos, destinos fuentes y procesos.

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

La simbología utilizada en el DD sera la siguiente:

Simbología	Significado
=	Se compone de
+	y
[1...1]	selecciona una de
{ ... }	iteraciones
(...)	opcional
* ... *	comentario

Diccionario de datos de los DFD's

Póliza = No.póliza + Fecha + {línea detalle} + descripción

línea detalle = Cuenta + clave + Saldo

No.póliza = {dígito}

Fecha = dd + mm + aa

descripción = {caracter}

Cuenta = Cta + Niv1 + Niv2 + Dig

Cta = {dígito}
Niv1 = {dígito}
Niv2 = {dígito}
Dig = {dígito}
Clave = ['RD'/'RH'/'MH'/'MD'/'PH'/'PD']
Saldo = {dígito} + "." + {dígito}
dd = [01/02 /31]
mm = [01/02 /12]
aa = [85/86 /99]
tipo = [1/2/3/4/5/6]
pol = {dígito}

Sección de archivos

COMAE = cta + niv1 + niv2 + dig + sh +sd + sma
COTEMP = pol + cta + niv1 +niv2 + dig + tipo + saldo
COMAY = pol + cta + niv1 + niv2 + dig + tipo +saldo
COCON = pol + con + fecha
COCORR = pol + con + fecha
COAUX = pol + cta + niv1 + niv2 + dig + tipo + saldo

Sección de sinónimos

con=descripción
sd = sh = sma = saldo

Miniespecificaciones.

Una miniespecificación, es una especificación de las primitivas funcionales con el objetivo de tener una información concisa de lo que se está desarrollando, no ambigua, lógica y no física, completa que indique qué hace el proceso y no cómo lo hace. Para el desarrollo de estas miniespecificaciones se usa el Español Estructurado el cual es un lenguaje de especificación considerado como un subconjunto del idioma español con un vocabulario limitado, una sintaxis de estatus limitada, y un conjunto limitado de maneras de combinar oraciones.

Es de vocabulario limitado porque solo se utilizan verbos en modo imperativo, existen palabras reservadas y se relaciona los datos (DFD y DD).

Una miniespecificación debe ser entendible para toda la gente y no tener expresiones algebraicas matemáticas sino que éstas deben ser narradas.

A continuación se presentan las miniespecificaciones de los procesos descritos en los DFD's

DEFINICION DE MINIESPECIFICACIONES

DEL SISTEMA DE CONTABILIDAD PC

PROCESO : CONTROLAR EL MENU PRINCIPAL

OBJETIVO: Seleccionar la actividad a realizar
dependiendo de la opción elegida por
el usuario.

Proceso: SISTEMA CONTABILIDAD.

REPITE

Desplegar el menú principal

Leer opción

CASO de opción igual a:

- 1: Actualizar y mantener el archivo maestro.
- 2: Capturar y diagnosticar.
- 3: Corregir pólizas.
- 4: Crear reportes.
- 5: Crear auxiliares y cierre de periodo.
- 6: Realizar estado de resultados.
- 7: Consultar pólizas.
- 0: Salir del sistema.

De otra manera:

Desplegar mensaje de error

HASTA opción =0

FIN DEL SISTEMA.

PROCESO : ACTUALIZAR Y MANTENER ARCHIVO MAESTRO

OBJETIVO:Seleccionar actividad a realizar
dependiendo de la opción elegida por
el usuario.

Proceso: ACTUALIZA Y MANTIENE MAESTRO

REPITE

Desplegar el menú mantenimiento

Leer opción

CASO de opción igual a:

- 1: Realiza altas en el maestro.
- 2: Realiza bajas en el maestro.
- 3: Modifica el maestro.
- 0: Salir del módulo actualiza
regresar al menú principal.

De otra manera:

Desplegar mensaje de error

HASTA opción =0

FIN ACTUALIZA MANTIENE.

PROCESO : REALIZA ALTAS

OBJETIVO: Dar de alta registros de cuentas en
la base de datos.

Proceso: REALIZA ALTAS.

REPITE

Desplegar pantalla altas

Leer cuenta y nombre

Valida cuenta

Actualiza base de datos

HASTA otra alta = 'no'

FIN REALIZA ALTAS.

PROCESO : REALIZA BAJAS

OBJETIVO:Dar de baja registros de cuentas en
la base de datos.

Proceso: REALIZA BAJAS.

REPITE

Desplegar pantalla bajas

Leer cuenta

Valida cuenta

Borra registro de la base de datos

HASTA otra baja = 'no'

FIN REALIZA BAJAS.

PROCESO : REALIZA CAMBIOS

OBJETIVO: Modificar registros de cuentas en
la base de datos.

Proceso: REALIZA CAMBIOS

REPITE

Desplegar pantalla cambios

Leer cuenta

Valida cuenta

Leer nuevos datos

Sustituir los datos anteriores por los nuevos
en la base de datos

HASTA otro cambio = 'no'

FIN REALIZA CAMBIOS.

PROCESO : CAPTURA Y DIAGNOSTICO

OBJETIVO: Capturar y diagnosticar pólizas de entrada.

Proceso: CAPTURA Y DIAGNOSTICO

REPITE

Desplegar el menú captura

Leer opción

CASO de opción igual a:

1: Captura y diagnostica póliza.

2: Diagnostica póliza.

0: Salir del proceso

regresar al menú principal.

De otra manera:

Desplegar mensaje de error

HASTA opción =0

FIN CAPTURA Y DIAGNOSTICO

PROCESO : CAPTURA

OBJETIVO: Capturar polizas de entrada.

Proceso: REALIZA CAPTURA

MIENTRAS otra captura = 'si'

MIENTRAS No. póliza no sea correcto

lee no.póliza

valida no.póliza

FIN MIENTRAS

MIENTRAS Clave <> 'RH' o 'RD'

MIENTRAS Clave y Cuenta no concuerden

MIENTRAS Clave no este correcta

Lee clave

Valida clave

FIN MIENTRAS

MIENTRAS Cuenta no este correcta

Lee cuenta

Valida cuenta

FIN MIENTRAS

Lee saldos

FIN MIENTRAS

Lee resúmenes

Lee concepto y fecha

Actualiza base de datos

Diagnostica póliza

FIN MIENTRAS

FIN CAPTURA.

PROCESO : DIAGNOSTICA POLIZA

OBJETIVO:Determinar si una póliza "cuadra" o "no
cuadra"

Proceso: DIAGNOSTICO

Leer No.póliza

Válida No.Póliza

SI la póliza es valida ENTONCES

Realiza sumas

Despliega resultados

SI la póliza cuadra ENTONCES

Actualiza la base de datos

mensaje igual a "cuadra"

SI NO

Mensaje igual a "no cuadra"

FIN SI

Despliega Mensaje

SI NO

Despliega mensaje de error

FIN DIAGNOSTICA POLIZAS.

PROCESO : CORRECCION DE POLIZAS CAPTURADAS

OBJETIVO: Seleccionar la actividad a realizar
dependiendo de la opción elegida por
el usuario.

Proceso: CORRIGE POLIZAS.

Lee No.póliza

Valida NO.póliza

SI el no.póliza es válido ENTONCES

REPITE

Desplegar el menú de correcciones

Leer opción

CASO de opción igual a:

- 1: Insertar registros.
- 2: Modificar registros.
- 3: Eliminar registros.
- 4: Eliminar pólizas.
- 0: Salir del sistema.

De otra manera:

Desplegar mensaje de error

HASTA opción =0

FIN DE CORRECCION DE POLIZAS.

PROCESO : INSERTA REGISTRO

OBJETIVO: Insertar registros de pólizas.

Proceso: INSERTA REGISTROS

Lee no.póliza

Valida No.póliza

SI no.póliza es correcta ENTONCES

Lee datos a insertar

Inserta registro a la base

SINO

Mensaje de error

FIN INSERTA REGISTROS.

PROCESO : MODIFICA REGISTRO

OBJETIVO: Modificar registros de pólizas.

Proceso: MODIFICA REGISTRO

Lee no.póliza a modificar

Valida No.póliza

SI no.póliza es correcta ENTONCES

Lee datos a modificar

Modifica registro de la base

SINO

Mensaje de error

FIN MODIFICA REGISTROS.

PROCESO : ELIMINA REGISTRO

OBJETIVO: Eliminar registros de pólizas.

Proceso: ELIMINA REGISTRO

Lee no.póliza a modificar

Valida No.póliza

SI no.póliza es correcta ENTONCES

Lee datos del registro a eliminar

Borra registro de la base

SINO

Mensaje de error

FIN ELIMINA REGISTROS.

PROCESO : ELIMINA POLIZAS

OBJETIVO: Eliminar pólizas de contabilidad

Proceso: ELIMINA POLIZAS

REPITE

Desplegar el menú elimina pólizas

Leer opción

CASO de opción igual a:

1: Elimina pólizas "malas".

2: Elimina pólizas "correctas".

0: Salir del proceso

regresar al menú correcciones.

De otra manera:

Desplegar mensaje de error

HASTA opción =0

FIN ELIMINA POLIZAS.

PROCESO : ELIMINA POLIZAS MALAS

OBJETIVO: Eliminar pólizas de temporal.

Proceso: ELIMINA POLIZAS MALAS

Lee no.póliza a eliminar

Valida No.póliza

SI no.póliza es correcta ENTONCES

Borra todos los registros correspondiente
al no.póliza leído

SINO

Mensaje de error

FIN ELIMINA POLIZAS MALAS.

PROCESO : ELIMINA POLIZAS CORRECTAS

OBJETIVO: Eliminar pólizas diagnosticadas como correctas

Proceso: ELIMINA POLIZAS CORRECTAS

Lee no.póliza a eliminar

Valida No.póliza

SI no.póliza es correcta ENTONCES

Borra todos los registros correspondiente
al no.póliza leído

SINO

Mensaje de error

FIN ELIMINA POLIZAS CORRECTAS.

PROCESO : CREAR REPORTES

OBJETIVO: Emitir reporte de directorio de cuentas

Proceso: CREAR REPORTES

REPITE

Desplegar el menú reportes

Leer opción

CASO de opción igual a:

- 1: Genera reportes numéricos
- 2: Genera reportes alfabéticos
- 0: Salir del proceso
regresar al menú principal.

De otra manera:

Desplegar mensaje de error

HASTA opción =0

FIN CREA REPORTES.

PROCESO : GENERA REPORTES NUMERICOS

OBJETIVO:Emitir reporte numérico de cuentas

Proceso:GENERA REPORTES NUMERICOS

Ordena Maestro por número de cuenta

Imprime

FIN GENERA REPORTES NUMERICOS.

PROCESO : GENERA REPORTES ALFABETICOS

OBJETIVO:Emitir reporte alfabético de cuentas

Proceso:GENERA REPORTES ALFABETICOS

Ordena Maestro por nombre de la cuenta

Imprime

FIN GENERA REPORTES ALFABETICOS.

PROCESO : REALIZA CIERRE DE PERIODO

OBJETIVO:Emitir reportes auxiliares y realizar cierre de periodo

Proceso:REALIZA CIERRE DE PERIODO

Emite Balanza de Comprobación

Emite Diario General

Emite Auxiliar Uno

Emite Auxiliar Dos

FIN REALIZA REPORTES CIERRE DE PERIODO.

PROCESO : EMITE BALANZA DE COMPROBACION

OBJETIVO:Emitir reporte de resúmenes de cuentas de mayor y datos del maestro

Proceso:EMITE BALANZA DE COMPROBACION

Resume registros de cuentas de mayor

FOR CADA registro de resumen de mayor HAZ

Lee del archivo maestro nombre y saldo anterior

Suma saldos

Imprime

FIN FOR CADA

Imprime totales

FIN EMITE BALANZA DE COMPROBACION.

PROCESO : EMITE DIARIO GENERAL

OBJETIVO: Emitir reporte de resúmenes de cuentas de
mayor

Proceso: EMITE BALANZA DE COMPROBACION

FOR CADA registro de resumen de mayor HAZ

Lee del archivo maestro nombre y saldo
anterior

Suma saldos

Imprime

FIN FOR CADA

Imprime totales

FIN EMITE BALANZA DE COMPROBACION.

PROCESO : EMITE AUXILIAR UNO

OBJETIVO: Emitir reporte de resúmenes de cuentas de
mayor y su desglose
Proceso: EMITE AUXILIAR UNO

FOR CADA registro de resumen de mayor HAZ

Lee los registros desglosados asociados

Lee datos del archivo maestro

Calcula totales

Imprime

FIN FOR CADA

Imprime totales

FIN EMITE AUXILIAR UNO.

PROCESO : EMITE AUXILIAR DOS

OBJETIVO: Emitir reporte de resúmenes de cuentas de
auxiliar y su desglose

Proceso: EMITE AUXILIAR DOS

FOR CADA registro auxiliar HAZ

Lee datos del archivo maestro

Lee fecha y concepto de los documentos que
hacen referencia a la cuenta de auxiliar

Calcula totales

Imprime

FIN POR CADA

Imprime totales

FIN EMITE AUXILIAR DOS.

PROCESO : REALIZA CIERRE DE PERIODO

OBJETIVO: Actualizar las cuentas de acuerdo a los
movimientos realizados durante el periodo
contable

Proceso: REALIZA CIERRE DE PERIODO

FOR CADA registro maestro HAZ

Lee datos del archivo maestro

Calcula saldo anterior

Actualiza archivo maestro

FIN POR CADA

FIN REALIZA CIERRE DE PERIODO.

PROCESO : REALIZA ESTADO DE RESULTADOS.

OBJETIVO: Emitir reporte de pérdidas y ganancias.

Proceso: REALIZA ESTADO DE RESULTADOS

FOR CADA registro tabla1 HAZ

Lee cuenta

SI la cuenta es diferente de cero ENTONCES

Busca en el maestro la cuenta

Imprime cuenta , nombre y saldo

Acumula o disminuye según sea el caso, el total.

SINO

Lee registro de tabla2

Imprime letrero

Imprime total

FINSI

FIN-POR-CADA

FIN REALIZA ESTADO DE RESULTADOS.

* La tabla 1 contiene los números de cuenta que aparecerán en el Estado de Resultados, además de la operación a realizar para calcular el total (suma o resta).

* La tabla2 contiene los letreros que deben aparecer en cada corte indicado en la tabla1 con número de cuenta '0000'.

PROCESO : REALIZA BALANCE GENERAL.

OBJETIVO: Emitir reporte de balance general.

Proceso: REALIZA BALANCE GENERAL

MIENTRAS no sea final de la tabla3 HAZ

Lee cuenta y tipo1

MIENTRAS tipo1 sea igual al anterior HAZ

MIENTRAS tipo2 sea igual al anterior HAZ

Busca en el maestro la cuenta

Imprime cuenta , nombre y saldo

Acumula o disminuye segun sea el caso, el total.

FIN MIENTRAS

Imprime la suma del corte

Acumula al gran total

Imprime total

FIN MIENTRAS

Imprime gran total

FIN-MIENTRAS.

FIN REALIZA BALANCE GENERAL.

* La tabla3 contiene los números de cuenta que aparecieron en el Balance General, además como su tipo. El tipo puede ser:

C (activo circulante)
A----- F(activo fijo)
 \
 D (activo diferido)

C (pasivo circulante)
F----- F(pasivo fijo)
 \
 D (pasivo diferido)

** Patrimonio

PROCESO : REALIZA CONSULTAS

OBJETIVO: Consultar pólizas capturadas, ya sean correctas o incorrectas y saldos en maestro.

Proceso: REALIZA CONSULTAS

REPITE

Desplegar el menú consulta

Leer opción

CASO de opción igual a:

- 1: Lee y ordena temporal
Despliega pólizas
- 2: Lee y ordena correctas
Despliega pólizas
- 3: Lee no de cuenta a consultar
Busca reg en maestro
Despliega datos
- 0: Salir del proceso
regresar al menú principal.

De otra manera:

Desplegar mensaje de error

HASTA opción =0

FIN REALIZA CONSULTAS.

Tablas de Condición. -(Decisión)

Una tabla de condición es una herramienta que nos ayuda en el caso de que existan varias condiciones que se deban aplicar simultáneamente o también en el caso en el que sea difícil juzgar si una política es o no completa .

Las tabla de condición esta formada por tres elementos :
Condiciones, Reglas y acciones.

Una Condición utiliza una variable de decisión la cual toma un conjunto de valores .Cada acción debe indicar como proceder para cada una de las reglas que se derive.

El formato de una tabla de condición es la siguiente:

Condiciones	Reglas = 2 ⁿ
C1:	
C2:	
•	
•	
•	
Cn	
Acciones	
A1:	
A2:	
•	
•	
Ai	

Para nuestro sistema de contabilidad la realización de una tabla de condición es necesaria para aclarar los conflictos que pudieran surgir en el proceso de validación de una cuenta y su tipo ,que se presenta en el momento de la captura ,ya que ambos tienen que cumplir con cierta correspondencia específica.

Tabla de condición para la validación de una cuenta y su tipo

Condiciones	Reglas							
Tipo = "mayor"	V	V	V	V	F	F	F	F
Tipo = "parcial"	F	F	F	F	V	V	V	V
Nivel1 = 0	V	V	F	F	V	V	F	F
Nivel2 = 0	V	F	V	F	V	F	V	F
A1:Cuenta y tipo correcto	V	F	F	F	F	V	F	V
A2:Cuenta y tipo no coinciden	F	V	F	V	V	F	F	F
A3:Tipo de cuenta es inválido	F	F	V	F	F	F	V	F

El número de cuenta esta formado por cuenta,nivel1,nivel2 y digito;para este proceso de validación se considera los campos nivel1 y nivel2 de la cuenta.

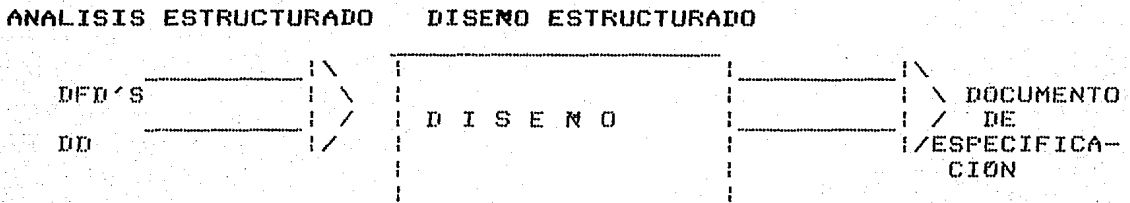
C A P I T U L O I V

D I S E Ñ O D E L S I S T E M A

El diseño del sistema es la etapa de transformación y consiste en el proceso de decidir cuales componentes interconectados y de que manera resuelven algún problema bien definido.

El diseño es la fase donde se indica "cómo" se va a resolver el problema planteado, a diferencia del análisis donde se especifica "qué" se desea resolver.

Como entrada al diseño del sistema se tienen principalmente los DFD's y el DD ,obtenidos en la etapa anterior ,y la salida que se produce es un documento de especificación estructurada.



El objetivo del diseño estructurado es el de desarrollar software a un costo mínimo y con mayor durabilidad ,esto implica crear sistemas flexibles, de alta calidad,eficientes,eficaces y que sean una etapa natural siguiente al análisis .

La filosofía del diseño estructurado, se basa en el principio de dividir un problema, en pequeños problemas con el fin de facilitar el desarrollo, el mantenimiento y las modificaciones sobre el mismo diseño . Se puede afirmar que el diseño estructurado busca atacar la complejidad de un sistema muy grande mediante el uso de dos procedimientos; particionando el sistema en "cajas negras" y organizando jerárquicamente el problema.

Una caja negra cumple con los siguientes características : Se conocen las entradas, se conocen las salidas que producirá, se conoce su función (que hace con las entradas para generar las salidas esperadas), y no es necesario conocer el cómo realiza dicha función.

Las ventajas de el uso de "cajas negras", es que éstas son muy fáciles de construir, su prueba se hace muy sencilla, se pueden corregir fácilmente y son muy entendibles.

Por otra parte la idea de jerarquizar es aún mas antigua que la de particionar ya que todos los sistemas complejos en el universo estan organizados en jerarquias de partes a su vez formadas por otras pequeñas partes .

Con las dos ideas anteriores, la de particionar y la de jerarquizar, el diseño estructurado conforma como su principal herramienta al Diagrama Estructurado, este diagrama ilustra la partición del sistema en módulos (cajas negras) mostrando su jerarquía, organización y comunicación.

Las ventajas de usar un diagrama estructurado es tener una herramienta para el diseño de tipo gráfico, particionable, riguroso pero a la vez flexible, que además sirve como documentación del

sistema, y es una gran ayuda al mantenimiento y modificación del sistema .

Antes de proceder a la realización del diagrama es necesario entender los conceptos esenciales que se expondrán a continuación.

DIAGRAMA ESTRUCTURADO.

Un diagrama de estructura está formado por módulos o cajas negras, entendiéndose como módulo a una colección de instrucciones de programa caracterizados por cuatro atributos: entrada y salida, función, mecanismo, y datos internos.

1. Entrada .- que es lo que necesita el módulo
- salida .- que es lo que regresa el módulo
2. Función .- que es lo que realiza con su entrada para producir la salida
3. Mecanismo .- como se realiza la función
4. datos internos.- datos a los que hace referencia de manera solitaria

El módulo tiene también otros atributos, como el nombre mediante el cual es referenciado dentro de toda la unidad.

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.

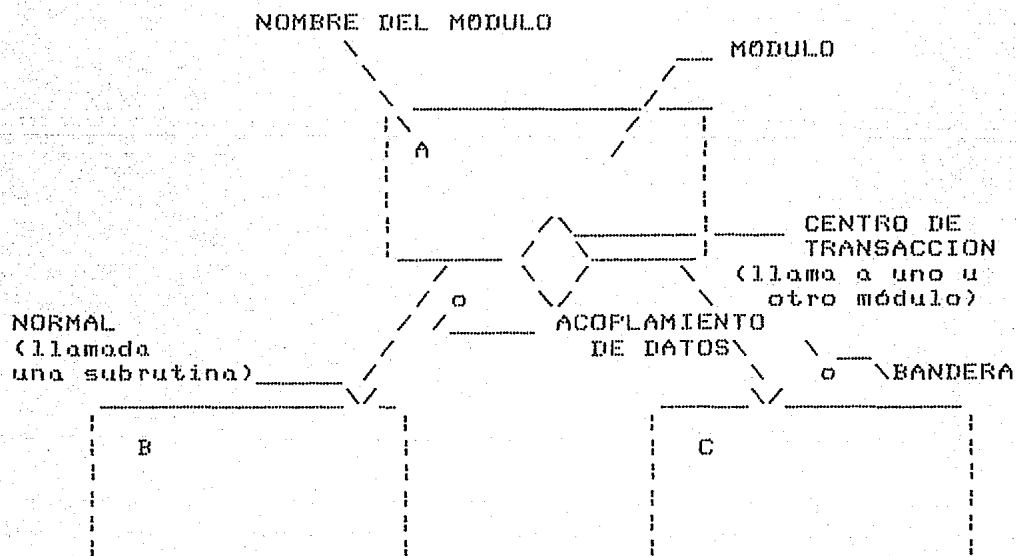
Conecciones entre módulos.

El símbolo que nos permite conectar módulos con módulos es

una flecha, esta flecha muestra una llamada a una subrutina y mediante el sentido de la flecha se puede saber cual es el módulo que invoca al otro módulo subordinado. Los datos estan representados por flechas mas pequeñas en las cuales el sentido de éstas indica de donde ha donde se realiza la transferencia, si la flechita tiene en un extremo un circulo relleno, indica que se trata de la transferencia de una bandera.

En el caso que la invocación de los módulos hijos sea opcional, esto se representa con un rombo en el módulo padre.

Simbología de diagramas de estructura



Cualidades de un buen diseño

Uno de los principios fundamentales del diseño estructurado es el hecho de particionar un gran sistema en módulos

manejables; de esta forma es vital que este particionamiento se realice de tal forma que se obtengan módulos que sean lo mas independiente posible entre sí , a este criterio se le conoce como "acoplamiento", y además se debe perseguir que cada módulo tenga una función bien definida y produzca una salida simple ,a este criterio se le conoce como "cohesión". Juntos acoplamiento y cohesión son los factores que sirven para distinguir de un buen diseño a uno malo. En un buen diseño lo que se pretende es que exista el minimo de acoplamiento entre módulos y se busca una máxima cohesión entre los elementos de cada módulo. Dada la importancia de los criterios de cohesión y acoplamiento, se expone a continuación una explicación de ambos términos.

Acoplamiento.

Este término se define como la medida de interdependencia entre módulos, entonces los módulos altamente acoplados estan unidos por interconexiones rígidas.

El diagrama de estructura persigue el desacoplamiento ya que al aumentar el acoplamiento el mantenimiento se hace complicado .

Podemos distinguir los siguientes tipos de acoplamiento:

- 1.- datos
- 2.- estampilla
- 3.- control
- 4.- común
- 5.- contenido

1) Acoplamiento de datos.

Dos módulos mantienen un acoplamiento de datos si su

comunicación se realiza a través de parámetros ,y estos parámetros son un dato unitario.

Este tipo de acoplamiento es el mas deseado ya que es necesario para la comunicación de datos entre módulos.

2) Acoplamiento por estampilla.

Dos módulos estan unidos a través de un acoplamiento por estampilla si se envían estructuras de datos de un módulo a otro,entendiendo como estructura de datos a un dato compuesto como lo es un registro,un arreglo,etc.

El problema inherente que trae consigo este tipo de acoplamiento es que crea dependencia entre otros módulos ,además de que tiende a exponer al módulo mas datos de los que necesita.

3) Acoplamiento de control.

Dos módulos estan acoplados por control si uno de ellos pasa a otro,información que intente controlar la lógica interna del otro .Además es necesario conocer como está organizado lógicamente el módulo invocado dado que el módulo invocador realiza una decisión.

En un acoplamiento de control,el módulo subordinado no es una caja negra por lo que este tipo de cohesión no es muy conveniente.

Elementos de control pueden ser banderas de control,nombres de módulo y etiquetas.

4) Acoplamiento común.

Dos módulos están bajo acoplamiento común si se refieren a la misma área global de datos.

Este tipo de acoplamiento es considerado malo ya que los programas se hacen difíciles de mantener y entender, si es difícil descubrir que dato debe ser cambiado al modificarse un módulo, es más difícil encontrar cuáles módulos deben ser modificados si un dato es cambiado.

5) Acoplamiento por contenido.

Dos módulos exhiben acoplamiento por contenido (o patológico) si uno de ellos hace referencia al interior del otro, por cualquier camino, es decir que este tipo de acoplamiento se da cuando un módulo altera alguna instrucción de otro módulo o se refiere a un dato de otro módulo.

El acoplamiento por contenido es el menos deseado ya que se pierde totalmente el concepto de caja negra dado que se fuerza a que un módulo conozca sobre el contenido e implementación del otro.

Cohesión.

La cohesión se define como una medida de fuerza de asociación de la consistencia de los elementos de un módulo, entendiéndose como elemento una instrucción o un grupo de instrucciones, una llamada a otro módulo, es decir un pedazo de código que realiza algún trabajo.

Además la cohesión junto con el concepto de acoplamiento nos permite medir la forma de realización de las particiones.

Un buen diseño se debe caracterizar por módulos que mantengan un alto grado de cohesión entre sus elementos ,es decir que estén genuinamente relacionados unos con otros y por otra parte se debe de evitar que los elementos de un módulo se relacionen con los de otro para evitar un alto grado de acoplamiento.

Se pueden distinguir los siguientes tipos de cohesión:

- 1.- Funcional
- 2.- Secuencial
- 3.- Comunicacional
- 4.- Procedimiento
- 5.- Temporal
- 6.- Lógica
- 7.- Coincidental

1) Cohesión Funcional.

Un módulo funcionalmente cohesivo donde todos sus elementos contribuyen a la ejecución de una y solo una tarea.

Ejemplos de módulos funcionales:

Calcula el coseno de un ángulo

Calcula el salario neto de un empleado

Verifica sintáxis alfabética

Es conveniente notar que todos estos módulos tienen un propósito fuerte .En un módulo completamente funcional los elementos estan relacionados por principios secuenciales.

2) Cohesión Secuencial.

Un módulo tiene cohesión de tipo secuencial si sus elementos

están envueltos en actividades de tal forma que la salida de una actividad sirve como entrada a la siguiente.

Un módulo con cohesión secuencial usualmente tiene buen acoplamiento y es muy fácil de mantener, tanto como un módulo con cohesión funcional, la única desventaja real que presenta el tipo cohesión secuencial, es que no es tan fácil de volver a usar en el mismo sistema o en otros, como uno con cohesión funcional, ya que contiene actividades que no son generales en su conjunto.

3) Cohesión comunicacional.

Un módulo comunicacionalmente cohesivo es aquel cuyos elementos contribuyen a actividades que utilizan los mismos datos de entrada o salida, es decir que todas las tareas están en función de un parámetro. Para mayor claridad se presentan los siguientes ejemplos:

Actividades

- a) Busca título del libro
- b) Busca precio del libro
- c) Busca código del libro
- d) Busca autor del libro

Todas las actividades anteriores están relacionadas, ya que siempre se refieren al mismo dato de entrada, "el libro", lo que hace que el módulo tenga una cohesión comunicacional.

Un módulo con este tipo de cohesión todavía es mantenible, aunque con más problemas que en los casos anteriores.

4) Cohesión de Procedimiento.

La cohesión de procedimiento se da cuando todos los elementos están envueltos en diferentes y, muy posiblemente,

tareas no relacionadas ,en el cual el control fluye de una actividad a la siguiente .Ejemplo

Escribe

Lee

Edita

Los módulos con cohesión de procedimiento están compuestos por varias funciones, las cuales, están parcialmente relacionadas unas con otras, de cualquier forma, probablemente estas funciones están relacionadas con funciones de otros módulos. Es típico que en un módulo, que presente este tipo de cohesión, los datos que llegan a él y los datos que regresa tienen alguna relación.

5) Cohesión temporal.

Los módulos que presentan cohesión temporal, son aquellos cuyos elementos están envueltos en actividades relacionadas en el tiempo. Un ejemplo clásico de este tipo de cohesión son los que presentan los módulos de inicialización.

Los módulos con cohesión temporal al igual que los que presentan cohesión por procedimiento, tienden a estar compuestos por funciones parciales donde la única interacción entre éstas, es que todas producen su salida en un tiempo definido.

6) Cohesión Lógica.

Un módulo con cohesión lógica, es aquel cuyos elementos contribuyen a actividades de la misma categoría general, en la cual la actividad o actividades a ser ejecutadas se seleccionan desde el exterior del módulo.

Un módulo que presenta este tipo de cohesión , contiene un

número de actividades de la misma clase ,al usar el módulo sólo se selecciona la parte que se necesita.

Un ejemplo de este caso de cohesión, es un menú o un manejador de errores, en estos casos este tipo de cohesión no es grave.

7) Cohesión coincidental.

La cohesión de tipo coincidental se presenta en un módulo cuando sus elementos contribuyen a actividades sin relación alguna entre ellas, es por esto que este tipo de cohesión es el peor de todos ya que provoca errores menos comprensibles, y el módulo se vuelve no mantenible.

Para poder determinar cual es el tipo de cohesión que presenta un módulo , podemos basarnos en el nombre de éste, tomando en consideración que el nombre del módulo, siempre debe describir fielmente la función del módulo.

Un módulo con cohesión funcional, se puede identificar porque en su nombre lleve un verbo imperativo(lee, calcula, deduce, etc).

Un módulo con cohesión secuencial se reconoce como un número de funciones ensambladas como "Valida transacción y actualiza registro maestro" .

Un módulo con cohesión comunicacional se distingue porque presenta funciones no secuenciales que trabajan con los mismos datos, por ejemplo: Calcula promedio y salario máximo del empleado.

La cohesión de procedimiento se conoce porque el módulo muestra nombres que indican que existen varias tareas no

relacionadas como "módulo switch" o "rutina de loop".

Un módulo con cohesión temporal muestra nombres que guardan relación con el tiempo como "final del job", "inicializa", "limpia".

La cohesión lógica se distingue porque la descripción de la proposición no contiene un objetivo específico, por ejemplo:

"edita módulo", "entrada del módulo".

Para finalizar un módulo cuyo nombre no dice nada, muestra claramente cohesión coincidental. ejemplo:

Rutina de proceso.

Rutina X-100-RTPQ.

Ya que se han analizado las herramientas que nos permiten evaluar y refinar nuestro diseño, el enriquecer éste ya que ha sido implementado puede convertirse en un proceso tedioso, afortunadamente el diseño estructurado proporciona dos muy buenas estrategias que derivan en un buen y rápido diseño, estas estrategias se conocen como Análisis de Transformación y Análisis por Transacción.

Análisis de Transformación.

Es una de las estrategias principales para diseñar sistemas altamente balanceados, también se le conoce como diseño de la transformación central. Para el análisis de transformación se requiere del diagrama de flujo de datos obtenido en la etapa de análisis del sistema, obteniendo como resultado un diagrama de estructura, en el que el módulo superior trabaja con datos altamente procesados, datos lógicos.

La estrategia se puede resumir de la siguiente forma:

1. Obtener el DFD del problema

2. Identificar la transformación central, esto puede realizarse siguiendo las ramas aferentes eferentes del DFD, hasta encontrar el punto en que los datos ¹ son independientes de los ² dispositivos de entrada-salida, los procesos entre estos puntos forman la transformación central.

3. Identificar las ramas principales de datos de entrada y salida. Determinar los puntos de máxima abstracción.

4. Diseñar la estructura a partir de la información previa con un módulo para cada rama principal de entrada y un módulo para cada rama principal de salida.

5. Para cada uno de los módulos de entrada, identificar la última transformación necesaria para producir los datos en la forma en que los regresa el módulo. Después identifique la forma de la entrada justo antes de la última transformación.

Para módulos de salida, identifique el primer proceso necesario para acercarse a la salida deseada con el formato deseado de resultados.

El análisis por transformación es la mejor estrategia para convertir un DFD en un diagrama de estructura, de cualquier forma existe una técnica suplementaria llamada análisis por transacción el cual es conveniente para realizar un diagrama de estructura de un sistema que procesa transacciones.

1. Módulo aferente es el que manda información de abajo hacia arriba (el jefe).

2. Módulo eferente manda información de arriba hacia abajo (hacia los subordinados).

Análisis por transacción.

Esta estrategia tiene dos usos principales, primero: permite simplificar un DFD muy complicado en pequeños DFD's, uno por cada transacción que el sistema procese, estos pequeños DFD's son suficientes para ser convertidos en el diagrama de estructura usando el análisis por transformación, segundo: el análisis por transacciones puede ser usado para combinar diagramas de estructura individual de transacciones separadas en un gran diagrama de estructura flexible a cambios.

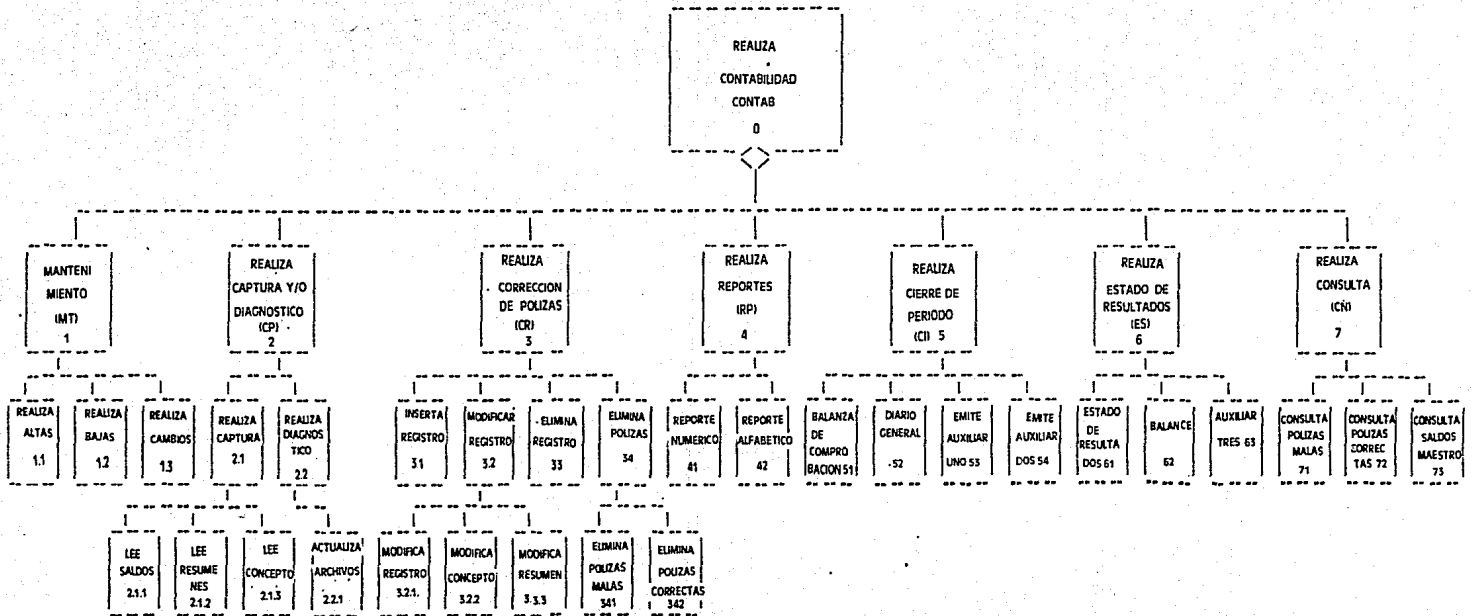
El análisis por transacción tiene el siguiente principio fundamental: separar las múltiples transacciones por tipo y no como un proceso común, en otras palabras se debe encontrar que tan similar o diferentes son los tipos de módulo, donde debe existir un módulo separado el cual es el responsable del proceso de cada tipo de transacción, a este módulo se le conoce como módulo de transacción, existen tantos módulos de transacción como tipos de transacciones haya en el sistema. Existe otro tipo de módulo el cual determina el tipo de cada transacción del sistema total, el cual es llamado módulo de transacción central.

Un tipo de centro de transacción se encuentra frecuentemente en sistemas en línea, en los cuales se le da un menú al usuario, para que seleccione una de varias opciones.

Normalmente es muy fácil reconocer transacciones, centros de transacciones, y burbujas de procesos de transacciones en un DFD simplemente con ver el diagrama, de cualquier forma, el flujo de datos entre las burbujas determina su tipo y siguiendo su ruta

en varias direcciones dependiendo de su tipo, se puede localizar una transacción encontrando el centro de transacción.

Para nuestro problema en particular, de acuerdo a sus DFD's realizados, en la etapa de análisis estructurado, podemos determinar, que en este caso, es necesario usar el método de análisis por transacción para realizar el diagrama de estructura.



Selección Del Lenguaje de Programación.

Después de tener bien definido el problema y haber diseñado los procedimientos que forman parte del sistema, es de suma importancia la realización de la selección del lenguaje en que se llevará a cabo la implantación del sistema.

El factor mas importante a considerar en la selección del lenguaje es la minimización del costo de vida de la programación incluyendo tanto el costo de desarrollo, como el de mantenimiento.

Entre los aspectos mas importantes a considerar dentro de la selección del lenguaje podemos considerar los siguientes: disponibilidad del compilador, tiempos en la realización de la programación, versatilidad, tiempo y costo de desarrollo, si cuenta con estructuras de control, transportabilidad, facilidad de aprendizaje, documentación entre otros.

Existen lenguajes que por su naturaleza quedan fuera del contexto de este análisis, tal sería el caso de lenguajes como Fortran el cual esta orientado hacia aplicaciones mas bien científicas que de cualquier otro tipo.

Tomando en cuenta que el sistema diseñado es de tipo meramente administrativo las opciones que se presentan son las siguientes:

- 1.- Uso de un lenguaje meramente administrativo como COBOL.
- 2.- Uso de un intérprete como Basic.
- 3.- Uso de un administrador de Base de datos para microcomputador.

1.-Las ventajas que se presentan al escoger la primera alternativa es que el lenguaje Cobol es muy estándar, además de que da muchas facilidades para el manejo de archivos, sin embargo presenta a mi parecer una gran desventaja, que no es un lenguaje que permita por naturaleza programar en forma estructurada ya que no tiene implementadas las estructuras repetitivas Repeat-Until, Do-While y For-To, por lo que realizar un sistema modular es complicado ya que en este tipo de lenguajes es muy común el uso de la instrucción GOTO, otra de las desventajas que presenta Cobol es que para realizar un sistema algo complicado las líneas de código son muchísimas mas que las que se tendrían si se usara cualquier otro lenguaje.

2.- El uso de un lenguaje como Basic, presenta la ventaja de que es un lenguaje estructurado y además como puede ser interpretado, la programación es mas rápida, por otra parte no ocupa mucho espacio en disco, sin embargo el manejo de archivos y generación de formas de captura y reportes es complicada.

3.- La tercera alternativa que se presenta es la de utilizar un manejador de base de datos para microcomputadora, para nuestro caso el manejador de base de datos a considerar es DBASEIII el cual presenta las ventajas propias de un manejador de base de datos en general, estas ventajas son las siguientes:

3.1. La posibilidad de administrar de una manera adecuada a las funciones interrelacionadas de la organización.

3.2. La descripción de la información esta contenida en la base de datos, independientemente de las funciones de programación, liberando así a los programadores del manejo de los

datos.

3.3 Al integrarse los datos de diferentes aplicaciones en una sola base, la información repetida puede ser identificada y la redundancia eliminada.

3.4 Una mayor disponibilidad de la información, surge no solo por la integración de los archivos de diferentes aplicaciones, sino también del desarrollo de nuevas aplicaciones que aprovechan la base de datos existente.

3.5 La posibilidad de ampliarse, sin necesidad de un ajuste al sistema.

Las desventajas que se presentan al usar un DBMS es que puede ser muy caro, y puede ocupar mayor cantidad de memoria.

Para el caso particular de DBASEIII el lenguaje de aplicación permite realizar sistemas altamente modulares, además de poseer muchísimas instrucciones que permiten actualizar archivos, y explotar la base de datos de una manera mucho más sencilla y también tiene mucha capacidad en lo que se refiere a información.

De acuerdo a las alternativas propuestas, he decidido que las ventajas que presenta el uso de DbaseIII, son muchas más que las que presenta el uso de cualquier lenguaje de programación, además que el tiempo de desarrollo será mucho menor que si se utilizara un lenguaje, dadas las facilidades para el manejo de archivos.

Características de DBASEIII.

El manejador de bases de datos DBASEIII, consta de 124 órdenes que permiten manipular la información y fue diseñado

para obtener un pleno rendimiento de las microcomputadoras de 16 bits y está escrito en lenguaje C.

DbaseIII puede manejar hasta 1000 millones de registros y hasta 128 campos por registro ,hasta 4000 caracteres puede contener un solo registro ,puede ordenar varios campos a la vez y trabajar hasta con 10 ficheros simultáneamente.Para darnos cuenta de la capacidad de almacenamiento,podríamos decir que si una base de datos contiene 1000 millones de registros de 128 campos cada uno, a la PC le llevaría mas de un mes leer una sola vez la base de datos y dicha base de datos ocuparía 30 millones de diskettes.

Las limitaciones que presenta DBASEIII,es que ha menudo, necesita información almacenada en disco para poder operar , lo que disminuye la velocidad del proceso que se esta ejecutando y además ese retraso es más evidente en un sistema basado en diskettes,que en un sistema con disco duro.

Otra desventaja que presenta DBASEIII es que no está diseñado para sistemas multiusuario o para sistemas que estén conectados en una red local.

Para utilizar DBASEIII se necesitará una computadora de 16 bits que corra los sistemas operativos PCDOS o MSDOS como la IBM PC ,Compaq,Columbia,Corona o EaglePC.

Cualquier microcomputadora que sea completamente compatible a nivel software con la IBM PC podrá usar DBASEIII.

La computadora debe tener como mínimo 256K de memoria y debe usar MSDOS o PCDOS versión 2.0 o posterior,el sistema debe contener dos unidades de diskette o una unidad, mas un disco rígido(debido al tamaño de los ficheros del disco del sistema de DBASEIII se hace impracticable usar DBASEIII en un sistema con

una sola unidad de disco). Se puede usar cualquier impresora que imprima como mínimo 80 columnas de texto .

Por último DbaseIII también permite transferir ficheros entre él y muchos de los programas más populares que están disponibles para la PC ,con la única limitante que el otro programa debe ser capaz de transferir información en un formato aceptable para el DBASEIII (ASCII delimitado o formato de datos de sistema (SFD)).

Por otra parte tomando en cuenta que DBASEIII es un intérprete se ha considerado la necesidad de optimizar el sistema ,para lo cual se usará el Clipper, como compilador de DBASEIII.

Características de Clipper

1) Es mucho menor el tiempo de ejecución del sistema que al usar un intérprete.

2) Mayor seguridad del código fuente ,además que éste es convertido a lenguaje máquina durante la compilación y ligado,este código de máquina el cual es el menor nivel de código posible en una computadora y este no se puede reconvertir a código fuente.

3) Permite usar hasta 64000 variables activas en memoria. y un máximo de 1024 campos por base de datos.

Por otra parte Clipper requiere de una computadora IBMXT o AT o compatibles con IBM PC y se debe de contar con los siguientes requerimientos:

- 1) Sistema operativo PC DOS o MSDOS versión 2.0 o mayor.
- 2) 256K de RAM.

3) 1 floppy disk y un disco duro o dos floppy disk.

Diseño de la base de datos.

Una de las actividades que se deben desarrollar en forma paralela al diseño estructurado es el diseño de la base de datos.

Antes de proceder al diseño es importante señalar las características generales de las bases de datos.

Los primeros esfuerzos en manejo de bases de datos, se iniciaron a fines de los años sesenta, por las necesidades que se tenían de proporcionar acceso efectivo y eficiente a los usuarios y/o programas de aplicaciones, a grandes archivos, los cuales requerían además integrarse en una base. Las estructuras de archivos convencionales, con sus limitaciones características de llaves únicas de acceso secuencial, o un solo tipo de registros, resultaban inadecuadas para ámbitos tan exigentes y ocasionaban tiempos de procesamiento y requerimientos de almacenamiento externo.

Los primeros diseños de sistemas de manejo de bases denominados jerárquicos, invertidas y de red aparecieron a fines de los años sesenta. Surgieron también los sistemas de bases/telecomunicaciones de datos, en los cuales las facilidades de la base y del proceso de datos están muy desarrollados.

Es de primordial importancia, el surgimiento inevitable de aplicaciones de la computadora y de sistemas de información centrados en bases de datos, en lugar de la orientación tradicional de archivos (y sus tradicionales desventajas).

Más recientemente se ha iniciado la integración de diversos sistemas computarizados y sus archivos asociados, dentro de sistemas de información más capaces y provechosos, centrados en

una base de datos con vias a satisfacer a las comunidades de usuarios .

Han surgido muchos retos debido a la necesidad de integrar dentro de bases de datos,archivos y datos diversos con mayor efectividad de costos para usuarios con diferentes prioridades y requerimientos.Entre estos retos se tiene la necesidad de estructuras de datos y posibilidades de organización de archivos,más capaces que las que proporcionan los sistemas de cómputo estándares a través de lenguajes como Cobol,Fortran,PL/1 ;independencia de datos entre programas de aplicación y estructuras particulares dadas a la base;habilidad para que los usuarios compartan datos,irredundancia de la información,alto nivel de rendimiento,eficiencia,seguridad e integridad.

Conceptos fundamentales.

La unidad más pequeña de datos es el campo .Un conjunto de datos elementales constituye un registro lógico o entidad .Un tipo de registro lógico es un registro con una constitución particular de datos elementales;un archivo es una colección de ocurrencias de un mismo tipo de registros y finalmente una base de datos es un conjunto de ocurrencias de varios tipos de registros, en el cual, tanto los tipos de registros como sus ocurrencias están interrelacionadas mediante relaciones específicas .

Una base de datos reside típicamente en dispositivos externos de almacén de acceso directo.

Un sistema de manejo de bases de datos DBMS,es un sistema de programas o software capaz de dar soporte y manejar un banco de

datos integrado.

Habilidad de Relacionar.

Es la habilidad para definir relaciones entre registros o entidades a nivel lógico de manera conveniente, tal como se hace para definir a los registros mismos. Las relaciones son tan importantes y tan susceptibles de identificación como cualquier registro o atributo de los datos y deben poder definirse y manejarse sin ambigüedades por el sistema de base de datos.

Flexibilidad de acceso.

La flexibilidad de acceso es la capacidad de lograr acceso a cualquier lugar de la base de datos en base a cualquier llave y calificación lógica mediante un lenguaje de alto nivel para consultas, no por procedimientos o mediante instrucciones de entrada/salida desde un programa escrito en algún lenguaje convencional de programación .

Seguridad.

Deben existir los mecanismos apropiados para asignar controlar y revocar los derechos de acceso (leer, insertar, borrar, cambiar) de cualesquier usuario a cualquier dato o subconjunto definido de la base.

Arquitectura de un sistema de manejo de bases de datos.

Existen tres tipos principales de modelos de bases de datos para definir registros o entidades y las relaciones que guardan, estos en las estructuras de las bases de datos lógicas:

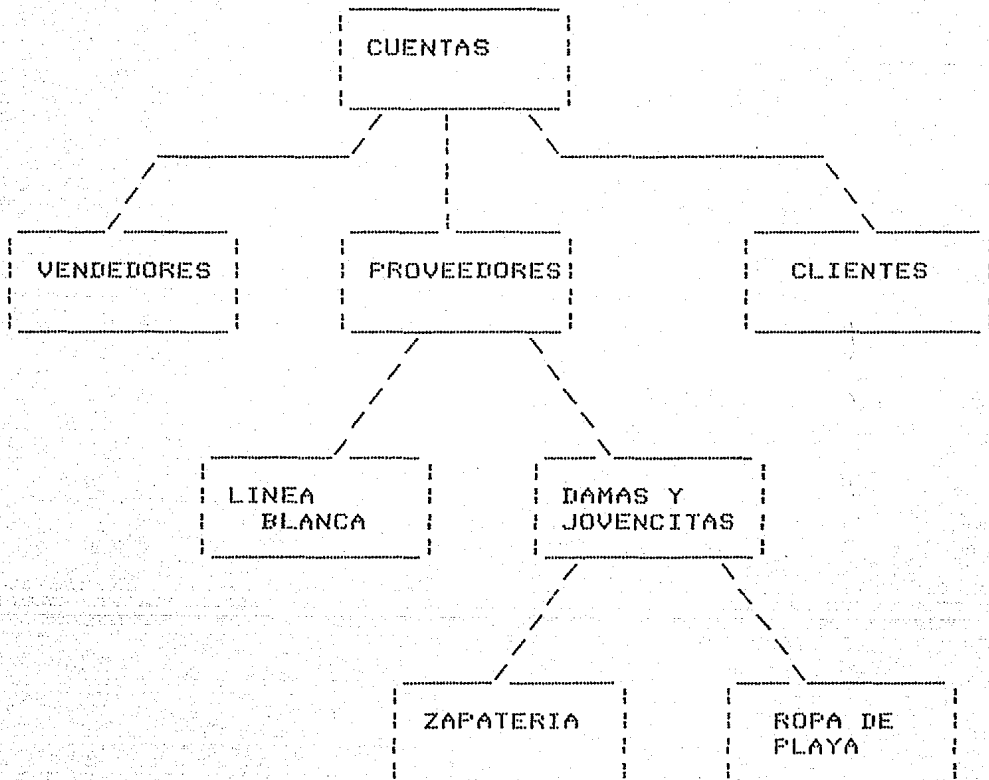
1. Modelo jerárquico
2. Modelo de red
3. Modelo relacional

Estos son los enfoques alternativos para visualizar y manipular datos a un nivel lógico, independientemente de cualesquiera estructuras físicas de soporte en que se basen.

Modelo Jerárquico.

El modelo jerárquico es una simple red ,constituida por una estructura arborescente en la cual todos los apuntadores apuntan en la dirección desde los hijos hacia el padre,es decir puede incluir una o mas entidades inferiores,las cuales pueden contener otras entidades a un tercer nivel y así sucesivamente,sin embargo en este tipo de estructura no pueden existir hijos compartidos por diferentes padres .

Modelo Jerárquico

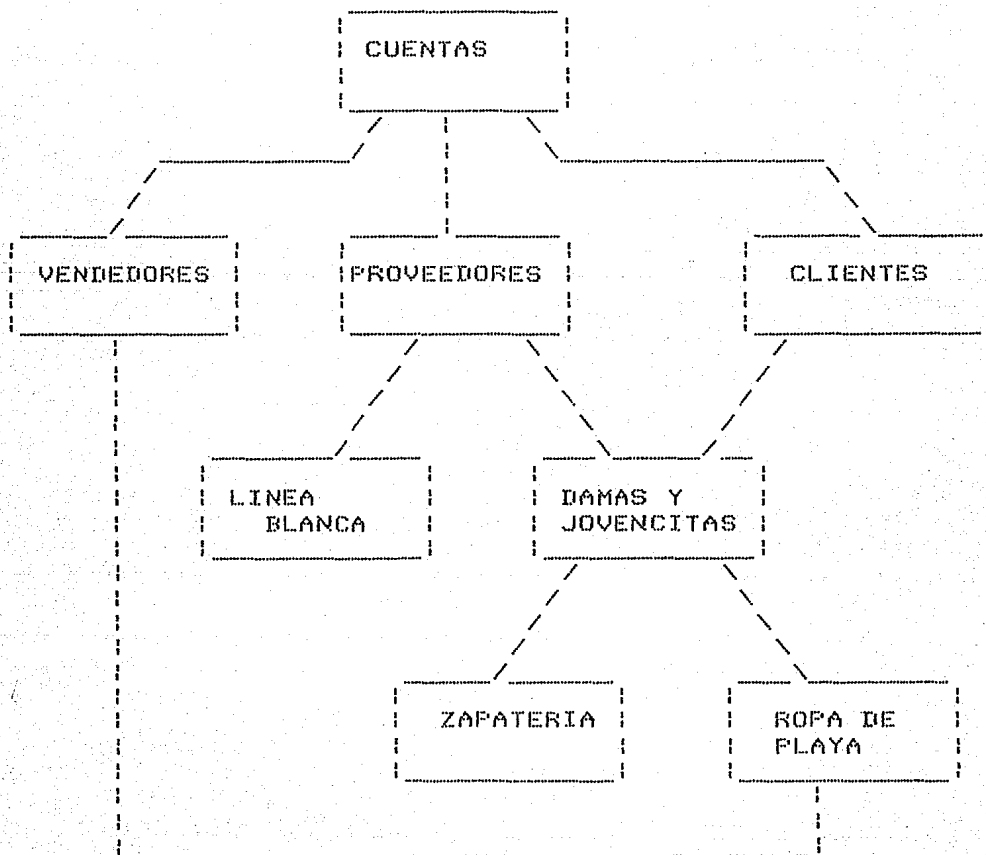


Modelo de datos de red o reticular.

El modelo de red se caracteriza porque todas las relaciones están restringidas a ser de muchos a uno, es decir que los nodos pueden tener uno o varios padres, esta restricción trae consigo el permitirnos usar un diagrama de datos muy simple, además de que la implementación de las relaciones también se hace muy sencilla.

Un ejemplo de este tipo de modelo es el que se presenta a continuación:

Modelo Reticular o de red.



CUENTAS (NO. ARTICULO, descripción, no vendedores, no. clientes)

dominio llave

2. A cada columna, esto es al dominio se le asigna un nombre diferente y esta constituido por valores del mismo dato elemental.

3. Todas las hileras son distintas, no se permiten duplicaciones.

4. Las hileras y las columnas pueden ordenarse en cualquier secuencia en cualquier momento sin que esto afecte el contenido de la formación de la semántica implicada.

Normalización.

La normalización es el proceso mediante el cual un diseñador de bases de datos puede transformar cualquier estructura no plana, arborea o de red a un conjunto de relaciones normalizadas es decir a un conjunto de relaciones planas que no tengan grupos repetitivos. Se dice que una relación no plana es no normalizada. Una relación no normalizada contiene al menos un dominio que es en realidad otra relación, una relación normalizada tiene únicamente dominios simples es decir dominios que no sean a su vez otra relación.

Un archivo que sea plano excepto por un grupo repetitivo se normaliza al quitar el grupo repetitivo y formar con el una relación separada.

La normalización no comprende aspectos de la organización física de la base de datos, únicamente trata con el ámbito del diseño lógico.

Para una base de datos pueden definirse tres tipos de las llamadas formas normales, estas formas proporcionan una mejora en las propiedades de la base de datos respecto a cambios en la misma.

Se dice que una relación se encuentra en primera forma normal si todos sus dominios son simples, es decir si es una tabla bidimensional o plana.

Toda relación normalizada esta automáticamente en primera forma normal. Un proceso adicional de normalización puede tomar una relación en primera forma normal y separarla en relaciones mas simples de las que se dicen estan en segunda forma normal. Otro proceso mas de normalización puede tomar una relación en segunda forma normal y separarlas en relaciones aún mas simples de las que se dicen que estan en tercera forma normal.

De una manera mas formal se define :

Una relación esta en 1FN (primera forma normal), si todos sus atributos son atómicos, definiendo como atributo atómico aquel que se da cuando existe una sola instancia.

Una relación se encuentra en 2FN (segunda forma normal) si esta en 1FN y todos los atributos no llave dependen de la llave completa.

Una relación se encuentra en 3FN(tercera forma normal) si esta en 2FN y no presenta transitividades bajo un conjunto de dependencias funcionales.

A depende funcionalmente de B si para cada valor de B hay cuando mas un valor de A.

El proceso de normalización se puede resumir de la siguiente manera:

1. Tomar la relación en primera forma normal y remover todas las dependencias funcionales incompletas mediante la aplicación de proyecciones apropiadas. Esto produce un conjunto de relaciones de segunda forma normal si cada llave candidata consiste de un solo atributo, entonces la relación en primera forma normal se encuentra automáticamente en segunda forma normal también.

2. Aplicar proyecciones a las relaciones en segunda forma normal, de tal manera que en las relaciones que resulten cada determinante sea una llave candidata o un atributo primo (un atributo primo es uno que sea miembro del conjunto de atributos que constituyen a una llave candidata de atributos múltiples) o en otras palabras, tal que los atributos no primos sean mutuamente independientes.

De acuerdo a toda la teoría de bases de datos resumida en las líneas anteriores se ha podido definir un proceso de diseño de bases de datos para que estas cumplan con las características óptimas de un buen diseño; este proceso se puede desglosar en tres etapas:

- Diseño conceptual
- Diseño lógico
- Diseño físico

Diseño conceptual.

El objetivo del diseño conceptual de la base de datos es obtener un modelo de las necesidades de información, independientemente de las características del sistema de cómputo.

El modelo conceptual de una base de datos es un enunciado estructurado de las entidades que deben almacenarse y de las relaciones que existen entre ellas .

Diseño lógico.

En el diseño lógico de la base de datos se presentan las entidades y las relaciones del modelo conceptual, utilizando las estructuras de algún sistema de administración de datos o las estructuras de un sistema de archivos.

Las relaciones se pueden expresar en un modelo E-R (entidad-relacionamiento) el cual es un modelo de datos no implementado en alguna máquina y sirve para el diseño de sistemas. Este modelo está compuesto por tablas y gráficas que utilizan la siguiente nomenclatura:

- 1.- Un rectángulo representa a una entidad.
- 2.- Los círculos representan a los atributos, los cuales están ligados a sus entidades mediante líneas.

3.- Los rombos representan a las relaciones entre entidades.

Diseño Físico.

En el diseño físico de la base de datos se hacen estudios de los tiempos de acceso para ver si corresponden a los requerimientos de tiempo de respuesta especificado por el usuario. Para mejorar los tiempos de acceso se pueden utilizar estructuras de mas bajo nivel que las que se utilizan a nivel lógico, o bien estructuras lógicas alternativas con redundancia.

Las dos primeras etapas en el diseño de una base de datos son siempre necesarias. En ocasiones, no se precisa del diseño físico, dependiendo del tipo de sistema de administración de datos o del tipo de sistema de archivos con que se cuente.

Para nuestro problema particular se desarrolló el siguiente diseño:

Diseño conceptual.

De acuerdo a lo especificado en la fase de análisis del sistema se necesita almacenar la siguiente información:

Las cuentas del periodo se almacenan en un archivo "maestro" el cual debe contener información sobre los saldos de las cuentas.

Es necesario almacenar las pólizas capturadas ,separando las correctas de las incorrectas.

Las pólizas incorrectas se almacenan en un archivo "temporal",guardando su descripción en un archivo "concepto".

Las pólizas correctas se clasifican según el número de cuenta al que hacen referencia en mayores y auxiliares,quedando las cuentas de mayor almacenados en el archivo de "mayor" y las de auxiliar en el archivo de "auxiliar".

La relación que guardan el archivo maestro con los archivos mayor y auxiliar esta dada por el número de cuenta.

La relación que guardan todos los archivos entre si a excepción del maestro esta dada por el No. de póliza.

Diseño Lógico.

Para el sistema de contabilidad PC se utiliza el administrador de base de datos DBASE III el cual nos permite realizar selecciones,,joins,totalizar registros,actualizar registros de manera automática entre otras cosas.Además es posible abrir 10 archivos a la vez donde el número máximo de campos es de 128,con un número total de caracteres en todos los campos de 4000,y un total de 254 caracteres por campo.,pudiendo tener 1 billón de registros en un archivo.

La definición lógica de la base de datos para el sistema de contabilidad requiere de los siguientes archivos:

Archivo Maestro.

Contiene información de cuentas de contabilidad.

Estructura:

COMAE (cta,niv1,niv2,dig,nom,sd,sh,sma)

Archivo temporal

Contiene información de pólizas incorrectas.

Estructura:

COTEMP (pol,cta,niv1,niv2,dig,tipo,saldo)

Archivo Concepto.

Contiene la descripción y fecha de las pólizas incorrectas

Estructura:

COCON (pol,con,fecha)

Archivo Correcta.

Contiene descripción y fecha de las pólizas correctas.

Estructura:

COCORR (pol,con,fecha)

Archivo Mayor.

Contiene datos de cuentas de mayor que tuvieron movimientos en el periodo.

Estructura:

COMAY (pol,cta,niv1,niv2,dig,tipo,saldo)

Archivo Auxiliar.

Contiene datos de cuentas de auxiliar que tuvieron movimientos en el periodo.

Estructura:

COAUX (pol,cta,niv1,niv2,dig,tipo,saldo)

Archivo Tabla1.

Contiene las cuentas que deben ser incluidas en el Estado de Resultados ,al igual que su efecto sobre el total (suma o resta).

Estructura:

COTA1 (cta,sig)

Archivo Tabla2.

Contiene los letreros que debe de contener el Estado de Resultados.

Estructura.

COTA2 (let)

Archivo Tabla3.

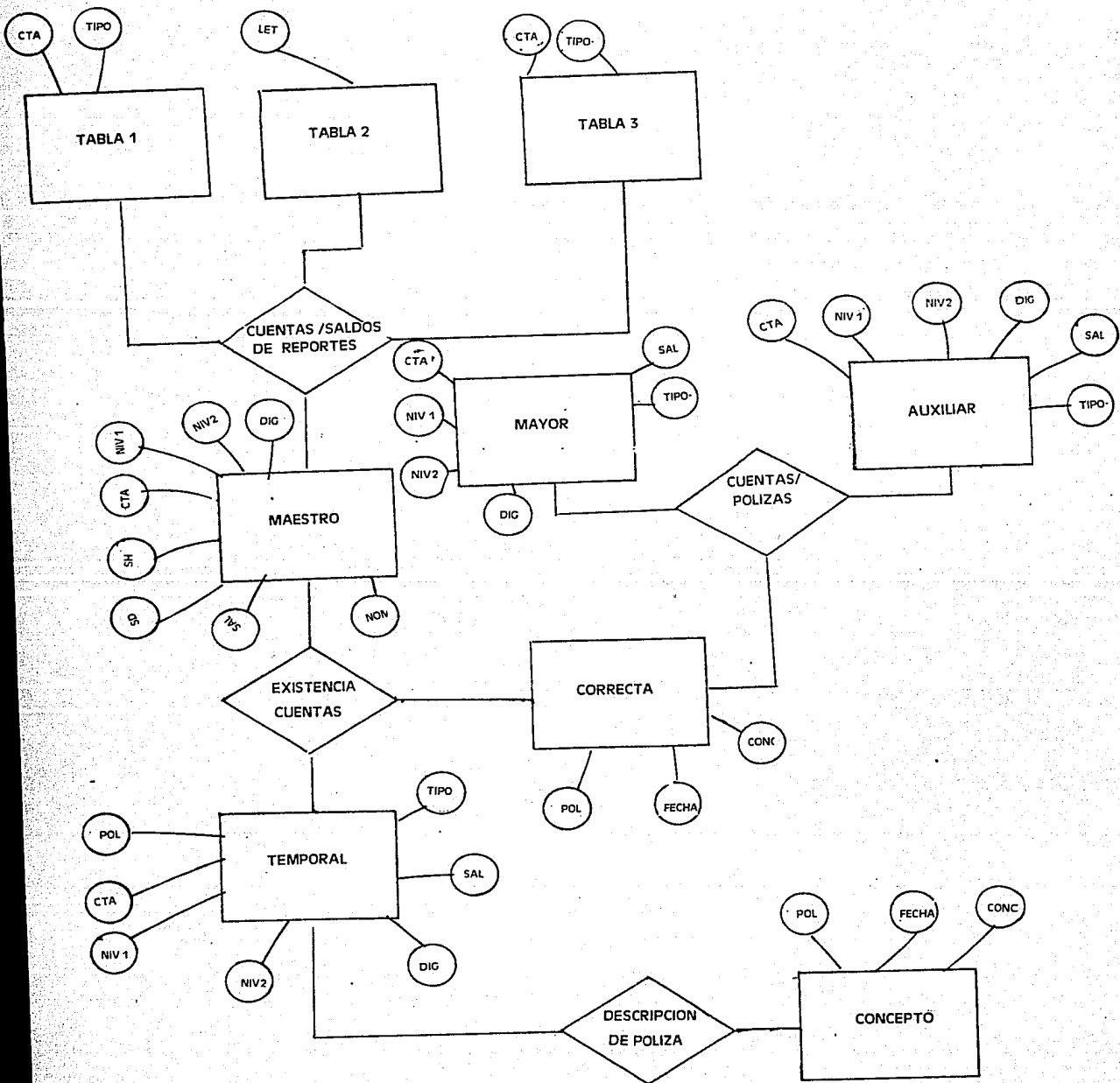
Este archivo se utiliza para el balance general ya que contiene las cuentas que deben de aparecer en este reporte, además de su tipo.

Estructura.

COTA3 (cta,tip1,tip2)

Las relaciones entre los archivos se muestran en el siguiente diagrama de Enter-Relación :

MODELO DE ENTERRACION
 DEL SISTEMA DE
 CONTABILIDAD



Diseño Físico.

En este problema no se tienen requerimientos que demanden un acceso a la base de datos de determinado tiempo, como sucede en sistemas de tiempo real.

C A P I T U L O V I M P L E M E N T A C I O N .

Después de que el sistema ha sido diseñado, la fase siguiente es la que corresponde a la de Implementación del sistema, durante esta etapa se deberá realizar la codificación y documentación de los programas especificados por el diseño del sistema.

Codificación.

La codificación consiste en la traducción de las especificaciones de proceso de cada módulo en instrucciones ejecutables por un lenguaje de programación.

El trabajo de codificación de programas ha permanecido en un estado de transición durante cierto número de años, originalmente era un arte ya que no se contaban con métodos de diseño, actualmente ya existen metodologías de programación como la de programación estructurada, la cual es una técnica para detallar procesos y cuando el lenguaje de programación lo permita se usará también como técnica de codificación.

Se pueden definir dos tipos de codificación: codificación TOP-DOWN o descendente y la codificación BOTTOM-UP o ascendente.

La codificación ascendente permite atender inicialmente los módulos de los niveles jerárquicos mas bajos, ya que estos contienen las operaciones mas primitivas del programa y permite sentar la base para de ahí construir las operaciones mas complejas.

De manera contraria opera la codificación descendente, es decir que primero se codificarán los niveles más altos de la

jerarquía y así sucesivamente hasta llegar a los niveles inferiores.

El sistema de contabilidad desarrollado se codificará usando la técnica de codificación top-down de tal forma que primero se codificará el menú principal ,después los menues siguientes en la jerarquía y después los procesos que son llamados por estos.

Además para mantener uniformidad en los nombres de los programas,se utilizará la siguiente convención,atendiendo al diagrama de estructura de la página 101 del presente documento.

Todos los archivos del sistema de contabilidad comenzarán con las siglas CO a excepción de aquellos ejecutables por el usuario final del sistema.El programa principal tendrá como nombre CONTAB.

Las siguientes dos siglas del nombre, indicarán la función que realiza el módulo subordinado inmediato al nivel cero de manera abreviada, en este caso si observamos nuestro diagrama de estructura contamos con siete módulos a nivel uno, los cuales se definen mediante las siguientes siglas:

MT - Mantenimiento

CF - Captura Y/O Diagnóstico

CO - Correcciones

RF - Reportes

CI - Cierre de periodo

ES - Estado de Resultados

CN - Consulta

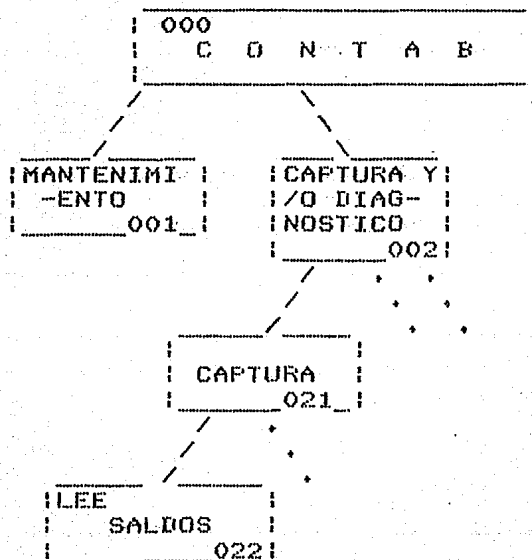
AU - Auxiliares * Para programas de uso general *

Los tres campos siguientes a la descripción de la función

Los tres campos siguientes a la descripción de la función del módulo, corresponden a la posición que guarda el programa dentro del diagrama jerárquico, de tal manera que el número 000 corresponde al primer bloque en la jerarquía que es el del programa principal CONTAB, para los módulos del primer nivel solo se usará un dígito, para los del nivel dos, dos dígitos y así consecutivamente. La numeración de los módulos se realizará de arriba hacia abajo y de izquierda a derecha.

Para que lo anterior quede claro se explicará el siguiente ejemplo:

Si tenemos que el módulo número 2 del diagrama que es el de captura y diagnóstico llama a un programa del nivel dos que realiza la captura y este a su vez llama a un programa que lee los saldos, en el diagrama de estructura se puede apreciar lo siguiente:



El nombre del programa que lee saldos quedara como sigue:

COCF211.PRG

De tal forma que el sistema de contabilidad PC esta formado por los siguientes programas fuente:

- CONTAB.prg
- COMT01.prg
- COMT11.prg
- COMT12.prg
- COMT13.prg
- COCF02.prg
- COCF21.prg
- COCF211.prg
- COCF212.prg
- COCF213.prg
- COCF22.prg
- COCF221.prg
- COCR03.prg
- COCR31.prg
- COCR32.prg
- COCR321.prg
- COCR322.prg
- COCR33.prg
- COCR34.prg
- COCR341.prg
- COCR342.prg
- CORP04.prg
- CORP41.prg
- CORP42.prg

- COC105.prg
- COC151.prg
- COC152.prg
- COC153.prg
- COC154.prg
- COES06.prg
- COC161.prg
- COC162.prg
- Archivos Auxiliares.
- COAU01.prg
- COAU02.prg
- COAU03.prg
- COAU04.prg
- COAU05.prg

Por otra parte para la codificación se utilizaron las miniespecificaciones obtenidas en la etapa de análisis y el diseño realizado del sistema, de tal manera de acuerdo a que a la selección del lenguaje realizada en la etapa de diseño, se optó por la utilización de DBASEIII, en la etapa de codificación se podrán usar de manera total las técnicas de programación estructurada, dado que DBASEIII posee dentro de su conjunto de instrucciones comandos que nos permiten configurar estructuras de decisión como el IF ,, ELSE , con sus respectivos niveles de anidamiento, además de contar también con la instrucción de CASE , para condiciones mutuamente exclusivas. También posee instrucciones que nos permiten realizar estructuras de repetición como el DO WHILE con sus respectivos anidamientos.

Las instrucciones anteriores nos permitirán obtener un sistema bien documentado gracias a la claridad de su estructura.

En cuanto a lo referente a la implementación de la base de datos utilizada, DbaseIII posee varias instrucciones para el manejo de bases relacionales como son la creación, modificación de estructura de la base de datos, sin tener que modificar los programas de aplicación.

Para nombrar los archivos que forman parte de la Base de Datos, se utilizará una estrategia similar a la usada para nombrar los programas de aplicación, con la diferencia que el nombre de un archivo cuenta con ocho caracteres, donde los dos primeros serán CO para indicar que se trata de archivos del sistema de contabilidad, las siguientes siglas corresponden al nombre abreviado del archivo de acuerdo al diseño realizado, y en caso de ser un archivo indexado se utilizarán las siguientes siglas para indicar el campo de ordenamiento. De esta manera el sistema de Contabilidad cuenta con los siguientes archivos de Base de Datos:

COMAE.dbf	* archivo maestro *
COTEMP.dbf	* archivo temporal *
COCON.dbf	* archivo concepto *
COCORR.dbf	* archivo correctas *
COMAY.dbf	* archivo de cuentas de mayor *
COAUX.dbf	* archivo de cuentas auxiliar *
COMAYTOT.dbf	* archivo de resúmenes de mayor *
COAUXTOT.dbf	* archivo de resúmenes de auxiliar *
COTA1.dbf	* archivo tabla 1 *

COTA2.dbf * archivo tabla 2 *
COTA3.dbf * archivo tabla 3 *
COERR.dbf * archivo de errores *

La extensión .DBF corresponde a un fichero de datos de DBASEIII, mientras que la extensión .NDX corresponde a un fichero de índices de DBaseIII .

Por lo que se refiere al nombramiento de las variables utilizadas por los programas de aplicación, se realizó de acuerdo a lo especificado en el diccionario de datos de la etapa de análisis estructurado.

Documentación.

Es conveniente que dentro del código fuente se presente cierta documentación explicando que es lo que se está realizando a manera de comentario, de tal forma que se facilite el proceso de mantenimiento.

El sistema de contabilidad FC como parte de su documentación presenta al principio de cada uno de los programas fuentes, datos como son: Nombre del programa, función, fecha de implantación, y nombre de la persona que lo realizó.

Por otra parte, a lo largo del código se encuentran pequeños comentarios que indican la función de alguna sección con el fin de procurar claridad, además de que el código se presenta sangrado para indicar principios y fines de bloques.

Si se desea una documentación más profunda en cuanto a la realización del sistema es conveniente hacer referencia a los documentos realizados durante la fase de análisis y diseño del mismo.

C A P I T U L O V I

I N T E G R A C I O N Y P R U E B A S

INTEGRACION.

El objetivo principal de esta etapa, es la integración funcional de los módulos de un programa de computadora, ajustándolos a las particularidades del sistema de explotación de cómputo.

La etapa de integración esta formada por dos aspectos: la manera de como se van a combinar los módulos para formar programas y el diseño de pruebas que permiten identificar errores de codificación .

Podemos definir los siguientes tipos de integración:

Integración tradicional o no incremental.

Por lo general este tipo de integración es la que sigue la siguiente secuencia.

-Primero, escribir y probar cada módulo separadamente que es lo que se llama "unit-test".

-Después reunir y probar todas las unidades agrupados en subsistemas (subsystem-test).

-Mas tarde se combinan todos los subsistemas de tal forma que se integre todo el sistema.

-Finalmente se prueba todo el sistema.

El problema que presenta esta forma de integración es que las interfases mas cruciales de sistemas son probadas mas tarde en el proyecto, cuando hay muy poco tiempo de corregir los errores que se pudieran encontrar, ya que los errores emergen hasta el término del proceso mismo.

Integración Incremental.

Esta forma de integración permite detectar con relativa facilidad los errores de programación relacionados con la intercomunicación entre módulos, ya que al integrarse un nuevo módulo ya probado individualmente a un conjunto que ya también fue validado, la presencia de algún error podrá atribuirse en primera instancia a la interconexión modular, además de permitir la identificación de errores en los módulos del sistema adicionados mas recientemente.

La implementación incremental se puede describir a través del siguiente algoritmo:

REPETIR HASTA que el sistema este completo

 Probar un módulo.

 Adicionar ese módulo a la combinación existente.

 Probar la nueva combinación.

FIN REPITE.

 Liberar el sistema.

Este tipo de integración da mas flexibilidad en cuanto al uso de recursos ya que se puede llevar a cabo por muchos caminos.

Podemos definir tres formas de integración incremental:

Integración TOP-DOWN Incremental.

Este tipo de integración consiste en probar el módulo que se encuentra en el nivel superior del diagrama de estructura de cada módulo subordinado será simulado por un módulo falso o stub. Un módulo "stub" es un módulo cuya función es rudimentaria (no hace nada, solo despliega letreros), y se usa en lugar del módulo subordinado real que aún no ha sido integrado. Cada módulo falso

será mas tarde reemplazado por un módulo real hasta llegar a la parte inferior del sistema.

Integración BOTTOM-UP Incremental.

La integración bottom-up se realiza de manera inversa a la top-down, se integra un módulo de los inferiores y se van probando los módulos en forma piramidal hasta llegar al de nivel superior.

La ventaja de este tipo de integración es que por la dirección en que se van probando los módulos ,primero se prueban las entradas y salidas físicas.

Integración SANDWICH Incremental.

Esta forma de integración se caracteriza porque combina las ventajas de los métodos de integración bottom-up y topdown. Primero se integra la mitad del sistema partiendo de la parte superior usando el método top-down y después se integra la otra mitad a través del método bottom-up. Esta técnica presenta la desventaja de que no es posible traslapar las etapas de diseño e integración y pruebas ya que no se puede realizar la integración hasta haber finalizado el diseño total del sistema.

El sistema de Contabilidad PC se integró mediante la técnica de integración incremental bottom-up ,pero considerando como módulos de mas alto nivel los de nivel uno, es decir para el módulo de mantenimiento ,primero se probaron por separado los módulos inferiores correspondientes al módulo uno, se integraron al módulo o módulos inmediatos superiores del módulo uno y así hasta tener integrado totalmente el módulo de mantenimiento.

De la misma manera como se integró el módulo de mantenimiento se integraron al sistema los seis módulos restantes

agrupando al final todos para ser llamados por "CONTAB", esto se pudo realizar de esta manera dado que la comunicación entre los módulos del sistema se da de izquierda a derecha, por lo que si el módulo cinco necesitaba datos o archivos creados en el módulo cuatro, estos ya existían sin tener la necesidad de generarlos de manera externa.

La única desventaja que se presentó es que como se probaron los módulos de abajo hacia arriba, en el caso de existir parámetros o variables globales requeridas por el módulo en prueba, estos se debían de generar para la prueba y cuando se realizaba la integración se debía de eliminar las declaraciones o instrucciones que nos habían servido para realizar esa generación de datos externos, pero esto se presenta contra la ventaja de que cuando se integra el módulo superior inmediato los datos generados por el módulo inferior pueden ser usados tal cual por el módulo en prueba.

Pruebas.

La etapa de pruebas del sistema consiste en la realización de un procedimiento que nos permita conocer si el sistema cumple con su objetivo, en situaciones diferentes, de acuerdo con lo especificado por el usuario.

Las pruebas se definen como un ejercicio del sistema con el fin de encontrar los errores que pudieran existir en éste. Esta fase del sistema va muy ligada a la integración, por lo que las dos etapas constituyen una sola. Primero se realizan pruebas de los módulos por separado, y después de realizarse la integración se deberá probar el sistema como tal.

La fase de pruebas al sistema es crucial y debe de ser

realista y tratar de cubrir muy bien todas las condiciones que se puedan presentar durante la operación del sistema.

A continuación se definirán los tipos de pruebas que existen:

Pruebas de Implantación..- Este tipo de pruebas deben de aplicarse a cada versión del sistema y tienen como objetivo encontrar errores de especificación, concepto y diseño en el ambiente real (equipo, sistema operativo, etc) en donde los programas serán finalmente instalados.

Pruebas del Sistema .- Las pruebas del sistema son realizadas cuando se ha concluido con la integración del sistema. Este tipo de pruebas deben de cubrir cada una de las facetas del sistema en producción :operaciones, interfases con el usuario funciones internas etc, y tiene como objetivo probar la compatibilidad del sistema con la documentación del usuario.

Pruebas de Aceptación .- Este tipo de pruebas, técnicamente son muy similares a las del sistema, pero políticamente son muy diferentes ya que las pruebas de aceptación a diferencia de las otras, requiere de la presencia del usuario final por lo que la preparación de estas pruebas quedarán completamente en manos del usuario o en el caso de que el sistema sea muy grande o muy complejo, conviene responsabilizar de estas pruebas a una tercera institución distinta del usuario y de la que desarrollo el producto .

Para la realización de casos de prueba se definen dos técnicas:

Pruebas de Caja Negra. Esta técnica indica la realización de

pruebas para determinadas entradas al sistema analizando si las salidas correspondientes a dichas entradas son las adecuadas.

Como se podrá observar, en la aplicación de esta técnica no interviene de ninguna manera con el código interno del sistema, es decir que los casos de prueba se realizan sin importar la forma en que fue desarrollado el sistema.

Dado que las posibles entradas a un sistema por lo general son infinitas, el hecho de observar el comportamiento del sistema a todas las entradas repercutiría en un aumento en el costo del producto además de realizarse una gran inversión de tiempo, es por esto que se recomienda para la realización de los casos de prueba, agrupar las entradas que cumplan con características similares. Estos grupos deberán cubrir entradas típicas, casos frontera, datos válidos e inválidos.

La selección de casos de prueba es un proceso muy minucioso y muchas veces depende de la habilidad del analista para que la agrupación de las entradas sea lo más acertada posible, ya que se deberá partir de que si una de las entradas del grupo produce una salida errónea todas las entradas de dicho grupo serán, también procesadas incorrectamente por el programa.

Pruebas de Caja Blanca. Esta técnica de pruebas, considera el código interno del sistema. Los casos de pruebas realizados a partir del diseño y código de los programas del sistema.

Los casos de prueba diseñados deben permitir constatar que el código satisface con las especificaciones del diseño.

Para elaborar los casos de prueba es importante la conceptualización de un programa de cómputo como una secuencia de rutas, donde la ruta seleccionada en un caso dado depende de los

datos de entrada .

Es claro que el número de rutas en un programa es infinito, por lo que no es factible el desarrollo de casos de prueba para cada ruta , es por esto que se recomienda la realización de casos de prueba donde se hagan llamadas a todas las posibles combinaciones de estado en las condiciones de cada punto de decisión del programa dado que una porción significativa de los errores en el código sucede en los puntos de decisión.

C A P I T U L O V I I

P U E S T A E N O P E R A C I O N .

La etapa de puesta en operación se inicia con la primera instalación del sistema integrado, después de que éste ya ha pasado todas las pruebas diseñadas. Para lograr la operación del sistema en esta etapa de su vida es donde se deberá realizar un **"Manual de Usuario"** el cual deberá permitir a los operarios del sistema desarrollado, el uso cabal y eficiente de éste, para así poder obtener los resultados esperados.

El Manual del Usuario es un instructivo para instalar y operar el sistema. Este documento debe de permitir al usuario la instalación del sistema en su propio equipo, la realización de la codificación de la información requerida por el sistema, y debe de mostrar al usuario la manera de operar el sistema, indicándole que es lo que puede esperar de cada proceso realizado.

Para el Sistema de CONTABILIDAD PC, en esta etapa de su vida se ha realizado el siguiente manual del usuario.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO.

SISTEMA DE CONTABILIDAD PC

MANUAL DEL USUARIO

DIRECCION GENERAL DE SERVICIOS DE COMPUTO PARA LA ADMINISTRACION

MANUAL DEL USUARIO

Introducción.

La Dirección General de Servicios de Cómputo para la Administración a través del departamento de Asesoría e Implantación de Sistemas ha desarrollado un sistema para la realización de los procesos contables de las distintas dependencias de la UNAM, con el fin de agilizar y facilitar al usuario este proceso.

El sistema fue desarrollado para ser utilizado en microcomputadoras de tipo PC o compatibles.

El propósito de este manual es guiar al usuario para la implementación y operación del sistema.

La primera parte de implementación esta formada por la codificación de la información y las formas de codificación.

La segunda parte (operación), esta formada por los elementos de operación y se explica brevemente los módulos del sistema.

Implementación.

Codificación de la información.

Codificar las cuentas del directorio en la forma correspondiente de acuerdo a las siguientes indicaciones:

Columna	Concepto	Descripción
1	Clave	Identifica la operación a realizar sobre el catálogo de cuentas. existen tres claves: 1 Alta 2 Baja 3 Cambio de Nombre
2 -10	Número de cuenta	Señalar el número de la cuenta con cuatro posiciones para el mayor, dos para el nivel, y tres para el auxiliar.

Columna	Concepto	Descripción
11	Dígito de control	Calcular el dígito de control de la siguiente manera: considerando los dígitos de la cuenta de izquierda a derecha ,se suman los que se encuentran en posiciones nones y se multiplican por siete;se suman los dígitos de las posiciones pares y se multiplican por tres;los resultados se suman y el dígito menos significativo será el dígito de control.
12 - 80	Nombre de la cuenta	Nombre de 1 a 69 caracteres.

Codificar las pólizas de contabilidad de la siguiente manera:

Columna	Concepto	Descripción
1 - 4	Número de póliza	Número consecutivo que mantendrá su secuencia creciente hasta el final del periodo.
5 - 10	Fecha de póliza	Fecha de elaboración de la póliza con el formato ddmmaa
11 - 12	Clave	Tipo de movimiento: MD = Mayor del Debe MH = Mayor del Haber PD = Parcial del Debe PH = Parcial del Haber RD = Resumen del Debe RH = Resumen del Haber
13 - 21	Número de cuenta	Cuenta a la que se le va a realizar el movimiento contable, utilizando cuatro posiciones para el mayor, dos para el nivel y tres para el auxiliar.

Columna	Concepto	Descripción.
22	Digito de control	Calcular el digito de control de la siguiente manera: considerando los digitos de la cuenta de izquierda a derecha ,se suman los que se encuentran en posiciones nones y se multiplican por siete;se suman los digitos de las posiciones pares y se multiplican por tres;los resultados se suman y el digito menos significativo sera el digito de control.
23 - 35	Importe Parcial	Importe de la cuenta de auxiliar a la que se hace referencia.
36 - 48	Importe del Mayor del Debe	Importe cargado a una cuenta de mayor.
49 - 61	Importe del Mayor del Haber	Importe abonado a una cuenta de mayor.

Al final de la póliza se deberá registrar los totales de los abonos y de los cargos, así como la descripción de la póliza.

Columna	Concepto	Descripción
36 - 48	Suma cargos	Total de los cargos.
49 - 61	Suma abonos	Total de los abonos.
	Concepto	Breve descripción del contenido de la póliza.

Operación del Sistema.

El sistema de contabilidad desarrollado fue realizado para su uso en una microcomputadora PC o compatible que puede ser con disco duro o con dos floppydisk .

Los requerimientos mínimos que debe presentar el equipo donde se operará el sistema son los siguientes:256K de memoria y la existencia dentro del directorio raíz del archivo Config.sys,el cual si se despliega deberá contener las siguientes líneas:

```
files = 20
```

```
buffers = 15
```

En el caso de que no existiera este archivo,éste se tendría que crear usando el editor de textos.

Para el caso de tener una microcomputadora con disco duro se recomienda crear un directorio donde quede guardada la información que integra al sistema de contabilidad ,se deberá copiar en ese directorio el diskette del sistema que será proporcionado al usuario .Este diskette contendrá diez archivos de extensión .dbf y los archivos CONTAB.EXE,NPERIOD.EXE e INICIA.EXE ,este último archivo deberá ser ejecutado al correrse por primera vez el sistema y creará archivos de extensión .NTX los cuales serán utilizados por el sistema de contabilidad .

El archivo NPERIOD.EXE deberá ser ejecutado al momento de iniciar un nuevo periodo, es decir que la ejecución de este archivo provocará que se elimine toda la información de pólizas capturadas en el periodo anterior para empezar otro.

El archivo CONTAB.EXE se deberá ejecutar cada vez que se desee correr el "SISTEMA DE CONTABILIDAD FC" .

Para ejemplificar como se realiza lo especificado anteriormente usando los comandos de MS-DOS se muestra el siguiente resumen; desglosándolo en pasos:

1. Creación del directorio CONTA:

C.MKDIR CONTA

2. Introducción al directorio CONTA:

C.CD\CONTA

3. Copia del diskette del sistema proporcionado por la DGSCA.

C.COPY A:*.*

4. Para ver cuáles son los archivos que existen en el directorio:

C.DIR/W

Si se tecldea el comando dir después del copy efectuado en el paso tres, el sistema desplegará lo siguiente:

Directory of C:\conta

```

*
CONTAB  EXE      **  INICIA  EXE      NPERIOD  EXE
COMAE   DBF      COTEMP  DBF      COCON    DBF
COCORR  DBF      COMAY   DBF      COAUX    DBF
COMAYTOT DBF      COAUXTOT DBF      COTA1    DBF
COTA2   DBF
13 File(s)
```

5.Ejecución de los archivos ejecutables.

C.INICIA

C.NPERIOD

C.CONTAB

Los pasos 1 y 3 solo deberán realizarse cuando se va a ejecutar por primera vez el sistema ,posteriormente solo será necesario teclar "CONTAB".

Para el caso de tener dos floppys el diskette del sistema deberá quedar en el drive A y en el drive B se guardarán los datos generados por el sistema.Para llamar a ejecución del sistema bastará con estar en el drive A e invocar al sistema con la palabra CONTAB.

El sistema esta constituido por siete módulos de tal forma que al teclar CONTAB se desplegará el siguiente menú:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

M E N U P R I N C I P A L

- 1 .- ACTUALIZACION Y MANTENIMIENTO
- 2 .- CAPTURA Y/O DIAGNOSTICO DE POLIZAS
- 3 .- CORRECCION DE POLIZAS
- 4 .- REPORTES
- 5.- CIERRE DE PERIODO
- 6.- ESTADO DE RESULTADOS
- 7.- CONSULTA DE POLIZAS
- 0.- FIN
- ELEGIR OPCION

De acuerdo a la opción seleccionada se ejecutará el módulo respectivo de la siguiente manera:

1. Módulo de Actualización y Mantenimiento.

Este módulo presenta el siguiente submenú:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

A C T U A L I Z A A R C H I V O M A E S T R O

1. ALTAS

2. BAJAS

3. CAMBIOS

0. FIN

- ELEGIR OPCION

1.1 ALTAS.

Solo se darán de alta las cuentas que no existan en maestro ,al seleccionar esta opción se presentará en pantalla lo siguiente:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

A L T A S

HORA : 11:26:37

CUENTA:

SUBCUENTA:

AUXILIAR:

DIGITO:

NOMBRE:

Después de dar la información anterior se desplegará :

OTRA ALTA ? S/N

1.2 BAJAS.

Solo se darán de baja las cuentas que tengan saldo final en
ceros ,al seleccionar esta opción se presentará en pantalla lo
siguiente

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

B A J A D E C U E N T A S

HORA : 11:26:37

CUENTA:

SUBCUENTA:

AUXILIAR:

DIGITO:

Enseguida preguntará:

ESTA CORRECTA LA INFORMACION ?

Posteriormente preguntará :

OTRA BAJA ? S/N

1.3 CAMBIOS.

Realiza el cambio en el nombre de la cuenta deseada.

Al seleccionar esta opción se presentará en pantalla lo siguiente:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

C A M B I O S

CUENTA:

SUBCUENTA:

AUXILIAR:

DIGITO:

En seguida preguntará:

ESTA CORRECTA LA INFORMACION ?

El sistema responde de la siguiente forma:

NOMBRE : Nombre de la cuenta

NUEVO NOMBRE ? : _____

2. Módulo de Captura y/o diagnóstico de pólizas.

Al escoger la opción de captura y/o diagnóstico del menú principal inmediatamente se desplegará el siguiente submenú:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

C A P T U R A Y / O D I A G N O S T I C O D E P O L I Z A S

1. CAPTURA Y DIAGNOSTICO
2. DIAGNOSTICO DE POLIZAS
0. FIN DE CAPTURA

- ELEGIR OPCION

2.1 Captura y diagnóstico.

A través de esta rutina es posible capturar y diagnosticar pólizas de contabilidad, donde inicialmente el número de la póliza a capturar y la fecha de ésta .

NUMERO DE POLIZA ? [0000]

FECHA ddmmaa : [- -]

Se valida el número de póliza y si éste es correcto se captura la clave y la cuenta:

CLAVE []

CUENTA : []

SUBCUENTA : []

AUXILIAR : []

DIGITO : []

La clave puede ser:

MH = Mayor del Haber

MD = Mayor del Debe

PH = Parcial del Haber

PD = Parcial del Debe

RD = Resumen del debe

RH = Resumen del Haber

Si la clave y el número de cuenta son válidos el sistema
pedirá el saldo:

CUENTA : _____
SALDO : \$ [\$\$\$\$\$\$\$\$\$0.00]

Una vez que haya terminado de proporcionar la información de todos los movimientos ,dar el importe de los resúmenes, con sólo teclear RD pedirá el importe del resumen del Debe y enseguida pedirá el resumen del Haber, de la siguiente manera:

R E S U M E N D E L D E R E

SALDO : \$

R E S U M E N D E L H A B E R

SALDO : \$

Después de haber leído los saldos de los resúmenes, el sistema pedirá al usuario que teclee el concepto correspondiente a la póliza capturada de la siguiente manera:

TECLEAR CONCEPTO

Una vez capturada la información, el sistema realiza el diagnóstico de la póliza, en caso de tener errores desglosa el tipo de error del que se trata, si esta correcta lo presenta de la siguiente forma:

DIAGNOSTICO

DEBE. SUMA DE MAYORES : 56.00
 DEBE. SUMA DE PARCIALES : 56.00
 DEBE. RESUMEN LEIDO : 56.00
 HABER. SUMA DE MAYORES : 56.00
 HABER. SUMA DE PARCIALES : 56.00
 HABER. RESUMEN LEIDO : 56.00

>>>>>>> POLIZA CUADRA <<<<<<<<

PULSAR CUALQUIER TECLA PARA CONTINUAR []

Al pulsar alguna tecla el sistema presenta la póliza capturada de la siguiente manera:

CVE	CUENTA	PARCIALES	MAYOR-DEBE	MAYOR-HABER
MD	1201000006	0.00	56.00	0.00
PD	1201010016	56.00	0.00	0.00
MH	1301000009	0.00	0.00	56.00
PH	1301015110	56.00	0.00	0.00
RD			56.00	
RH				56.00

PULSAR CUALQUIER TECLA PARA CONTINUAR []

Inmediatamente después de pulsar alguna tecla el sistema desplegará la siguiente pantalla:

CONCEPTO : Compra de Equipo de cómputo

FECHA : 02-02-1987

DESEA OTRA CAPTURA S/N ?

2.2 Diagnóstico.

Esta opción sirve para diagnosticar una póliza que ya ha sido capturada y corregida previamente, con teclear la opción 2 del menú de captura se presenta lo siguiente en pantalla:

NO. POLIZA : []

Al ser capturado el número de póliza si este es correcto se presentarán en pantalla los diagnósticos de la misma forma, como salen en la parte de diagnóstico de la opción de captura y/o diagnóstico.

3. Módulo de Correcciones.

El uso de este módulo permite al usuario corregir las pólizas capturadas que al momento de diagnosticarlas hayan tenido errores, para entrar a este módulo basta con teclear el número "3" en el menú principal y automáticamente aparecerá el siguiente submenú de correcciones:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

C O R R E C C I O N D E P O L I Z A S

1. INSERTA REGISTROS
 2. MODIFICA REGISTROS
 3. ELIMINA REGISTROS
 4. ELIMINA POLIZA
 0. FIN DE CORRECCIONES
- ELEGIR OPCION

NUMERO DE POLIZA A MODIFICAR ?

3.1 Inserta registros.

Este proceso nos permite insertar registros a pólizas capturadas previamente pidiendo la información de la siguiente forma:

I N S E R C I O N D E R E G I S T R O S

CUENTA:

SUBCUENTA:

AUXILIAR:

DIGITO:

TIPO (MD=1,PD=2,MH=3,PH=4):

TIPO :

SALDO:

3.2.1 Modifica Registros .

Esta parte del sistema permite al usuario modificar información de registros de pólizas ya existentes y diagnosticadas como incorrectas, al elegir la opción "2" del menú de modificaciones el sistema presentará el siguiente menú:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

M O D I F I C A C I O N

- 1. MODIFICA REGISTROS
- 2. MODIFICA CONCEPTO
- 0. FIN

- ELEGIR OPCION

La opción de "modifica registros" nos muestra la información actual del registro a modificar y nos permite colocarnos en el campo que se desea cambiar para teclear ahí la información deseada, presentando la siguiente pantalla:

M O D I F I C A C I O N D E R E G I S T R O S

POLIZA :001

CUENTA : SUBCUENTA:

AUXILIAR: DIGITO:

TIPO (MD=1,PD=2,MH=3,PH=4):

Si el registro, existe el sistema desplegará el saldo del registro:

SALDO:\$ 4555.99

3.2.2 Modifica Concepto.

El sistema también nos permite modificar el concepto de la póliza deseada con teclar la opción "2" del menú de modificación de registros, al pedir esta opción el sistema nos pedirá el número de póliza, desplegará el concepto actual y pedirá el nuevo concepto de la siguiente manera:

M O D I F I C A C I O N D E C O N C E P T O

POLIZA NUM: 001

CONCEPTO ACTUAL : COMPRA DE EQUIPO DE COMPUTO

NUEVO CONCEPTO :

3.3 Eliminación de registros.

Esta parte del sistema permite al usuario borrar registros de pólizas diagnosticadas como incorrectas, para esto el sistema pide al usuario teclar los datos del registro a eliminar a través de la siguiente pantalla:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

E L I M I N A C I O N D E R E G I S T R O S

POLIZA NUM : 0001

CUENTA:

SUBCUENTA:

AUXILIAR:

DIGITO:

TIPO(MD=1,PD=2,MH=3,PH=3)

3.4 Eliminación de pólizas.

Es posible mediante esta opción borrar una póliza ya sea correcta o incorrecta a partir del siguiente menú:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

E L I M I N A P O L I Z A S

1. ELIMINA POLIZAS CAPTURADAS
2. ELIMINA POLIZAS CORRECTAS
0. FIN

- ELEGIR OPCION

Al elegir cualquiera de las dos primeras opciones el sistema pedirá el número de la póliza a ser eliminado y al terminar el proceso mandará un mensaje de que realizó bien la operación.

NUMERO DE POLIZA :

POLIZA ELIMINADA DE TEMPORAL

4. Módulo de Reportes

Esta parte del sistema se encarga de emitir el directorio de cuentas ,ya sea ordenadas de forma numérica ó alfabéticamente. Al invocar este módulo con la opción "4" del menú principal, el sistema desplegará el siguiente menú de reportes:

R E P O R T E S D E C U E N T A S

1. REPORTE NUMERICO DE CUENTAS
2. REPORTE ALFABETICO DE CUENTAS
0. FIN DE REPORTES

- ELEGIR OPCION

Al elegir cualquiera de las dos primeras opciones ,antes de emitir el reporte el usuario deberá teclear la fecha con la que aparecerá el reporte de acuerdo al formato ddmmaa ,de no ser así el reporte aparecerá con la fecha del sistema:

FAVOR DE HABILITAR LA IMPRESORA

FECHA:

Después de dar la fecha el sistema desplegará el siguiente mensaje:

***** IMPRIMIENDO REPORTE NUMERICO *****

ó

***** IMPRIMIENDO REPORTE ALFABETICO *****

5. Módulo de Cierre de Período.

La función de este módulo es la de emitir los reportes de fin de periodo como son :la Balanza de Comprobación, el Diario General y dos reportes auxiliares: Auxiliar Uno y Auxiliar Dos.

Al elegir la opción "5" del menú principal el sistema pedirá la fecha con la que aparecerán los tres reportes:

FAVOR DE HABILITAR LA IMPRESORA

FECHA:

El usuario deberá teclear la fecha con el formato ddmmaa ,después de haberlo realizado y si la impresora esta habilitada se emitirán los tres reportes antes mencionados, uno tras otro empezando por la balanza de comprobación.

6. Módulo de Estado de Resultados

Este módulo se encarga de emitir, al final del cierre de periodo, tres reportes que constituyen los Estados Financieros y que son:

- Estado De Resultados
- Balance General
- Anexo al Balance o Auxiliar Tres

Al seleccionar la opción "6" del menú principal, el sistema pedirá al usuario la fecha con la que aparecerá el reporte:

FAVOR DE HABILITAR LA IMPRESORA

FECHA:

Después de aceptar la fecha el sistema emitirá, uno por uno los tres reportes antes mencionados.

7. Módulo de Consulta de Pólizas.

La función de este módulo es la de permitir al usuario consultar las pólizas correctas y las incorrectas, así como los saldos de cuentas, para esto el sistema ofrece el siguiente submenú:

SISTEMA DE CONTABILIDAD PC

FECHA : 02-02-87

HORA : 11:26:37

C O N S U L T A D E P O L I Z A S

1. FOLIZAS ERRONEAS
 2. FOLIZAS CORRECTAS
 3. SALDO DE CUENTAS
 0. FIN DE CONSULTA.
- ELEGIR OPCION

7.1 Pólizas Erróneas.

Si elegimos la opción "1" del menú anterior, el sistema presentará al usuario todas aquellas pólizas que hayan sido capturadas pero que se hayan diagnosticado como incorrectas, estos datos se presentarán de la siguiente forma:

NUM POLIZA	NOMBRE	FECHA
0001	Compra de Equipo	08-02-87
0002	Papelería	08-02-86
.

7.2 Pólizas Correctas

Si el usuario elige la opción "2" del menú de consultas, el sistema presentará al usuario las pólizas capturadas y correctas, de la misma manera que lo hizo para la opción "1"

7.3 Saldo de Cuentas

De elegir la opción "3" del menú de consulta, el usuario podrá consultar los saldos de cualquier cuenta, el usuario para esto deberá de proporcionar el número de cuenta a consultar de la siguiente manera:

CUENTA:	SUBCUENTA:
AUXILIAR:	DIGITO:

Después de haber leído el número de cuenta, el sistema proporcionará al usuario la siguiente información:

CUENTA:100100010

SALDO MES ANTERIOR: \$ 0.00

SALDO HABER : \$ 0.00

SALDO DEBE : \$ 0.00

8. Fin.

La última opción del menú principal permite al usuario despedirse del sistema de contabilidad PC, al teclear la opción '0' el sistema desplegará el siguiente mensaje:

```

/-----/
|               |
|               |
|      F I N    |
|               |
|      D E L    |
|               |
|      S I S T E M A
|               |
|               |
|-----|
/-----/

```

Y regresará el control al sistema operativo, terminando así la ejecución del Sistema de Contabilidad PC.

C A P I T U L O V I I I M A N T I E N I M I E N T O.

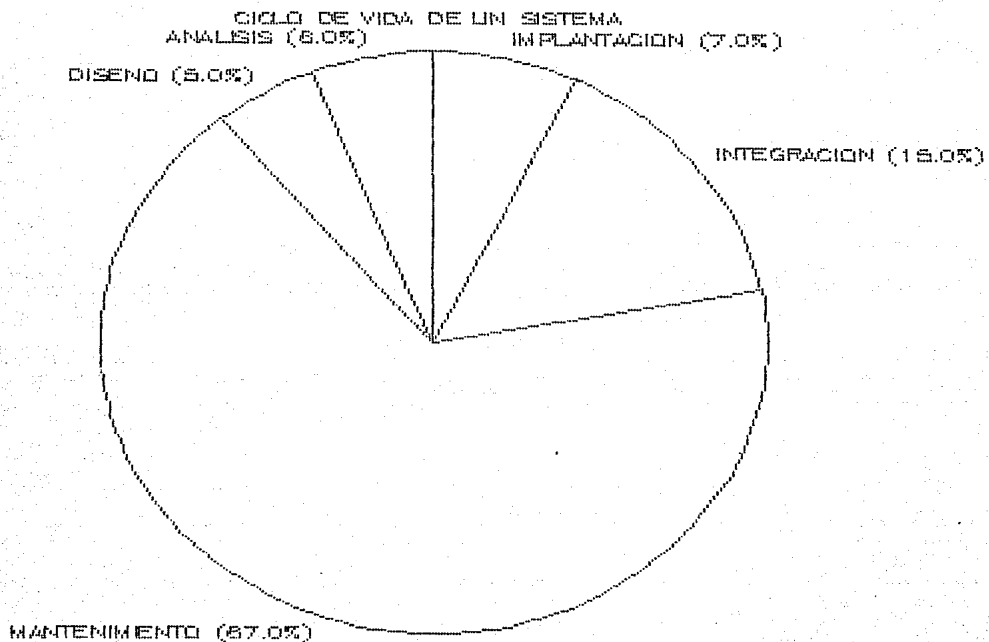
La última etapa de la vida de un sistema es conocida como mantenimiento, y éste se realiza después de que ha sido liberado el sistema durante lo que resta de su vida. En realidad el mantenimiento se define como el proceso de expandir o compactar los productos de software a menor costo.

Los problemas que se presentan durante el mantenimiento de un sistema están dados por la falta de documentación, la incompatibilidad existente entre la documentación y el diseño y la dificultad que se puede presentar al tratar de entender y sobre todo de modificar el sistema.

En realidad se puede afirmar que la realización de las etapas de la vida del sistema anteriores a la etapa de mantenimiento son de suma importancia, ya que el hecho de tratar de realizar algún número de cambios razonables después de liberado el sistema es más frustrante y costoso que si estos se realizaran a través del desarrollo de éste.

El mantenimiento de un sistema es el proceso más caro realizado durante el ciclo de vida de un sistema, para tener una idea del costo de este proceso se presenta a continuación un diagrama de pie, donde se define el porcentaje de costo de cada una de las etapas del ciclo de vida del sistema:

GRAFICA DE COSTO DE LAS ETAPAS DEL



Dado el gran costo de esta etapa, se considera muy importante la realización de sistemas funcionalmente modulares, como se definió en el capítulo de diseño, y el uso de técnicas que nos permitan alargar la vida del sistema al realizar un mejor mantenimiento, estas técnicas se dividen en dos grupos:

- Técnicas de tipo administrativo .- Donde se involucra el concepto de "Administración de la Configuración", la cual consiste en la realización de un seguimiento íntegro de un sistema para acumular toda la historia del sistema en una base de datos (Bitácora del sistema). Esta base de datos deberá proporcionar los cambios

efectuados en cierto periodo, los recursos de los que hace uso, y una biblioteca de programas.

- **Técnicas de tipo técnico :** Estas técnicas están enfocadas al uso de gente experta para realizar el mantenimiento y el desarrollo de los sistemas considerando hasta donde sea posible el futuro mantenimiento.

Por otra parte podemos definir la existencia de los siguientes tipos de mantenimiento:

- **Mantenimiento Perfectivo.**

Este tipo de mantenimiento es aquel que enriquece el sistema con las necesidades del usuario.

- **Mantenimiento Adaptivo.**

El mantenimiento adaptivo es el que se refiere a cambios en el medio ambiente del sistema como son el Sistema operativo, Hardware, Base de datos, etc.

- **Mantenimiento Correctivo.**

El mantenimiento correctivo consiste simplemente en la corrección de errores que se presenten en el sistema.

En cuanto a los tipos de mantenimiento definidos se tienen los siguientes grados de incidencia durante la vida del sistema:

Mantenimiento Correctivo	20%
Mantenimiento Adaptivo	25%
Mantenimiento Perfectivo	55%

Durante todo el desarrollo del sistema de Contabilidad

PC, siempre se consideraron los posibles problemas que pudieran surgir relacionados con el mantenimiento del sistema, es por esto que se enfatizó en la realización de un sistema modular.

Si nos referimos al mantenimiento perfectivo del sistema en estudio, se puede considerar que entre los cambios a realizar en el sistema cuando se presente un nuevo usuario se encuentra la modificación de los títulos de reportes.

La manera para realizar este cambio en los títulos de los reportes es la siguiente: Entrando a DbaseIII, se deberá editar el archivo titulo.dbf y escribir el título deseado, éste podrá tener hasta 130 caracteres.

Otro tipo de cambio a realizar de tipo perfectivo son los referentes a los números de cuenta que el usuario desee manejar en los Estados financieros y los formatos de estos reportes, para esto se tienen los archivos COTA1.dbf, COTA2.dbf y COTA3.dbf, los cuales se deberán modificar bajo los siguientes criterios:

Estado de Resultados.

Para darle mantenimiento a este reporte existen dos archivos auxiliares: COTA1 y COTA2, el archivo COTA1.dbf contiene las cuentas que deben de aparecer en el reporte. Los cortes a realizar en el reporte para la impresión de un título están representados a través del número de cuenta "0000", cuando llegue a un corte recopilará del archivo Conta2.dbf el letrero correspondiente y lo escribirá. En el caso de los letreros de signo, aparecerá el título "menos" o "mas", si después del número de cuenta en el campo

llamado sig ,se pone el signo (+ ó -)en donde se desee que aparezca siguiendo una relación de la operación. Para finalizar en el campo signo correspondiente al último corte deberá contener el simbolo "=", y el último registro del archivo ,deberá contener la cuenta afectada por el estado de resultados,conteniendo en el campo signo el simbolo "*".

Deberá aparecer información si después del número de cuenta del campo cuenta hoy un cambio en la operación que continúa y se desea que aparezca un letrero de "menos" o "mas".

Para que quede mas claro,se listará el archivo COTA1.dbf del reporte del sistema:

ARCHIVO COTA1.dbf

Reg.	N	CTA	SIG
1	5101		-
2	6001		
3	6002		
4	0000		
5	6003		
6	0000		
7	6004		
8	6005		
9	0000		
10	6006		+
11	5103		
12	0000		
13	6007		
14	6008		
15	6009		
16	6010		
17	6011		
18	6012		
19	6013		
20	5102		
21	0000		=
22	3003		*

El archivo COTA2.dbf contiene los letreros que deberán

aparecer en cada corte ,indicado en COTA1 por el número de cuenta 0000.

El archivo COTA2.dbf del sistema queda definido de la siguiente manera:

ARCHIVO COTA2.dbf

Reg N.	LETRERO
1	UTILIDAD NETA
2	UTILIDAD BRUTA
3	UTILIDAD DE OPERACION
4	UTILIDAD FINANCIERA
5	UTILIDAD O PERDIDA

Balance General

Para el balance general, las cuentas que son consideradas para la elaboración de este reporte se encuentran contenidas en el archivo COTA3.dbf, el cual contiene tres campos: cuenta, tip1 y tip2. El primer campo llamado cuenta es de 4 dígitos y contiene el número de la cuenta, el segundo campo llamado tip1 , deberá contener el tipo 1 de cuenta, el tercer campo llamado tip2 contiene el tipo2 de la cuenta.

Los tipos de cuenta se definen como sigue:

tipo 1

A	ACTIVO
F	PASIVO
*	PATRIMONIO

tipo 2

C	CIRCULANTE
F	FIJO

D DIFERIDO
* Para cuentas de patrimonio.

La combinación de los tipos 1 y 2 darán los diferentes tipos de cuenta que se presentan en el reporte del Balance General:

Activo Circulante
Activo Fijo
Activo Diferido
Pasivo Circulante etc.

El listado del archivo COTA3.dbf del sistema actual es el siguiente:

Reg. N	CTA	TIP1	TIP2
1	1001	A	C
2	1002	A	C
3	1003	A	C
4	1004	A	C
5	1101	A	C
6	1201	A	F
7	1202	A	F
8	1301	A	F
9	1302	A	F
10	1303	A	F
11	1401	A	D
12	2101	F	C
13	2102	F	C
14	2201	F	F
15	3001	*	*
16	3002	*	*
17	3003	*	*

La forma de modificar los archivos de tabla dependiendo de las necesidades de cada usuario se deberá realizar desde DbaseIII, dando la siguiente secuencia de instrucciones:

```
C > DBASEIII  
      .USE COTA1  
      .EDIT
```

Después del comando edit aparecerá el primer registro del archivo donde se podrá modificar el contenido de sus campos, para pasar al siguiente registro bastará con darle "enter", al finalizar la edición se dará "ctrl end" para salir de editor y aparecerá un punto "." .Si se desea modificar otro archivo se realizará la misma secuencia explicada anteriormente. Para abandonar la sesión en DBASEIII se dará QUIT.

El sistema no se habrá modificado y sólo se deberán de actualizar los archivos titulo, COTA1, COTA2 y COTA3, evitando la necesidad de corregir los programas fuentes y de recompilar todo el sistema.

Por lo que se refiere al mantenimiento adaptivo del sistema de Contabilidad PC ,éste fué desarrollado para ser utilizado bajo el sistema operativo MSDOS o PCDOS versión 2.0 o mayor, esto es dado por las características de DbaseIII, por otro lado en el caso de existir cambios en cuanto al número de versión del sistema operativo ,los cambios a realizar en el sistema serían mínimos, en el remoto caso de que existieran, dado que las instrucciones utilizadas en los programas de aplicación son muy generales.

A P E N D I C E A.

CARACTERISTICAS DEL EQUIPO UTILIZADO.

La MPF-C computadora personal, esta basada en su diseño en un microprocesador INTEL de 16 bits 8088, el cual es uno de los microprocesadores de 16 bits más populares actualmente.

En esta PC se puede correr una gran cantidad de software de aplicación y paquetes como son symphony ,lotus123, wordstar, framework, lenguajes como Basic, Fortran, Pascal, C, Cobol etc.

MPF-PC ofrece al usuario varios modelos de acuerdo a las múltiples necesidades de los usuarios ;las principales diferencias entre estos modelos son que unas constan de floppy disk y otras tienen disco duro (llamado "XT").

El disco duro provee de mayor capacidad de almacenamiento, con mayor eficiencia y un más rápido acceso de datos que en un floppy disk.

La capacidad de la memoria RAM depende del modelo de la máquina :

Para Floppydisk se tiene los siguientes modelos:

Modelo	Tamaño de la RAM
502	64 Kbytes
512	128 Kbytes
522	256 Kbytes
532	384 Kbytes

542	512 Kbytes
552	640 Kbytes

Para disco duro se tienen los siguientes modelos:

Modelo	Tamaño de la RAM
50XT	64 Kbytes
51XT	128 Kbytes
52XT	256 Kbytes
53XT	384 Kbytes
54XT	512 Kbytes
55XT	640 Kbytes

Si la capacidad de la memoria RAM esta abajo de los 256KB se puede expandir fácilmente a 640KB instalando chips adicionales de memoria o una de las tarjetas de expansión opcional en el sistema y ajustando algunos switches del tablero del sistema.

HARDWARE

El hardware básico de una PC esta formado por:

- 1.- La unidad del sistema que incluye la memoria.
- 2.- El teclado.
- 3.- La pantalla.

Unidad del Sistema.

La unidad del sistema constituye el núcleo de la PC y contiene todos los circuitos básicos de la PC en la tarjeta del sistema entre estos circuitos se encuentra la memoria.

La memoria es el lugar donde se almacenan datos y programas. Existen dos clases de memoria :ROM y RAM .

La memoria ROM (read only memory) se emplea para almacenar programas y datos que han de quedar en forma permanente en la PC

,el contenido de la memoria ROM no se borra al apagar la máquina.

La memoria RAM (Random Access Memory) se emplea para almacenar la mayor parte de los programas y datos empleados en la ejecución de un programa. El contenido de la memoria RAM se pierde si se apaga la máquina.

La memoria ROM tiene una capacidad de 1024 bytes y la RAM, dependiendo del modelo, puede tener una capacidad desde 64K hasta 640K.

Teclado.

El teclado de la PC es similar al de una máquina de escribir con algunas teclas de funciones especiales.

Monitor.

La PC puede desplegar su salida a través de un monitor de video monocromático, o un monitor de color o una pantalla de tv sería necesario el uso de un modulador.

Dispositivos de almacenamiento masivo.

Los dispositivos para almacenamiento masivo le permiten a la PC almacenar datos y programas en forma permanente. La facilidad para acceder la información almacenada en cada uno de estos dispositivos depende del dispositivo empleado.

-Grabadora de Cassette .

Una grabadora de cassette normal puede usarse con el PC para almacenar datos y programas.

-Unidad de discos flexibles.

Los discos flexibles presentan ciertas ventajas sobre las cintas, ya que pueden almacenar mas cantidad de información y puede accederse a ella de forma mas fácil.

Las unidades de cara sencilla almacenan 160 Kbytes por disco, las unidades de disco de doble cara almacenan 320 Kbytes de información.

-Unidad para discos rígidos.

El medio mas rápido y con mayor capacidad para almacenamiento masivo de datos, que puede asociarse al PC, es la unidad para discos rígidos. Esta unidad puede almacenar millones de bytes.

El equipo que se utilizó para el desarrollo del sistema de contabilidad PC cuenta con una memoria RAM de 512K, una ROM de 8K, seis slots de expansión, disco duro de 10 Megabytes y una unidad de disco flexible para diskettes de 5 1/4".

SOFTWARE DEL PC.

El software del PC se divide en tres partes: Sistemas Operativos, lenguajes de programación y programas de aplicación.

Sistema Operativo.

Un sistema operativo es un programa que controla el hardware de la PC. Una PC sin unidades no necesita un sistema operativo proporcionado desde el exterior, dado que una versión muy sencilla esta contenida en la memoria RAM del PC. Los sistemas que usan disco por el contrario necesitan que la máquina se cargue con un sistema operativo al encenderse.

Los sistemas operativos mas importantes para la PC son el MSDOS y el CP/M86. La mayor parte del software se ejecutará mediante el uso de estos dos programas.

Lenguajes de Programación.

Si intenta escribir sus propios programas para el PC, necesita ejecutar un programa denominado interprete ,compilador o ensamblador dependiendo del lenguaje que intente usar .Los lenguajes de alto nivel mas importantes que se pueden ejecutar en la PC son: BASIC, Cobol, Fortran, y Pascal.

Programas de Aplicación.

Los programas de aplicación son aquellos que se ejecutan para un trabajo determinado. Pueden adquirirse para ejecutarlos en la microcomputadora o pueden escribirse por el usuario empleando un determinado lenguaje como el Basic o el Pascal.

Dentro de este tipo de software se encuentran los procesadores de textos, las hojas electrónicas ,manejadores de bases de datos, etc.

Sistema Operativo MSDOS.

MSDOS es el sistema operativo bajo el cual trabaja la microcomputadora personal .

Al momento de encender la PC se cargará el sistema operativo el cual residirá en disco duro o en un diskette en el drive A al momento en que el sistema carga el archivo COMMAND.COM ,se presentará la siguiente pantalla:

```
MS-DOS Version 2.11
Copyright 1981, 1982 Microsoft Corp.
```

```
Command v. 2.02
Current date is wed 1-01-1981
Enter new date:
```

Se deberá teclar la fecha bajo el formato mm-dd-aa

Si la fecha proporcionada es correcta el sistema desplegará:

```
Current time is 0:00:14.32
Enter new time:
```


Se deberá proporcionar la hora bajo el formato hh:mm

Si la hora es correcta, la máquina responderá con el prompt: "C>" y a partir de entonces podremos comunicarnos con la máquina usando comandos de MSDOS.

Directorios.

El concepto de directorio se utiliza para almacenar y clasificar archivos, sobre todo si la máquina es utilizada por varios usuarios o se cuenta con varias aplicaciones.

Podemos considerar a un directorio como archiveros donde podremos guardar nuestra información para que no se mezcle con la de los demás usuarios, de tal forma que en el directorio de contabilidad se tendrá toda la información referente a esa rama.

La estructura de directorios que se tiene en MSDOS es de tipo jerárquica donde el directorio principal es llamado directorio raíz y es referenciado por "\".

Para el uso del sistema operativo MSDOS cuenta con los siguientes tipos de comandos:

Comandos de tipo externo:

Son aquellos que residen en el disco como archivos ejecutables, por lo que deben ser leídos antes de ser ejecutados. Cualquier archivo que tenga extensión .COM .EXE o .BAT es considerado como un comando externo.

Los comandos externos de MSDOS son los siguientes:

CHDSK Analiza un disco y produce un informe de estado

[d:] CHDSK

DISKCOPY Permite copiar todo un disco hacia otro

DISKCOPY [d:] [d:]

FIND Busca una serie de caracteres en archivos

FIND <string> [[nombre de archivo]]

FORMAT Formatea un disco

FORMAT d:

EXE2BIN

Convierte los archivos .EXE en archivos con formato binario.

EXE2BIN <archivo.exe> [d:]<archivo.com>

MORE

Muestra la pantalla completa de datos.

MORE

PRINT

Manda a impresión un archivo de texto mientras se realiza el proceso de otros comandos de MSDOS.

PRINT [Nombre del archivo a imprimir]

RECOVER

Recupera un archivo o todo un disco que contenga sectores dañados.

RECOVER <nombre de archivo>

ó

RECOVER [d:]

SDRT Ordena los datos de un archivo y así los guarda en un archivo o los muestra por terminal.

SDRT <archivo no ordenado> archivo ordenado

SYS

Transfiere los archivos del sistema operativo MSDOS del disco que se encuentra en el drive de

default hacia el disco del drive especificado.

Comandos Internos.

Los comandos internos son los mas simples y se pueden ejecutar inmediatamente ya que pertenecen a el procesador de comandos.

A continuación se presentan los comandos internos de MSDOS, así como su función:

CHDIR Change directory

Este comando nos permite cambiarnos de directorio.

Si por ejemplo el directorio de trabajo es \Bin\User\Juan y deseamos cambiarnos a otro directorio (\Bin\User\Juan\Formas) se tendría que introducir :

```
CHDir \Bin\User\Juan\Formas
```

Si no se especifica ninguna ruta después del comando CHDIR, ésta instrucción nos mostrará el nombre del directorio de trabajo .

COPY

Este comando permite copiar uno o mas archivos a otro disco o al mismo disco con los mismos nombres o nombres diferentes.

La sintáxis de este comando es la siguiente:

```
COPY <d:\pathname\archivo fuente> <d:\pathname\archivo destino>
```

d: drive A,B ó C

pathname: ruta de directorio

Si el nombre del archivo destino no se especifica, el copy se realizará en el drive de default y el archivo se copiará con el mismo nombre del archivo original.

Si el archivo fuente se encuentra en el drive de default y el nombre del archivo destino no es especificado el copy será abortado.

DIR

El comando dir muestra los archivos existentes bajo el directorio en el cual se esta trabajando.

DIR \dir1 Muestra los archivos bajo el directorio llamado dir1.

CLS

Limpia la pantalla.

DATE

Proporciona la fecha del sistema y da oportunidad de cambiarla.

DEL

Borra los archivos indicados

DEL <lista de archivos separados por blanco>
ejem: DEL *.com
(* = todos)

DISKCOPY

Permite copiar todo un disco hacia otro

DISKCOPY [d:] [d:]
(d = drive A: o B:)

FORMAT

Formatea un disco

FORMAT A:

MKDIR

Crea un directorio.

MKDIR <nombre de directorio>

PATH

Busca archivos por lotes y mandatos en directorios.

PATH [<pathname>[; <pathname>]...]

PROMPT

Establece un nuevo símbolo indicador

PROMPT [texto indicador]

REN

Cambia de nombre a un archivo

REN <nombre actual> <nombre nuevo>

REN PRUEBA.COB NUEVO.COB

RMDIR

Borra todo un directorio que debe estar vacío.

RMDIR \CONTA\DGSCA

TIME

Proporciona la hora del sistema y da la oportunidad de cambiar este.

TYPE

Permite desplegar el contenido de un archivo.

VER

Despliega la versión del sistema operativo en uso.

El sistema operativo también nos da la facilidad de crear archivos de comandos que serán de tipo BAT.

B I B L I O G R A F I A

ADVANCED PROGRAMMER'S GUIDE

FEATURING DBASEIII AND DBASEII.

Luis Castro, Jay Hanson y Tom Retting.

Edit. Ashton Tate.

AN INTRODUCTION TO DATABASE SYSTEMS

C.J. Date.

Edit. Adisson Wensley.

PRINCIPLES OF DATA BASE SYSTEM

Ullman Jeffrey D.

Edit. Computer Science Press, Inc.

APLIQUE DBASEIII.

Edward Jones.

Edit. Mc Graw Hill.

SISTEMA DE ADMINISTRACION DE BANCOS DE DATOS.

Alfonso F. Cárdenas.

Edit. Limusa .

DESARROLLO Y ADMINISTRACION DE PROGRAMAS DE COMPUTADORA.

Victor Gerez , Mauricio Mier, Rolando Nieva y Guillermo Rodriguez.

Edit. Cecsua.

INSIDE THE IBM PC

ACCESS TO ADVANCED FEATURES AND PROGRAMMING.

Peter Norton.

Edit Brady ,

MICROSOFT MS-DOS

OPERATING SYSTEM USER'S GUIDE.

Multitech Industrial Corp.

MPF-PC USER'S GUIDE.

Multitech Industrial Corp.

PRIMER CURSO DE CONTABILIDAD.

Elias Flores.

Edit. Trillas.

THE PRACTICAL GUIDE TO STRUCTURED SYSTEM DESIGN.

Meilir Page Jones.

Edit. Ed Yourdon Press .