

34
2es.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

CAMPUS ACATLAN

"DESARROLLO DE UN SISTEMA CONTABLE PARA EL INSTITUTO DE INGENIERIA"



MEMORIA DE DESEMPEÑO PROFESIONAL

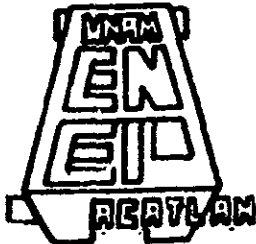
QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN MATEMATICAS APLICADAS Y COMPUTACION

P R E S E N T A :

ERIKA QUIROZ ARREDONDO

ASESOR: ING. MANUEL GUTIERREZ SEDANO.



ACATLAN, EDO. DE MEX.

1998.

TESIS CON FALLA DE ORIGEN

267816



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A ese Ser Supremo

Gracias, por darme la oportunidad de existir y la capacidad para lograr lo que hasta ahora he logrado...

A mi Padre

Rafael Quiroz M.

A quien admiro por su gran espíritu de aventura y libertad, y quien con su cariño me ha enseñado lo linda que es la vida...

A mi Madre

Celia Arredondo A.

A quien admiro por su gran fortaleza y el saber afrontar los problemas sin darles la espalda; y quien con su amor me ha enseñado a ver la vida con gran optimismo....

A mi esposo

Gerardo Medina G.

Quien me ha apoyado incondicionalmente y me ha enseñado que con el don de la paciencia y el Amor es posible lograr eso que parecía no alcanzable.
Gracias, por haber estado conmigo en algunos de los momentos más difíciles de mi vida....

A mi hijo

Luis Gerardo Medina Quiroz

Quien se ha convertido en el centro de mi existir...
Quien ha llenado mi vida de alegría y con quien he aprendido lo hermoso de dar Amor sin límite alguno....

A mis hermanos

Rafael

De quien admiro su enorme corazón y su valentía para enfrentar la vida después de los fracasos....

J. Gabriel

Por ser tan noble y quien a pesar de todo, me ha apoyado con su cariño desinteresado...

Edgar

A quien tanto admiro por ser fuerte y firme en todos los sentidos,
y quien a pesar de su corta edad ha demostrado su gran madurez....

Edna

Quien ha sabido ser para mí una gran amiga y
a quien le agradezco el apoyar mis "locuras"....

A mi tía

Elida Quiroz

Quien siempre estuvo a mi lado y quien me ha apoyado incondicionalmente....

A toda mi familia

Porque con ellos he aprendido la gran fuerza que da la unión....

A mis mejores amigos

Gracias, por ser como son y siempre extenderme
la mano cuando lo he necesitado....

A tí...

Porque a pesar de no poder estar físicamente conmigo,
sé que desde donde te encuentres estás a mi lado y
nunca dejaremos de sentir que estamos juntos;
pues siempre estas en mi Pensamiento y mi Corazón....

A Todos los que hicieron posible la culminación de este trabajo, mil Gracias....

INDICE

AGRADECIMIENTOS

INDICE

RESUMEN	I
INTRODUCCIÓN	II
METODOLOGÍA	VI
CAPÍTULO 1.- BASES DE DATOS	1
1.1 CONCEPTOS Y APLICACIONES	1
1.2 IMPORTANCIA.....	9
CAPÍTULO 2.- PROGRAMA CONTABLE	13
2.1 SISTEMA DESARROLLADO.....	13
2.1.1 <i>Objetivos del sistema y resultados esperados</i>	13
2.1.2 <i>Descripción breve del sistema</i>	24
2.1.3 <i>Explicación de procedimientos y funciones</i>	33
2.1.4 <i>Estructuras de la base de datos</i>	44
RESULTADOS	67
CONCLUSIONES	68
ANEXO 1.- CONCEPTOS BÁSICOS DE CONTABILIDAD	I
ANEXO 2.- PAQUETES CONTABLES	V
ANEXO 3.-PROGRAMACION ORIENTADA A OBJETOS	XI
CONCEPTOS BÁSICOS	XI
ANEXO 4.-BREVE INTRODUCCIÓN A CLIPPER 5.01	XIX
DEFINICIÓN Y CARACTERÍSTICAS.	XIX
ENFOQUE A LA PROGRAMACIÓN ORIENTADA A OBJETOS.....	XXIII
BIBLIOGRAFIA	XXVII

R E S U M E N

El objetivo de la elaboración de este trabajo es demostrar como gracias a la aplicación de nuevos y cada vez más bondadosos lenguajes de programación, combinados con las nuevas técnicas de programación, se logra una optimización de los sistemas y una mayor funcionalidad de los mismos. En este caso se habla del lenguaje de programación Clipper 5.01 y de la técnica de la programación orientada a objetos.

La metodología empleada para el análisis, diseño e implantación del sistema es la conocida como Yourdon Structured Method (YSM).

Los resultados obtenidos al final del trabajo fueron sumamente satisfactorios, pues se logró una funcionalidad buena del sistema y una pronta aceptación de los usuarios al mismo, esto gracias al acoplamiento en lo más posible a los estándares de presentación de pantallas y de captura de datos de otros sistemas ya conocidos por ellos. También se logró una reducción de los archivos físicos para los departamentos involucrados, una entrega de resultados más rápida y una carga de trabajo para los usuarios menor.

Finalmente, se concluyó: el objetivo fue alcanzado pues con la aplicación del lenguaje Clipper 5.01 y la utilización de la técnica de la programación orientada a objetos se tuvo un tiempo de acceso y de respuesta rápido, además, la utilización de código fuente fue menor, logrando tener un sistema de fácil mantenimiento y en un tiempo corto.

INTRODUCCIÓN

A través de los últimos años, instituciones de educación superior, entre ellas, la Universidad Nacional Autónoma de México, han tenido un gran crecimiento y diversificación de actividades, haciendo de su administración una función vital.

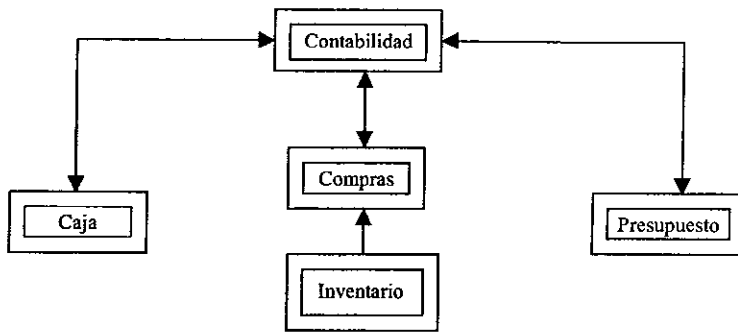
Con esta misma diversificación de tareas resulta importante contar con métodos y sistemas, los cuales permitan el control adecuado de la labor administrativa para obtener una agilización de trámites y procedimientos así como la obtención de información oportuna y veraz.

En el presente caso, nuestra máxima casa de estudios requiere contar con procesos administrativos los cuales debido a su naturaleza, comprenden un volumen elevado de datos; que al ser procesados arrojan información importante, requiriendo la utilización de sistemas automatizados y modernos.

El Instituto de Ingeniería anexo a la Facultad del mismo nombre, originó con el tiempo la *necesidad de contar con un programa contable para agilizar la labor administrativa cada vez más compleja y difícil de realizar de manera manual*. En efecto, en dicho instituto se generaron muchos problemas debidos a la escasa capacitación de algunos empleados en materia contable, originando elevado costo en el manejo de información y la ineficaz veracidad y rapidez de la información debido a su extemporaneidad.

El sistema contable en dicho instituto era meramente manual; manejado por un contador y dos pasantes de contabilidad adscritos al departamento de contabilidad; quienes interactuaban con personal de caja, de compras e inventario y, del presupuesto.

Un diagrama de dicho flujo de información se esquematiza de la siguiente manera:



Con base a este flujo de información la entrega de resultados de manera central era tardío, en ocasiones hubo pérdida de los mismos, originándose evidentemente la reconstrucción del proceso y naturalmente la pérdida de horas de trabajo para todas las personas involucradas. Adicionalmente se dió el problema de almacenar toda la documentación contable, cuyo espacio físico era ya insuficiente.

Por todos estos problemas se planteó al Instituto de Ingeniería una solución consistente en automatizar el sistema contable; con lo cual se lograría la integración del procesamiento de datos, teniendo con esto, una transferencia total de actividades, tales como la creación, la manipulación, el almacenamiento, la recuperación, la distribución y, el control de la información en un dispositivo electrónico, disminuyendo la necesidad del espacio físico para el archivo e incrementando la productividad de los departamentos involucrados.

Primeramente, se analizó la posibilidad de comprar un paquete existente en el mercado, pero en ese momento apenas fueron apareciendo algunos paquetes contables; de los cuales se investigaron precios, características y capacidades para saber si podían adaptarse a las necesidades de los usuarios del instituto y estar seguros de su transportabilidad al tener que interactuar con los sistemas existentes de caja, de presupuesto y de compras e inventarios. Se realizó una lista de los

productos que podían cubrir estas necesidades; estableciendo a la vez costos de adaptación de los mismos, las interfases a desarrollar para la interconexión entre los otros sistemas arriba mencionados, el costo del soporte técnico así como el costo de la capacitación tanto al personal usuario como al personal técnico; así mismo se hizo una estimación de los tiempos para tener listo el paquete.

De la información que se obtuvo se llegó a la conclusión de que la mejor opción sería el desarrollo del sistema directamente por el personal del instituto; debido básicamente al alto costo estimado en la compra de algún paquete, así como el tiempo de entrega.

Al plantear el desarrollo de un programa contable para el Instituto de Ingeniería el primer problema que surgió fue la necesidad de establecer un lenguaje de programación, el cual permitiera tener tiempos de respuesta y de accesos rápidos. Tomando como experiencia el hecho de que los otros sistemas con los cuales iba a interactuar estaban programados en el lenguaje Clipper Summer'87 se investigó la posibilidad de crear el sistema contable con la nueva versión Clipper 5.01, analizando si con ésta podría hacerse más eficiente el sistema comparativamente con los ya existentes.

La investigación arrojó valores positivos para crear el sistema contable, teniendo la característica de usar la técnica de la programación orientada a objetos, razón por la cual se decidió utilizar este lenguaje para el desarrollo del sistema contable.

Como un resumen de este trabajo se pueden señalar dos capítulos: Capítulo uno.- Bases de Datos y, Capítulo dos.- Programa Contable.

En el primero se estudian los conceptos, la aplicación y, la importancia que tienen las bases de datos en el desarrollo de sistemas computacionales en la actualidad.

En el segundo, se habla sobre el sistema desarrollado en el Instituto de Ingeniería. Aquí se contemplan tanto la organización y los objetivos del instituto como los objetivos del sistema mismo y, el problema a resolver.

Así mismo, se analizan las funciones y los procedimientos principales del sistema, mostrando las estructuras de las principales bases de datos y los archivos de ordenamiento básicos de éstas. También se indican los requerimientos para correr el sistema contable, tanto de hardware como de software.

Se cuenta con cuatro anexos importantes:

Anexo Uno: Conceptos Básicos de Contabilidad

En este anexo se habla de los conceptos básicos de la contabilidad.

Anexo Dos: Paquetes Contables

En este anexo se mencionan las características principales de los paquetes contables que fueron analizados durante la fase de definición.

Anexo Tres: Programación Orientada a Objetos

A lo largo de este anexo se abarcan los conceptos importantes de la técnica de la programación orientada a objetos. Se indican algunas de las ventajas de esta técnica y, sus objetivos principales.

Anexo Cuatro: Breve Introducción a Clipper 5.01

En este anexo se da una breve historia del lenguaje de programación Clipper, su definición y sus ventajas.

METODOLOGÍA

La metodología empleada para el desarrollo del sistema contable es conocida como Yourdon Structured Method (YSM).

Esta metodología se eligió después de analizar todos los aspectos que involucraba el sistema a desarrollar, los cuales pueden observarse en el capítulo 2 Programa Contable; pues es una metodología muy bien planteada que logró satisfacer lo necesario.

Los pasos más importantes seguidos fueron los siguientes:

1) Estudio de la situación actual del problema. Aquí se estudiaron:

- a) Los objetivos de los Departamentos involucrados (de contabilidad, de caja, de compras e inventarios y de presupuesto), así como del instituto mismo,
- b) Los equipos de cómputo y el software con los que contaba el Instituto,
- c) Las actividades de los empleados del departamento de contabilidad, de caja, de presupuesto y de compras e inventarios. Quienes iban a interactuar con el sistema.

2) Justificación del costo.- se determinaron las necesidades primordiales, así como los beneficios a obtener:

- a) Identificación de necesidades de los usuarios, a través de entrevistas y cuestionarios informales,
- b) Identificación de tareas básicas de los usuarios.
- c) Beneficios de la automatización del sistema.

3) Investigación del Sistema.

- a) Establecimiento de objetivos del sistema,
- b) Establecimiento de la funcionalidad del sistema,

- c) Análisis de la interfase con el usuario,
- d) Análisis de su mantenimiento y adaptabilidad,
- e) Análisis de la flexibilidad y portabilidad del sistema, así como su reusabilidad,
- f) Observación del ambiente de hardware y del software,
- g) Análisis de su coexistencia con otros sistemas,
- h) Investigación de los estándares de los sistemas con los cuales va a interactuar,
- i) Análisis de los procedimientos generales de los sistemas con los cuales va a interactuar,
- j) Análisis de los documentos de entrada,
- k) Análisis de los documentos y pantallas de salida,
- l) Estudio de la documentación existente,
- m) Estimación del hardware y del software necesarios.

4) Definición del Sistema.- Se conceptualiza un modelo basado en los resultados de la fase uno:

- a) Revisión de los estándares de los sistemas con los cuales va a interactuar; se establecieron los estándares convencionales en base a los primeros, se determinaron los formatos de las pantallas y de los reportes; se estableció la navegación entre pantallas.
- b) Diseño del modelo; se analizaron los módulos y subrutinas a realizar. Se definieron los procedimientos globales. Se consideraron los accesos de seguridad.
- c) Determinación de la interfase gráfica con el usuario. Se determinaron los formatos de las pantallas, los procedimientos para archivar la información, los de respaldo y los de recuperación de la misma.
- d) Realización de un estimado del espacio a ocupar por el sistema.

- e) Definición de los campos de la base de datos. Se definieron las pantallas y los reportes, así como su secuencia.

5) Desarrollo del Sistema.

- a) Realización del diseño de la base de datos, de los procedimientos globales y locales, de los archivos índices, de los reportes y de los estándares. Se analizaron los colores, las formas y los sonidos para el diseño final de las pantallas.
- b) Diseño del acceso, la seguridad y la salida del sistema.
- c) Preparación de la documentación, tanto la del usuario como la técnica.
- d) Capacitación a los usuarios en el uso y mantenimiento del sistema contable
- e) Diseño del plan de implementación del sistema. Eligiéndose la técnica de la implementación directa, en este caso, el sistema nuevo entra en su totalidad a producción, pero llevándose un control manual como se hacía anteriormente, para así comprobar que los datos que se arrojan son los correctos.

6) Implementación del Sistema.- El objetivo de esta fase es dejar el sistema en un ambiente de producción:

- a) Establecimiento de los códigos de usuario y los passwords para los usuarios en producción,
- b) Actualización de los parámetros de especificación del sistema.
- c) Generación del sistema para ser instalado en producción.
- d) Instalación del sistema en producción.
- e) Establecimiento de procedimientos que reporten problemas.

CAPÍTULO 1 .- BASES DE DATOS.

1.1 CONCEPTOS Y APLICACIONES.

Una **base de datos** la podemos entender como una colección de datos similares a la cual más de una persona tiene acceso y que puede emplearse para más de un propósito, para satisfacer las necesidades de uno o más usuarios.

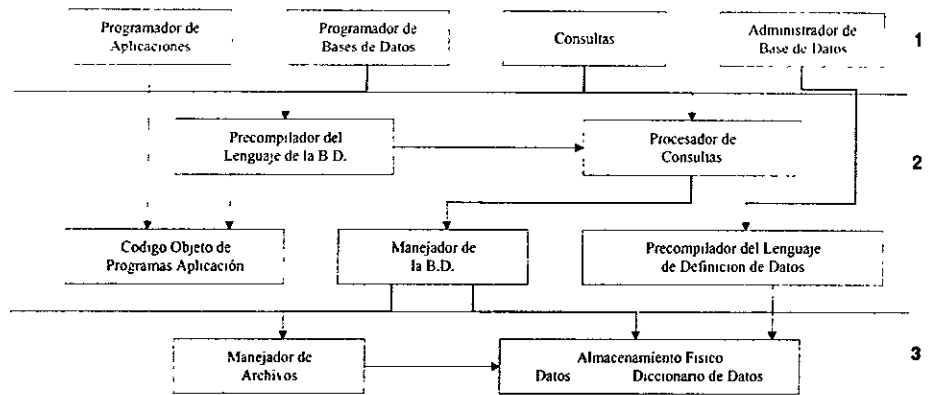
Algunas de las ventajas de utilizar una base de datos son:

- 1) Ahorro de recursos al reunir los datos y almacenarlos sin duplicarlos,
- 2) Acceso a los datos, en caso necesario, es rápido y confiable,
- 3) Inconsistencia en los datos es mucho menor, evitando con ello errores.

Para el manejo de una base de datos es necesario un sistema llamado **sistema manejador de bases de datos (DBMS.- DataBase Management System)**, el cual puede considerarse como un conjunto de recursos, encargado básicamente de las siguientes funciones :

- 1) Obtener y mantener datos para poder operarlos y así tomar una decisión,
- 2) Mantener seguridad de la base de datos al reforzar las medidas de privacidad e integridad, para así prevenir fallas en programas.

La estructura general de un sistema de bases de datos puede observarse en el siguiente esquema:

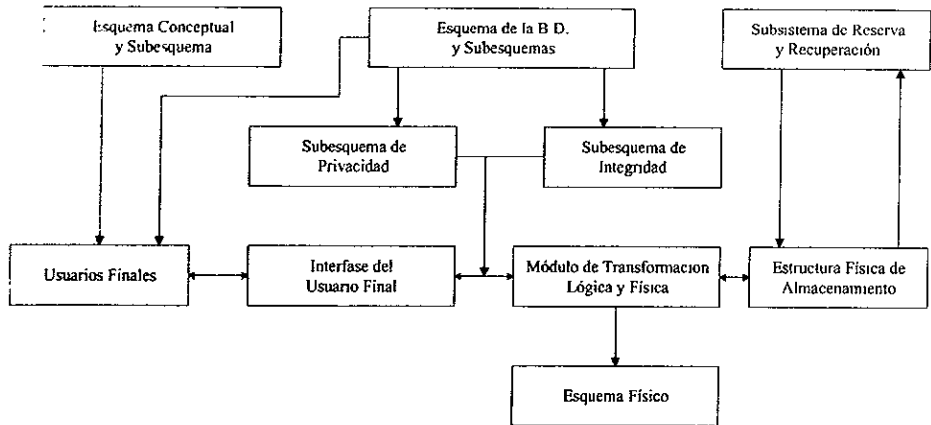


1) Nivel de usuarios

2) Nivel de DBMS (DataBase Management System)

3) Nivel de la Base de Datos

La arquitectura ideal de una base de datos puede considerarse compuesta de nueve módulos que indican las diferentes funciones o medios disponibles, que a continuación se presentan:



Claves:

- Indica el flujo de los datos de la Base de Datos
- Indica el Flujo de los datos de referencia
- Indica una restricción en el flujo de los datos

Para el buen diseño de una base de datos se sugiere formular esquema:

El **esquema** es el conjunto de entidades y atributos que definen a la base de datos.

Los esquemas utilizados para el diseño de una base de datos son básicamente tres :

- 1) **Esquema conceptual**, lo cual se considera como una descripción abstracta y general de una parte del universo que representan los datos de la base de datos, esto es, una descripción de la realidad y no de los datos.

Por ejemplo: El esquema conceptual podría contener:

- a) una lista de los tipos de entidades implicados (empleados, departamentos, máquinas, piezas, etc.),
- b) una lista de las restricciones entre estos tipos de entidades (empleados que trabajan en departamentos , máquinas que se construyen a partir de piezas, etc.) y,

- c) una lista de las restricciones de integridad que se aplican (si el empleado A dirige al empleado B y el empleado A trabaja en el departamento D, entonces el empleado B debe trabajar en el departamento D).
- 2) **Esquema lógico de base de datos.**- puede considerarse como la descripción de los datos que están almacenados en la base de datos y que indica qué elementos de los datos están almacenados y cuáles son los accesos entre estos elementos. Para esto se recomienda al diseñador elaborar un diccionario de datos importantes, para así tener la información más importante como es: nombre, sinónimos, definición, tipo, formato, restricciones de acceso, relaciones con otros elementos.
- 3) **Esquema físico (interno)** se refiere a una descripción de la estructura física de la base de datos, la cual contiene el tipo, el nombre y la longitud de los campos. Se puede ver como un fichero secuencial indexado. De donde podemos identificar algunos conceptos importantes:
- a) registro.- el cual es un conjunto de datos relativos, formado por campos;
 - b) campo.- es un conjunto de datos del mismo tipo;
 - c) campo clave.- sirve de plataforma para determinar el orden de clasificación y/o relación en la base de datos;
 - d) entidad.- es el objeto sobre el cual se almacena información (por ejemplo: alumno, pieza);
 - e) atributo.- es la propiedad o propiedades que posee una entidad (por ejemplo: color, sueldo).

En seguida se da un ejemplo que contempla los tres tipos de esquemas:

- 1) **Esquema conceptual:** el universo consta de 20 compañías, cada una de éstas emplea entre 50 y 200 empleados, los cuales trabajan para una única compañía y tienen una edad única.

- 2) **Esquema lógico:** la estructura lógica de los datos está basada en el modelo de datos relacional y consta de una relación única con los dominios <<nombr es de los empleados>>, <<edad>>, <<compañía>>.
- 3) **Esquema interno:** La relación se cumple como un fichero secuencial indexado con el nombre de los empleados como clave.

Otros conceptos importantes dentro de lo que son las bases de datos son:

La **independencia de datos** indica hasta qué punto la estructura física de almacenamiento de un sistema de base de datos es independiente de los programas de aplicación. Es importante que un sistema de base de datos contenga datos independientes, para así, poder ampliarse o trasladarse a una nueva estructura física de almacenamiento produciendo un efecto mínimo en el resto del sistema.

La **privacidad** de una base de datos está definida como una propiedad de la base de datos que refleja hasta qué punto los datos están protegidos contra el acceso no autorizado.

La privacidad debe de considerar las funciones siguientes:

- a) identificar al usuario,
- b) listar los módulos del esquema de la base de datos a los que el usuario desea acceder y,
- c) consultar las restricciones de acceso para ver si se le da opción a entrar.

La **seguridad** dentro de una base de datos se da mediante mecanismos diseñados para asignar niveles de acceso a usuarios, para evitar que alguien que no posea una clave de acceso pueda llegar a acceder algún dato. Aquí se dan las restricciones de acceso.

La **integridad** de una base de datos la protege contra modificaciones no válidas (a diferencia de las no autorizadas como en el punto anterior).

Las **rutinas de recuperación** detallan la recuperación de la base de datos, después de una falla del sistema. Siempre es recomendable realizar copias de los archivos críticos, para que, en casos extremos puedan recuperarse.

Considerando la importancia de los DBMS, se menciona a los más conocidos a nivel comercial:

IMS (Information Management System); Sistema de gestión de la Información.- Este es un sistema jerárquico creado por IBM. El IMS proporciona un diccionario de datos como herramienta; proporciona reserva y recuperación por medio de programas de utilerías, permite el acceso a una base de datos.

IDMS.- El esquema de la base de datos se define por el uso de un lenguaje de descripción de datos (DDL - Data Description Language-), similar a COBOL. El IDMS proporciona un lenguaje de control de medios de dispositivos (DMCL - Device Media Control Language-), utilizado para especificar detalles de la transformación del modelo lógico en recursos de almacenamiento.

ADABAS.- Es un DBMS orientado a redes locales; a pesar de tener características relacionales, puede interconectarse con COBOL, FORTAN, PL/1 o ENSAMBLADOR; para el procesamiento en línea puede interconectarse con TSO, CICS, INTERCOMM, TASKMASTER y ENVIRON/1.

LINC II.- Es un DBMS orientado a mainframes. Es creado por UNISYS; también puede ser utilizado por equipos UNIX. Es de tipo relacional. Proporciona un diccionario de datos como herramienta, así como una reserva y recuperación por medio de programas de utilerías. Genera códigos fuentes en COBOL.

DBASE.- Es un DBMS relacional escrito en lenguaje C. Puede contener hasta 1000 millones de registros, con hasta 128 campos cada registro y, con un máximo de 400 caracteres por campo.

CLIPPER.- Considerado como un compilador de programas en dBASE, hasta su versión 5.0, pues a partir de ella se observa como un lenguaje de programación, debido a algunas características particulares que presenta como son entre otras, el manejo de la técnica de la programación orientada a objetos teniendo cuatro clases pre-construidas (TBrowse, TBColumn, Get y, Error).

FOXBASE.- Es un manejador de base de datos relacional; se puede utilizar en sistemas como MSDOS, UNIX y VAX; puede emplearse en redes locales. Se tiene una buena organización de la memoria y de los dispositivos de entrada/salida.

PARADOX.- Trabaja con SQL integrado; contiene su propio lenguaje de programación llamado PAL (Paradox Application Language). Puede importar y exportar bases de datos de y a los siguientes formatos : Quattro Pro, dBASE, Reflex, Lotus, Symphony, ASCII.

SQL.- Es considerado como un DML (Data Management Language), esto es, un lenguaje manejador de datos. Es de tipo relacional. Puede unir o relacionar hasta ocho bases de datos.

INGRES.- Es un DBMS relacional. Funciona bajo el sistema operativo UNIX, funciona en ordenadores DEC PDP-11 y VAX. Puede tener interfases con Páscal, Fortran y C.

DB2.- Admite el lenguaje SQL y QUERY-BY-EXAMPLE. Es de IBM.

NDB: (Data Mapping Program).- Es un DBMS de IBM.

Los componentes importantes de un DBMS son:

- 1) Un lenguaje de definición fácil de usar,
- 2) un sistema de diccionario de datos,
- 3) un lenguaje de definición de esquemas de base de datos de propósito general,
- 4) una estructura de almacenamiento de datos,
- 5) un sistema de privacidad,
- 6) un sistema de integridad,
- 7) un sistema de reserva y recuperación,
- 8) un generador de programas de aplicación.

Algunas de las limitantes de los DBMS actuales son:

- a) el usuario del DBMS debe estar envuelto tanto en los detalles de instalación como en la especificación del sistema,

- b) el usuario debe estar familiarizado con la estructura de almacenamiento,
- c) la independencia de los datos no está garantizada,
- d) las interfases del usuario final no son sencillas,
- e) algunas veces hace los problemas más severos,
- f) no identifica los datos redundantes,

1.2 IMPORTANCIA.

Las bases de datos tuvieron su origen en la industria y, poco después fueron disparadas por los avances tecnológicos de almacenamiento físico de los datos.

Primeramente los datos se almacenaban externamente en cartas y cintas de papel; esto se realiza alrededor de 1945.

En 1959 se emplean los llamados archivos de datos, en los lenguajes de programación Fortran, COBOL, Ensamblador y, lenguajes máquina. En relación al hardware, las máquinas son lentas, grandes y, de memoria limitada; se utilizan las tarjetas perforadas.

En 1961, surgen los primeros sistemas manejadores de datos (DBMS), pero no tienen un auge considerable.

En 1964 siguen utilizándose archivos de datos para el almacenamiento físico de los datos, pero con lenguajes más sencillos de usar como son Basic y Algol. En relación al hardware, se emplean los transistores y con ello se logra tener una memoria más aceptable; comienza a desarrollarse la cinta de papel y, con ello los medios de almacenamiento se vuelven más prácticos. El tamaño de las máquinas disminuye considerablemente.

En los años 70's, aparece el concepto de base de datos, empleándose para su manejo lenguajes de programación más estructurados como lo son Pascal, ADA, Módula y C, aparece el DBMS de IBM CODASYL (Comité de lenguajes de sistemas de datos), principalmente. En relación al hardware, se utilizan los circuitos integrados; las cintas magnéticas hacen su aparición; se da una manipulación de grandes almacenamientos de datos, empleándose técnicas de estructuras de almacenamiento.

En los años 80's, aparecen los lenguajes de cuarta generación y, comienza el auge de los manejadores de base de datos (DBMS - DataBase Management System). Surge el SQL, dBASE,

Clipper, FoxBase y, muchos otros. Se da el manejo de las bases de datos en red. En relación al hardware, aparecen los chips (circuitos integrados de alta integración). Surgen las computadoras personales (PC's); los medios de almacenamiento secundario tienen mayor capacidad de almacenamiento (discos duros, discos flexibles). Se da un incremento considerable de la memoria RAM (memoria de acceso aleatorio). En las microcomputadoras va de 128 Kb a 16 Mb, en las minicomputadoras va de 128 Kb a 32 Mb y, en las macrocomputadora va de 2 Mb a 64 Mb. Así mismo, se logra un incremento en la memoria secundaria; en las macrocomputadoras va de 200 Mb a 1.2 Gb, en las minicomputadoras de 20 Mb a 600 Mb y, en las microcomputadoras de 5 Mb a 360 Mb.

En los años 90's, surgen las bases de datos "inteligentes". Se da un desarrollo de DBMS para redes locales; se emplea la técnica de programación orientada a objetos para optimizar el manejo de las bases de datos; se emplea la técnica de multimedia. En relación al hardware, aparecen las LAN'S (redes de área local) y, dentro de los dispositivos de almacenamiento secundarios el CD-ROM.

La importancia de utilizar las bases de datos puede ser vista a través de las siguientes ventajas que se obtienen a partir de su manejo:

- 1) Los datos pueden emplearse de formas múltiples, esto es, el acceso puede darse para varios usuarios a la vez.
- 2) Se protege la inversión intelectual; esto se logra al diseñar en forma simple la base de datos.
- 3) Las bases de datos tienen bajo costo.
- 4) Se minimiza la redundancia de datos.
- 5) El usuario tiene la respuesta en el tiempo requerido.
- 6) Los datos presentados al usuario son entendibles y claros.
- 7) Se logra una facilidad de uso, pues basta con presionar teclas para el acceso a la información.

- 8) Se tiene una rápida atención a consultas.
- 9) Las interfaces con el pasado y/o el futuro son relativamente sencillas.
- 10) La actualización de las bases de datos se da con precisión y coherencia.
- 11) Se puede tener seguridad y privacidad para los datos.
- 12) Se tiene una independencia física de los datos, es decir, los dispositivos de hardware son independientes a la base de datos.
- 13) Se tiene con una independencia lógica de los datos, esto es, se puede agregar un campo a la base de datos sin necesidad de reestructurarla totalmente.
- 14) Las relaciones y ordenamientos de las bases de datos controlan en gran medida la redundancia de los datos.
- 15) Se elige la mejor técnica para lograr la rapidez adecuada de acceso, algunas de éstas pueden ser :
 - a) Procesamiento en lotes, mejor conocido como "proceso batch": la actualización de la base de datos es de forma secundaria; puede ser por el uso de archivos de paso, los cuales después de terminado un proceso se "vacían" en la base de datos correspondiente, para actualizarla;
 - b) procesamiento en línea: la actualización de la base de datos es inmediata.
- 16) Se tiene una búsqueda dinámica, esto se debe a la rapidez de exploración.
- 17) Se puede tener una normalización de los datos, esto es, puede tenerse el manejo del campo dentro de la aplicación, con su mismo nombre.
- 18) Cada campo puede explicarse en un diccionario de datos.
- 19) Se puede recuperar la información de una forma relativamente fácil.

De esto se desprenden las características deseables para un sistema automatizado :

- a) documentación técnica del sistema,

- b) documentación para el usuario,
- c) facilidad de uso: muestra la flexibilidad de las nuevas aplicaciones,
- d) manual de mensajes de errores: el usuario debe saber el mensaje de error que arroja el sistema en caso de que introduzca una respuesta u opción incorrecta,
- e) Sencillez de mantenimiento: los cambios que se requieran hacer al sistema, pues este nunca permanecerá estático, deben ser fáciles y, si algún módulo se vuelve absoluto, poder aislarlo y desecharlo o modificarlo según convenga,
- f) debe tener un buen tiempo de respuesta para el usuario,
- g) el software debe ejecutarse rápidamente y las estructuras de datos empleadas deben minimizar el espacio físico de almacenamiento de los datos.

Actualmente, se está dando una tendencia a lo que se le llama sistemas manejadores de bases de datos de quinta generación, lo que desea de estos DBMS es que tengan una mayor flexibilidad de independencia de datos, y que posean mecanismos de verificación de integridad más poderosos. Y que tengan interfaces de usuario final sencillas para que los mismos usuarios fuesen capaces de aprenderlas y utilizarlas con pocas horas de instrucción.

CAPÍTULO 2 .- PROGRAMA CONTABLE

2.1 SISTEMA DESARROLLADO

2.1.1 Objetivos del sistema y resultados esperados

La administración actual y la dinámica que presenta una institución como la UNAM y, en particular, el Instituto de Ingeniería dependencia de nuestra Máxima Casa de Estudios; lugar en el cual será implantado el sistema a desarrollar para el manejo de su contabilidad; hacen necesaria la elaboración cada vez más continua de sistemas computarizados que optimicen la toma de decisiones.

En la etapa del análisis se establecen los objetivos del Instituto de Ingeniería en donde se va a trabajar.

El Instituto de Ingeniería forma parte del Sistema de Investigación Científica de la Universidad Nacional Autónoma de México, cuyas funciones básicas son las siguientes:

- 1) Realizar investigación fundamental y aplicada.
- 2) Formar investigadores en ingeniería.
- 3) Apoyar a la formación de profesores y las tareas docentes de la Facultad de Ingeniería.
- 4) Estudiar problemas de interés nacional.
- 5) Contratar investigaciones, estudios y asesorías sobre problemas específicos, cuando así se contribuya al cumplimiento de las funciones anteriores.
- 6) Colaborar con otras instituciones de la UNAM.
- 7) Difundir los resultados de sus investigaciones.
- 8) Llevar a cabo las actuaciones necesarias para realizar funciones precedentes.

Las jerarquías existentes dentro del Instituto de Ingeniería pueden observarse con la siguiente descripción:

Primeramente, se encuentra la DIRECCION cuyo objetivo es la de mantener, la de administrar y, la de lograr los objetivos de la Institución.

En segundo plano, de forma directa tiene bajo sus órdenes a tres subdirecciones y tres secretarías:

a) SUBDIRECCION 1. La cual se conforma por cinco coordinaciones.

- Estructuras.- realiza estudios analíticos y experimentales sobre el comportamiento de estructuras y materiales expuestos a diversas acciones que afectan la vida útil de las construcciones.

- Geotecnia.- trata sobre el comportamiento mecánico de suelos y enrocamientos, el comportamiento de estructuras térreas.

- Mecánica aplicada.- efectúa investigaciones teóricas y aplicadas para analizar la respuesta inelástica de edificios ante sollicitaciones sísmicas.

- Sismología.- intenta explicar los fenómenos sísmicos.

- Vías Terrestres.-estudia los sistemas de transporte en aquellos aspectos relacionados con diseño geométrico y estructural, operación y conservación de carreteras y aeropistas.

b) SUBDIRECCION 2.- Se compone de las coordinaciones:

- Automatización.- estudia diferentes aspectos de hardware, programación y algoritmos, para la implantación de sistemas de control de tiempo real de procesos industriales.

- Instrumentación.- realiza estudios relacionados con la electrónica e instrumentación aplicada.

- Mecánica, térmica y fluidos.- se estudian teórica y experimentalmente, y se diseñan mecanismos de tipo industrial para automatización de procesos, dispositivos mecánicos y bombas.

- Sistemas.- se preocupa de la planeación de sistemas.

c) SUBDIRECCION 3. Compuesta por las siguientes coordinaciones:

- **Hidráulica.**- estudia la forma de aprovechar la geohidrología, los fenómenos transitorios, la geotermia, la hidráulica fundamental, la hidráulica de la superficie, la fluvial y la marítima.

- **Ambiental.**- realiza investigaciones sobre el agua potable, el control de la contaminación del agua subterránea, el tratamiento de aguas residuales y, la contaminación de aire.

Las secretarías son las siguientes:

a) **ACADEMICA.**- Administra la situación de los becarios y del personal académico que labora dentro del Instituto.

b) **TECNICA.**- Consta de tres departamentos:

- **Coordinación editorial.**- la cual sirve de enlace entre el personal académico y la información documental; tiene a su cargo la publicación de las Series del Instituto de Ingeniería; proporciona material fotográfico para registrar y comunicar resultados obtenidos en diferentes investigaciones; y da servicios de venta de publicaciones.

- **Coordinación de carpintería y mecánica.**- donde se elaboran dispositivos, máquinas y sus elementos diseñados por los investigadores.

- **Mantenimiento de edificios.**- conserva las instalaciones.

c) **ADMINISTRATIVA.**- Formada por cuatro departamentos:

- **Contabilidad.**- se lleva el control contable del Instituto.

- **Personal.**- Se maneja toda la administración que involucra al personal del Instituto.

- **Informática.**- Encargada de la realización y mantenimiento del software requerido para la administración del Instituto

- **Servicios Administrativos.**- Organización y planeación de la administración del Instituto.

En esta fase de análisis se llevó a cabo un estudio detallado de la información recopilada, después de tener varias entrevistas con el contador, encargado de llevar el control manual de la contabilidad del instituto, con la persona encargada de la caja, con la persona de presupuesto y con la persona encargada del almacén para solicitarles su ayuda sobre los requerimientos que debería de cubrir el sistema. Cabe señalar que las entrevistas aquí realizadas fueron de forma espontánea, es decir, no se formuló un cuestionario formal para ellas; además de que fueron muchas y muy variadas.

Para conocer a fondo el problema, además de las entrevistas realizadas, se tuvo que familiarizar con el control manual que, hasta ese momento se llevaba a cabo, para lo cual también participaron las personas inmersas de una u otra manera en la contabilidad central, de ellas se obtuvo en gran medida las bases para entender las terminologías y claves usadas, así como los procedimientos de los movimientos que realizará el sistema computarizado, con base al papel que realiza cada usuario. Con todo esto se lograron identificar las necesidades de los mismos.

Con esta información se estimó el alcance y la duración del proyecto; así como la asignación de recursos al mismo. El alcance es el desarrollo del Sistema de Contabilidad que tenga inmerso el actual Sistema de Caja y pueda tener interfases con el Sistema de Presupuesto y, el Sistema de Compras e Inventario. Por otro lado, se estimó un tiempo de análisis, de diseño, de desarrollo y de implementación de cinco meses, teniendo en cuenta que van a participar tres personas en este proyecto.

Continuando, con la investigación del sistema se establecen los objetivos de éste.

Objetivos generales del sistema contable :

- 1) Computarizar el sistema manual de contabilidad del Instituto de Ingeniería para optimizar el control y manejo de la información , así como optimizar los tiempos de obtención de resultados.
- 2) Obtener la información necesaria en forma oportuna y veraz.
- 3) Cubrir los diversos niveles de requerimientos en la administración de la contabilidad, con la información necesaria que permita apoyar veraz y oportunamente los trámites cotidianos, el seguimiento de la información, la planeación y la toma de decisiones.
- 4) Mantener en una base de datos la información requerida necesaria, reduciendo los volúmenes de los archivos físicos tradicionales.

Objetivos particulares del sistema :

- 1) Crear la base de datos necesaria para disponer de los mecanismos de almacenamiento, actualización y recuperación de la información; así como la expedición de reportes necesarios y, las consultas pertinentes.
- 2) Estandarizar el flujo de información e integrar los módulos necesarios.
- 3) Contar con una estructura de datos y procesos para soportar los cambios dinámicos en el manejo de los trámites.
- 4) Disponer de los mecanismos para la actualización de los procedimientos y la desconcentración de trámites.
- 5) Preparar la base de datos que permita a través del tiempo darle una mayor utilidad a nuestro sistema.
- 6) Almacenar solamente la información necesaria para evitar los archivos de volumen tradicionalmente enormes.

Las características que debe tener el sistema son las siguientes:

- 1) La base de datos estará constituida por el conjunto de archivos relacionados que almacenarán información que permita:
 - a) dar el soporte a los procedimientos de los diversos trámites que atienda la dependencia,
 - b) controlar el acceso y actualización de la información (lo cual se realiza a través de verificar claves y passwords, tanto para el acceso a la red como para el acceso al sistema en particular),
 - c) proporcionar los elementos necesarios para la planeación y control de la información,
 - d) facilitar la toma de decisiones.

- 2) El sistema propuesto será generado bajo el lenguaje de Clipper 5.01, después de analizar los puntos que a continuación se presentan:
 - a) magnitud del problema a resolver. Con el análisis se observó que el problema a resolver requería de un gran número de estructuras de la base de datos, abiertas. de archivos índices de ordenamiento, de relaciones entre dichas estructuras y con otras manejadas por otros sistemas, así como un gran volumen de información a consultar y a dar de alta. Todo esto lo puede manejar Clipper 5.01,
 - b) recursos humanos para el desarrollo del sistema. En este punto se observó con cuántas personas se contaba para el manejo del sistema. En este caso, son tres,
 - c) recursos de cómputo. Dentro de este punto se consideró tanto el hardware como el software con el que cuenta el Instituto para evitar la compra de recursos no necesarios,

- d) tiempo para su realización. Se hizo un estimado sobre el tiempo necesario que se requería para desarrollar el sistema.
- 3) Entonces el sistema deberá estar complementado por el conjunto de programas de aplicación utilizando el lenguaje Clipper 5.01, apoyándose en la técnica de la programación orientada a objetos.
- 4) Los programas de aplicación deberán permitir:
- a) una o más funciones para el control de acceso a usuarios de la red,
 - b) uno o más módulos para la actualización de la información,
 - c) otro u otros módulos para la recuperación y modificación de los datos,
 - d) uno o varios procedimientos para la eliminación y depuración de información no deseada (esta se realiza primeramente en forma lógica solamente, para que, una vez terminada la sesión se realice en forma física); esto implica, una eliminación en lotes y no en línea, sin embargo, en algunos casos la eliminación se podrá realizar en línea,
 - e) uno o más módulos encargados para la consulta de la información que se desee,
 - f) uno o varios procedimientos para la expedición de reportes,
 - g) uno o más módulos para los cortes requeridos (mensuales y/o anuales). Esto significa colocar en estructuras mensuales y/o anuales las información para poder tener acceso después a ellas, evitando con esta separación el crecimiento excesivo de una sola estructura, corriendo el riesgo de perder información o tener un proceso de datos muy lento,
 - h) otro u otros módulos encargados del acceso a consultas a catálogos ya elaborados.

- 5) Las estructuras de la base de datos guardarán la información asociada a las pólizas de diario, pólizas de ingreso, pólizas cheque, etc.; para el control de la contabilidad del Instituto de Ingeniería.
- 6) El sistema debe manejar las estructuras de la base de datos de tal forma que permita su mantenimiento y acceso a la información de una manera ágil y sencilla, en línea o batch de acuerdo a las modalidades de actualización y recuperación de información requerida.
- 7) El sistema es capaz de atender a la vez a varios usuarios de la red local del Instituto de Ingeniería. Esto es, deberá ser multiusuario.

Dadas las necesidades y los objetivos perseguidos por esta dependencia, se considera la creación de estructuras primarias o básicas, que se encarguen de almacenar la información y los datos necesarios, en un segundo término, las estructuras que fungirán como catálogos, los archivos que almacenarán los requerimientos esenciales para el arranque del sistema, así como los parámetros del mismo.

Con respecto a la definición del sistema y comenzando con la fase de **diseño**, teniendo como base los resultados arrojados del análisis, se establecen los siguientes componentes para el mismo:

Una estructura primaria se encargará de guardar los datos generales e importantes de las pólizas, la cual será de carácter anual, se ha de llamar "POLIZA+año de ejercicio.DBF" (Archivo General de pólizas), el cual deberá contener básicamente: la descripción de la póliza, la fecha en que se comprobó la póliza (si se ha comprobado), la fecha de la póliza, el importe de la póliza, el nombre a quien se expidió la póliza cheque o de cualquier otro tipo, el número de memorandum, el número del cheque expedido, la situación de la póliza y, el tipo de la póliza.

Una segunda estructura de la base de datos será el acumulado mensual de las pólizas, su nombre está formado "POD_+año de ejercicio+mes de ejercicio.DBF" y, en su estructura podrá verse: la cuenta contable, la subcuenta de la cuenta, la subsubcuenta de la cuenta contable, el número de la

póliza que afecta a la cuenta contable, el importe de la cuenta, el tipo de la póliza que afecta a la cuenta, el tipo de movimiento.

Otra estructura primaria de la base de datos necesaria es la de acumulados mensuales de las cuentas contables, considerando los cargos y abonos a ellas, el nombre de este archivo será "ACU+mes de ejercicio+año de ejercicio.DBF"; la información que se contemplará es: la cuenta contable, la subcuenta de la cuenta contable, la subsubcuenta de la cuenta contable, el abono actual de dichas cuentas, el cargo actual de las cuentas contables, el abono anterior de las cuentas, el cargo anterior de estas mismas cuentas.

Otra estructura de igual importancia es la de los acumulados anuales de las cuentas contables; cuyo nombre será "ACUPOL+año de ejercicio.DBF" y la información contenida podrá ser semejante a la de los acumulados mensuales, se considera: la cuenta contable, la subcuenta de la cuenta, la subsubcuenta de la cuenta, los cargos iniciales de las cuentas, los abonos iniciales de las mismas cuentas, el tipo de la cuenta.

Una quinta estructura será la encargada de contener la información de los documentos comprometidos de pago; su nombre se formará como "COMPRO+año de ejercicio.DBF"; la información básica contenida en ésta será: la fecha de comprobación de la póliza, la fecha de la póliza, el número de la póliza, el tipo de la póliza, el importe original de la póliza, el número de memorandum, el número de comprobación de la póliza, el tipo de comprobación de la póliza, el saldo de la póliza.

La estructura de la base de datos que contendrá la información de los gastos de las pólizas, será de carácter mensual, su nombre se formará de la siguiente manera "GAST+año de ejercicio+mes de ejercicio"; la información almacenada: año de ejercicio (año en el cual se está trabajando), año de la póliza, el cargo de la póliza, el número de la póliza, el tipo de la póliza, el número de memorandum, el costo de la póliza, la descripción de la póliza, el número de proyecto al cual afecta (en caso de que

afecte directamente a algún proyecto), la coordinación al que pertenece la persona responsable del proyecto, el grupo de gasto que afecta dicho proyecto, la forma de afectación de la póliza, la partida que afecta el proyecto, la referencia de la póliza, la situación de la póliza, si la póliza ha sido reembolsada o no.

La estructura que almacenará todos los movimientos de los bancos, contendrá la siguiente información, su nombre será ser "MOVIBANCO.DBF" : el número de la cuenta, la fecha del movimiento, la fecha del cheque, la ficha de depósito, el importe del cheque, el tipo de ajuste.

Una de las estructuras de la base de datos que se ha contemplado fungirá como catálogo de las cuentas contables; llevará por nombre "CUENCONT.DBF" y contendrá la información: la cuenta contable, la subcuenta de la cuenta, la subsubcuenta de la cuenta contable, la descripción de la cuenta contable, el tipo de la cuenta, si la cuenta afecta a los proyectos, si la cuenta afecta a comprometidos de gastos.

Un archivo encargado de almacenar los parámetros del sistema llevará por nombre "MEMOCONT.MEM", cuya información contenida será: números de las últimas pólizas, año de ejercicio, fecha de corte.

El archivo esencial para ejecutar el sistema será el "ENLACE.MEM", pues contendrá los valores de las rutas de acceso de algunas variables que se manejarán en el sistema.

Concluyendo, el sistema será versátil y flexible, siendo compatible con versiones anteriores de Clipper como lo es la versión del verano del 87; pudiendo ser actualizado en un futuro en una forma fácil y práctica bajo las nuevas versiones de este lenguaje, pues se realizará con la visión de utilización de la técnica de la programación orientada a objetos que se maneja en Clipper 5.01

Los requerimientos mínimos para instalar el sistema contable son los siguientes:

1) Para la implantación en una red local como lo es el caso del Instituto de Ingeniería el cual cuenta con un red Novell IPX/SPX versión 4.0 LAN: con tarjeta Netware Ethernet NE1000 versión 4.0Ec). Formada por:

- un servidor (en este caso con dos discos duros de capacidad de 120Mb y 80Mb respectivamente), con un procesador 486 y una velocidad de 66 Mhz.

- Una terminal de trabajo (no necesariamente con disco duro), pero con un mínimo de 640K de memoria RAM.

- Una impresora (IBM graphics, Epson o Lasser).

- Teniendo particiones en los discos duros en un drive "H:\\" para el uso y generación de la base de datos.

2) Para la implantación en una PC, se requiere de disco duro, con una capacidad mínima de 50Mb y con un mínimo de 640K de memoria RAM.

Con respecto al software en red es requerido es el siguiente:

- Sistema operativo de la red Novell y sistema operativo (MS-DOS V 3.1 en adelante).

- Ambiente de :

PATH=Z:.;Y:.;X:.;W:.;T:.;S:.;P:.;N:.;H:

CLIPPER=F44

_MONITOR=B (monocromático) o,

_MONITOR=C (color)

2.1.2 Descripción breve del sistema.

El diseño del sistema en forma técnica fue concebido así:

1) **CATALOGOS** : este módulo se encargará del mantenimiento y las consultas a los catálogos existentes como lo son, el de cuentas contables y el de las cuentas bancarias.

1.1) **Cuentas contables** : esta opción contempla las siguientes subopciones:

- **Alta** : sirve para adicionar nuevas cuentas contables al catálogo.

- **Modificaciones** : se realizarán las modificaciones requeridas en el catálogo de cuentas contables.

- **Eliminación** : se podrá eliminar la información que ya no resulte necesaria en el catálogo de cuentas contables.

- **Consulta** : obtención de la información completa contenida en el catálogo de cuentas contables.

- **Reportes** : se ofrecerán dos tipos de reportes:

a) **cuenta completa** : en este caso, el reporte expedido será de todas las cuentas contables existentes en el catálogo sin considerar el orden alfabético interno de las mismas,

b) **cuenta alfabética** : para esta opción, el reporte expedido será de todas las cuentas contables existentes en el catálogo considerando el orden alfabético interno de las mismas

1.2) **Cuentas bancarias** : esta opción contempla las siguientes subopciones:

- **Alta** : es para adicionar nuevas cuentas bancarias al catálogo.

- **Modificaciones** : en ésta se realizarán las modificaciones requeridas en el catálogo de cuentas bancarias.

- **Eliminación** : se podrá eliminar la información que ya no resulte necesaria en el catálogo de cuentas bancarias.

- **Consulta** : obtención de la información completa contenida en el catálogo de cuentas bancarias.

- **Reportes** : en este caso, se expide el reporte del catálogo de cuentas bancarias del Instituto de Ingeniería.

2) **INICIO EJERCICIO** : Con esta opción se dará el saldo de arranque de las cuentas para iniciar el sistema otro año, esto se logrará con el corte contable anual. Contará con las siguientes subopciones:

2.1) **Alta** : En esta opción se adicionarán nuevos saldos a las cuentas.

2.2) **Modificaciones** : realizará los cambios pertinentes a los saldos de arranque de las cuentas.

2.3) **Eliminación** : se eliminarán los saldos sobrantes u obsoletos de las cuentas.

2.4) **Consulta** : se visualizará la información de los saldos de arranque de las cuentas.

2.5) **Reporte** : reporte de las cuentas con su correspondiente saldo de arranque

3) **CAPTURA POLIZAS** : En esta opción se darán de alta las pólizas requeridas a las estructuras correspondientes; constará de las siguientes subopciones:

3.1) **Alta** : es la opción para la captura de los diferentes tipos de pólizas (pólizas cheque, pólizas de diario, pólizas de ingreso, etc.).

3.2) **Modificaciones** : se realizarán los cambios pertinentes o necesarios a las diferentes pólizas que hayan sido ingresadas previamente.

3.3) **Eliminación** : se borrarán o darán de baja las pólizas no requeridas por el usuario.

3.4) **Consulta** : se visualizará en pantalla la o las póliza(s) que el usuario solicite al sistema.

- 4) **CONCILIACION** : Esta opción se refiere a la conciliación bancaria, la cual es un reporte expedido por el banco con el importe y la fecha en la cual se pagó el cheque correspondiente a la póliza de cheque; esta será una estructura con carácter anual, esto es, solamente se generará cada año o corte anual. Constará de las siguientes subopciones:
- 4.1) **Captura cheques** : se adicionarán las pólizas cheques que se necesiten.
 - 4.2) **Captura otra póliza** : se darán de alta las pólizas requeridas que no sean tipo cheque y que se utilicen para la conciliación bancaria.
 - 4.3) **Modificación cheques** : se realizarán los cambios pertinentes sobre las pólizas cheque ya capturadas en la conciliación bancaria.
 - 4.4) **Modificación otra póliza** : se salvarán los cambios realizados a pólizas de otro tipo diferente al de cheque.
 - 4.5) **Cancela cheque/póliza** : en esta opciones podrán realizarse las cancelaciones de cheques o pólizas necesarias.
 - 4.6) **Comisiones** : son cantidades extras que cobra el banco por concepto de comisión.
 - 4.7) **Intereses** : son los que carga el banco a la cuenta por algún movimiento realizado.
 - 4.8) **Consulta** : se podrá visualizar en pantalla cualquier póliza capturada durante la conciliación bancaria.
 - 4.9) **Reporte** : este reporte contendrá todas las pólizas de la conciliación bancaria y, sus saldos correspondientes.
- 5) **REPORTES** : en este caso, se expedirán todos los reportes de tipo contable, como son:
- 5.1) **Libro de diario** : se podrá explicar el reporte de todas las pólizas de cheque de todas las cuentas bancarias, considerando detalles como póliza, cheque, la descripción de la póliza, la cuenta de esta, su cargo y su abono.
- **Generar impresión** : se dará opción a generar el reporte correspondiente.

- **Imprimir** : podrá mandarse a impresora el reporte generado.

5.2) **Auxiliares Bancos** : se obtendrá el reporte de las pólizas cheque de alguna cuenta bancaria; se considerará información como lo es el número de cheque, la póliza, su descripción, el cargo y abono de la misma y su saldo.

- **Generar impresión** : se generará el reporte correspondiente a los auxiliares.

- **Imprimir** : mandará a impresora el reporte generado.

5.3) **Auxiliar mensual** : a partir de este reporte puede obtenerse la información de la afectación de alguna póliza a subcuentas contables

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

5.4) **Auxiliar anual** :

- **Generar impresión** : se podrá generar el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

5.5) **Resumen pólizas** : se desglosarán todos los movimientos por fecha y cuenta que correspondan a un mes; considerándose la descripción de la póliza, la fecha y número de la misma, la cuenta de bancos con su cargo y abono correspondientes, la cuenta de acreedores con su cargo y abono, la cuenta de resultados de ejercicio con su cargo y abono, las cuentas de costo de proyectos patrocinados y proyectos no patrocinados con su cargo y abono independientes y, la cuenta de gastos indirectos con su cargo o abono respectivo y los saldos acumulados de dichas cuentas.

- **Generar impresión** : se generarán el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

5.6) **Clientes** : se obtendrán los avances de los contratos de los patrocinadores a una determinada fecha y una determinada vigencia; contendrá el contrato, el patrocinador, el porcentaje

de avance tanto del día primero del mes solicitado como del último día, así como también el total de avances por todos los patrocinadores.

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : se podrá mandar a impresora el reporte generado.

5.7) **Mano de Obra** : contendrá los montos pagados al personal por cada quincena tanto del monto dado por la UNAM, como las vacaciones y el aguinaldo pagados; así como el monto por la nómina interna, las vacaciones y el aguinaldo correspondientes y, las becas.

- **Generar impresión** : se expedirá el reporte correspondiente.

- **Imprimir** : imprimirá el reporte generado.

5.8) **Gastos Presupuesto** : contendrá la descripción de todos los cargos de las No pólizas correspondientes a algún mes. Indicará los gastos indirectos y los fines específicos; contemplará el proyecto, el concepto, la referencia, el importe y la partida.

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

5.9) **Saldos** : se arrojarán los saldos de las cuentas con sus subcuentas y sus subsubcuentas, el concepto, el cargo mensual, el abono mensual, el saldo anterior y, el saldo acumulado.

- **Generar impresión** : se podrá generar el reporte correspondiente.

- **Imprimir** : se mandará a impresora el reporte generado.

- **Consulta** : solamente presentará en pantalla la información solicitada de los saldos.

5.10) **Balance comparativo** : presentará la información de las cuentas, su saldo al mes solicitado como inicial, los movimientos existentes al segundo mes solicitado, el aumento o disminución en los saldos y, finalmente los saldos finales.

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

5.11) **Balanza comprobación** : será un reporte de carácter mensual donde se presentarán los saldos de las cuentas del mes anterior al solicitado, los movimientos al mes requerido, contendrá los cargos y abonos de la cuenta, los saldos de las cuentas al mes solicitado con sus columnas de debe y haber y, los saldos generales.

- **Generar impresión** : se expedirá el reporte correspondiente.

- **Imprimir** : imprimirá el reporte generado.

5.12) **Balance general** : contendrá las cuentas y subcuentas con sus saldos correspondientes y todos los saldos generales, así como algunas relaciones necesarias para poder observar de donde se ha obtenido dicha información; será de carácter mensual.

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado

6) **UTILERIAS** : Esta opción contendrá algunas de las funciones específicas del sistema, como lo son:

6.1) **Corte contable** : es el corte anual que se realizará a nivel contabilidad, con el fin de generar nuevas estructuras, así como también, inicializar los saldos de arranque para el siguiente año contable.

6.2) **Parámetros del sistema** : se contemplan algunas claves con las cuales trabaja el sistema, como son el año de ejercicio, la fecha de corte del mes, el porcentaje del IVA (impuesto), la cuenta y subcuenta de los bancos, la cuenta de resultados, la cuentas de orden, la cuenta, subcuenta y subsubcuenta de impuestos, la cuenta y subcuenta de comprobaciones y, los últimos números de pólizas.

Las opciones accedidas por la caja chica, por medio de los parámetros I, J :

1) **CATALOGOS** : esta opción aplica igual que en el Sistema de Contabilidad antes descrito.

2) **CAPTURA POLIZAS** : Por medio de esta opción se actualizará la información de las pólizas, tanto cheque como de ingreso y todos sus diferentes tipos. Contendrá las siguientes subopciones:

2.1) **Alta** : para la captura de los diferentes tipos de pólizas (pólizas cheque, pólizas de diario, pólizas de ingreso, etc.).

2.2) **Modificaciones** : los cambios a las diferentes pólizas ya ingresadas previamente.

2.3) **Eliminación** : se borrarán o darán de baja las pólizas no requeridas por el usuario.

2.4) **Consulta** : se visualizará en pantalla la o las póliza(s) que el usuario solicite al sistema.

2.5) **Comprobación gastos** : por medio de esta subopción se podrá realizar la comprobación de los gastos introducidos a través de pólizas cheque, para ello se contará con otro submenú;

- **Alta** : se darán de alta pólizas de comprobación.

- **Modificación** : cambios a las pólizas de comprobación ya ingresadas.

2.6) **Reembolso de pólizas** : se realizarán los reembolsos pertinentes a las pólizas que lo requieran; no contará con subopciones.

3) **CONCILIACION** : Esta opción se referirá a la conciliación bancaria, la cual es un reporte expedido por el banco con el importe y la fecha en la cual se pagó el cheque correspondiente a la póliza de cheque; esta será una estructura con carácter anual, esto es, solamente se generará cada año o corte anual. Consta de las siguientes subopciones:

3.1) **Captura cheques** : se adicionarán las pólizas cheques que se necesiten.

3.2) **Captura otra póliza** : se darán de alta las pólizas requeridas que no sean tipo cheque y que se utilicen para la conciliación bancaria.

3.3) **Modificación cheques** : cambios a las pólizas en la conciliación bancaria.

3.4) **Modificación otra póliza** : se salvarán los cambios realizados a pólizas de otro tipo diferente al de cheque.

3.5) **Cancela cheque/póliza** : en esta opciones podrán realizarse las cancelaciones necesarias.

3.6) **Comisiones** : son cantidades extras que cobra el banco por concepto de comisión.

3.7) **Intereses** : los intereses que carga el banco a la cuenta por algún movimiento realizado.

3.8) **Consulta** : se podrá visualizar en pantalla cualquier póliza capturada durante la conciliación bancaria.

3.9) **Reporte** : este reporte contendrá todas las pólizas de la conciliación bancaria y, sus saldos correspondientes.

4) **REPORTES** : se obtendrán los reportes necesarios para el usuario:

4.1) **Auxiliares Bancos** : se obtendrá el reporte de las pólizas cheque de alguna cuenta bancaria; se considerará información como lo es el número de cheque, la póliza, su descripción, el cargo y abono de la misma y su saldo.

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

4.2) **Auxiliar mensual** : a partir de este reporte podrá obtenerse la información de la afectación de alguna póliza a subcuentas contables.

- **Generar impresión** : se expedirá el reporte correspondiente.

- **Imprimir** : imprimirá el reporte generado.

4.3) **Auxiliar anual** :

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

4.4) **Saldos** : arrojará los saldos de las cuentas con sus subcuentas y sus subsubcuentas, el concepto, el cargo mensual, el abono mensual, el saldo anterior y, el saldo acumulado.

- **Generar impresión** : se generará el reporte correspondiente.

- **Imprimir** : mandará a impresora el reporte generado.

- **Consulta** : solamente presentará en pantalla la información solicitada de los saldos.

4.6) **Honorarios** : arrojará las relaciones por pago de honorarios del Instituto de Ingeniería; contendrá el registro federal de causantes de la persona que percibe el monto, el nombre del beneficiario, el número de póliza que le corresponde, la percepción, el IVA (impuesto), la suma, el impuesto y, el importe de la póliza cheque expedida.

- **Generar impresión** : se expedirá el reporte correspondiente.

- **Imprimir** : podrá imprimirse el reporte generado.

5) **UTILERIAS** :Esta opción contendrá algunas de las funciones específicas del sistema:

5.1) **Corte de Caja** : es el corte mensual que se realizará a nivel caja interna del Instituto de Ingeniería, con el fin de generar nuevas estructuras, así como también, inicializar los parámetros de arranque para el siguiente mes de captura.

Parámetros del sistema : se contemplan algunas claves con las cuales trabajará el sistema, como son el año de ejercicio, la fecha de corte del mes, el porcentaje del IVA (impuesto), la cuenta y subcuenta de los bancos, la cuenta de resultados, la cuentas de orden, la cuenta, subcuenta y subsubcuenta de impuestos, la cuenta y subcuenta de comprobaciones y, los últimos números de pólizas.

2.1.3 Explicación de procedimientos y funciones

El procedimiento principal del cual se desprende la llamada a todos los demás que componen al sistema lleva como nombre "MENUCONT.PRG". Este se encarga de verificar que existan los archivos necesarios para arrancar el sistema; tanto el de nombre "ENLACE.MEM", el cual contiene los valores de algunas variables utilizadas para indicar la ruta de las estructuras de la base de datos a acceder; como del de nombre "MEMOCONT.MEM", el cual contiene los parámetros básicos del sistema, como los son la fecha de corte, el año de ejercicio, los números de las últimas pólizas capturas dependiendo de su tipo, entre otra información importante.

MENUCONT() llama a las siguientes funciones:

- 1) **Cierre()**.- Esta función se encarga de cerrar todas las estructuras de la base de datos que se hayan abierto durante la ejecución del sistema, así como de limpiar todos los campos.
- 2) **Dfarcoc()**.- Esta rutina es la encargada de llenar el arreglo a presentar con el catálogo de las coordinaciones; el cual se empleará durante la ejecución del sistema.
- 3) **Borrabasu()**.- Función encargada de borrar las estructura de paso que genere el sistema, así como los índices de éstas.
- 4) **Dfarre()**.- Esta función llena los arreglos de los menús del sistema, realiza la llamada a las funciones o procedimientos restantes.
- 5) **Ldiario()**.- Esta rutina se encarga de la generación del reporte del libro diario utilizado por contabilidad; lo que es lo mismo, es el vaciado de todas las pólizas tipo cheque de las cuentas contables, considerando el número de la póliza, de cheque, descripción de la póliza, su cuenta, el cargo y abono a las cuentas requeridas, etc.. Los pasos que sigue a grandes rasgos son:

- a) genera una estructura de paso de nombre nompaso¹+0,
 - b) la relaciona con la estructura de pólizas generales,
 - c) relaciona la estructura de pólizas generales con la de cuentas contables.
- 6) Imprepo().- Esta función se encarga de mandar a impresora un reporte previamente generado. Es llamado en todos los módulos para mandar a impresión un documento generado.
- 7) Auxmayor().- Genera el reporte para el auxiliar de bancos enviando todos los cargos y descargos de las cuentas contables manejadas por el Instituto de Ingeniería; la información básica que presenta es el número de cheque, el de la póliza, el cargo y abono de la cuenta contable y, el saldo de ésta. Abre las estructuras de mensual de pólizas, la de pólizas anual y, la acumulada anual.
- 8) Auxmayoc().- Este procedimiento es el encargado de generar el reporte del resumen de movimientos del banco por fecha de pólizas por cuentas contables, esto es, se puede obtener la información de la afectación de alguna póliza a subcuentas contables. Los pasos que sigue son los siguientes:
- a) genera la estructura nompaso+2 en base a la de pólizas generales anuales,
 - b) genera la estructura nompaso+h tomando como base la estructura del acumulado mensual de pólizas,
 - c) se genera la estructura nompaso , cuya estructura es la misma que la de pólizas mensual,
 - d) por último, se genera la estructura que es copia del acumulado anual de pólizas,

¹ Cada vez que hagamos referencia a **nompaso** diremos que es el nombre del usuario que en ese momento accesa al sistema

9) Rsaldos().- Con esta rutina se generan los reportes de los saldos, esto es, la balanza de comprobación del principio del año (enero) al mes solicitado. La secuencia de pasos básica es la siguiente:

- a) relaciona la estructura anual de acumulado de pólizas con el acumulado de pólizas mensual,
- b) copia a una estructura de paso (nompaso) las cuentas, subcuentas y subsubcuentas,
- c) copia en otra estructura de paso (nompaso+1) la información de nompaso,
- d) relaciona el la estructura nompaso+1 con la de cuentas contables.

10) Balcomp().- Esta función es la encargada generar el reporte del balance general comparativo. Presenta la información de las cuentas, su saldo al mes solicitado como inicial, los movimientos existentes al segundo mes solicitado, el aumento o disminución en los saldos y, los saldos finales. Los pasos relacionados con éste son a los que a continuación se hace referencia:

- a) relaciona la estructura acumulada anual de pólizas con la de acumulado mensual de pólizas,
- b) copia a una estructura de paso (nompaso) las cuentas, subcuentas y subsubcuentas,
- c) copia en otra estructura de paso (nompaso+1) la información de nompaso,

11) Balzcomp().- Este procedimiento es el encargado de generar la balanza de comprobación.

Es un reporte de carácter mensual donde se presentan los saldos de las cuentas del mes anterior al solicitado, los movimientos al mes requerido, conteniendo los cargos y abonos de la cuenta, los saldos de las cuentas al mes solicitado con sus columnas de debe y haber y, los saldos generales. Los pasos principales que sigue son los siguientes:

- a) relaciona la estructura acumulada anual de pólizas con la de acumulado mensual de pólizas,

- b) copia a una estructura de paso (nompaso) las cuentas, subcuentas y subsubcuentas,
- c) copia en otra estructura de paso (nompaso+1) la información de nompaso.

12) Balgral().- Dentro de esta rutina se obtiene el balance general de contabilidad. En su contenido se tienen las cuentas y subcuentas con sus saldos correspondientes y todos los saldos generales, así como algunas relaciones necesarias para poder observar de donde se ha obtenido dicha información; es de carácter mensual, aunque cabe, señalar que la impresión puede obtenerse en el momento que se desee.

13) Avan_cli().- Esta función es utilizada para la generación del reporte de clientes que tiene el Instituto de Ingeniería en Cartera; por medio de éste puede observarse el avance de los contratos de los mismos a un mes dado. En este reporte se realiza un filtrado de los contratos con fecha de vigencia del año solicitado por el usuario.

14) Totmdo().- A través de la llamada a este procedimiento se obtiene el reporte de mano de obra del personal que labora en el Instituto de Ingeniería. Los pasos principales son los siguientes:

- a) Se abre la estructura de generales de todo el personal que labora en dicho Instituto,
- b) se calculan las percepciones e impuestos retenidos del personal con plaza, así como las becas y los honorarios pagados por el Instituto de Ingeniería y, por la UNAM.

15) Repaviso().- En esta rutina se genera el reporte de conciliación de gastos mensuales en presupuesto para contabilidad; este reporte puede obtenerse tanto por número de póliza como por tipo de gasto. Los pasos básicos para su generación son:

- a) abre la estructura general de los proyectos,
- b) separa los gastos indirectos de los gastos directos,

- c) llama a la rutina "Impnopol": Esta rutina imprime la descripción de los cargos correspondientes a un mes; clasifica la información en: ingresos DGAPA, ingresos UNAM, fines específicos, apoyos
- 16) Honorari().- Por medio de este procedimiento se logra la obtención de la impresión de la relación de honorarios pagados por el Instituto de Ingeniería. La secuencia de pasos seguidos es:
- a) copia a una estructura nompaso+0 la información filtrada de la estructura anual de pólizas, con respecto al año y mes solicitados por el usuario,
 - b) relaciona la estructura de pólizas mensual con la de pólizas anual,
 - c) filtra el contenido de la estructura nompaso+0 tomando como base la información contenida en los parámetros del sistema (para mayor información al respecto puede referirse posteriormente a los mismos).
- 17) Fcatcucb().- Esta función es la encargada de dar mantenimiento al catálogo de cuentas contables; para lo cual se auxilia de otras rutinas:
- a) alta_ccb().- con esta rutina se da de alta en la estructura de cuentas contables alguna cuenta que se requiera, esto es, valida el rango del campo (tanto el inferior como el superior), valida que exista el nivel superior en el caso de dar de alta una subcuenta, valida para evitar la redundancia de los datos.
 - b) modi_ccb().- modifica una cuenta dentro de la estructura de cuentas contables, para ello sigue las validaciones anteriores.
 - c) baja_ccb().- elimina una cuenta de la estructura de cuentas contables. Solamente verifica que exista la cuentas que se desea dar de baja.
 - d) cons_ccb().- con esta rutina puede desplegarse el catálogo de cuentas contables del Instituto de Ingeniería.

e) `repo_ccb()`.- se obtiene el reporte general del catálogo de cuentas contables.

18) `Fcatbn()`.- Con este procedimiento se logra dar mantenimiento a la estructura que contiene las cuentas bancarias que maneja el Instituto de Ingeniería. Para ello, al igual que la función descrita anteriormente, se auxilia de otras rutinas:

a) `alta_cbb()`.- con esta rutina se da de alta en la estructura de cuentas bancarias alguna cuenta que se requiera, hace lo siguiente: valida el rango del campo (tanto el inferior como el superior), valida que exista el nivel superior en el caso de dar de alta una subcuenta, valida para evitar la redundancia de los datos.

b) `modi_cbb()`.- modifica una cuenta dentro de la estructura de cuentas bancarias, para ello sigue las validaciones anteriores.

c) `baja_cbb()`.- elimina una cuenta de la estructura de cuentas bancarias. Solamente verifica que exista la cuentas que se desea dar de baja.

d) `cons_cbb()`.- con esta rutina puede desplegarse el catálogo de cuentas bancarias del Instituto de Ingeniería.

e) `repo_cbb()`.- se genera el reporte general de catálogo de cuentas bancarias.

19) `Fncappol()`.- Esta función es la encargada de dar mantenimiento a las estructuras relacionadas con el control de las pólizas diferentes que maneja el Instituto de Ingeniería. Para esto se auxilia en las siguientes rutinas:

a) `alta_poli()`.- esta rutina es la encargada de adicionar una póliza en las estructuras siguientes: en la acumulado mensual de pólizas (`Acupol+mes`), en la de pólizas mensual (`polizas+año+mes`), en la de gastos mensual (`gast+año+mes`), en caso de ser alguna comprobación en la de (`compro+año`).

El reporte de la misma se auxilia de otras subrutinas: `accproye()`.- es la subrutina que lleva los parámetros requeridos para saber el tipo de operación que se realizará, ya sea afectación a proyectos

de investigación, afectación a proyectos de investigación y a anticipos a reserva de comprobar, o solamente afectación a anticipos a reserva de comprobar. En el primero de los casos se llama a la función `revicpro()` la cual es la encargada de realizar la afectación a proyectos o a comprometidos. Si afecta a proyectos, pide los datos de la póliza y valida que la suma de los gastos sea igual al importe que se introdujo al principio de la captura, llama a la rutina `al_proy()` que observa si es un cargo o descargo al proyecto; realiza ya sea uno u otro, tanto a cuentas contables, como a proyectos de investigación afectados por la operación. Para ello, abre las estructuras de cuentas contables y la de pólizas mensual.

Así mismo, verifica si afecta a comprometidos, a través de la función `fproy()` que valida que el tipo de póliza sea "CH" (cheque) o "GB" (giro bancario), pues en este caso sí afectaría a comprometidos y, en su caso, abre la estructura `compro+año` y, adiciona la póliza con su correspondiente movimiento, aunque también pueden ser devoluciones, esto implica, tipo de póliza "IN" (de ingreso) o "PD" (póliza de diario), para lo cual llama a la rutina `fnpoling()` por medio de la cual se capturan las fichas de depósito relacionadas con la comprobación de comprometidos; esta rutina llama a la vez a otra de nombre `altcuenta()` la cual se encarga de validar que la cantidad a comprobar sea menor o igual al saldo de la cuenta; realizando también los cálculos necesarios para la afectación del saldo resultante. Para este tipo de conciliaciones se crea una estructura auxiliar de nombre "CH000+mes.año", para control de las subcuentas contables que se vean afectadas por comprobaciones o devoluciones.

Para el caso en que se afecte tanto a proyectos de investigación como a anticipos a reserva de comprobar se llama a la función `Fmemo()`, la cual se encarga de efectuar la comprobación de gastos; para la captura de los memorandum afectados llama a la rutina `fnpolcomp()`, dentro de la cual se valida que el monto a comprobar sea igual al saldo inicial (el calculado en la estructura de `compro+año`), después se prosigue con la captura de notas de gastos para lo cual se recurre a la

función de nombre actlnotas(), dentro de esta se realiza la tanto la captura de los proyectos de investigación afectados como el cargo o descargo (según se requiera) a los mismos. Finalmente, se hace un llamado a la rutina conocida como alt_pol() para realizar la adición de la póliza en la estructura de compro+año, con su afectación al saldo.

En el último de los tres casos, es decir, cuando solamente se afecte a anticipos a reserva de comprobar se llama a la función Fnreem(), la cual se encarga de realizar el reembolso de gastos de la póliza requerida. Para la captura de pólizas reembolsadas se hace un llamada al procedimiento fnpolreem(), el cual verifica que la cantidad a reembolsar sea igual al monto de la póliza de ingreso; así como también al del monto en notas reembolsadas; crea una estructura de paso con el nombre nompaso a partir de la de gastos mensual (gast+año+mes) y, en ella se descargan los gastos cargados originalmente, para finalmente, sumar las dos estructuras originales de gastos y así obtener la actualización de éstos.

b) Modi_poli().- En este procedimiento se modifican las pólizas que se requieran

siguiendo los pasos de (a). También puede obtenerse el reporte de la póliza si se requiere.

c) Baja_poli().- Esta función se encarga de eliminar una póliza dada; solamente verifica su existencia, en la estructura de gastos.

d) Cons_poli().- Esta rutina presenta en pantalla la pólizas solicitadas por el usuario.

e) Conspond() .- Por medio de esta rutina obtenemos las consultas a anticipos a reserva de comprobar, así como de préstamos. Dentro de esta opción se puede mandar a impresora la consulta.

20) Cardes() .- Con esta función se logra el acceso a cargos y descargos, esto es, realiza la transferencia de gastos; tanto a cuentas contables y/o proyectos. Sus actividades primarias son:

- a) abre la estructura de gastos,
 - b) abre la estructura con la información general de proyectos,
 - c) abre la estructura de ejercicios de mensuales, con respecto a las afectaciones a proyectos,
 - d) crea la estructura de nombre "CARGO.DBF" para realizar cargos y descargos,
 - e) adiciona a la estructura de gastos los registros filtrados de la estructura CARGO,
 - f) no permite transferencias en préstamos o en anticipos a reserva de comprobar.
- 21) Concich() .- Por medio de este módulo se realiza la conciliación bancaria de cheques de todas las cuentas bancarias que maneja el Instituto de Ingeniería. Abre el archivo de conciliaciones, se realiza la afectación a las pólizas cheques correspondientes.
- 22) Fcomint() .- Por medio de este procedimiento se anexan los movimientos extras del banco, como lo son, las comisiones y, los intereses. Abre la estructura del acumulado mensual de las cuentas contables, llama a la rutina fnfichas(), por medio de la cual captura fichas de depósito, abre la estructura de cuentas bancarias y, en ella, reemplaza el saldo inicial y final de la cuenta correspondiente, borra el registro afectado en la estructura.
- 23) Cons_cns() .- Por medio de esta rutina accedemos la consulta de la conciliación bancaria de pólizas de tipo cheque. Existen tres tipos de consultas. Abre la estructura de conciliaciones, presenta la opción por número de cheque donde indexa la estructura por número de cheque, otra opción es por fecha de pago, donde indexa la estructura por fecha de pago expedida por el banco, en un reporte dado, la tercera opción de consultas es por fecha de conciliación, en este caso, ordena la estructura por fecha en que se ha realizado la conciliación bancaria.
- 24) Repconci() .- A través de esta función se logra la obtención del reporte de la conciliación bancaria. Abre la estructura de conciliaciones, abre la estructura de cuentas bancarias

manejadas por el Instituto, copia en un archivo de paso nompaso, la estructura del acumulado mensual de cuentas contables y, la información contenida en el rango solicitado por el usuario.

25) Finiejer() .- Con este módulo mostramos el inicio de ejercicio de las cuentas contables del Instituto de Ingeniería. Para su ayuda llama a las rutinas:

- a) alta_ini().- es la función encargada de adicionar a las cuentas contables su correspondiente monto para el inicio de ejercicio y, en su caso, lo calcula automáticamente. Para esto, abre la estructura de acumulado anual y, sobre ella, realiza las altas.
- b) modi_ini().- Modifica el monto de las cuentas contables correspondiente a su inicio de ejercicio,
- c) baja_ini().- elimina el inicio de ejercicio de alguna cuenta contable que ya no se requiera,
- d) cons_ini().- esta rutina sirve para mostrar en pantalla el inicio de ejercicio de las cuentas contables.
- e) repo_ini().- se puede generar el reporte del inicio de ejercicio de las cuentas contables del Instituto de Ingeniería.

26) Corta() .- Por medio de este procedimiento podemos realizar en forma automática el corte mensual manejado por la caja chica del Instituto de Ingeniería. Abre en la estructura del acumulado mensual, abre la estructura general de pólizas, abre la estructura del inicio de ejercicio de las cuentas contables, abre la estructura que contiene los cargos y abonos de las pólizas, se encarga de abrir el catálogo de cuentas contables para validaciones, abre la estructura de conciliaciones bancarias.

Para continuar, se auxilia de la rutina actlcheq(), la cual valida las pólizas que correspondan al periodo solicitado y, las adiciona a la estructura de conciliaciones indicado, por otro lado, reinicializa

los cargos y los abonos de las cuentas contables en el acumulado mensual; generando en el archivo de comprobaciones anual todas las pólizas que estén pendientes de comprobación. Esta misma función se encarga de actualizar los saldos iniciales para el siguiente año.

27) Cortacont() .- Esta función de lo que se encarga es de, realizar automáticamente el corte anual de contabilidad. Actualiza los nuevos saldos mensuales anteriores del acumulado de pólizas anual, actualiza los saldos iniciales para siguiente año de trabajo, como en el caso del corte de caja mensual.

28) Cgastpry() .- A través de esta rutina obtenemos la consulta de los gastos de los proyectos de investigación del Instituto de Ingeniería. Abre la estructura de descripciones de gastos anual, se abre la estructura de ejercicio de los proyectos de investigación, de carácter anual, se abre la estructura de presupuesto de proyectos anual.

29) Cpry_fin() .- Con este procedimiento puede accederse la consulta a los saldos de los proyectos de investigación, como en el caso anterior, puede imprimirse esta consulta.

30) Parm sist() .- Con esta función puede tenerse un control de los parámetros primordiales manejados por el sistema contable. Se accesa al archivo "MEMOCONT.MEM", el cual contiene la fecha de corte, el año de trabajo actual, el número de los tipos de pólizas correspondientes, entre otra información importante; realiza un acceso al archivo de nombre "CCORTE.MEM", el cual contiene entre otras cosas, la fecha de corte del sistema tanto del corte mensual de la caja, como del corte anual de contabilidad.

2.1.4 Estructuras de la base de datos

La base de datos creada para el sistema de Contabilidad contiene las siguientes estructuras:

f) ACU+año+mes.DBF. La cual contiene los acumulados mensuales de las cuentas contables, considerando los cargos y abonos a ellas; su esquema es el siguiente:

NOMBRE DEL CAMPO	TIPO	LONGITUD
ABON_ACT	Numeric	14, 2
ABON_ANT	Numeric	14, 2
CARG_ACT	Numeric	14,2
CARG_ANT	Numeric	14,2
CCUENTA	Character	4
SCCUENTA	Character	4
SSCCUENTA	Character	4

1) ABON_ACT .- almacena el monto del abono actual de la cuenta contable, es de tipo numérico y su rango va de 0000000000.00 a 9999999999.99.

2) ABON_ANT .- almacena el monto del abono anterior al movimiento de la cuenta contable, es de tipo numérico y su rango va de 0000000000.00 a 9999999999.99.

3) CARG_ACT .- almacena el monto del cargo actual de la cuenta contable, es de tipo numérico y su rango va de 0000000000.00 a 9999999999.99.

4) CARG_ANT .- almacena el monto del cargo anterior al movimiento de la cuenta contable, es de tipo numérico y su rango va de 0000000000.00 a 9999999999.99.

5) CCUENTA .- guarda la clave de la cuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

6) SCCUENTA .- guarda la clave de la subcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

7) SSCCUENTA .- guarda la clave de la subsubcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

El archivo índice del cual se sirve para su ordenamiento lleva por nombre acu+año+mes.NTX, cuya clave está formada por: nume_cta+dtos(fecha_cche)+ficha_dep.

II) ACUPOL+año.DBF. Es el archivo de los acumulados anuales de las cuentas contables;

cuyo esquema es el que a continuación se muestra:

NOMBRE DEL CAMPO	TIPO	LONGITUD
ABON_INI	Numeric	14,2
CARG_INI	Numeric	14,2
CCUENTA	Character	4
SCCUENTA	Character	4
SSCCUENTA	Character	4
TIPO_CUEN	Numeric	1

1) ABON_INI .- almacena el monto del abono inicial de la cuenta contable, es de tipo numérico y su rango va de 0000000000.00 a 9999999999.99.

2) CARG_INI .- almacena el monto del cargo inicial de la cuenta contable, es de tipo numérico y su rango va de 0000000000.00 a 9999999999.99.

3) CCUENTA .- guarda la clave de la cuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

4) SCCUENTA .- guarda la clave de la subcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

5) SSCCUENTA .- guarda la clave de la subsubcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

6) TIPO_CUEN .- es de tipo numérico de longitud de dos, guarda el tipo de la cuenta; valida que su valor sea 1 en el caso de cuenta de cargo y, 2 en el caso de cuenta de abono.

El archivo índice del cual se auxilia para su ordenamiento tiene el nombre de acupol+año.NTX, cuya llave es: ccuenta+scuenta+ssccuenta.

III) COMPRO+año.DBF. Es el archivo que contiene los gastos comprometidos del año de ejercicio. Su esquema es el siguiente:

NOMBRE DEL CAMPO	TIPO	LONGITUD
FECHA_COMP	Date	8
FECHA_POL	Date	8
IMPORT_ORI	Numeric	14,2
NO_MEMO	Numeric	5
NUME_POL	Numeric	4
NUME_POLC	Numeric	4
SALDO	Numeric	14,2
TIPO_POL	Character	2
TIPO_POLC	Character	2

1) FECHA_COMP .- este campo es de tipo fecha y, se utiliza para almacenar la fecha en que se realiza la comprobación de la póliza.

2) FECHA_POL .- este campo es de tipo fecha y, se utiliza para almacenar la fecha en es capturada la póliza.

3) IMPORT_ORI .- en este campo de tipo numérico se almacena el importe original de la póliza al ser ingresada. Su rango es de 00000000000.00 a 99999999999.99.

4) NO_MEMO .- es de tipo numérico y de longitud de 5, por lo cual, su rango va de 00000 a 99999, sirve para almacenar el número de memorandum.

5) NUME_POL .- en este campo se almacena el número de la póliza. Es de tipo numérico de longitud 4, por lo tanto su longitud va de 0000 a 9999.

6) NUME_POLC .- en este campo se almacena el número de la póliza de comprobación. Es de tipo numérico de longitud 4, por lo tanto su longitud va de 0000 a 9999.

7) SALDO .- este campo es tipo numérico, sirve para almacenar el saldo de la póliza. Su rango es de 00000000000.00 a 99999999999.99.

8) TIPO_POL .- en este campo se almacena el tipo de la póliza. Es de tipo carácter de longitud 2. Su rango es ("CH" en caso de ser de tipo cheque, "IN" en caso de ser póliza de ingreso, "PD" si es póliza de diario, "GB" si es giro bancario).

9) TIPO_POLC .- en este campo se almacena el tipo de la póliza de comprobación. Es de tipo carácter de longitud 2. Su rango es ("GS" en caso de ser póliza de anticipo a reserva de comprobar, "PR" si es póliza de préstamo).

Los archivos índices en los cuales se basa para sus ordenamientos son los siguientes:

a) copol+año.NTX cuya clave se forma por tipo_pol + str(ume_pol,4) + dtos(fecha_pol) + dtos(fecha_comp) y,

b) comem+año.NTX cuya llave se encuentra estructurada de la forma tipo_polc + str(ume_polc,4) + dtos(fecha_comp).

IV) GAST+año+mes.DBF. En esta estructura se guardan todos los gastos que se tengan durante un mes, por el Instituto de Ingeniería. El esquema que presenta es el siguiente:

NOMBRE DEL CAMPO	TIPO	LONGITUD
TIPO_POL	Character	2

TPO_POLC	Character	2
ANO_EJERC	Character	2
ANO_POL	Character	2
CARGO	Character	1
COO	Character	3
COSTO	Numeric	14,2
DESCRIP	Character	40
GRUPO_IJ	Character	3
NOPROY	Character	4
NUME_POL	Numeric	4
PARTIDA	Character	3
PART_PROY	Character	3
REFERENCIA	Character	10
SEREEMBOL	Logical	1
SIT_DOCU	Character	1

1) TIPO_POLC .- en este campo se almacena el tipo de la póliza. Es de tipo carácter de longitud 2. Su rango es ("CH" en caso de ser de tipo cheque, "IN" en caso de ser póliza de ingreso, "PD" si es póliza de diario, "GB" si es giro bancario).

2) TIPO_POLC .- en este campo se almacena el tipo de la póliza de comprobación. Es de tipo carácter de longitud 2. Su rango es ("GS" en caso de ser póliza de anticipo a reserva de comprobar, "PR" si es póliza de préstamo).

3) ANO_EJERC .- es de tipo carácter y longitud de 2. En él se almacena el año de ejercicio de la póliza original. Su rango se valida de tipo numérico, por lo cual va de 00 a 99.

4) ANO_POL .- es de tipo carácter y longitud de 2. En él se almacena el año de la póliza. Su rango se valida de tipo numérico, por lo cual va de 00 a 99.

5) CARGO .- este campo contiene el tipo de cargo que se aplica a la póliza. Es de tipo carácter y de longitud de 1. Su rango es ("I" si el cargo es por parte del Instituto de Ingeniería o "U" si es por parte de la UNAM).

6) COO .- este campo de tipo carácter y longitud igual a 3. Contiene la clave de la coordinación a la que pertenece el proyecto de investigación que se ve afectado. Su rango va de 000 a 999, debido a que se valida que sea numérico.

7) COSTO .- en este campo se almacena el costo o importe total de la póliza. Es de tipo numérico y de longitud 14.2, por lo cual su rango va de 00000000000.00 a 9999999999.99.

8) DESCRIP .- como su nombre lo indica, en este campo se almacena la descripción de la póliza ingresada. Su tipo es carácter y tiene una longitud de 40.

9) GRUPO_II .- este campo contiene la clave del grupo de gasto que se afecte con la póliza. Es de tipo carácter de longitud 3, pero su validación se realiza como si fuese numérico, por lo que su rango va de 0.0 a 9.9.

10) NOPROY .- en este campo se almacena el número de proyecto que se ve afectado por el movimiento de la póliza. Es de tipo carácter y de longitud igual a 4, sin embargo, su validación se hace como si fuera numérico. Su rango va de 0000 a 9999.

11) NUME_POL .- este campo almacena el número de la póliza, es de tipo numérico y longitud 4, por ello, su rango va desde 0000 hasta 9999.

12) PARTIDA .- es de tipo carácter y longitud 3. Se encarga de almacenar la clave de la partida del grupo de gasto afectado. Se valida de tipo numérico por lo que su rango va de 000 a 999.

13) PART_PROY .- es de tipo carácter y longitud 3. Se encarga de almacenar la clave de la partida del grupo de gasto del proyecto afectado. Se valida de tipo numérico por lo que su rango va de 000 a 999.

14) REFERENCIA .- en este campo se almacena la referencia de la póliza. Es de tipo carácter y de longitud 10.

15) SEREEMBOL .- es de tipo lógico. Se encarga de almacenar la información de si la póliza fue reembolsada o no. Su rango es (.T. si se reembolsó la póliza y .F. en caso contrario).

16) SIT_DOCU .- este campo contiene la situación del documento o de la póliza. Es de tipo carácter y longitud 1. Su rango es de ("C" en caso de estar comprometida y, "E" en caso de haber sido ejercida).

El archivo índice utilizado para su ordenamiento lleva por nombre gast+año+mes.NTX y se forma con la llave ano_poliza + tipo_pol + str(ume_pol,4).

V) MOVBANCO.DBF. Se encarga de guardar la información relacionada a los movimientos de los bancarios; su esquema se muestra a continuación:

NOMBRE DEL CAMPO	TIPO	LONGITUD
COSTO	Numeric	14,2
FECHAQ	Date	8
FECHA_CCHE	Date	8
FICHA_DEP	Character	10
NUME_CTA	Character	4
TIPO_AJUS	Character	1

1) COSTO .- en este campo se almacena el costo o importe total del movimiento de la cuenta bancaria. Es de tipo numérico y de longitud 14.2, por lo cual su rango va de 0000000000.00 a 9999999999.99.

2) FECHAQ .- este campo es de tipo fecha y almacena la fecha en que se realiza el movimiento.

3) FECHA_CCHE .- en este campo se almacena la fecha de la póliza cheque. Es de tipo fecha.

4) FICHA_DEP .- es de tipo carácter y de longitud de 10. Contiene el número de la ficha de depósito. Se valida como numérico por lo cual se rango va de 0000000000 a 9999999999.

5) NUME_CTA .- contiene el número de la cuenta bancaria, es de tipo carácter y de longitud igual a 4; se valida como si fuese de tipo numérico por ello, su rango va de 0000 a 9999.

6) TIPO_AJUS .- este campo es de tipo carácter y longitud igual a 1. Contiene el tipo de movimiento que es; su rango va de ("I" en caso de ser interés o, "C" en caso de ser una comisión bancaria).

El archivo de ordenamiento lleva por nombre movbanco.NTX y su llave se conforma por nume_cta.

VI) POD_año+mes.DBF. Este archivo contiene la información referente a las pólizas, es de carácter mensual y, su esquema es el siguiente:

NOMBRE DEL CAMPO	TIPO	LONGITUD
CUENTA	Character	4
IMPORTE	Numeric	14,2
NUME_POL	Numeric	4
SCCUENTA	Character	4
SSCCUENTA	Character	4
TIPO_MOV	Character	1
TIPO_POL	Character	2

1) CUENTA .- guarda la clave de la cuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

2) IMPORTE .- en este campo de tipo numérico se almacena el importe original de la póliza al ser ingresada. Su rango es de 0000000000.00 a 9999999999.99.

3) NUME_POL .- este campo almacena el número de la póliza, es de tipo numérico y longitud 4, por ello, su rango va desde 0000 hasta 9999.

4) SCUENTA .- guarda la clave de la subcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

5) SSCUENTA .- guarda la clave de la subsubcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

6) TIPO_MOV .- este campo es de tipo carácter y longitud igual a 1. Contiene el tipo de movimiento a la cuenta; su rango es ("A" en caso de que se abone o "C" en caso de que se realice un cargo).

7) TIPO_POL .- en este campo se almacena el tipo de la póliza. Es de tipo carácter de longitud 2. Su rango es ("CH" en caso de ser de tipo cheque, "IN" en caso de ser póliza de ingreso, "PD" si es póliza de diario, "GB" si es giro bancario).

Para esta estructura se cuenta con dos archivos índices de ordenamiento:

a) pod_+año+mes.NTX cuya llave está formada por los siguientes campos tipo_pol + str(ume_pol,4) + tipo_mov + ccuenta + sccuenta + sscuenta

b) pod2+año+mes.NTX en este caso la clave de ordenamiento es ccuenta + sccuenta + sscuenta + tipo_pol + str(ume_pol,4) + tipo_mov.

VII) POLIZA+año.DBF. Contiene la información anual sobre las pólizas de las cuentas del Instituto de Ingeniería; el esquema de esta estructura es:

NOMBRE DEL CAMPO	TIPO	LONGITUD
DESC_POL	Memo	10
FECHA_COMP	Date	8
FECHA_POL	Date	8
IMP_CHEQUE	Numeric	14,2
NOMCHEQUE	Character	50
NO_MEMO	Numeric	5
NUMCHEQUE	Character	10
NUME_POL	Numeric	1
SIT_POL	Numeric	1
TIPOL_POL	Character	2

1) DESC_POL .- en este campo se almacena de forma breve la descripción de la póliza. Es de tipo memo con una longitud de 10.

2) FECHA_COMP .- este campo es de tipo fecha. Contiene la fecha de comprobación de la póliza.

3) FECHA_POL .- este campo es de tipo fecha. Contiene la fecha de la póliza.

4) IMP_CHEQUE .- en este campo de tipo numérico se almacena el importe original de la póliza a cheque. Su rango es de 0000000000.00 a 9999999999.99.

5) NOMCHEQUE .- este campo contiene el nombre del cheque o a quien se le expidió. Es de tipo carácter con una longitud de 50.

6) NO_MEMO .- en este campo se almacena el número de memorandum. Es de tipo numérico y cuenta con una longitud de 5, por lo cual su rango va de 00000 a 99999.

7) NUMCHEQUE .- este campo contiene el número del cheque. Es de tipo carácter y de longitud igual a 10. Como se valida de tipo numérico su rango va de 0000000000 a 9999999999.

8) NUME_POL .- este campo almacena el número de la póliza, es de tipo numérico y longitud 4, por ello, su rango va desde 0000 hasta 9999.

9) SIT_POL .- es de tipo numérico con longitud 1. Contiene la situación de la póliza. Su rango es (1 si la póliza está vigente ó 2 si la póliza está cancelada).

10) TIPO_POL .- en este campo se almacena el tipo de la póliza. Es de tipo carácter de longitud 2. Su rango es ("CH" en caso de ser de tipo cheque, "IN" en caso de ser póliza de ingreso, "PD" si es póliza de diario, "GB" si es giro bancario).

Para esta estructura se cuenta con dos archivos índices de ordenamiento:

a) poliza+año.NTX cuya llave está formada por lo siguiente $\text{right}(\text{dtoc}(\text{fech_pol},2) + \text{str}(\text{month}(\text{fech_pol},2) + \text{tipo_pol} + \text{str}(\text{nume_pol},4),$

b) poli2+año.NTX en este caso la clave de ordenamiento es $\text{tipo_pol} + \text{str}(\text{nume_pol},4) + \text{dtos}(\text{fech_pol})$.

VIII) CUENCONT.DBF. Esta estructura se ha designado para guardar la información relacionada a las cuentas contables utilizadas por el Instituto de Ingeniería, pero solamente la necesaria para que funcione como catálogo. Su esquema:

NOMBRE DEL CAMPO	TIPO	LONGITUD
CUENTA	Character	4
SCCUENTA	Character	4
SSCCUENTA	Character	4
DES_CUEN	Character	40
TIPO_CUEN	Numeric	1
AFEC_PROY	Numeric	1
AFEC_COMP	Numeric	1

1) CUENTA .- guarda la clave de la cuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

2) SCUENTA .- guarda la clave de la subcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

3) SSCUENTA .- guarda la clave de la subsubcuenta contable, es de tipo carácter, pero se valida la captura de valores numéricos, su rango va de "0000" a "9999".

4) DES_CUEN .- como su nombre lo indica, en este campo se almacena la descripción de la cuenta contable ingresada. Su tipo es carácter y tiene una longitud de 40.

5) TIPO_CUEN .- este campo contiene el tipo de cuenta contable. Es de tipo numérico y longitud igual a 1. Su rango es (1 si es acreedora ó 2 si es deudora).

6) AFEC_PROY .- este campo es de tipo numérico con una longitud de 1. Contiene la información de la cuenta (1 si afecta a proyectos de investigación ó 2 en caso de que no los afecte).

7) AFEC_COMP .- este campo es de tipo numérico con una longitud de 1. Contiene la información de la cuenta (1 si afecta a anticipos a reserva de comprobar ó 2 en caso de que no los afecte).

Otros archivos ocupados por el sistema como catálogos son los siguientes; en estos casos solamente describiremos los campos que de alguna forma afectan al sistema contable:

Para esta estructura se cuenta con dos archivos índices de ordenamiento:

a) cuencont.NTX cuya llave está formada por lo siguiente ccuenta+scuenta+sscuenta,

b) cuennomb.NTX en este caso la clave de ordenamiento es ccuenta + scuenta + sscuenta + upper(desc_cuen).

D) DATOSGEN.DBF. Contiene los datos generales de los proyectos. Su esquema se presenta a continuación:

NOMBRE DEL CAMPO	TIPO	LONGITUD
AVAN_ACANT	Numeric	3
AVAN_ACU	Numeric	3
AVAN_MES	Numeric	3
COORD	Character	3
COS_PRES	Numeric	12
FECHAINI	Date	8
FECHATERR	Date	8
FECHATER	Date	8
FECHA_CAP	Date	8
GPOT1	Character	4
GPOT2	Character	4
GPOT3	Character	4
JEFEPROY	Character	12
NOMPROY	Character	30
NOMPROY1	Character	30
NOPROY	Character	4
SITUACION	Character	1
SUD	Character	1
TIPOPROY	Character	1

1) COORD .- este campo de de tipo caracter y longitud igual a 3. Contiene la clave de la coordinación a la que pertenece el proyecto de investigación que se ve afectado. Su rango va de 000 a 999, debido a que se valida que sea numérico.

2) NOPROY .- en este campo se almacena el número de proyecto que se ve afectado por el movimiento de la póliza. Es de tipo caracter y de longitud igual a 4, sin embargo, su validación se hace como si fuera numérico. Su rango va de 0000 a 9999.

3) SITUACION .- en este campo se almacena la situación del proyecto de investigación. Es de tipo caracter con una longitud de 1. Su rango es de ("V" en caso de que esté vigente, "C" si fue cancelado y/o "T" si ha sido terminado).

Para esta estructura se cuenta con dos archivos índices de ordenamiento:

a) dg.NTX cuya llave es coord+noproj,

b) dg_proy.NTX en este caso la clave de ordenamiento es noproj.

II) EJE_año+mes.DBF. Esta estructura guarda la información de los ejercicios a los proyectos; esto es, la afectación a ellos. Su esquema:

NOMBRE DEL CAMPO	TIPO	LONGITUD
ANO_EJERC	Character	2
ANO_POLIZA	Character	2
CAMBIO	Numeric	14
CARGO	Character	1
COO	Character	3
COSTOU	Numeric	14
COSTO	Numeric	14
CTDAD	Numeric	14
CUENTA	Character	3
DESCRIP	Character	40
EN_QUE	Character	40

FECHAP	Date	8
FECHAQ	Date	8
FEHCA_MOV	Date	8
GRUPO_II	Character	3
NOPROY	Character	4
NUME_POL	Numeric	4
NUMEFACT	Character	10
PARTIDA	Character	3
PART_PROY	Character	3
PROGRAMA	Character	2
REFERENCIA	Character	10
SIT_DOCU	Character	1
SUBPROG	Character	2
TGRUPO	Character	1
TIPO_DOC	Character	2
TIPO_POL	Character	2

1) ANO_EJERC .- es de tipo caracter y longitud de 2. En él se almacena el año de ejercicio de la póliza original. Su rango se valida de tipo numérico, por lo cual va de 00 a 99.

2) ANO_POLIZA .- es de tipo caracter y longitud de 2. En él se almacena el año de la póliza. Su rango se valida de tipo numérico, por lo cual va de 00 a 99.

3) CARGO .- este campo contiene el tipo de cargo que se aplica a la póliza. Es de tipo caracter y de longitud de 1. Su rango es ("I" si el cargo es por parte del Instituto de Ingeniería o "U" si es por parte de la UNAM).

4) COO .- este campo de de tipo caracter y longitud igual a 3. Contiene la clave de la coordinación a la que pertenece el proyecto de investigación que se ve afectado. Su rango va de 000 a 999, debido a que se valida que sea numérico.

5) COSTO .- en este campo se almacena el costo o importe total de la póliza. Es de tipo numérico y de longitud 14.2, por lo cual su rango va de 00000000000.00 a 9999999999.99.

6) EN_QUE .- como su nombre lo indica, en este campo se almacena la descripción de la póliza ingresada. Su tipo es caracter y tiene una longitud de 40.

7) GRUPO_II .- este campo contiene la clave del grupo de gasto que se afecte con la póliza. Es de tipo caracter de longitud 3, pero su validación se realiza como si fuese numérico, por lo que su rango va de 0.0 a 9.9.

8) NOPROY .- en este campo se almacena el número de proyecto que se ve afectado por el movimiento de la póliza. Es de tipo caracter y de longitud igual a 4, sin embargo, su validación se hace como si fuera numérico. Su rango va de 0000 a 9999.

9) NUME_POL .- este campo almacena el número de la póliza, es de tipo numérico y longitud 4, por ello, su rango va desde 0000 hasta 9999.

10) PARTIDA .- es de tipo caracter y longitud 3. Se encarga de almacenar la clave de la partida del grupo de gasto afectado. Se valida de tipo numérico por lo que su rango va de 000 a 999.

11) PART_PROY .- es de tipo caracter y longitud 3. Se encarga de almacenar la clave de la partida del grupo de gasto del proyecto afectado. Se valida de tipo numérico por lo que su rango va de 000 a 999.

12) REFERENCIA .- en este campo se almacena la referencia de la póliza. Es de tipo caracter y de longitud 10.

13) SIT_DOCU .- este campo contiene la situación del documento o de la póliza. Es de tipo caracter y longitud 1. Su rango es de ("C" en caso de estar comprometida y, "E" en caso de haber sido ejercida).

14) TIPO_POL .- en este campo se almacena el tipo de la póliza. Es de tipo caracter de longitud 2. Su rango es ("CH" en caso de ser de tipo cheque,"IN" en caso de ser póliza de ingreso,"PD" si es póliza de diario,"GB" si es giro bancario).

III) PRES_PRO.DBF. Catálogo que contiene el presupuesto general de los proyectos de investigación, cuyo esquema se presenta a continuación:

NOMBRE DEL CAMPO	TIPO	LONGITUD
CDACUM_ANT	Numeric	15 2
CD_AN_UN	Numeric	15 2
CDCOMPR	Numeric	14 2
CDCOMPUNAM	Numeric	14 2
CDEJER	Numeric	14 2
CDEJERUNAM	Numeric	14 2
CDPRESU_P	Numeric	14 2
GRUPO_II	Character	3
NOPROY	Character	4
NOPROYG	Character	4
PARTIDA	Character	3
PORCT1	Numeric	3
PORCT2	Numeric	3
PORCT3	Numeric	3

TGRUPO	Character	1
--------	-----------	---

1) CDEJER .- este campo almacena la cantidad ejercida al proyecto de investigación afectado.

Es de tipo numérico con una longitud de 14.2, por lo cual su rango va de 00000000000.00 a 99999999999.99.

2) GRUPO_II .- este campo contiene la clave del grupo de gasto que se afecte con la póliza.

Es de tipo caracter de longitud 3, pero su validación se realiza como si fuese numérico, por lo que su rango va de 0.0 a 9.9.

3) NOPROY .- en este campo se almacena el número de proyecto que se ve afectado por el movimiento de la póliza. Es de tipo caracter y de longitud igual a 4, sin embargo, su validación se hace como si fuera numérico. Su rango va de 0000 a 9999.

4) PARTIDA .- es de tipo caracter y longitud 3. Se encarga de almacenar la clave de la partida del grupo de gasto afectado. Se valida de tipo numérico por lo que su rango va de 000 a 999.

El archivo índice empleado es pr_p_g_i.NTX cuya clave está formada por noproy + grupo_ii + tgrupo + partida.

IV) TOD_PART.DBF. Esta estructura contiene la información de las partidas correspondientes a los grupos de gastos para el instituto. Su esquema es:

NOMBRE DEL CAMPO	TIPO	LONGITUD
GRUPO_II	Character	3
PARTIDA	Character	3
CONCEPTO	Character	70
DESC_COMP	Character	20
PART_PROY	Character	3
CONC_PROY	Character	32

CARA_IMPOR	Numeric	1
CON_FACTU	Numeric	1
SITRANSFI	Numeric	1
TIOPAR	Character	1
FACT_ANTG1	Numeric	3
FACT_ANTG2	Numeric	3

1) GRUPO_II .- este campo contiene la clave del grupo de gasto que se afecte con la póliza. Es de tipo caracter de longitud 3, pero su validación se realiza como si fuese numérico, por lo que su rango va de 0.0 a 9.9.

2) PARTIDA .- es de tipo caracter y longitud 3. Se encarga de almacenar la clave de la partida del grupo de gasto afectado. Se valida de tipo numérico por lo que su rango va de 000 a 999.

3) PART_PROY .- es de tipo caracter y longitud 3. Se encarga de almacenar la clave de la partida del grupo de gasto del proyecto afectado. Se valida de tipo numérico por lo que su rango va de 000 a 999.

El archivo de ordenamiento es tod_pax.NTX con la llave partida+grupo_ii.

V) DGPTGRAL.DBF. Como se puede apreciar en el esquema, esta estructura contiene los cobros, contratos y, otros datos importantes de los patrocinadores de los proyectos de investigación.

NOMBRE DEL CAMPO	TIPO	LONGITUD
ANO_CONTRA	Character	2
COBRANZA	Character	90
COBRO_ACD	Numeric	14 2
COBRO_ACT	Numeric	14 2
COORD	Character	3

COPATRO	Numeric	15
COST_CONT	Numeric	14 2
COS_DLL	Numeric	14 2
CUENTA_ING	Character	3
FECHATER	Date	8
FIN_VIGEN	Date	8
INI_VIGEN	Date	8
NOCONTRAT	Character	20
NOMCONTRA	Character	120
NOREV	Character	1
PATROC	Character	7
RESPONCONT	Character	30
REV	Character	1
SIT_CONTRA	Numeric	1
TIOPAT	Character	3

1) ANO_CONTRA .- es de tipo caracter y longitud de 2. En él se almacena el año de contrato del proyecto de investigación. Su rango se valida de tipo numérico, por lo cual va de 00 a 99.

2) COORD .- este campo de de tipo caracter y longitud igual a 3. Contiene la clave de la coordinación a la que pertenece el proyecto de investigación que se ve afectado. Su rango va de 000 a 999, debido a que se valida que sea numérico.

3) FIN_VIGEN .- este campo es de tipo fecha; contiene la fecha probable de término del proyecto.

4) INI_VIGEN .- este campo es de tipo fecha; contiene la fecha de inicio del proyecto.

5) NOCONTRAT .- este campo almacena el nombre o clave del contrato del patrocinador del proyecto de investigación. Es de tipo caracter con una longitud de 20.

6) PATROC .- en este campo se almacena la clave del patrocinador del proyecto de investigación. Su tipo es caracter y su longitud de 7.

El archivo índice para esta estructura es gptgr1.NTX formado por la llave nocontrat+norev.

VII) UNIRES.DBF. En esta estructura se guarda la información general relacionada con las coordinaciones. Su esquema se presenta a continuación:

NOMBRE DEL CAMPO	TIPO	LONGITUD
ABREVCOO	Character	8
COORD	Character	3
DIG_UR	Character	1
NOMCOORD	Character	50
RESPON	Character	40
SUBD	Character	1

1) COORD sirve para guardar la clave de la coordinación. Al igual que SUBD es un dato de tipo caracter pero al cual solamente se permite introducir dígitos. Su rango (0,1,2,...,999).

2) NOMCOORD conlleva el nombre de cada una de las coordinaciones. Su rango es hasta de 50 caracteres pues es el límite de la longitud de campo.

3) RESPON contiene el nombre del responsable de la coordinación. Su rango es de hasta 40 caracteres.

4) SUBD contiene el número clave de la subdirección, a pesar de ser de tipo caracter por comodidad en la indexación se restringe a que su contenido es un dígito. Rango {0,1,2,3,4,5,6,7,8,9}.

Para ordenamiento se auxilia del archivo índice unires2.NTX cuya clave es coord.

VIII) CUENTASB.DBF. Esta estructura se ocupa como catálogo de cuentas bancarias. Su

esquema se muestra a continuación:

NOMBRE DEL CAMPO	TIPO	LONGITUD
NUME_CTA	Character	10
NOMBANCO	Numeric	1
ANO_ASIG	Character	2
CUENTA_B	Character	10
RFC_RESPO	Character	14
RESP_CTA	Character	30
SUBCODIGO	Character	13
ULTHOJA	Numeric	3
ULTSALDO	Numeric	14 2
SALDO_REAL	Numeric	14 2
ULTCHEQUE	Character	10
ULTFICHA	Character	10
ULTFECHA	Date	8
NUME_SIG	Numeric	4
NUME_SGL	Character	2
FONDO_CTA	Numeric	2

1) NUME_CTA .- este campo contiene el número de la cuenta bancaria. Es de tipo caracter con una longitud de 10. Por ser su validación numérica su rango va de 0000000000 a 9999999999.

2) RESP_CTA .- en este campo se almacena el nombre del responsable de la cuenta bancaria.

Su tipo es caracter y su longitud es de 30.

3) SALDO_REAL .- este campo contiene el saldo total de la cuenta bancaria. Es de tipo numérico, su rango es de 0000000000.00 a 9999999999.99.

Para esta estructura se tiene el archivo índice cuentasb.NTX cuya llave es nume_cta.

Con respecto a la capacitación, ésta se dará a los usuarios en tres días para explicar el funcionamiento del sistema.

La **implantación** del sistema se hará en forma total ya que este es un sistema completamente nuevo que no sustituye a otro.

RESULTADOS

- 1) Se logró la adecuada captura de los datos así como su modificación y/o eliminación; considerando su respectiva validación.
- 2) Las consultas y reportes que fueron solicitados por las personas que intervinieron en el análisis, funcionan adecuadamente.
- 3) La información obtenida a través del sistema contable, se obtiene de una forma veraz y oportuna, por lo cual ayuda a la toma de decisiones a las personas que así lo requieren.
- 4) Con la aplicación de las bases de datos y utilizando la herramienta de desarrollo Clipper 5.01, se logró una disminución en el tiempo de acceso a los datos y la respuesta del sistema fue inmediata; así como también se logró la disminución de los archivos físicos tradicionales.
- 5) Con el manejo de la técnica de la Programación Orientada a Objetos, se logró una disminución en los programas fuentes, así como una simplificación y reutilización de los mismos, por lo cual, su mantenimiento será mucho más fácil de realizar.

CONCLUSIONES

- 1) El objetivo del presente trabajo fue alcanzado.
- 2) El aplicar una metodología durante las etapas de análisis, diseño e implantación fue de una valiosa ayuda.
- 3) Se obtuvo como resultado un sistema amigable y muy flexible para ser migrado a otras plataformas y arquitecturas.
- 4) El reto más importante que se presentó fue el escaso presupuesto asignado al proyecto, ya que se tuvo que invertir en la compra de la actualización del software (Clipper 5.01), en incrementar los recursos del servidor que se encontraba (memoria y discos duros) y, en la adquisición de memoria y discos duros para tres microcomputadoras de desarrollo.
- 5) Algo de lo más valioso que se adquirió, además de la satisfacción de ver el sistema funcionando adecuadamente, que ha sido aceptado de buen agrado por el usuario y de saber que sigue actualmente satisfaciendo necesidades de la gente que interactúa con él; es el saber que dentro de la Universidad se puede estar seguro que el trabajar en equipo deja muy buenos resultados.
- 6) Como se mencionó anteriormente, el sistema sigue funcionando actualmente en el Instituto de Ingeniería de la UNAM.
- 7) Este sistema ha sido adecuado y se ha implantado en el Posgrado de Ingeniería de la UNAM y en el Centro de Instrumentos de la UNAM.
- 8) La gente que se encuentra dando mantenimiento a este sistema contable está analizando la posibilidad de migrarlo a otra plataforma, ya que el Instituto está tendiendo a la utilización del lenguaje Delphi.

ANEXO 1.- CONCEPTOS BÁSICOS DE CONTABILIDAD

Conceptos Básicos.

La contabilidad surge de la necesidad misma del hombre de llevar la cuenta de sus bienes, de las transacciones que realizaba, etc.; pero fue hasta 1494 en Italia cuando el monje franciscano Fray Luca Pacioli publica sus métodos, procedimientos y recursos contables en su obra "*Summa de Arithmetica, Geometrica, Proportioni Et Proportionalita*".

La **contabilidad** puede entenderse como una técnica constituida por métodos, procedimientos e instrumentos que aplicados con un conocimiento y una habilidad conducen al registro, clasificación y resumen de los efectos financieros que provocan las operaciones que realiza una empresa.

Una **empresa** es una entidad socioeconómica en la cual se combinan los factores de la producción con el fin de comprar y vender o comprar, transformar y vender satisfactores.

Los **factores de la producción** son la tierra, el trabajo y el capital, aunque la teoría económica actual también considera a las materias primas, la organización y la técnica.

Existen diferentes clasificaciones de las empresas:

Por el número de personas integrantes se clasifican en a) individuales y, b) colectivas.

Por la procedencia del capital en a) privadas y, b) públicas.

Por la actividad desarrollada a) comerciales y, b) industriales.

Por su finalidad se clasifican en a) lucrativas y, b) de servicio.

Los conceptos básicos de la contabilidad son: lo que se conoce como los **estados financieros**, que son documentos básica y esencialmente numéricos, con los cuales se muestra ya sea, la situación financiera de la empresa (por medio de lo que se conoce como **balance general**) o los resultados de su operación (conocidos como **estado de resultados**).

SALIR DE LA BIBLIOTECA
ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Como **activo** se **entiende** al total de los recursos de que dispone la empresa para llevar a cabo sus operaciones.

Entonces, el **pasivo** es el total de deudas contraídas por la empresa.

Otro término importante dentro de los conceptos de contabilidad es el que se conoce como **capital contable**, el cual es la suma de las aportaciones de los propietarios modificada por los resultados de la operación de la empresa, esto es, el capital social más las utilidades o menos las pérdidas. Como **capital social** se entiende a la suma de las aportaciones de los socios.

- 1) El balance general, como se mencionó muestra el estado en que se encuentra la empresa en un momento determinado con respecto a la obtención del dinero y su empleo con relación al logro del objetivo para el cual fue creada. Este balance general es un documento contable en el cual se observa el equilibrio existente entre el total de los recursos de la empresa y la suma de las deudas, más las aportaciones de sus propietarios. Basándose en su activo, su pasivo y su capital contable cuantificados en una fecha determinada. La fórmula que puede ejemplificar esto es:

$$A = P + CC$$

Entonces el esqueleto representativo sería:

BALANCE GENERAL DEL 01 DE OCT. AL 01 DE NOV.	
ACTIVO	PASIVO
Caja	Proveedores
Bancos	CAPITAL CONTABLE
Clientes	Capital social
	Utilidades o pérdidas

Dentro de la contabilidad se encuentra un término conocido como **teoría de la partida doble**, la cual sigue estas reglas:

- a) Siempre que se registren uno o varios cargos deben registrarse uno o varios abonos por el mismo importe.
- b) Siempre que se registren uno o varios abonos deben registrarse uno o varios cargos por el mismo importe.

Una **cuenta** es el reflejo de los cargos o abonos que sufre el concepto afectado por las operaciones que realiza la empresa.

El **libro mayor** es el documento que contiene a todas las cuentas contables.

Las partes de una cuenta son las siguientes:

- a) **Debe.**- se representa en el lado izquierdo de la cuenta, es un **cargo** que implica un aumento del activo o una disminución de pasivo o capital contable.
- b) **Haber.**- se representa en el lado izquierdo de la cuenta, esto es un **abono** que implica un aumento de pasivo o de capital contable o una disminución de activo.

Cuando una cuenta tiene registrados cargos y abonos se dice que esta en movimiento.

Como **saldo** vamos a entender el resultado de la diferencia aritmética entre los cargos y los abonos.

El **resultado en ejercicios anteriores** es el acumulado de las utilidades y pérdidas de ejercicios anteriores que han quedado pendientes de aplicarse o de amortizarse.

Al concluir cada ejercicio y después de determinar la utilidad o la pérdida generada durante el mismo, se corre lo que se conoce como **asiento de cierre** que demuestra que la suma de los saldos deudores es igual a la suma de los saldos acreedores.

Una **balanza de comprobación** es un documento que se elabora con el fin de contar con todos los elementos necesarios para verificar la aplicación correcta de la teoría de la partida doble. Esta indica los movimientos y saldos de la cuenta durante un periodo contable.

Ejemplo:

CAJA

Abono	Cargo
45000	13150
Saldo: 31850	

BALANZA DE COMPROBACIÓN AL 7 DE DICIEMBRE DE 1990

CUENTA	MOVIMIENTOS		SALDO
	Abono	Cargo	
CAJA	45000	13150	31850 (+)
FAMSA	25000	335000	310000 (-)

A un estado financiero de pérdidas y ganancias se le conoce con **estado de resultados**, este muestra la utilidad y pérdida que sufre el capital contable o patrimonio de la empresa como consecuencia de las operaciones practicadas durante un periodo de tiempo.

Al excedente de los ingresos sobre los costos se le llama **utilidad bruta**.

Pérdida bruta es el excedente de los costos sobre los ingresos.

ANEXO 2.- PAQUETES CONTABLES

Hace varios años nadie pensaba en el desarrollo que tendría la industria de la computación. Con el paso del tiempo, gracias al desarrollo tecnológico y a la constante baja de precios, muchas empresas que no podían instalar un gran centro de cómputo vieron la posibilidad de adquirir microcomputadoras para automatizar algunos de sus procesos.

No es casualidad que uno de los primeros de estos procesos fuese la contabilidad, punto crítico de muchas organizaciones. En México los sistemas contables se han convertido en una herramienta indispensable dada la constante evolución de las leyes fiscales.

Al hacer el análisis también se contempló la posibilidad de comprar algún paquete contable ya existente en el mercado, por lo cual se investigó con algunos proveedores las posibilidades. Teniendo en cuenta qué tan grande sería nuestro catálogo de cuentas, cuántos niveles de cuentas debería de soportar, qué cantidad de pólizas se capturan en un mes, qué reportes se necesitan con mayor frecuencia. Los puntos que se solicitaron para que cubrieran los paquetes según las necesidades de los usuarios fueron:

- 1) Captura de datos: donde puede considerarse el catálogo de cuentas, de conceptos y los saldos iniciales.
- 2) El módulo de pólizas: que permita dar cargos y abonos en cero y negativos, numeración de las pólizas en forma manual o automática.
- 3) Consultas por pantalla: estados financieros, pólizas del periodo y de fecha valor, cuentas con auxiliares y de mayor.
- 4) Reportes impresos: balanza de comprobación, mayor, auxiliar de mayor, relación de cuentas, diario de pólizas, catálogos principales.

- 5) Procesos eventuales: cierres mensuales y anuales, apertura del siguiente ejercicio, generación de pólizas de cierre anual y respaldos de información.
- 6) Comunicación con otros programas: si permiten exportar datos con el objeto de que sean aprovechados por otros sistemas.
- 7) Manuales: tanto de usuario como técnico.
- 8) Capacitación: si va incluida en el precio o tiene un costo extra.
- 9) Instalación de la base y el soporte: si cuenta con alguna garantía de un tiempo de soporte, si dejan instalada la base y funcionando, cuántas copias podemos tener.

Los programas que se cotizaron fueron los siguientes:

- 1) **Aspel-COI** .- de APEMEX S.A. de C.V. Su año de aparición es en 1979. Requiere de un sistema operativo MS-DOS o Redes. Está desarrollado en el lenguaje C-BASIC. Necesita 256Kb de memoria RAM como mínimo. No es necesario utilizar espacio en el disco duro. La capacitación es a través del distribuidor. Puede conectarse a otros sistemas. Su precio de lista es de \$210 dólares. Este paquete se descartó porque el lenguaje con el cual se desarrolló no es conocido por los miembros del Instituto, esto implica capacitación extra en este lenguaje y, con ello costos extras al proyecto.
- 2) **Contabilidad 2000** .- de SISTEMAS ESTRATÉGICOS SA de CV. Su año de aparición es en 1987. Requiere de un sistema operativo MS-DOS o Redes. Está desarrollado en el lenguaje dBASE III. Necesita 512Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra. No puede conectarse a otros sistemas. Su precio de lista es de \$250 dólares. Este paquete se descartó porque no se puede conectar a otros sistemas.
- 3) **Conta-Flex** .- de EDITORIAL CONTAFLEX SA de CV. Su año de aparición es en 1985. Requiere de un sistema operativo MS-DOS o Redes o Unix/Xenix. Está

desarrollado en el lenguaje DATA-FLEX. Necesita 384 Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro, mínimo 10 Mb. La capacitación no es necesaria, existe una ayuda en línea. Puede conectarse a otros sistemas. Su precio de lista es de \$600 dólares. Este paquete se descartó porque el lenguaje con el cual se desarrolló no es conocido por los miembros del Instituto, esto implica capacitación extra en este lenguaje y, con ello costos extras al proyecto; además el costo es muy elevado.

- 4) **Contabilidad General** .- de ASESORIA Y DES. CREATIVO SA de CV. Su año de aparición es en 1988. Requiere de un sistema operativo MS-DOS. Está desarrollado en el lenguaje Turbo BASIC. Necesita 512Kb de memoria RAM como mínimo. No es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra, con una póliza de mantenimiento. Puede conectarse a otros sistemas. Su precio de lista es de 40 salarios mínimos diarios. Este paquete se descartó porque el lenguaje con el cual se desarrolló no es conocido por los miembros del Instituto, esto implica capacitación extra en este lenguaje y, con ello costos extras al proyecto; la capacitación del paquete es con costo extra; tiene un precio elevado.
- 5) **Contabilidad General** .- de LÁNDA DE MEXICO SA de CV. Su año de aparición es en 1990. Requiere de un sistema operativo MS-DOS. Está desarrollado en el lenguaje Clipper. Necesita 512Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro, mínimo 10 Mb. La capacitación es sólo de la instalación. No puede conectarse a otros sistemas. Su precio de lista es de \$300 dólares. Este paquete se descartó porque no puede conectarse a otros sistemas.
- 6) **Contabilidad General DAC** .- de DIAGS. ADMN. POR COMPUTADOR SA. Su año de aparición es en 1983. Requiere de un sistema operativo MS-DOS. Está desarrollado

en el lenguaje C. Necesita 256Kb de memoria RAM como mínimo. No es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra. Puede conectarse a otros sistemas. Su precio de lista es de \$350,000,000.00MN. Este paquete se descartó porque el costo de la capacitación era muy elevado.

- 7) **Contabilidad Lomas** .- de DISTRIBUIDORA CITLALI SA de CV. Su año de aparición es en 1987. Requiere de un sistema operativo MS-DOS. Está desarrollado en el lenguaje Turbo Pascal Ensamblador. Necesita 256Kb de memoria RAM como mínimo. No es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra. No puede conectarse a otros sistemas. Su precio de lista es de \$250 dólares. Este paquete se descartó porque el costo de la capacitación era muy elevado; no puede conectarse a otros sistemas.
- 8) **Contabilidad MG** .- de COM. DE MICROPRODUCTOS SA de CV. Su año de aparición es en 1986. Requiere de un sistema operativo MS-DOS. Está desarrollado en el lenguaje Better BASIC. Necesita 256Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra. No puede conectarse a otros sistemas. Su precio de lista es de \$195 dólares. Este paquete se descartó porque el costo de la capacitación era muy elevado; el lenguaje en el que se encuentra desarrollado no es conocido por los miembros del Instituto, lo que provocaba un costo extra en capacitación; y no puede conectarse a otros sistemas.
- 9) **Contavision** .- de TECNOSISTEMAS DE HERMÓSILLO SA. Su año de aparición es en 1985. Requiere de un sistema operativo MS-DOS o Redes. Está desarrollado en el lenguaje COBOL. Necesita 128Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro. La capacitación incluye un curso, instalación y mantenimiento. Puede conectarse a otros sistemas. Su precio de lista es de \$195

dólares. Este paquete se descartó porque el lenguaje en el que se encuentra desarrollado es desconocido por los miembros del Instituto lo que provoca un costo extra por capacitación al proyecto.

10) **Multisoft Contabilidad.**- de MULTISIS, MODULARES DE COMP. SA. Su año de aparición es en 1985. Requiere de un sistema operativo MS-DOS o Redes o Xenix. Está desarrollado en el lenguaje RM-COBOL. Necesita 512Kb de memoria RAM como *mínimo*. Es necesario utilizar espacio en el disco duro, mínimo 10 Mb. La capacitación tiene un costo extra. Puede conectarse a otros sistemas. Su precio de lista es de \$195 dólares. Este paquete se descartó porque el lenguaje en el que se encuentra desarrollado es desconocido por los miembros del Instituto lo que provoca un costo extra por capacitación al proyecto; además el costo de la capacitación del paquete era muy elevado.

11) **PACO .-** de FRALC. Consultores SC. Su año de aparición es en 1988. Requiere de un sistema operativo MS-DOS o Windows o Redes. Está desarrollado en el lenguaje C ambiente Windows. Necesita 640Kb de memoria RAM como mínimo. Es necesario *utilizar espacio en el disco duro, mínimo 10 Mb*. La capacitación tiene un costo extra al igual que el soporte. Puede conectarse a otros sistemas. Su precio de lista es de \$295 dólares. Este paquete se descartó porque el costo de la capacitación era elevado.

12) **STAR-Con Empresarial .-** de INFORMÁTICA, ING. Y ADMON. SA. Su año de aparición es en 1988. Requiere de un sistema operativo MS-DOS o Redes o Xenix o Star-Net. Está desarrollado en el lenguaje HOT. Necesita 512Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra. Puede conectarse a otros sistemas. Su precio de lista es de \$245 dólares. Este paquete se descartó porque el lenguaje en el que se encuentra desarrollado es

desconocido por los miembros del Instituto lo que provoca un costo extra por capacitación al proyecto; además el costo por capacitación era muy elevado.

- 13) **Sistema Int. De Contabilidad** .- de ADSISTEM SA de CV. Su año de aparición es en 1987. Requiere de un sistema operativo MS-DOS. Está desarrollado en el lenguaje dBASE III. Necesita 640Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro, mínimo 10 Mb. La capacitación tiene un costo extra. Puede conectarse a otros sistemas. Su precio de lista es de \$600 dólares. Este paquete se descartó porque el costo de la capacitación era elevado al igual que el precio mismo del paquete.
- 14) **Sistema de Cont. RRYCSA** .- de ROGELIO REYNA Y CIA SA de CV. Su año de aparición es en 1985. Requiere de un sistema operativo MS-DOS. Está desarrollado en el lenguaje GW-BASIC. Necesita 256Kb de memoria RAM como mínimo. Es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra. No puede conectarse a otros sistemas. Su precio de lista es de \$195 dólares. Este paquete se descartó porque el lenguaje en el que se encuentra desarrollado es desconocido por los miembros del Instituto lo que provoca un costo extra por capacitación al proyecto.
- 15) **Software TEA/GL** .- de GRUPO TEA SA. Su año de aparición es en 1985. Requiere de un sistema operativo MS-DOS. Está desarrollado en el lenguaje dBASE III. Necesita 512Kb de memoria RAM como mínimo. No es necesario utilizar espacio en el disco duro. La capacitación tiene un costo extra. Puede conectarse a otros sistemas. Su precio de lista es de \$600 dólares. Este paquete se descartó porque el costo del paquete y de la capacitación eran elevados.

ANEXO 3.-PROGRAMACION ORIENTADA A OBJETOS

Conceptos Básicos

Un programa completo consta de dos partes: las instrucciones ejecutables (o programa) y, los datos sobre los que actúan estas instrucciones. Dependiendo de la forma de organización de los datos y de los programas se consideran tres formas fundamentales de programar:

- a) **Programación procedural o por procedimiento:** Esta programación se caracteriza porque las instrucciones que componen al programa se ejecutan secuencialmente en un orden preestablecido. Además, el programa se divide en una serie de módulos o rutinas distribuidos en forma jerárquica; en donde los datos están desorganizados y solamente proporcionan valores que sirven para realizar cálculos o, en el mejor de los casos, para decidir la secuencia de ejecución de las instrucciones del programa.
- b) **Programación lógica:** En un lenguaje no procedural (de programación lógica), por ejemplo, utilizando el lenguaje PROLOG, las instrucciones no tienen un orden prefijado, pues reciben el control de un programa procedural que decide el orden en que ha de recibir control cada una de las instrucciones del programa; no existe una jerarquía entre estas reglas.
- c) **Programación orientada a objetos:** En la década de los ochenta, surge un nuevo conjunto de técnicas llamada *programación orientada a objetos (POO)*. La **programación orientada a objetos** ha surgido del concepto de la programación lógica, es decir, resolver problemas a través de inferencia en lugar de lo procedural. Entonces, la POO no es un lenguaje de programación, sino una técnica o estilo de programar cuyo objetivo consiste en disminuir los tiempos de programación, así como la estrategia para el diseño de aplicaciones y, desde luego, el mantenimiento de las mismas.

En la POO también existen programas, pero éstos están desorganizados; en esta técnica los datos y procedimientos son encapsulados en forma de objetos, con el fin de construir componentes de software discretos que visualicen y participen en el medio ambiente que refleje el comportamiento real de un objeto o evento.

La POO surge básicamente por la necesidad creciente de administrar grandes cantidades de datos interrelacionados, los cuales son manejados a través de un sistema. Su diseño es un proceso de descomposición de un programa en objetos estableciendo las relaciones entre ellos.

Esta técnica de programación tiene sus orígenes en los lenguajes como el LIPS y, el ALGOL; no pasando desapercibido PROLOG, que intenta manejar inferencias en lugar de algoritmos. A finales de los años sesentas, aparece SIMULA 68 basado en ALGOL, pero que ha sido el germen de muchos lenguajes actuales; por medio de este lenguaje se intenta simular lo que sucede en el mundo real; a través de objetos que responden a mensajes externos. Poco después surge SMALLTALK en 1972, que es el lenguaje más famoso de la POO, surgiendo con éste los conceptos de objeto, herencia, clases, mensajes y agrupamiento.

Para 1975, se comienza a manejar la POO como tal, después continúa desarrollándose esta técnica y, para 1986 se tiene un método para el Diseño Orientado a Objetos y, para 1989 surge el Análisis Orientado a Objetos.

Los paradigmas de programación usuales son básicamente los siguientes :

- a) **Enfoques lineales** .- en donde se puede incluir a los lenguajes no procedurales o estructurados.
- b) **Enfoques procedurales** .- aquí se engloba a los lenguajes que presentan una estructuración en su programación con el uso de funciones o procedimientos. Su objetivo es encontrar el algoritmo óptimo lógico basado en procedimientos, como ejemplos de estos lenguajes tenemos a Fortran, Pascal, C.

- c) **Enfoques modelados de datos (modular)** .- se consideran a los lenguajes que son estructurados pero que se basan principalmente en módulos. En este paradigma se encuentran definidos en un archivo los tipos de datos, las operaciones y las interfases necesarias. Se pueden ocultar los datos a través de los módulos para que éstos no puedan ser modificados.
- d) **Enfoques de tipos de datos abstractos (TDA)** .- Son los datos definidos por el programador; pueden crearse los tipos de datos que se deseen.
- e) **Enfoques de Programación Orientada a Objetos** .- Se parte de lo general a lo particular, con el uso de la herencia. Se analizan los objetos que se necesitan, se proporciona un conjunto completo de operaciones para cada clase; se hacen generales y, se utiliza la herencia.

Los lenguajes se pueden categorizar en la forma siguiente, de acuerdo al grado de utilización de la técnica de POO en su generación de código:

- 1) **Lenguajes Orientados a Objetos (puros)** : Un lenguaje de programación puro no permite salirse del entorno, esto es, no permite la ejecución de la instrucción CALL u otra equivalente. Estos manejan exclusivamente métodos. Por ejemplo Smalltalk, Eiffel, Actor.
- 2) **Lenguajes Híbridos:** Los cuales permiten que coexistan funciones y métodos. Esto es, permite salirse del entorno; pudiéndose mezclar las dos formas de programar en proporciones indiferentes. Como puede ser C++, Object C, Pascal Object.
- 3) **Procedurales:** Son lenguajes caracterizados por su manejo exclusivo de funciones. Como lo son Pascal, Algol, C, entre otros.

Un lenguaje POO debe poseer tres características esenciales para que pueda ser llamado así: deberá estar basado en objetos, deberá estar basado en clases y, ser capaz de heredar clases.

Como parte complementaria de las metodologías de análisis y diseño surge el concepto de **durabilidad** el cual intenta cuantificar qué es lo que permite sobrevivir a una aplicación a un rediseño. Este concepto cuenta con cinco cuantificaciones principales.

Una de estas cuantificaciones es la **abstracción** la cual consiste en la diferenciación de atributos de un objeto en particular; por lo cual, si un desarrollador es capaz de obtener una abstracción adecuada del módulo que está analizando, en el futuro podrá cambiarlo en la forma que desee y sin demasiada dificultad.

Una cuantificación más es la **generalidad** dentro de la cual se ve la necesidad de desarrollar funciones balanceadas, que permitan manejar cualquier situación en el futuro.

La tercera cuantificación es la **encapsulación**, la cual es similar a la abstracción, pero en menor escala y, cuyo objetivo es ocultar información acerca del objeto para el resto del sistema, siendo la "célula" de la POO.

Otra cuantificación igualmente importante es la **pureza en el código** lo cual implica lograr un algoritmo fácil de leer.

La quinta y última cuantificación es la **cohesión** que se refiere a reducir en lo posible la interfase entre módulos.

Un **Objeto** es un conjunto de datos que se encuentran encapsulados por el código que opera en ellos. El comportamiento de un objeto está sujeto o descrito en su código y en el estado actual en que se encuentre dicho objeto (sus atributos) representados por su información, la cual está protegida de alteraciones por parte de otros objetos. Los objetos se consideran abstracciones a un nivel más alto que un dato o un procedimiento; pues ellos contienen datos y procedimientos relacionados (asociados).

Las **propiedades**: distinguen a un objeto de los demás que forman parte de la organización, éstas pueden ser heredadas.

Los **métodos** son operaciones que pueden realizarse al objeto.

El **atributo** es algún dato o estado de información para el cual cada objeto en una clase tiene su propio valor. Esto es, son las variables y características de los objetos.

El **servicio** sirve como indicación para que un objeto se comporte de una forma específica, tomando como datos sus atributos; esto es, son métodos aplicados a los atributos.

La **clase** es un objeto encargado de crear objetos basados en reglas preestablecidas, es decir, en base a un modelo previamente definido. Esto implica ver a la clase como una constructora de objetos; porque a partir de ella se describe a uno o más objetos similares.

La **constructora** es un método que permite crear instancias u objetos de una clase específica.

La **instancia** es un objeto descrito por una clase en particular; porque dado que una clase es una constructora de objetos, éstos pueden ser de diferente tipo, sin embargo fueron creados por una misma clase. Por ende, un objeto puede considerarse como una instancia de la clase, el cual posee la implementación de la interfase con los datos.

Las **variables de instancia** se refiere a los datos o atributos encapsulados dentro de un objeto, esto significa que, corresponden a la información la cual pertenece a una instancia u objeto.

La **herencia (inherencia)** esta cualidad de la POO aplica el concepto de compartir código, logrando que el lenguaje sea quien lo reutilice y no el programador. Es una propiedad de los objetos.

El **método** es el código que opera sobre los objetos o instancias de una clase determinada; es decir, le indican al objeto cómo se desea que realice una operación para el acceso a los datos. Esto implica una función u operación intrínseca a alguna clase.

El **mensaje** es un mecanismo mediante el cual se comunican los objetos; éste permite a los métodos enviar o recibir datos. Lo que es lo mismo, los mensajes son indicaciones enviadas a los objetos para obtener información sobre sus atributos y/o servicios.

El **polimorfismo** es la reacción diferente de diferentes objetos a un mismo mensaje, esto es, se tiene la posibilidad de construir varios métodos diferentes que tengan el mismo nombre y poderlos utilizar al mismo tiempo.

La **propiedad** un objeto puede ser dueño de otro objeto y, por tanto, controlar su comportamiento.

La **ejecución (binding)** es el tiempo que tarda un objeto en responder ante un mensaje; considerando que, en cuanto el objeto recibe un mensaje éste debe interpretarlo y efectuar un método específico.

La **subclase** es una clase que posee todas las propiedades de su clase padre, así como también sufre los cambios hechos a éste.

La **superclase** es la clase padre de una subclase, esto es, es el antecesor de una clase.

La **agrupación apriori** es la forma de agrupación más rápida, pues durante la ejecución el objeto verifica la referencia directa (mensaje) y la ejecuta (método).

La **agrupación aposteriori** esta agrupación se refiere a la capacidad de asimilar cualquier tipo de actividad, esto es, las variables de instancia de cualquier objeto dentro de un sistema pueden responder a cualquier mensaje enviado. Esta forma de agrupación es la más lenta.

La metodología de POO posee algunas ventajas en el desarrollo del software como lo son:

- a) se logran modelos más naturales,
- b) la modularidad,
- c) la extensibilidad,
- d) la eliminación de redundancias,
- e) la facilidad de reutilización.

Los principales objetivos que se buscan con la aplicación de la POO son principalmente aumentar la calidad del software resultante y, disminuir su costo.

Como se sabe las fases de vida del software son, básicamente las siguiente: análisis, diseño, codificación, prueba y, mantenimiento.

En la programación clásica, ante un proyecto de desarrollo de software, se plantea el establecimiento de un modelo que defina el problema a resolver; este modelo subdivide al problema en un conjunto de subproblemas más sencillos, que los programadores traducen en un código jerárquico que cumpla con los requisitos finales.

Dentro de la POO, los datos son los que establecen las jerarquías, con lo cual los modelos se establecen de forma similar a la manera de razonar del hombre, siendo mucho más natural. El diseño de un modelo debe conceptualizarse como un conjunto de objetos que permitan el funcionamiento del mismo; por ejemplo, para la construcción de un automóvil necesitamos varios objetos, como la carrocería, el motor, el sistema de arranque, etc., y el desempeño o funcionamiento del mismo se realizará mediante la comunicación entre los diferentes objetos.

Con lo que respecta a la siguiente ventaja, la **modularidad**, ésta es un concepto clave dentro de la evaluación del software y uno de los principales objetivos para los diseñadores de aplicaciones.

Se dice que un programa es modular si se compone de módulos independientes y robustos, lo cual permite una mayor facilidad a la reutilización de componentes, así como la verificación y mantenimiento de los mismos.

Continuando con las ventajas que proporciona la POO, se tiene la **extensibilidad**, la cual se refiere que para que un sistema bajo la aplicación de la programación orientada a objetos, esté bien diseñado, es importante que el modelo sea correcto, pues el análisis en esta metodología es lo más importante y crítico de todo el ciclo de vida del software.

Otra ventaja de la utilización de la POO es la **eliminación de redundancias**. La propiedad clave que permite la eliminación de redundancias es la herencia, pues evita la definición múltiple de propiedades comunes a muchos objetos.

Por último, la ventaja de la **reutilización de código**. Esta ventaja estimula la reutilización de código en lugar de reinventarlo nuevamente, esto es, los módulos diseñados pueden utilizarse nuevamente en aplicaciones similares.

Gracias a la reutilización del código se reduce el tiempo empleado en el desarrollo de sistemas, además de decrementar considerablemente el tiempo de respuesta de los mismos.

La POO contrariamente a lo que pudiera pensarse, requiere de una mayor capacidad de los programadores, por lo cual tiende a agrandar la diferencia entre los buenos y los malos; por otro lado, es una técnica atractiva, pero peligrosa en las manos de un programador incompetente.

Algunas aplicaciones propensas a la utilización de la POO pertenecen a los siguientes campos:

a) **Interfaces de usuarios:** el usuario actual es una persona ajena a la programación, en su gran mayoría, que no desea sumergirse en él; por esto la proliferación de interfaces avanzadas que utilizan el ratón como ejemplo fundamental de interacción entre el usuario y el ordenador, sustituyendo al teclado, y que son capaces de generar estructuras complejas de presentación de la información tales como las siguientes:

- ventanas gráficas y de texto,
 - caja de selección simple: ventana que presenta varias alternativas verticales, de las que puede seleccionar una moviendo el cursor o por medio del ratón,
 - caja de selección múltiple: ventana que presenta varias alternativas verticales,
 - barra de acción: ventana que presenta varias alternativas horizontales,
 - caja de diálogo: ventana que presenta varias preguntas.
- b) **Gráficos:** es la representación de objetos a través de puntos y/o píxeles.

ANEXO 4.-BREVE INTRODUCCIÓN A CLIPPER 5.01

Definición y Características.

La versión inicial de Clipper fue liberada en el año de 1985, después en 1986 aparece Clipper Autumn'86. Para el verano de 1987 aparece la versión más difundida hasta el momento Clipper Summer'87 y, actualmente la última versión en el mercado mexicano es la de Clipper 5.02.

Clipper es considerado como un compilador de programas realizados en el manejador de bases de datos dBASE, hasta su versión 5.0, pues a partir de ella se observa como un lenguaje de programación, debido a algunas características particulares que presenta como son, entre otras, el manejo de la técnica de la programación orientada a objetos teniendo tres clases preconstruidas (TBrowse, TBColumn, Get y, Error).

Nantucket (empresa creadora de Clipper, hasta la versión 5.01) recomienda las siguientes especificaciones para que una aplicación de Clipper corra correctamente: Un equipo IBM PS/2, AT, XT, PC ó 100% compatible, la capacidad mínima de memoria RAM es de 512 Kb, es indispensable el disco duro para el desarrollo de aplicaciones (1.5 Mb o más), requiere una versión de sistema operativo MS-DOS igual o superior a la 3.1, puede funcionar tanto en ambiente monousuario como en ambiente multiusuario.

Clipper consta básicamente de dos etapas:

- a) crear el código fuente en cualquier editor y,
- b) compilación, ligado de utilerías y/o librerías, creando archivos .OBJ y .EXE.

Así mismo, permite realizar interfases con otros lenguajes de manera sencilla, principalmente con C y ensamblador. Tiene cuatro fases: a) edición, b) compilación, c) enlace y, d) trazado de errores (rtlink).

Las ventajas primordiales a partir de la versión 5. son las siguientes:

- 1) El procesador permite el uso de directivas de compilación para declarar constantes, funciones, comandos definidos por el usuario, compilación condicional, etc.
- 2) El preprocesador es importante, pues incluye archivos, sustituye la información de un archivo en otro durante la compilación; además puede darse una compilación condicional de los archivos fuentes, sustitución de textos como son constantes simbólicas y pseudofunciones.

La inclusión de archivos se realiza con la instrucción:

```
#include "nombre del archivo".
```

La sustitución de textos por la instrucción:

```
#define "nombre definición".
```

- 3) Tiene un nuevo enlazador `RTLINK` que aporta nuevas funcionalidades entre las que destacan los overlays dinámicos y las librerías preenlazadas.²
- 4) Arrays n-dimensionales y de dimensionamiento dinámico.
- 5) Enfoque a la Programación Orientada a Objetos; Clipper 5. cuenta con cuatro clases de objetos predefinidas: `GET`, `TBROWSE`, `ERROR` y `TBCOLUMN`.
- 6) Existen nuevos modos `LOCAL` y `STATIC` para la declaración de variables.
- 7) Los operadores de Clipper 5. son muy similares a los utilizados en el lenguaje C.
- 8) Se da la posibilidad de tener funciones y procedimientos estáticos; esto implica, que el mismo nombre puede emplearse para dos procedimientos diferentes, siempre que se declaren como `STATIC` (estáticos).
- 9) Se cuenta con una documentación en línea, a través de las Norton Guides, residente en memoria.
- 10) El depurador nuevo permite la ejecución del programa mientras se visualiza el código fuente del mismo.
- 11) Se logra el empleo configurable de la memoria expandida.
- 12) Contiene un manejador de memoria virtual (`VMM`), para que a las aplicaciones nunca les falte memoria donde correr.
- 13) Cuenta con nuevos tipos de datos. En la versiones anteriores se manejaban tipos de datos; numérico, lógico, carácter, fecha y, memo; ahora a las nuevas versiones (5.0 y superiores), se añaden tres nuevos tipos de datos: los bloques de código (`CodeBlocks`) éstos son apuntadores a código compilado, encargados de hacer referencia a piezas completas de código compilado además de poder ser utilizados en funciones, procedimientos, paso de parámetros, etc.; otro de los nuevos tipos de datos es `NIL` el cual es el valor por "default" asignado a las variables públicas (anteriormente era `.F.`) y,

² **NOTA:** El `RTLINK` fue sustituido por el antes ligador `LINK` del sistema operativo DOS 5. para buscar mayor eficiencia

variables de tipo objetos, las cuales son referencias (apuntadores) de la instancia de una clase particular.

- 14) Cuenta con el manejo de overlays estáticos y dinámicos. Los overlays son necesarios cuando las aplicaciones de Clipper tienen un tamaño muy grande. Los overlays reservan espacio en memoria para código que no puede ser adaptado dentro de la memoria disponible en un tiempo de corrida. La extensión de estos archivos es .OVL
- a) **Overlays estáticos.**- apartan el espacio de memoria del overlay más grande y lo mantienen constante aunque el overlay accesado en ese momento sea de menor tamaño.
 - b) **Overlays dinámicos.**- son más versátiles que los estáticos, pues ocupan el espacio de memoria que cada uno de ellos requiera.

Las **capacidades del sistema** son las siguientes:

- número máximo de registros por base de datos es 1 billón
- número máximo de campos por registro 1024
- número máximo de caracteres por campo 32Kb
- dígitos máximos en un campo numérico 30
- campos memo 64 Kb
- precisión en operaciones de cálculo 16 dígitos
- número máximo de caracteres en una clave de indexación 256
- número máximo de índices por área de trabajo 15
- número de variables de memoria públicas o privadas 2048
- número de variables locales o estáticas RAM libre
- número máximo de arrays 2048
- número máximo de elementos por dimensión de array 4096
- número máximo de ficheros abiertos (DOS 3.3) 250

Las **variables de entorno** son importantes, CLIPPER las utiliza para determinar el modo en que se asignarán objetos en memoria y la cantidad de ésta que se reservará para cada una de sus acciones; las más importantes de la versión 5. son las siguientes:

- **SET CLIPPER: El entorno Run-Time:** El mandato SET proporciona a la variable una serie de valores que le servirán para ajustar internamente los parámetros de gestión de memoria.
- **SET CLIPPERCMD: Opciones implícitas del compilador:** Determina la lista de opciones del compilador que de forma implícita se ejecutarán cuando se compile cualquier programa.

- **SET INCLUDE: Ubicación de los ficheros de cabecera:** Determinan el disco y directorio donde el compilador buscará los ficheros de cabecera (.CH) usados por los programas.

- **SET LIB: Ubicación de las librerías:** Determina el disco y el directorio donde el enlazador buscará las librerías (.LIB) usadas por los programas.

- **SET OBJ: Ubicación de los módulos objetos (.OBJ):** Determina el disco y directorio donde el enlazador buscará los objetos generados por el compilador.

- **SET PLL: Ubicación de las librerías preenlazadas:** Determina el disco y directorio donde en tiempo de enlace y ejecución se buscarán los ficheros que contienen las librerías preenlazadas.

- **SET RMAKE: Las opciones implícitas de RMAKE:** Determina la lista de opciones de RMAKE que se forma implícita se ejecutarán cuando usemos este utilitario para el desarrollo de aplicaciones.

- **SET RTLINKCMD: Opciones implícitas de RTLIK:** Determina las opciones del enlazador RTLINK que de forma implícita se ejecutarán cuando compilemos cualquier programa.

- **SET TMP: Ubicación de los ficheros temporales del compilador y el enlazador:** Determina el disco y directorio donde en tiempo de enlace y ejecución se depositarán los ficheros temporales usados por el compilador y el enlazador.

Es importante conocer que el compilado de clipper se divide en dos partes: el preprocesador y, la compilación.

El preprocesador, como se vió anteriormente, es una de las herramientas más poderosas de Clipper; aquí se puede tener código fuente y trasladarlo dentro de las llamadas a funciones primitivas de Clipper; pueden crearse sintaxis de comandos alternativos creados por el usuario, constantes manifiestas (macros) y, pséudofunciones.

El trabajo del compilador es el mismo que el del código del preprocesador y, el de crear un código objeto compatible con Microsoft/Intel, para la mayoría de las partes este proceso es automático, requiriendo pequeñas intervenciones por parte del programador.

Como Clipper es un lenguaje estructurado con el uso de algunas técnicas estructuradas se puede obtener un mayor aprovechamiento de las bondades de este lenguaje.

Los componentes principales de las técnicas estructuradas son los siguientes:

- a) **Análisis Estructurado:** Es una colección de herramientas de modelado que permiten a los progmdadores reemplazar especificaciones funcionales clásicas con modelos que el usuario pueda comprender.

- b) **Diseño e Implementación TOP-DOWN:** Diseño de un sistema mediante la partición de funciones mayores, que se particionan a su vez, en subfunciones menores.
- c) **Diseño Estructurado:** Es un conjunto de técnicas y directrices que ayudan al diseñador a distinguir entre los "buenos" y los "malos" diseños. Este diseño consiste en la composición de un sistema en módulos pequeños independientes.
- d) **Programación Estructurada:** Es el enfoque de programar todas las construcciones lógicas que componen un programa elaborado bajo Clipper en tres formas básicas:
 - Estatutos imperativos sencillos, tales como READ, GET, INPUT y, estatutos algebraicos como $A:=B+C$.
 - Estatutos de condición individual, IF...THEN...ELSE y, condición múltiple CASE...ENDCASE.
 - Estatutos de iteración (ciclos), DO WHILE y, REPEAT...UNTIL.

Enfoque a la Programación Orientada a Objetos

La durabilidad de un sistema está dada por las metodologías de análisis y diseño utilizadas. En Clipper se manejan varios conceptos de la programación orientada a objetos, tales como:

El **Objeto** es un tipo de dato que permite coleccionar un conjunto de datos de diferentes tipos y un conjunto de procedimientos (código asociado a los datos) que interactúan con los datos coleccionados. Esto es, es un **contenedor** de variables y funciones.

El **mensaje** es la forma de comunicarse con el objeto. Los mensajes son enviados por un objeto y recibidos por otro; éstos determinan el comportamiento de un objeto. Por ejemplo: si se tienen un objeto que gestione la cola de impresión de un sistema, el mensaje para dicho objeto sería "comienza la impresión".

Las **variables de instancia** en Clipper, son las variables que continen un objeto creado; entonces, se hace referencia a las variables de instancia como las variables de un objeto.

En Clipper, los **métodos** son las funciones que están contenidas en los objetos y que operan sobre las variables de instancia de dichos objetos.

Clipper 5.01 permite crear objetos de ciertas clases predefinidas.

Una **clase** en Clipper se da cuando un objeto es definido con atributos y servicios, de tal forma que una clase puede ser descrita como un modelo para la creación de objetos.

En Clipper 5.0 no se pueden crear clases en forma directa, por lo cual no se pueden crear "arreglos especiales", sino utilizar los ya definidos por Nantucket. Para ello, se cuenta con cuatro clases predefinidas: a)TBrowse, b)TBcolumn, c)Get y, d)Error; las cuales permiten codificar en un gran porcentaje, aplicaciones orientadas a objetos.

Los objetos, al igual que los arreglos, se manejan referencialmente; por lo cual, una variable no puede contener a un objeto pero sí una referencia a dicho objeto.

Un ejemplo de esto puede ser:

```
#Define UNO  tabla_ar[ 1]
#Define DOS  tabla_ar[ 2]
#Define TRES  tabla_ar[ 3]
#Define CUATRO tabla_ar[ 4]
#Define CINCO tabla_ar[ 5]
```

Function MAin()

```
Local tabla_ar := { .T.,.F.,"A","B","C" }
Qout( CINCO )      /* Imprime la letra C */
```

Return(NIL)

Para poder obtener información y asignar nuevos valores de los objetos, se utilizan los mensajes (los cuales pueden ser variables de instancia y métodos). En Clipper, es posible enviar mensajes a los objetos a través del operador **send**; el cual trabaja solamente con objetos. El símbolo asociado a este operador se representa con dos puntos ":". Su sintaxis es la siguiente:

```
< objeto > : < mensaje > [( < parámetros > )]
```

donde:

< objeto > es la referencia (variable de memoria) a un objeto creado.

< mensaje > puede ser una variable de instancia o un método.

Esto puede leerse como: ". objeto, te envío un mensaje".

Las **constructoras**, llamadas también "Funciones de Clase", permiten crear instancias u objetos de una clase específica. En Clipper las dos clases predefinidas existentes son: **TBrowse** y **TBColumn** cuyo objetivo es desplegar información de forma tabular, la cual puede ser tablas, bases de datos, arreglos, archivos de texto, etc.

Entonces, las constructoras son funciones que regresan como resultado una referencia a un objeto, esto es, permiten crear objetos que son instancia de una clase. Por ejemplo, la constructora **TBrowseNew()** crea una instancia de la clase **TBrowse** heredando sus atributos y servicios.

La Clase **Tbrowse**.- Esta clase es la abstracción del evento que consiste en desplegar información en forma tabular (renglones y columnas), manteniendo una barra posicionadora para poder observar la diferente información seleccionada y, controlar su desplegado a través del teclado. Se diseñó para sustituir el DBEIDT() de CLIPPER Summer'87.

Esta clase tiene dos principales constructoras:

- **TBrowseNew()** y,

- **TBrowseDB()**

Ambas reciben cuatro parámetros que corresponden a las coordenadas del área en donde deseamos visualizar la información. La diferencia que existe entre las constructoras se da en la codificación de los métodos que permiten posicionar el apuntador al renglón en donde aparece la barra de cursor.

La sintaxis de estas constructoras es:

TBrowseNew(<renglón superior>,<columna izquierda>,<renglón inferior>,<columna derecha>)

TBrowseDB(<renglón superior>,<columna izquierda>,<renglón inferior>,<columna derecha>)

La Clase **TBColumn**.- Esta constructora es denominada TBColumnNew() y no tiene sentido en solitaria por ser una herramienta de la clase TBrowse; se utiliza para crear columnas, su sintaxis es la siguiente:

TBColumnNew(<título>,<bloque de código>)

el título es cualquier expresión de tipo carácter para identificar el título de la columna. El parámetro de bloque de código (CodeBlock), indica la información que se desea desplegar.

Por ejemplo, si se desea desplegar en pantalla una columna con título "Clave" con contenido de todos los campos de la base de datos, de un campo llamado "_clave" y otra columna con título "Nombre" y, cuyo contenido sean todos los registros la misma base de datos, de un campo llamado "_nombre", se deberían construir dos objetos de dos clases diferentes:

```
Local _browse:=TBrowseDB(0,0,10,60)
```

```
Local _column1:=TBColumnNew("Clave",{[_clave]})
```

```
Local _column2:=TBColumnNew("Nombre",{[_nombre]})
```

La Clase **GET**.- El sistema GET es el mecanismo que maneja Clipper para la captura y edición de información. Con la clase GET podemos crear nuevos objetos para la edición y programarlos según nuestras necesidades.

La edición a través de objetos se produce del modo siguiente:

- 1) se crean tantos objetos GET como variables vayan a editarse.
- 2) El paso de la edición de un objeto a otro se realiza a través de la técnica de pasa el foco; en donde, quien posee el foco es el GET activo, es decir, el que se está editando en ese momento.

La Clase **ERROR**.- El sistema de manejo de errores en Clipper 5.01 es un mecanismo que permite controlar cualquier evento en el cual pueda ocurrir una anomalía debido al código. La clase **ERROR** permite la creación de objetos de error. Su forma de trabajo es la siguiente:

Cuando en la ejecución de un programa se produce un error, Clipper crea un objeto **ERROR**, el cual se pasa como parámetro al bloque manejador de errores utilizado en **ERRORBLOCK()**; después se le asigna un nuevo nombre cualquiera al nuevo objeto **ERROR** y creamos una función que maneje los errores que se pueden producir para indicársela al bloque como el código a ejecutar.

BIBLIOGRAFIA

- 1) ALFONSECA MANUEL y ALCALÁ ALFONSO.

Programación Orientada A Objetos. Teoría y Técnicas OPP para Desarrollo de Software.
Editorial ANAYA Multimedia.
Madrid , 1992.

- 2) ARAIZA Y WEAVER.

Contabilidad .Sistemas y Procedimientos.Actualizaciones, Inventarios y Automatización.
Editorial Mc-Graw Hill.
México, 1986.

- 3) ARAIZA y WEAVER.

Contabilidad. Sistemas y Procedimientos. Elementos de los Estados Financieros.
Editorial Mc-Graw Hill.
México, 1986.

- 4) BLAIR G., GALLER y Otros.

Object Oriented Languages Systems And Applications.
Editorial PITMN.
Gran Bretaña, 1991.

- 5) DEEN S. M.

Fundamentos de los Sistemas de Bases de Datos.
Editorial Colección de la Ciencia Informática (GG).
Barcelona, 1987.

- 6) FROST R.

Bases de Datos y Sistemas Expertos .Ingeniería del Conocimiento.
Editorial Ediciones Diaz de Santos, S.A.
Madrid, 1989.

- 7) FROST R.
Bases de Datos y Sistemas Expertos.
Editorial Ediciones Díaz de Santos, S.A.
Madrid, 1989.
- 8) GERALD E. PETERSON.
Object Oriented Computing. Vol.1.
Editorial Computer Society Press.
USA, 1987.
- 9) GERALD E. PETERSON.
Object Oriented Computing. Vol.2.
Editorial Computer Society Press.
USA, 1989.
- 10) LARSON JAMES A.
Database Management. Tutorial.
Editorial Computer Society Press.
New York , 1987.
- 11) MARÍN QUIRÓS FRANCISCO y Otros.
CLIPPER 5. Referencia Rápida.
Editorial Grupo Eidos. Coedición Macrobit-ra-ma.
México, 1991.
- 12) MARÍN QUIRÓS FRANCISCO y Otros.
CLIPPER Técnicas, Aplicaciones y Rutinas de Programación.
Editorial Macrobit.
México, 1990.

13) NANTUCKET.

Manual de CLIPPER. Manual de referencia.

Editorial Nantucket.

USA, 1987.

14) TILEY W. EDWARD.

CLIPPER Programmer's Reference.

Editorial QUE.

USA, 1991.

15) TILEY W. EDWARD.

Using CLIPPER. Tercera edición.

Editorial QUE.

USA, 1992.

16) YOURDON.

Information Systems Analysis Workshop. Lexture Notes Versión 3.01.

Editorial Prentice-Hall Building.

New Jersey, 1991.

17) YOURDON.

Systems Method. Model-Driven Systems Development.

Editorial Prentice-Hall Building.

New Jersey, 1993.

Revistas :

- 1) Revista mensual PC/TIPS. Año 4. No.42, julio 15 de 1991.
- 2) Revista mensual PC/TIPS. Año 5. No.56, septiembre 15 de 1992.
- 3) Revista mensual PC/TIPS. Año 5. No.53, julio 15 de 1992.
- 4) PC MAGAZINE En Español. Guía Independiente Para el Usuario de Computadoras Volúmen 3-
Numero 9. Septiembre 1992.
- 5) PC Computing. En Español. Año 3. No. 25. Marzo 1990.
- 6) PC Computing. Special 90-page Setion. Vol. 5. No. 2. February 1992.