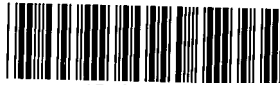


T  
519.7  
ALE



\*D-I 9578\*

ESCUELA SUPERIOR  
POLITECNICA DEL LITORAL

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Instituto de Ciencias Matemáticas.**

**“ELABORACIÓN Y DISEÑO DE UN SISTEMA CONTABLE A PARTIR DE  
UN LENGUAJE DE CUARTA GENERACIÓN.”**

**Tesis de Grado**

Previa la obtención del Título de:

**INGENIERO EN ESTADÍSTICA INFORMÁTICA.**

Presentada por:

**Jorge Alberto Alejandro Molina.**



Guayaquil – Ecuador.

AÑO

1999

## AGRADECIMIENTO.

Al Ingeniero Daniel Izquierdo, al Ingeniero Pedro Reyes, y la Srta. Bárbara Barriga, los cuales sin tener una relación directa con el desarrollo de esta tesis de grado me brindaron en todo momento su ayuda desinteresada.

## DEDICATORIA.

A Dios por darme el don del entendimiento, a mis padres por ser mis apoyo incondicional, y a mis amigos por mostrar su preocupación y paciencia en todo momento.

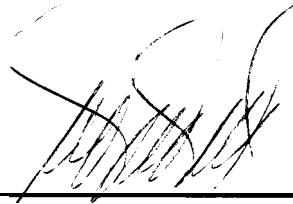
TRIBUNAL DE GRADUACIÓN.



---

**Ing. Félix Ramírez.**

DIRECTOR DEL INSTITUTO DE  
CIENCIAS MATEMÁTICAS.



---

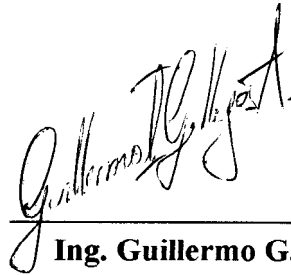
**Ing. Luis Rodríguez Ojeda**

DIRECTOR DE TESIS.

---

**Ec. Milton Triana.**

VOCAL



---

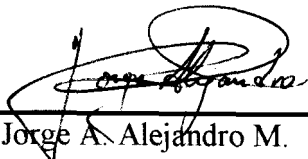
**Ing. Guillermo Gallegos.**

VOCAL.

## DECLARACIÓN EXPRESA.

“La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.”

(Reglamento de Graduación de la ESPOL.)



Jorge A. Alejandro M.

## RESUMEN.

Partiendo de un estudio de mercado de los sistemas contables, el cual habiendo sido orientado a un área específica que fue el de los colegios de la ciudad de Guayaquil, se obtuvieron por medio de una encuesta, los parámetros necesarios para delinear un sistema computacional que cubra dichos requerimientos y a su vez permita generalizar su uso a cualquier otra área en la que se necesite una contabilidad general automatizada. Dicha encuesta fue distribuida entre los colegios de la ciudad correspondientes a una muestra seleccionada aleatoriamente. Por otro lado, efectuando un estudio de las herramientas disponibles en el mercado para el desarrollo de programas, fue seleccionada una que por sus características permitía hacer efectivo el cubrir las falencias detectadas en el estudio antes mencionado y también daba la posibilidad de ampliación futura del sistema desarrollado.

La elaboración del trabajo al que se hace mención se lo dividió en 4 etapas: análisis, diseño, programación y pruebas. En cada una de ellas el objetivo primordial fue el hacer efectivo el proceso contable tratando de darle al usuario la facilidad de diseñar desde parte de la entrada de los datos (en el diseño del plan de cuentas), hasta parte de la salida (en la personalización de los reportes).

La idea de poder personalizar los reportes se basa en el hecho de que en el momento de que lo que se desee mostrar sean datos históricos o resúmenes de la información en la base de datos de las transacciones, éstas puedan ser especificadas por algún campo tal como fecha, cuenta, o periodo.

# Índice General.

Portada .....	I
Agradecimiento .....	II
Dedicatoria .....	III
Tribunal de <b>Graduación</b> .....	IV
<b>Declaración</b> Expresa .....	V
Resumen .....	VI
<b>Índice</b> General .....	VIII
<b>Índice</b> de Figuras .....	XII
<b>Índice</b> de Tablas .....	XIV
<b>Introducción</b> .....	15
1 <b>Objetivos</b> y Alcance .....	16
1.1 <b>Objetivos</b> .....	16
1.1.1 <b>Objetivo</b> General .....	16
1.1.1.1 <b>Integridad</b> Garantizada de los Datos .....	17
1.1.1.2 <b>Aprovechar</b> al <b>Máximo</b> las Inversiones Existentes en <b>Equipo</b> .....	18
1.1.1.3 <b>Incorporar</b> La Posibilidad de <b>Ampliación</b> en el <b>Futuro</b> .....	19
1.1.1.4 <b>Desarrollo</b> <b>Dinámico</b> y <b>Económico</b> Posible .....	19
1.1.1.5 <b>Diseño</b> Para un Alto Rendimiento.....	20
1.1.2 <b>Objetivos</b> Especificos .....	21
1.2 <b>Alcance</b> .....	22
1.2.1 <b>Flexibilidad</b> .....	23



1.2.2	Productividad.....	24
1.2.3	Proteccibn Contra Errores. ....	24
1.2.4	Módulos Disponibles.....	25
1.2.5	Pases Automáticos Al Mayor General. ....	26
1.2.6	Preparación Automática De Informes. ....	27
2	<b>Introducción.</b> .....	28
2.1	¿Qué Son Los Lenguajes De Programación?.....	28
2.2	Un planteamiento desde un punto de vista de la ingeniería. ....	32
2.3	Elección Del Lenguaje. ....	33
2.4	Evolución De Los Lenguajes De Programacibn. ....	34
3	<b>Estudio De Mercado De Los Sistemas Contables.</b> .....	42
3.1	Generalidades. ....	42
3.2	Objetivo General. ....	43
3.3	Objetivos Específicos.....	43
3.4	Acerca De La Encuesta y Los Resultados Obtenidos. ....	44
3.5	Anhllis Del Entorno. ....	47
3.5.1	Anhllis De La Competencia. ....	47
3.5.2	Sustitutos. ....	53
4	<b>Anhllis del Sistema.</b> .....	55
4.1	Generalidades. ....	55
4.2	El Enfoque De Los Diagramas De Flujo De Datos. ....	55
4.3	Anhllis Por Medio Del Diccionario De Datos. ....	61

5 <b>Diseño Del Sistema</b> .....	71
5.1 <b>Diseño De La Entrada.</b> .....	71
5.2 <b>Diseño De La Salida.</b> .....	78
6 Programacion y Pruebas. ....	86
6.1 De La Programacion En General.....	86
6.2 Programas y Sentencias Usadas. ....	88
7 <b>Documentación y Uso Del Sistema.</b> .....	94
7.1 Del Menu Principal.....	94
7.1.1 Menu <b>Archivo.</b> .....	95
7.1.2 El Menu Edicibn. ....	97
7.1.3 El Menu Comprobantes. ....	97
7.1.4 El Menu Mayorizacion. ....	99
7.1.5 El Menu <b>Listados</b> .....	99
7.1.6 El Menu Estadisticas. ....	100
8 Conclusiones y Recomendaciones.....	102
8.1 <b>Conclusiones.</b> .....	102
8.2 <b>Recomendaciones.</b> .....	104
<b>Apéndices.</b> . . . . .	105
Apendice A. Formulario de la Encuesta Realizada a la muestra de los colegios de la ciudad de Guayaquil.. . . . .	106

Apéndice B. Informe De Colegios .....	108
Apéndice C. Ventana Del Programa Account Pro ® .....	111
Bibliografía.....	112

## Índice de Figuras.

Figura 2.1	Generación de los lenguajes de programación. ....	36
Figura 2.2	Ejemplo de código en lenguaje ensamblador. ....	36
Figura 2.3	Muestra de código en Java. ....	38
Figura 2.4	Ejemplo de código en un lenguaje de cuarta generación. ....	39
Figura 4.1	Sub-Sistemas dependiendo del tipo de negocio. ....	56
Figura 4.2	Primer Diagrama de Contexto.....	57
Figura 4.3	Segundo Diagrama de Contexto.....	58
Figura 4.4	Diagrama de nivel 0. ....	59
Figura 4.5	Explosión del Diagrama del Procedimiento 2 .....	60
Figura 5.1	Diseño de la entrada efectiva: Definición del Plan de Cuentas. ....	73
Figura 5.2	Diseño de la entrada efectiva: Plan de Cuentas .....	74
Figura 5.3	Diseño de la entrada efectiva de datos: Definición de grupos. ....	74
Figura 5.4	Diseño de una entrada efectiva de datos: Ingreso de cuentas. ....	75
Figura 5.5	Diseño de la entrada efectiva: Definición de Comprobantes. ....	77
Figura 5.6	Diseño de una entrada efectiva: Ingreso de Transacciones .....	77
Figura 5.7	Diseño de la Salida: Personalización de los reportes.. ....	79
Figura 7.1	Menú Principal. ....	94

Figura 7.2 Menti <b>Archivo</b> .....	95
Figura 7.3 Plan de Cuentas.....	95
Figura 7.4 Cerrar <b>Período</b> .....	96
Figura 7.5 Menti <b>Edición</b> . . . . . , . . . . .	97
Figura 7.6 Menti Comprobantes.....	97
Figura 7.7 Menti <b>Mayorización</b> . . . . .	99
Figura 7.8 Menti Listados.....	99
Figura 7.9 Menti Estadísticas.. . . . .	100

## Índice de Tablas.

Tabla I.	Formato de Reporte de Cuentas .....	80
Tabla II.	Formato Todas las Cuentas. ....	80
Tabla III.	Formato de Reporte de Movimiento de Diario.....	81
Tabla IV.	Formato de Reporte por Cuenta. ....	82
Tabla V.	Formato de Balance de Comprobación. ....	83
Tabla VI.	Formato de Estado de Pérdidas y Ganancias.. ....	83
Tabla VII.	Formato de Balance General. ....	84

## INTRODUCCIÓN.

Parte de la **información** que se pudo inferir **del análisis** de los sistemas de contabilidad, al menos en el area de **los** colegios, mostro que la mayoría **funcionan** en ambientes (plataformas) no **actuales**. Aparte de ello, son sistemas que le presentan **muchas** limitaciones al usuario, el **cual**, **está** obligado a depender directamente **del** programador. Este **hecho**, que es casi un patron **común**, a pesar de ser **molesto**, ha sido aceptado y sobrellevado por las instituciones.

Fue entonces que surge el deseo de elaborar un **programa** que represente una **solución** basado en el **análisis** de requerimientos y especificaciones estudiados y mencionados en el **párrafo** anterior y dejar planteado la forma de mejorarlo.

La **motivación** final para el desarrollo **del programa** se debe a que dicho sistema debe tener una serie de relaciones (sub – sistemas, **procesos**, actividades, **funciones**, recursos, formatos, **documentos**, libros, etc.) que puedan articularse **como** medios **para** lograr que funcionen adecuadamente, de acuerdo con el **objeto** social de la entidad. En todas **existen** una serie de areas que defmen unos **procesos** a seguir; normalmente, una opera, otra administra su **situación interna** y otras asesoran a las anteriores; pero existe un area donde se concentra toda la **información**: contabilidad.

De **ahí** el que **haya llegado** a constituirse en un tema de **análisis** y desarrollo.



# CAPÍTULO I

## **Objetivos y Alcance.**

### **1.1 Objetivos.**

#### **1.1.1 Objetivo General.**

*Creación de una solución contable que funcione bajo un sistema operativo comercial, el cual se genere lo más rápido y productivo posible aprovechando una herramienta de desarrollo visual que crea automáticamente código que es compilado en un lenguaje de cuarta generación.*

Los objetivos específicos del programa a desarrollarse están en torno a las facilidades y limitaciones que este va a presentar, a fin de establecer con claridad luego el alcance que éste persigue.

Para ello se planteará brevemente las diferentes consideraciones que se deben hacer al momento de delimitar la funcionalidad de la aplicación con la herramienta a usarse para su desarrollo.



Entre estas consideraciones mencionadas está la base para el diseño, lo cual lo constituye la integridad de los datos que es una característica importante al momento de escoger un programa que maneja información.

Seguido es importante mencionar lo útil que es la integración entre los utilitarios de oficina con cualquier instrumento de análisis o procesamiento de datos, ya que de ser de otro modo, cualquier proceso que incluya la necesidad de esta información retrasará el normal avance de la tarea en determinadas ocasiones.

El vertiginoso avance tecnológico, en cuanto al manejo de la información, permite establecer altos a las barreras de la ineficiencia pero a su vez se puede convertir en el pozo sin fondo en el cual nos ahogan las innovaciones que a diario se realizan. Una aplicación debe siempre dar muestras claras que no con facilidad se va a convertir en parte del activo improductivo de una empresa, cualquiera que fuera la naturaleza de esta, haciendo que se incurran en más gastos de los que se hubieran presupuestado a su debido momento la o las partes interesadas en el nuevo paquete desarrollado.

Finalmente, otro punto relevante es la necesidad de que todos estos aspectos se encuentren juntos en un solo producto y no representen obstáculos para alcanzar el objetivo final que es una aplicación que se ajuste a las necesidades del usuario.

Para un enfoque más detallado podríamos hacer mención de cada uno de los puntos citados anteriormente:

#### **1.1.1.1 Integridad garantizada de los datos.**

El uso de una herramienta para la realización de un programa que procesa información de una base de datos se basa primero en la integridad de los datos con los que se trabaja, ya esto permite que las reglas que rigen la

coherencia de datos, específicamente las relaciones entre las claves principal y externa de tablas distintas sea resuelta mediante reglas a nivel de campo y a nivel de registro definidas por el usuario.

Para esto se hace uso de un **Generador de integridad referencial** el cual le ayuda a definir reglas para controlar cómo se insertan, actualizan o eliminan registros de tablas relacionadas estableciendo desencadenantes para controlar cómo se insertan, actualizan o eliminan registros de tablas relacionadas.

Teniendo en cuenta que los desencadenantes son código de evento a nivel de registro que se ejecuta después de una inserción, una actualización o una eliminación, es posible adjuntar distintas acciones a los distintos eventos. Los desencadenantes se ejecutan al final, después de las reglas y no se ejecutan durante las actualizaciones con almacenamiento en arreglos temporales para luego actualizar la tabla de origen. Todo esto permite tener una idea de la potencialidad de lo garantizado que puede ser la antes mencionada integridad de los datos.

#### **1.1.1.2 Aprovechar al máximo las inversiones existentes en equipo.**

Hoy en día es notorio que aún cuando en las organizaciones existen medios y formas de transmisión de información muy actualizados, éstos no son empleados de manera eficiente haciendo que la infraestructura instalada no se corresponda con la utilización de ésta.

Las herramientas de desarrollo que tienen el componente visual hacen que la aplicación entregada sea comprendida de una mejor manera desde sus primeros prototipos hasta el producto final. Mas, esto se complementa, cuando el programa a ser utilizado permite intercambiar información con otras aplicaciones de uso diario y más aún si hacen que la presentación de la

misma permita ser incorporada en un informe sin la necesidad de ser nuevamente procesada.

Esto hace que la inversión en el equipo sea aprovechada y dirigida hacia la culminación de la tarea sin tener que repetir pasos.

#### **1.1.1.3 Tncorporar la posibilidad de ampliación en el futuro.**

Uno de los principales requerimientos de los programas actuales es con cuánta posibilidad se puede ampliar su radio de acción sin que esto signifique un cambio completo de la herramienta ya elaborada.

El sistema a desarrollarse permite al usuario (empresa u organización grande o pequeña) con facilidad especificar nuevas formas de ampliación, siempre que dicha ampliación no signifique un cambio radical o afecte directamente a las especificaciones originales.

En todo caso es dable la posibilidad de que en el peor de las situaciones un cambio total permita aprovechar el sistema ya existente.

#### **1.1.1.4 Desarrollo dinámico y económico posible.**

Uno de los factores que inciden en el costo de los sistemas que se encuentran en la actualidad en el mercado es el tiempo de elaboración que éstos han llevado al hacerlos. Luego de ello el tiempo requerido para efectuar la instalación y las respectivas pruebas, no siempre han resultado ser poco significativas a tal punto que el producto instalado no funciona completamente como debería dejando insatisfechas las necesidades del usuario.

Al emplear una herramienta que en su diseño y programación se basa en la especificación de parámetros, esto permite que el usuario desde el inicio

pueda tener desde un prototipo preliminar hasta una serie de actualizaciones con nuevas versiones que simplemente representan avances claramente separables en módulos.

Si consideramos lo anterior, es fácil darse cuenta que el costo por el sistema no solo se va a reducir sino que al usuario le es otorgado un valor agregado que le permite seguir paso a paso el desarrollo del producto final.

#### **1.1.1.5 Diseiio para un alto rendimiento.**

Aunque parte del presente trabajo no es el desarrollo de una aplicación que funcione bajo red, la herramienta a ser seleccionada para su elaboración permite generar una aplicación cliente - servidor rápida y de alto rendimiento partiendo del mismo producto mono usuario ya terminado mediante el proceso de “subida de las tablas principales al servidor” (Upsizing).

Esto brinda la posibilidad al programador a centrarse únicamente en la elaboración del programa y luego preocuparse por la parte del acceso compartido.

Además de lo antes mencionado, implica aprovechar la enorme velocidad del motor de la herramienta de desarrollo. Esto se consigue con nuevas técnicas tales como el uso de acceso a datos basados en conjuntos o la generación de consultas en base a parámetros para descargar solamente los datos necesarios; la ubicación de tablas en la plataforma óptima y el aprovechamiento de procedimientos tanto hechos en la herramienta de desarrollo como los almacenados en forma remota.

Antes de hacer uso de las nuevas técnicas es necesario analizar los sistemas que piensa utilizar. Al diseñar una aplicación local o de servidor de archivos, se deben determinar las consultas, los formularios, los menús y los informes

que la aplicación va a utilizar o crear. Cuando se diseña una aplicación, sea o no cliente - servidor, se debe llevar a cabo el análisis habitual del sistema, así como un análisis adicional relacionado específicamente con las aplicaciones.

Es necesario plantearse dónde se ubicarán los datos utilizados por las consultas, los formularios, los menús y los informes, y cómo se accederá a esta información.

### **1.1.2 Objetivos Específicos.**

Pues como hasta aquí no se han formalizado los objetivos de este trabajo, a continuación se anotan de una manera clara y concreta cada uno de ellos, haciendo énfasis en que lo que se va a desarrollar es una herramienta contable, que lejos de ser un programa más de contabilidad, abarca los principios básicos y específicos de una aplicación elaborada a partir de un análisis y diseño que posteriormente se van a justificar.

- 1) Diseñar un sistema contable que permita al usuario llevar a cabo un proceso contable básico, pero completo; que sea funcional y adaptable a las necesidades desde un estudiante hasta una pequeña empresa que pueda trabajar con las limitaciones que este presente y que serán detalladas más adelante.
- 2) Analizar las principales necesidades de los usuarios en cuanto a los sistemas contables en el mercado local.
- 3) Estimar las posibilidades de la introducción de un nuevo programa de propósito general que pueda ser fácilmente adaptado a las especificaciones de quien lo emplee.

- 4) Comparar los productos sustitutos que se pueden adquirir en el mercado en base a la funcionalidad, adaptabilidad, y capacidad de ampliación, bien sea que se los adquiriera en el mercado local o a través de Internet.
- 5) Efectuar un análisis de las relaciones existentes entre las diferentes actividades en el proceso contable y los procesos que estos definen a fin de delimitar, ahora sí, las opciones del programa a ser elaborado.
- 6) Diseñar en función de un estándar que permita al usuario realizar las especificaciones para que este no se sienta esclavizado a la lógica del programador.
- 7) Elegir la herramienta más versátil para el desarrollo de la aplicación contable entre las distintas posibilidades existentes en el mercado, comparando sus características, a fin de que la elección sea justificada.
- 8) Plantear las posibilidades de desarrollo a partir de la aplicación a ser generada como incentivo a futuras generaciones desarrolladoras de programas (software) nacional.

Si bien pueden parecer muy ambiciosos los objetivos antes planteados, a lo largo del desarrollo de la presente tesis de grado se tratará en lo posible de incluir todo tipo de referencia a fin de que cualquier duda sea aclarada satisfactoriamente.

## **1.2 Alcance.**

El alcance de la tesis que se está presentando, se basará exclusivamente en los límites dentro del contexto de la contabilidad general, ya que el principal propósito es el de permitir al usuario llevar a cabo un proceso contable

básico pero completo; es decir, que se cubrirá el proceso desde el ingreso de los asientos generados por alguna transacción externa hasta la elaboración de los estados financieros principales que son el estado de resultados y el balance general.

Si bien es cierto que en el mercado encontramos toda una variedad de posibilidades para que alguien nos “realice un programa” el cual no se basa en un análisis exhaustivo previo para culminar con una serie de reuniones de entrega hasta verificar tanto la satisfacción del cliente e implantar la confiabilidad en el producto entregado, sí contamos con diferentes herramientas de desarrollo; mas estas brindan a quien las utilice diferentes ventajas y desventajas entre ellas, haciendo significativo en ciertos casos aquello que se tome como punto de partida para llegar a establecer una decisión final. Estas ventajas y desventajas se basan en los criterios que se mencionan a continuación:

### **1.2.1 Flexibilidad.**

Uno de los limitantes en los programas que se encuentran en el mercado es su falta de flexibilidad en cuanto a los requerimientos antes, durante y después de haber sido desarrollada la aplicación.

Si bien es cierto que lo que se va a desarrollar está alineado con el proceso de la contabilidad general, este presentará la flexibilidad necesaria al menos para suplir las principales falencias que encontremos en los sistemas de contabilidad existentes, no solo con el afán de cumplir con necesidades corrientes, sino que basados en un análisis justificado se dejarán sentadas las bases para el desarrollo de un programa que cubra las expectativas presentes y dé claras muestras de que fácilmente se pueden prever la falta de satisfacción de éstas.

### **1.2.2 Productividad.**

La productividad del sistema a ser elaborado será tal que pueda ser comparado con la de cualquier producto que se encuentre en el mercado actualmente. Esto es, su productividad lo podrá hacer lo suficientemente competitivo que podría ser lanzado a su mercado objetivo, el cual más adelante será claramente definido.

En un mercado tan competitivo como es el de los programas (software) a nivel internacional, el producto nacional está ya teniendo una presencia considerable dentro de lo que podría llamarse desde ahora el mercado potencial de cualquier desarrollador. Mas en este mercado son pocos los programas que existen a manera de “soluciones empresariales” que no distan considerablemente de ser una herramienta apta para un sistema de información que permita al usuario tener la información en el momento y no depender, una vez más, del programador.

Esta característica del programa constituye la base de la diferenciación del resto de los que no la poseen y permite que la productividad de cualquier actividad relacionada con la contabilidad (que son casi todas) sea beneficiada y no entorpecida por los desajustes que presenta el programa utilizado.

### **1.2.3 Protección contra errores.**

Todos los puntos que aquí se anotan son evidentemente importantes, pero la base de utilización de un producto cualquiera es la confianza que uno tenga en él; esto es, supongamos que un día en el computador hacemos la consulta de las cuentas por pagar que están a vencimiento próximo y nos encontramos con la sorpresa que no nuestras deudas relucen por mora cuando periódicamente efectuamos la misma consulta. La pregunta es,



entonces, ¿se equivocó al realizar la operación el programa o el operador?. Es más, que tal si se tiene un sistema de contabilidad que hace una transacción y no se da cuenta que la fecha del sistema ha sido cambiada y sin darse cuenta afecta a un período pasado. Eso realmente provocaría un gran caos y es precisamente lo que se desea evitar que suceda al referimos a la protección contra errores. Dicha protección abarcará desde la validación de la entrada de datos hasta los resultados obtenidos los cuales al ser personalizados por el usuario, deberían solo mostrar lo que es solicitado.

Como la programación a ser empleada va a basarse en la especificación de parámetros desde el inicio, por lo que será mucho más sencillo el realizar un control de errores desde el punto de vista de cómputo. Por otro lado, desde el punto de vista contable como antes se mencionó, el control de errores se realizará tanto en las validaciones como en lo que respecta a los asientos contables de tal manera que el resto de las operaciones vayan siendo admitidas de acuerdo a las especificaciones puntuales de la Contabilidad.

#### **1.2.4 Módulos disponibles.**

Los sistemas contables existentes brindan a los usuarios que los adquieren diversos tipos de procedimientos agrupados en módulos. Estos, solo representan una ventaja para poder dividir funciones específicas; mas el éxito, se alcanza cuando todos los elementos del sistema, es decir los módulos, se entrelazan entre sí dando como resultado una contabilidad integrada.

La idea de la división en módulos de un programa es el facilitar al programador el dividir las opciones del sistema a ser desarrollado,

permitiendo así tener una mejor relación con el cliente para que pueda efectuar un seguimiento al cumplimiento de lo pactado.

Ahora bien, tratándose del programa de contabilidad a ser desarrollado, este únicamente se basará en el módulo principal, dejando abierta la posibilidad de que fácilmente se puedan agregar los módulos restantes como: Cuentas por pagar; Cuentas cobrar; Inventario; etc.

En cuanto a los reportes (otra de las falencias de los sistemas actuales de los que se tratará más adelante), estos podrán ser diseñados por el usuario y modificados en cualquier momento de una manera fácil, aprovechándonos que la herramienta en la que se desarrollará es visual.

### **1.2.5 Pases automáticos al mayor general.**

El mayor general es el centro de un sistema integrado. En este se mantienen todas las cuentas del mayor general.

Entre las opciones que se incluyen en este módulo están las características del catálogo de cuentas, así como editarlo; registrar las operaciones; imprimir el diario; pasar al mayor las operaciones; ver y/o imprimir el balance de comprobación; etc.

El contador crea el catálogo de cuentas diciéndole al computador el nombre, el número, el saldo normal y el tipo de cada cuenta. Dicho catálogo puede modificarse en tanto no afecte a la integridad de los datos.

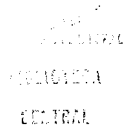
Como luego de toda operación contable, antes de cualquier tipo de reporte que se requiera, es necesario que la transacción asentada sea pasada automáticamente al mayor general.

Así se complementarían los módulos como cuentas por pagar, cuentas por cobrar, nómina, etc. , que no forman parte del programa final mas pueden adicionarse sin mayor problema.

### **1.2.6 Preparación automática de informes.**

Una ventaja de emplear una herramienta visual es la de poder reducir la complejidad de las tareas necesarias, así como la capacitación requerida para elaborar determinada acción.

Este es el caso de los informes que son empleados en la contabilidad y que son generados de acuerdo a lo especificado por el contador o la persona encargada de manejar la contabilidad mediante la utilización de la solución que se va a desarrollar.



## CAPÍTULO 2.

# Introducción.

### 2.1 ¿Qué son los lenguajes de programación?

Los lenguajes de programación son un vehículo de comunicación entre los humanos y las computadoras. El proceso de codificación - comunicación mediante un lenguaje de programación - es una actividad humana. Es por ello que las características psicológicas del lenguaje afectan directamente a la calidad de la comunicación.

Varias características aparecen como resultado del diseño de un lenguaje de programación. Aunque estas características no se pueden medir de forma cuantitativa, se reconoce su manifestación en todos los lenguajes de programación. Dichas características se anotan a continuación:

- **La uniformidad** indica el grado en que un lenguaje usa una notación consistente, aplica restricciones aparentemente arbitrarias o incluye excepciones a reglas sintácticas o semánticas.

□ **La ambigüedad** de un lenguaje de programación es percibida por el programador. Un compilador siempre interpreta una sentencia de una única forma, pero el lector humano puede interpretar la sentencia de formas diferentes.

La falta de uniformidad y la presencia de ambigüedad psicológica, normalmente, van juntas. Si un lenguaje de programación muestra aspectos negativos de estas características, el código fuente será menos legible y la traducción desde el diseño más problemática.

**Lo compacto** que sea un lenguaje de programación es un indicativo de la cantidad de información orientada al código que se debe retener en memoria humana. Entre los atributos del lenguaje que miden lo compacto que es, se encuentran:

- Soporte a las construcciones estructuradas.
- Tipos de palabras claves que se puedan utilizar.
- Variedades de tipos de datos.
- Número de operadores aritméticos y lógicos.
- Número de funciones incorporadas.

Aunque esta característica aparenta hacer robusto a un lenguaje de programación, esto no es tan cierto al momento de tomarnos un tiempo a comparar ciertos lenguajes; por ejemplo, el APL es un lenguaje de programación excepcionalmente compacto. Sus concisos y potentes operadores permiten implantar complejos procedimientos aritméticos lógicos con relativamente poco código. Desgraciadamente, al ser tan compacto, APL también se convierte en un lenguaje difícil de leer y de comprender, pudiendo llegar a una pobre uniformidad.

Las características de memoria humana tienen un fuerte impacto en la forma en que se usa un lenguaje. La memoria y el reconocimiento humano se pueden dividir en campos sinestético y secuencial. La memoria sinestética nos permite recordar y reconocer las cosas como un todo. Por ejemplo reconocemos una cara humana instantáneamente; no evaluamos concisamente cada una de sus partes antes de reconocerla. La memoria secuencial, en cambio, proporciona una forma de reconocer el siguiente elemento de un secuencia (como por ejemplo la siguiente línea de una canción, dadas las líneas precedentes). Cada una de estas características de memoria afectan a las características de los lenguajes de programación denominadas localización y linealidad, que se tratan a continuación.

La localización es una característica sinestética de un lenguaje de programación. La localización se potencia cuando las sentencias se pueden combinar en bloques, cuando las construcciones estructuradas se pueden implantar directamente y cuando el diseño y el código resultante son altamente modulares<sup>1</sup> y cohesivos<sup>2</sup>. Una característica que viola la localización es aquella que aporta o induce al manejo de excepciones por ejemplo el procesamiento de la condición ON en PL/I o ERR= en versiones extendidas de FORTRAN.

La linealidad es una característica psicológica que se asocia con el concepto de mantenimiento de un ámbito funcional. Osea la percepción humana se facilita cuando se encuentra una secuencia lineal de operadores lógicos. Las grandes ramificaciones ( y de alguna forma, por qué no, los grandes bucles)

---

<sup>1</sup> Programa dividido en componentes con nombres y ubicaciones determinados, que se denominan módulos y que se integran para satisfacer los requisitos del problema.

<sup>2</sup> Un módulo cohesivo es aquel que ejecuta una tarea sencilla de un procedimiento de programa y requiere poca interacción con procedimientos que ejecutan otras partes de un programa.

violan la linealidad del procesamiento. De nuevo la implantación directa de las construcciones estructuradas ayudan a la linealidad de un lenguaje de programación.

Por otro lado, la capacidad que presentan las personas para aprender un lenguaje de programación se ve afectada por lo que se conoce como la tradición. Para ilustrar esto podríamos afirmar que un programador con experiencia en FORTRAN o en ALGOL no tendrá mucha dificultad para aprender PL/I, PASCAL o C. Estos últimos lenguajes siguen una tradición establecida por los primeros. Las construcciones son similares, la forma es compatible y se mantiene en cierta forma el “formato” del lenguaje de programación. Sin embargo si se propone al mismo individuo que aprenda APL, LISP, o SMALLTALK, se rompería la tradición y la curva del tiempo de aprendizaje sería más pronunciada.

La tradición también afecta al grado de innovación durante el diseño de un lenguaje de programación. Aunque frecuentemente se proponen nuevos lenguajes, las formas de esos nuevos evolucionan muy despacio. Por ejemplo PASCAL es un pariente cercano de ALGOL. Sin embargo una importante innovación del lenguaje PASCAL es la implantación del tipo de datos definidos por el usuario, una forma que no existía en anteriores lenguajes relacionados por tradición con el lenguaje PASCAL. ADA, un lenguaje que también ha surgido de la tradición ALGOL-PASCAL, se extiende más allá de esos lenguajes con una gran variedad de estructuras y tipos innovadores.

Las características de los lenguajes de programación afectan de forma importante a nuestra capacidad para aprenderlos, aplicarlos y mantenerlos. El lenguaje de programación da color a nuestra forma de pensar sobre programas y, de forma inherente, limita la forma en que nos comunicamos

con la computadora. El grado en que esto sea bueno o malo aún está por determinar.

## 2.2 Un planteamiento desde el punto de vista de la ingeniería.

En función de las necesidades que pueda tener un proyecto de desarrollo de programación, entre los lenguajes de programación se pueden establecer ciertas características:

- Facilidad de traducción del diseño al código, proporciona una indicación de cómo se aproxima un lenguaje de programación a la representación del diseño. Un lenguaje que implante directamente las construcciones estructuradas, que incluya estructuras de datos sofisticadas, una entrada – salida de datos (E/S) especializada posibilidades de manipulación de bits y construcciones orientadas a objetos, hará que la traducción del diseño al código fuente sea mucho más fácil, siempre y cuando se especifiquen esos atributos en el diseño.
- Lo portátil del código fuente es una característica de los lenguajes de programación que se pueden interpretar de tres formas:
  - 1) El código fuente puede ser trasladado de un procesador a otro y de un compilador a otro sin ninguna o muy pocas modificaciones.
  - 2) El código fuente permanece inalterado cuando cambia su entorno de funcionamiento, por ejemplo cuando se cambia a una nueva versión de un sistema operativo.
  - 3) El código fuente puede ser integrado en diferentes paquetes de programas sin que prácticamente se requieran las modificaciones debidas a las características propias del lenguaje de programación.



- La disponibilidad de herramientas de desarrollo, la cual puede acortar el período de tiempo requerido para la generación del código fuente y puede mejorar la calidad del código en general. Muchos lenguajes de programación pueden ser adquiridos con un conjunto de herramientas que incluyen: compiladores con depuradores<sup>3</sup>, ayudas de formato para el código fuente, facilidades de edición incorporadas, extensas bibliotecas de sub – programas par-a una gran variedad de áreas de aplicación, correctores, compiladores cruzados para desarrollo de microprocesadores, capacidad de macroprocesador, herramientas de ingeniería inversa y otras.
- La facilidad de mantenimiento del código fuente, la cual es críticamente importante par-a cualquier esfuerzo no trivial de desarrollo de un proyecto de programación. El mantenimiento no se puede llevar a cabo hasta que no se entiende al programa. Anteriores elementos de configuración del programa proporcionan un fundamento para la facilidad de comprensión, ya que el código fuente final debe ser leído y modificado de acuerdo a los cambios en el diseño. La facilidad de traducción del diseño al código es un elemento importante en la facilidad de mantenimiento del código fuente. Además las propias características de documentación de un lenguaje tienen una fuerte influencia sobre el mantenimiento.

## 2.3 Elección de un lenguaje.

La elección de un lenguaje de programación para llevar a cabo el desarrollo de un proyecto de esta índole, merece una especial atención dado que hay que tomar en cuenta tanto el aspecto de la ingeniería como el psicológico. Pero por muy buena intención que se tenga de obtener la configuración perfecta que acople tanto el recurso humano (desarrollador) como el físico

---

<sup>3</sup> Parte del conjunto del compilador y el entorno de desarrollo que permite al programador hacer un seguimiento al código fuente en tiempo de ejecución del programa.

(computador e infraestructura de comunicacion), el solo hecho de que se cuente con un solo lenguaje disponible o si el cliente elige uno en particular hace que tanto la ingenieria, al menos en la elección, como la psicología, queden relegadas a un segundo plano.

Meek, propone una filosofía general para cuando se tenga que elegir algún lenguaje de programación:

*“... el arte de elegir un lenguaje comienza con el propio problema planteado, al decidir cuáles son sus requisitos y su importancia relativa, ya que probablemente será imposible satisfacerlos todos por igual (con un único lenguaje) . . . se deben medir los lenguajes disponibles teniendo al lado una lista de requisitos...”*

Existen diversos criterios par-a la evaluación del lenguaje para que sea el apropiado, entre los cuales podemos anotar los siguientes:

- Área de aplicación general
- Complejidad algoritmica y computacional.
- Entorno en el que se ejecutara el programa.
- Consideraciones de rendimiento.
- Complejidad de las estructuras de datos.
- Conocimiento de la plantilla de desarrollo de software.
- Disponibilidad de un buen compilador.

Existen distintos lenguajes que dependiendo de su area de aplicacion son elegidos. Y es precisamente ese el criterio más predominante para el momento de la evaluación.

Hoy en día se tienen en el mercado un sin número de lenguajes de programación; unos siguen la línea de la tradición, mientras que otros tratan de avanzar tan rápido como les permite la tecnología circundante. Por esta razón podemos encontrar que existen organizaciones en las que se prefiere siempre mantener el mismo lenguaje de programación que todo el grupo de desarrolladores conoce y da la casualidad que fue aquel que aprendieron en su época, como una clara muestra de la resistencia al cambio.

Sin embargo en la actualidad, lenguajes de programación orientados a que el usuario se tenga que preocupar más por resolver el problema en sí, como los llamados de cuarta generación y los de quinta generación (lenguajes visuales), están entrando muy fuertemente preocupando a quienes no se pueden deshacer del factor psicológico que les impide inclinarse por ellos.

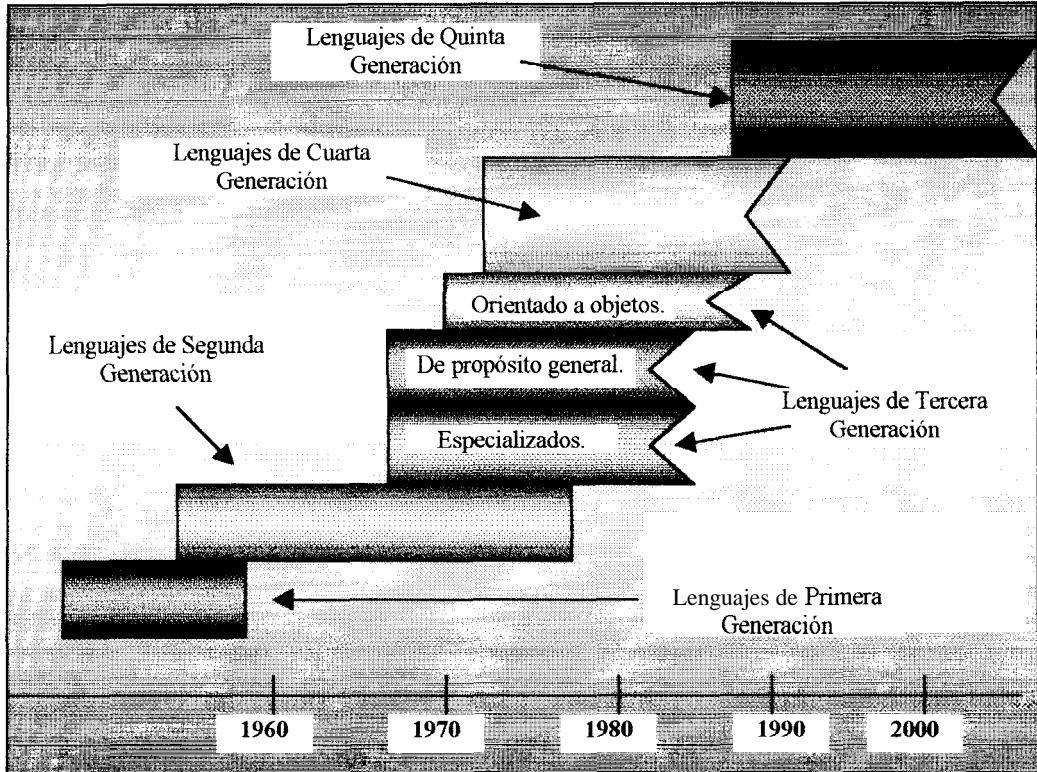
## **2.4 Evolución de los lenguajes de programación.**

Existen cientos de lenguajes de programación que han sido aplicados, en uno u otro momento, en algún esfuerzo serio de desarrollo de software. Cualquier clasificación de los lenguajes de programación será siempre un tema de discusión. En muchos casos un lenguaje podrá legítimamente aparecer en más de una categoría. Para el propósito de esta tesis se planteará la forma histórica en la que han ido evolucionando y apareciendo los lenguajes a fin de encaminarnos hacia el planteamiento del que vaya a ser el elegido para el desarrollo de la aplicación contable.

Al hablar de la evolución de los lenguajes de programación es necesario considerar las abreviaturas que se emplean para representarlos.

En la industria de la computación, las abreviaturas que son ampliamente utilizadas para representar los mayores pasos en la evolución de los

lenguajes de programación, como se representa en la figura 2.1, son los siguientes:



*Fig. 2.1 Generación de los lenguajes de programación.*

**1GL** o lenguaje de primera generación, fue (y ah es) el lenguaje de máquina o el nivel de instrucciones y datos con los que el procesador directamente trabaja, que en una computadora convencional, son cadenas de ceros y unos.

**2GL** o lenguaje de segunda generación es el lenguaje ensamblador. Una típica instrucción de un 2GL se observa como sigue:

```

ADD 12,8

```

**Figura 2.2** Ejemplo de código en lenguaje ensamblador

El ensamblador convierte estas instrucciones en lenguaje de máquina.

Hay autores que consideran al lenguaje ensamblador como uno de **primera generación**<sup>4</sup>, ya que **presenta un nivel de abstracción** muy bajo y asigna la jerarquía de segunda **generación** a los lenguajes desarrollados a finales de los años 50 y principios de los 60 y han servido como base para todos los lenguajes de programación modernos. A ellos los caracteriza el amplio uso de bibliotecas de programas y la gran familiaridad y aceptación. Nadie pone en duda que FORTRAN, COBOL, ALGOL y de alguna forma BASIC son lenguajes base.

De todos modos ya fue aclarado previamente al inicio de la **sección** que cualquier **clasificación** es ampliamente discutible, mas son una buena base **para** enmarcar el desarrollo y la **evolución** de estos lenguajes de programación.

3GL o lenguaje de tercera **generación** es un lenguaje de programación de alto nivel tal como PL/I, C, o Java. También denominados lenguajes de programación **moderna** o estructurada , **están** caracterizados por sus potentes posibilidades mediante el uso de procedimientos y de estructuración de datos. Los lenguajes de esta **clase** se pueden dividir en tres amplias categorías:

- Lenguajes de alto nivel de propósito general, como PASCAL, C, ALGOL.
- Lenguajes de alto nivel orientados a objetos, tales como C++, SmallTalk, Eiffel
- Lenguajes especializados, como por ejemplo LISP, PROLOG, FORTH.

---

<sup>4</sup> Tal es el **caso** de Roger Pressman, en su libro Ingeniería del Software, un enfoque **práctico**. Tercera edición.

Los lenguajes de propósito general y orientados a objetos tienen características de programación estructurada, herencia, sub – rutinas, etc., mientras que los lenguajes especializados son caracterizados por su inusual formulación sintáctica que ha sido especialmente diseñada para una aplicación particular.

Una instrucción de Java luce como se muestra en la figura siguiente:

```
public boolean handleEvent (Event evt) {  
    switch (evt.id) {  
        case Event.ACTION_EVENT: {  
            if ("Try me".equals(evt.arg)) {
```

Figura 2.3 Muestra de código en Java.

Un compilador convierte las instrucciones de un específico lenguaje de alto nivel en lenguaje de máquina. En el caso de Java el resultado es llamado bytecode el cual es convertido en un apropiado lenguaje de máquina por una máquina virtual Java que corre como una parte de una plataforma de un sistema operativo.

Un lenguaje de tercera generación requiere una considerable cantidad de conocimiento de programación.

4GL o lenguaje de cuarta generación ha sido diseñado para estar más cerca del lenguaje natural que el lenguaje de tercera generación. Lenguajes para tener acceso a bases de datos son frecuentemente descritos como 4GLs.

Una instrucción propia de un lenguaje de cuarta generación se ve así:

***EXTRACT ALL CUSTOMERS WHERE "PREVIOUS PURCHASES" TOTAL MORE THAN \$1000***

**Figura 2.4 Ejemplo de código en un lenguaje de cuarta generación.**

A lo largo de la historia del desarrollo de los programas, siempre se ha intentado generarlos a un nivel mayor de abstracción. Los lenguajes de primera generación trabajaban a nivel de instrucciones de máquina, el cual era y es el menor nivel de abstracción posible. Los lenguajes de segunda y tercera generación han subido el nivel de representación de los programas de computadora., pero aún hay que especificar distintos procedimientos algorítmicos perfectamente detallados. Durante la década de los 80, los lenguajes de cuarta generación han elevado más el nivel de abstracción.

Los lenguajes de cuarta generación, al igual que los lenguajes de inteligencia artificial, contienen una sintaxis distinta para la representación del control y para la representación de las estructuras de datos. Sin embargo un 4GL representa estas estructuras en un mayor nivel de abstracción, eliminando la necesidad de especificar los detalles algorítmicos.

Los 4GL son lenguajes de dominio específico, es decir, que lo que uno de ellos podría ejecutar con cierta instrucción, otro de ellos simplemente genere un error de compilación porque no ha sido diseñado para interpretar el código de la misma manera que el anterior.

Los lenguajes de cuarta generación combinan características de procedimientos y de los no orientados a los procedimientos. Es decir, el lenguaje permite al usuario especificar condiciones con sus correspondientes acciones (componente orientado a los procedimientos), mientras que al

mismo tiempo, se pide al usuario que indique el resultado deseado (**componente** no orientado a los procedimientos), encontrando los detalles orientados a los procedimientos mediante la **aplicación** de su conocimiento del dominio específico.

5GL o lenguaje de quinta generación es programación que utiliza una visual o gráfica **interfaz**<sup>5</sup> desarrollada para crear código fuente que es usualmente compilado con un compilador de lenguaje de tercera o cuarta **generación**.

Microsoft®, Borland®, IBM®, y otras compañías hacen programas que utilizan lenguajes de quinta generación para desarrollar aplicaciones en Java, por ejemplo: la programación visual a uno le **permite** con facilidad tener una visión de las clases de objetos, herencias, y arrastrar **iconos**<sup>6</sup> para ensamblar los componentes del programa.

Microbrew AppWare® e IBM's VisualAge® for Java® son ejemplos de lenguajes de quinta generación.

La **codificación** se traduce al **diseño** detallado a un lenguaje de programación que, por último es transformado en instrucciones ejecutables por la máquina. Las características psicológicas y técnicas de un lenguaje de programación afectan a la facilidad de la **traducción del diseño** y al **esfuerzo** requerido para la prueba y el mantenimiento.

El estilo es un atributo importante **del** código fuente que puede determinar la inteligibilidad de un programa. Los elementos de estilo incluyen la **documentación interna**, los **métodos de declaración de datos**, los procedimientos de **construcción** de sentencias y las técnicas de **codificación**

---

<sup>5</sup> Entorno de desarrollo o desarrollado para que se haga efectiva la **interacción** entre el programa y el usuario.

<sup>6</sup> Objeto gráfico que hace referencia a otro objeto, programa o procedimiento.



de la entrada y salida (E/S) de datos. En todos los casos, la claridad y la sencillez son características clave.

Aunque la eficiencia puede ser un requisito extremadamente importante, se debe recordar siempre que un programa eficiente que sea ininteligible tiene un valor muy cuestionable.

## **CAPÍTULO 3.**

# **Estudio De Mercado De Los Sistemas Contables.**

### **3.1 Generalidades.**

Para poder tener una idea de lo que debería incorporarse al programa a ser desarrollado para que satisfaga las necesidades de los usuarios fue necesario realizar un estudio de mercado el cual no solo buscaba hallar esa información, sino que también trataba de identificar los principales y más usados sistemas de contabilidad, los más disponibles al público; así como las posibles herramientas de desarrollo.

### **3.2 Objetivo General.**

## 3.2 Objetivo General.

Investigar sobre la aceptación de un nuevo sistema de contabilidad a partir de una evaluación directa de la competencia, así como la a través de la opinión y crítica de los actuales usuarios a sus correspondientes sistemas contables.

## 3.3 Objetivos Específicos.

El estudio realizado tuvo como fin determinar:

- Los parámetros para delinear un nuevo sistema de contabilidad en términos de la forma de pago usual
- Tiempo que tiene el sistema en uso
- Ambiente de funcionamiento
- Proporción de aceptación de un sistema con características nuevas.
- Opciones que consideran necesita el sistema en uso.

Adicionalmente se evaluó a la competencia con sus productos en el mercado local y se hizo una inspección de las posibilidades de adquirir un sistema contable a través de un medio común y de bajo costo como lo es el Internet.

### 3.4 Acerca de la encuesta y los resultados obtenidos.

Para realizar el estudio se escogió como mercado objetivo al de los colegios particular-es de la ciudad de Guayaquil. La razón por la que el mercado se segmentó de los fiscales, fue porque ellos llevan una contabilidad estatal y las preguntas no podían ser las mismas para ambos grupos, y por ende tampoco se las podía contrastar entre ellas.

Una vez seleccionado el grupo objetivo, se elaboro el marco muestral a partir de la lista de colegios que se muestra en el APÉNDICE B, en el cual constan todos aquellos colegios que potencialmente entrarían en el estudio a realizarse.

De entre esos colegios basados en una de las preguntas del formulario a ser utilizado en la encuesta (concretamente la pregunta 6) se determino el tamaño de la muestra a ser tomada dando como resultado 21 colegios.

De la información obtenida de aquellos que respondieron al formulario el análisis que se pudo hacer fue el siguiente:

- Todos manejan la contabilidad mediante un sistema de cómputo.
- No existe una forma predominante de pago, dándose el caso que es suministrado por un organismo del cual depende la institución a fm de

regula el manejo de las cuentas, como es el caso de las instituciones militares.

- Ninguno de los sistemas contables es actualizado, teniendo por lo menos 1 año de uso.
- Funcionan la mayoría en red, sobre plataformas poco funcionales o sobre sistemas operativos obsoletos.
- El 75% de las instituciones prefiere la adaptación de un sistema contable de acuerdo a sus requerimientos, partiendo de uno general.
- En todos los sistemas contables existentes, existe por lo menos un requerimiento insatisfecho. Entre los más necesitados están el análisis comparativo de periodos y el informe de índices. Seguido hay una clara muestra de que falta flexibilidad para la elaboración de consultas e informes en general.
- Todos los sistemas en uso tuvieron un valor de más de mil dólares.
- En promedio seis personas están a cargo o hacen uso del sistema de contabilidad en la institución, siendo la variación de este número bien amplia, en un rango de 3 a 10 personas.
- Así mismo el número promedio de niveles que manejan en su plan de cuentas es de 7 variando en un rango de 5 a 9 niveles.

Si bien es cierto que por la carencia de un numero considerable de respuestas desde el punto de vista de la estadística es imposible establecer intervalos de confianza que permitan tener una mejor estimación, los resultados obtenidos son contundentes y demuestran claramente que existe un mercado que no está satisfecho con el programa que maneja su contabilidad.

Cabe recalcar que esta falencia evidenciada también muestra que a pesar de ser el departamento de contabilidad el eje central de las decisiones tanto administrativas como financieras, es, por lo menos, uno de los que no se encuentra en la mira de actualizar a un ambiente de funcionamiento más seguro. Esta aseveración es basada en el hecho de que la contabilidad se maneja sobre plataformas que no facilitan la adaptación a un cambio tecnológico.

La parte interesante que será tema de análisis posterior será la correspondiente a los informes que se requieren en determinado momento durante el ejercicio contable y que los actuales programas en uso presentan una clara inflexibilidad a consultar única y exclusivamente lo que se estableció al momento de la programación.

## 3.5 Análisis del entorno.

### 3.5.1 Análisis de la competencia.

Los principales proveedores actuales de sistemas de contabilidad podría decirse son todas aquellas empresas dedicadas a la elaboración de programas de contabilidad los cuales se encuentran en el mercado, como es el caso de:

- TECHNOTEL Inc., con **MÓNICA**, version 5.00;
- POLIGRÁFICA, con su producto **COMPUVENUS**;
- SYNFO – Q , con su programa **EVA**.

De ellos nos referiremos al COMPUVENUS por ser el que abarca las características de los demás y porque es del que menos las personas que lo han utilizado han manifestado algún tipo de insatisfacción.

Este producto está desarrollado en FoxPro® para D.O.S. Sus especificaciones son bien detalladas por el departamento de mercadeo y sus datos técnicos muestra su funcionamiento y adaptabilidad a diferentes plataformas de trabajo como lo son D.O.S., Windows95®, Red Novell®, Red Windows95, Red Windows NT®.

Para cuando se estaba realizando el presente trabajo era uno de los productos en mayor boga dentro del mercado local, razón por lo cual fue

elegido para compararlo con las necesidades detectadas y mencionadas anteriormente, así como a partir de ello delinear un programa acorde con el alcance de esta tesis.

Ofrece a los usuarios un sistema integrado (modular) compuesto por:

- Bancos y cuentas por pagar.
- Cuentas por cobrar.
- Contabilidad general.
- Facturación y Estadística.
- Inventario y órdenes de compra.

Dentro de cada módulo permite efectuar transacciones y obtener diferentes tipos de reporte, tales como:

1) Módulo de Bancos y Cuentas por Pagar.

Permite realizar las siguientes transacciones:

- Depósitos.
- Notas de débito / crédito bancarias.
- Emisión e impresión de cheques.

Los reportes que se obtienen de este módulo son:



- Saldos bancarios al dia.
- Movimientos de un banco.
- Resumen de retenciones.
- Listados de Cheques por pagar.
- Análisis de saldos (facturas vencidas y por vencer)
- Comprobantes de egreso, ingreso y notas de débito / credito.
- Facturas de un proveedor.

## 2) Módulo de Cuentas por Cobrar.

Entre las transacciones tenemos:

- Abonos y cancelacion de una factura.
- Notas de débito / credito.
- Depositar cheques.
- Ingresar cheques protestados.

Los reportes que se obtienen son:

- Diario de cuentas por cobrar.
- Análisis de saldos (facturas por vencer y vencidas por cobrar)

- Documentos pendientes de clientes.
- Estado de cuentas de clientes.
- Listados de cheques fechados posteriormente (post – fechados )
- Listado de cheques protestados.

### 3) Módulo de Contabilidad General.

Entre las transacciones que realizan está:

- Registrar comprobantes.

Los reportes son:

- Plan de cuentas.
- Diario de movimientos.
- Comprobantes de diarios.
- Estados de cuentas.
- Balance de comprobación.
- Mayor general.
- Estados financieros.

### 4) Módulo de Facturación y Estadísticas.

Las transacciones que permite hacer son:

- Pedidos.
- Facturas.
- Notas de ventas.
- Devoluciones.
- Notas de entrega.

Los reportes:

- Cuadre de caja.
- Diario de ventas.
- Ventas no canceladas.
- Lista de precios.
- Ventas de un producto.
- Movimientos de ventas.

#### 5) Módulo de Inventario y Órdenes de compra.

Sus transacciones son como se anota:

- Ingreso a bodega.



- Salida de bodega.
- Transferencias.
- Órdenes de compra.

Y sus reportes:

- Catálogo de productos.
- Diario de inventario.
- Existencias en bodegas.
- Kardex de mercadería.
- Balance de mercadería.
- Hoja para la toma de inventario.

Finalmente, en el **aspecto** monetario es de resaltar que el valor de adquisición es de S/.800.000,00 + I.V.A. ; precio el cual lo hace relativamente barato en el mercado, sobre **todo** por lo muy funcional que es, ya que, no solo trata de abarcar todas las necesidades de **información** dentro del proceso contable, sino que **también** ofrece la posibilidad de trabajar con varias empresas al mismo tiempo, y lo limita **únicamente** el espacio del disco duro.

### 3.5.2 Sustitutos

Considerado aparte está el estudio de lo que en el mercado podría decirse constituiría un sustituto ya que brinda las mismas posibilidades que un sistema contable comercial, pero presenta ciertas restricciones. Concretamente nos referiremos a los productos tales como hojas electrónicas que permiten llevar una contabilidad casi manual, aunque se puede automatizar ciertos procesos.

Otra posibilidad son los programas de evaluación<sup>7</sup> (Shareware) que se encuentran disponibles por Internet, tal como el Account Pro ® que brinda al usuario que lo descargue de la dirección <http://www.accsoft-ch.com/downacc.htm> la posibilidad de tener un programa que efectúe todo el proceso contable con la salvedad de que no se puede modificar el encabezado de los informes.

Por otro lado parte de los programas que vienen de obsequio por la compra de un computador COMPAQ® o IBM® está el Quicken® de Intuit. Este programa le muestra al usuario toda una gama de posibilidades para, al menos en teoría, conectarse con el banco de la localidad para efectuar directamente, en tiempo real, las consultas bancarias. El inconveniente es

---

<sup>7</sup> Programas o versiones incompletas de estos para que las personas de forma gratuita las descarguen y las evalúen.

que esa característica solo la tienen disponible ciertos bancos y no es de fácil adaptabilidad.

En el apéndice C se muestra un ejemplo de una ventana que utiliza el programa Account Pro® para el ingreso de transacciones. Así mismo el apéndice D ilustra una la presentación del programa Peachtree®, que es otro de los disponibles por Internet, pero la diferencia es que este no posee una versión de evaluación disponible para ser descargada. Se caracteriza este último porque es un programa que se integra con los de la familia Office. Así se puede relacionar con las hojas de cálculo de Excel®, los documentos de Word®, como también con Microsoft Exchange® y Microsoft Outlook®.

En cuanto a precios, el Quicken, versión completa, tiene un costo de 150 U.S. \$ más el I.V.A.

## CAPÍTULO 4

# **Análisis Del Sistema.**

### **4.1 Generalidades.**

El sistema debe tener una serie de relaciones (sub – sistemas, procesos, actividades, funciones, recursos, formatos, documentos, libros, etc.) que puedan articularse como medios para lograr que funcionen adecuadamente, de acuerdo con el objeto social de la entidad. En todas existen una serie de áreas que definen unos procesos a seguir; normalmente, una opera, otra administra su situación interna y otras asesoran a las anteriores; pero existe un área donde se concentra toda la información: **contabilidad.**

### **4.2 El enfoque de los diagramas de flujo de datos.**

Entre los sub – sistemas , dependiendo del tipo de negocio, están los que se muestran en la figura 4.3.

Dentro de un sistema económico administrativo cada empresa constituye los sub – sistemas que, por su naturaleza, le sean indispensables para poder cumplir su misión, sus objetivos, y metas, con el fin de posicionarse en el medio, con las estrategias de estudios de tiempos y movimientos, del recurso humano y de los recursos materiales que permitan lograr la efectividad y economía deseada, por ello el gráfico anterior tan solo ilustra un caso de los tantos en que puede representarse la relevancia del departamento contable.

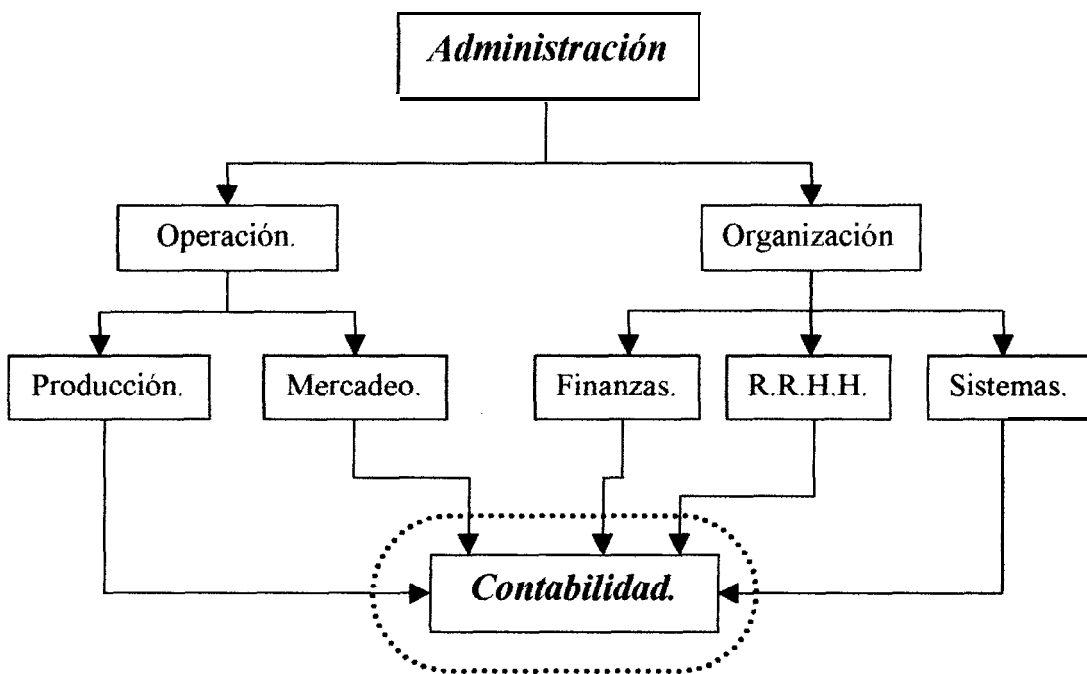


Figura 4.1 Sub-Sistemas dependiendo del tipo de negocio.



Las actividades comerciales originaron intercambios, generando excedentes en productos que cultivaban determinados sectores de producción, que a la vez necesitaban otros; cada vez se hacían más complejos estos cambios, y apareció un patrón de equivalencia general que dio lugar a los modelos, formatos y procedimientos.

Posteriormente se presentó la necesidad de ir organizando y relacionando toda la información en las diferentes áreas, para llegar a centralizarla en una de ellas: el área contable.

Por ser el presente proyecto un sistema cuyo propósito es, dentro de las limitaciones mencionadas en el primer capítulo, procesar una contabilidad general, dicha contabilidad se puede representar mediante el siguiente diagrama de contexto:

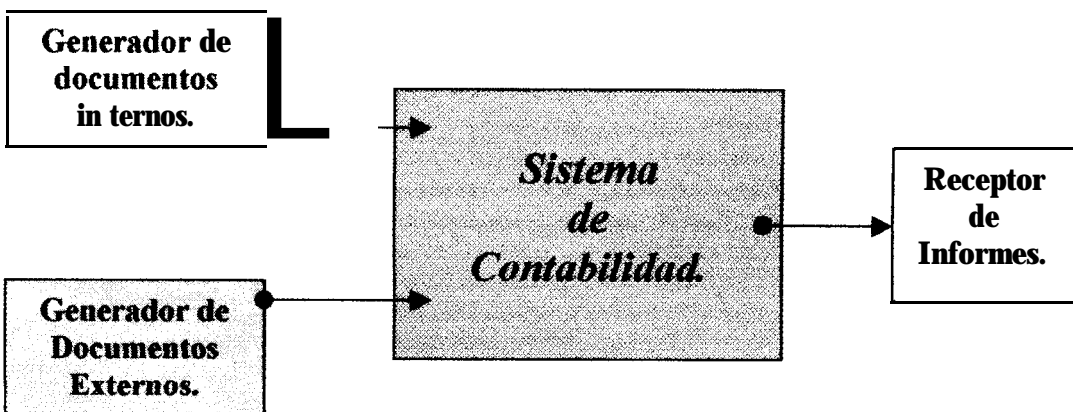


Figura 4.2 Primer Diagrama de Contexto.

Pues bien, en este primer diagrama encontramos que lo único que se está detallando es el proceso contable correspondiente a la entrada y salida de información, pero no se está considerando un componente importante que es el manejo de las cuentas por motivos de cambios en nombre de la cuenta, la anulación de una de estas, etc., lo cual haría necesario considerar el lugar de origen de ese tipo de transacciones.

Por ello es menester considerar qué ente externo al Sistema de Contabilidad es el generador.

Si lo analizamos un poco más detenidamente, el origen de este tipo de transacciones, que forman parte indiscutiblemente del proceso, se originan en el departamento Administrativo – Financiero, el cual es el que autoriza para que se lleve a cabo una manipulación directa de las cuentas.

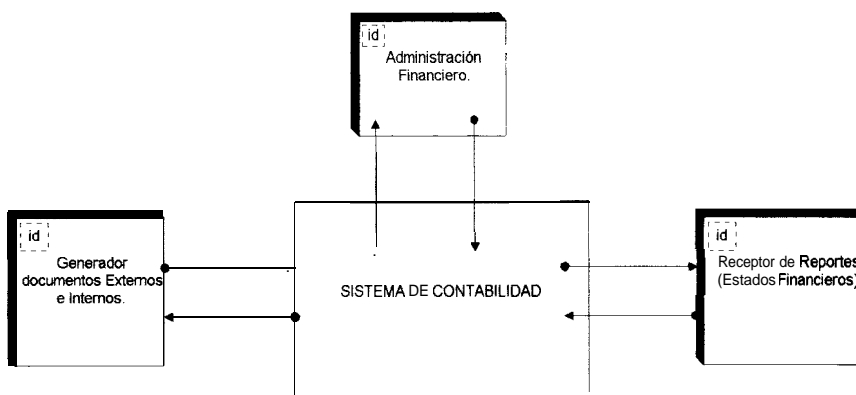


Figura 4.3 Segundo Diagrama de Contexto.

Seguido se detallan los diagramas correspondientes a partir del de Contexto:

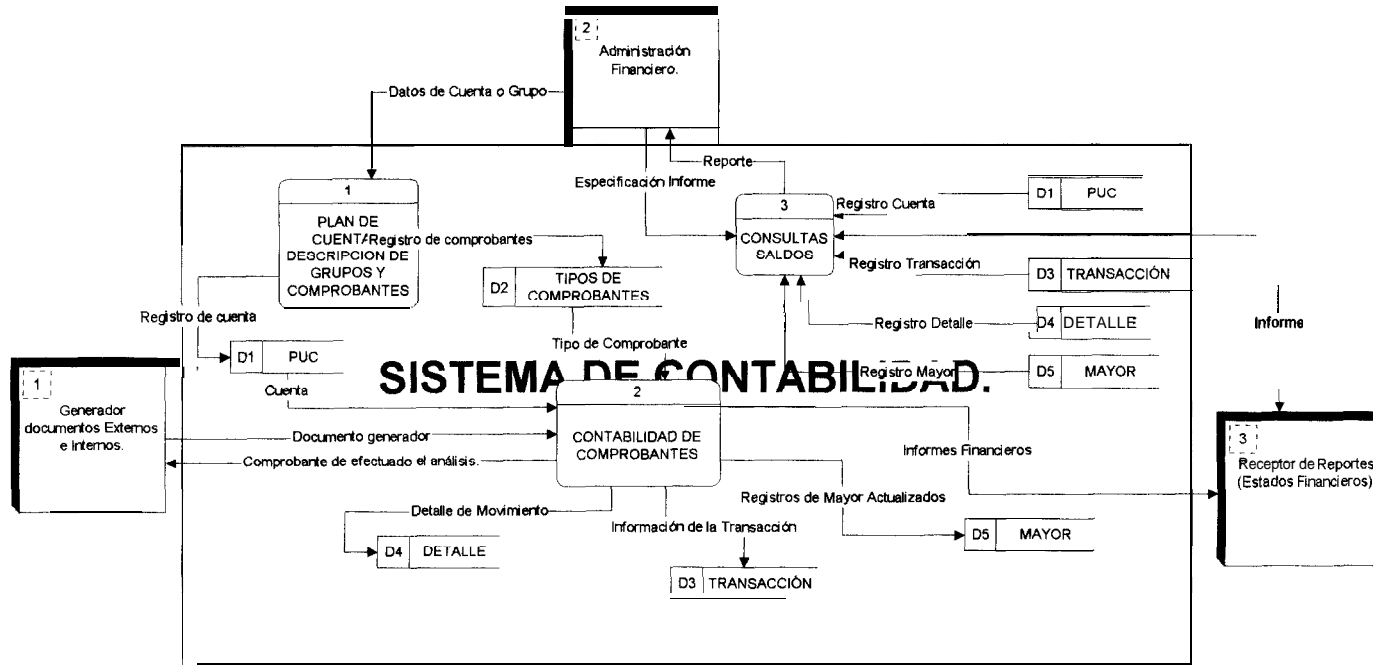


Figura 4.4 Diagrama de nivel 0.

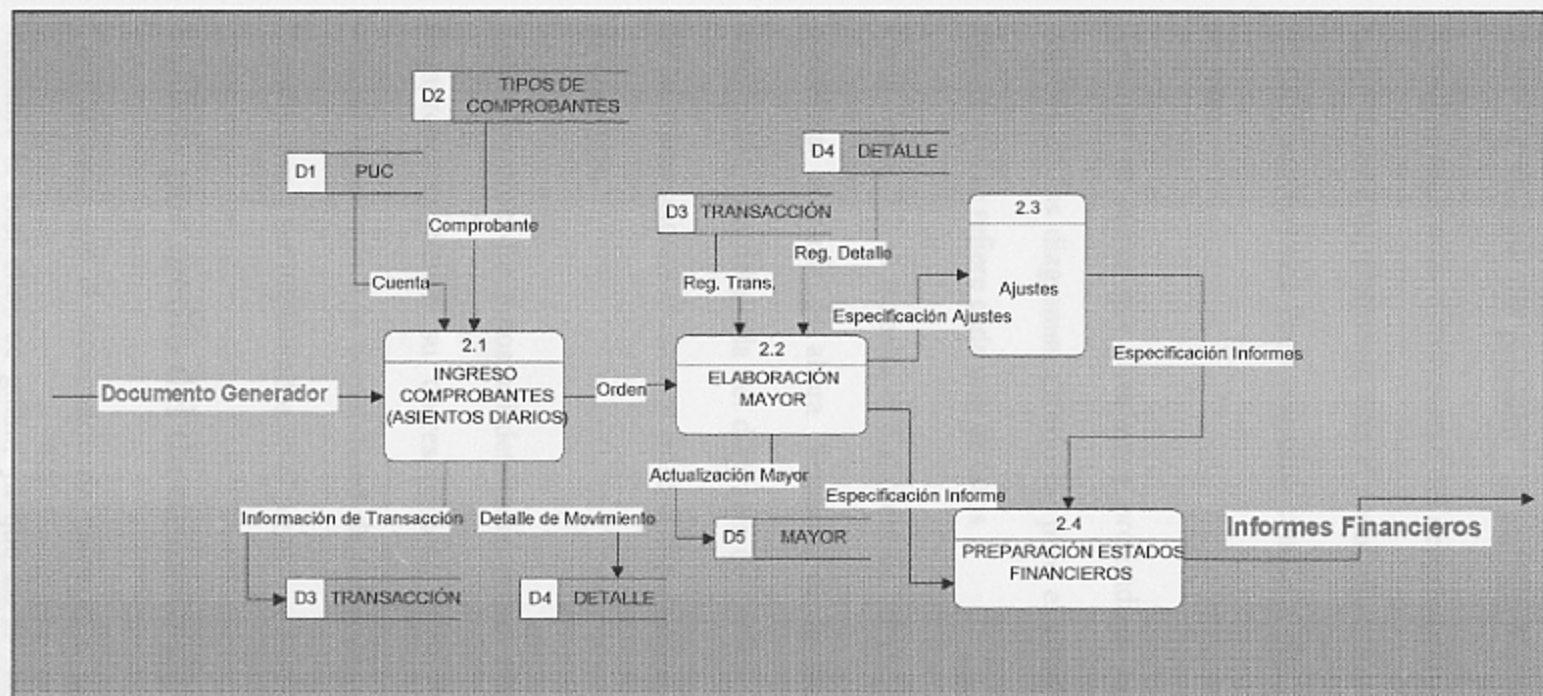


Figura 4.5 Explosión del Diagrama del Procedimiento 2

Como fue ilustrado, el diagrama de contexto planteado inicialmente dejaba fuera del alcance del proyecto la elaboración de informes y ejecución de consultas realizadas directamente del departamento financiero. Es más, como a cargo de este departamento está el crear, modificar y eliminar cuentas, carecer de esta característica iba a constituir una falla que no nos podíamos dar el lujo de tener.

También es cierto que podríamos profundizar tanto como se desee en los detalles de los diagramas anteriores, pero el propósito es primeramente dejar claro a qué se refiere cada uno de éstos.

En la figura 4.2 se muestra el diagrama de contexto completo como se lo va a considerar a partir de ahora. Seguido, en la figura 4.3, encontramos una descripción más detallada del diagrama de contexto pero con los componentes de las transacciones existentes entre el departamento de contabilidad y el resto de entidades.

Todos los procedimientos ahí detallados como: Plan de Cuentas, Consultas, y Comprobantes tienen su correspondiente especificación, pero es de vital utilidad la referida a la contabilidad de los comprobantes para definir ciertos lineamientos a ser implantados en el programa a ser desarrollado.

### **4.3 Análisis por medio del diccionario de datos.**

Considerando que el diccionario de datos es la herramienta de los analistas mediante la cual ellos se guían; ya que en 61 recolectan datos, coordinan y

confirman lo que significa un termino de dato específico que permite generalizarse a fin de ser claramente comprendido por cada elemento de la organización inmiscuida en el programa en desarrollo, vamos a mostrar el contenido de nuestro Diccionario de Datos, el cual ha sido obtenido directamente a partir del mismo lenguaje de programación elegido.

### Tabla: puc

Clave principal: id cuenta

#### camp0

id	N(5)	N	newid()
cod_cuenta	C(9)	Y	
descripcio	M(4)	N	
cod_grupo	I(4)	N	
nombre cue	C(30)	N	
fecha	D(8)	N	

#### Índice

id_cuenta	id
cod_grupo	cod_grupo
cod_cuenta	cod_cuenta
nom_cuenta	nombre cuenta

### Tabla: detallenniveles

Clave principal: id

#### camp0

id	I(4)	N	newid()
----	------	---	---------

nivel I(4) N

ancho I(4) N

### Índice

id	id
nivel	nivel

### Tabla: ids

Clave principal:

#### Campo

nextid B(8) N

Tabla C(20) N

mascara C(20) N ""

#### ndice

Tabla	Tabla
-------	-------

### Vista: detnivVista

#### Campo

nivel I

ancho I

id I

### Tabla: naturaleza

Clave principal: naturaleza

**Campo**

id	I(4)	N
naturaleza	C(10)	N

**Índice**

naturaleza	id
------------	----

**Tabla: tipocuenta**

Clave principal: tipoc

**camp0**

id_tipoc	I(4)	N
tipo_cuent	C(10)	N
fnndice		
tipoc	id tipoc	

**Vista: planVista****Campo**

id	N(5)
cod_grupo	I
cod_cuenta	C(9)
nombre_cue	C(30)
descripcio	M
fecha	D

**Tabla: varios**



Clave principal:

**Campo**

camp0            C(15) N

valor            I(4) N

**Tabla: transactiont**

Clave principal: id

**camp0**

id                I(4) N                            newid()

num\_tran        I(4) N                            newtran()

num comp        I(4) N

fecha            D(8) N

cod comp        I(4) N

concept0        M(4) N

**Índice**

id                                id

num\_tran                        num\_tran

num comp                        num\_comp

cod\_comp                        cod\_comp

**Tabla: grupos**

Clave principal: id

**Campo**

id                I(4) N                            newid()

codigog        I(4) N



nombreg	C(10)	N	
saldo	C(20)	N	
fn dice			
id			id
codigo			codigog
nombreg			nombreg

### Vista: gruposVista

#### Campo

nombreg	C(10)
saldo	C(10)
codigog	I

### Tabla: comp

Clave principal: id

#### Campo

id	I(4)	N	newid()
cod_comp	I(4)	N	
nomb comp	C(20)	N	
descripcio	M(4)	N	
fecha_crea	D(8)	N	
next comp	I(4)	N	

### Índice

id	id
codigo _corn	cod_comp

**Tabla: trancomp**

Clave principal: id

**Campo**

id	I(4)	N	newid()
num tran	I(4)	N	
num comp	I(4)	N	
cod_cuenta	C(9)	N	
vdebe	N(15)	N	
vhaber	N(15)	N	
mayor	L(1)	N	

**Índice**

id	id
num_tran	num_tran
num_comp	num_comp
cod_cuenta	cod cuenta

**Vista: transcompVista****Campo**

num tran	I
num comp	I

**Vista: trancomppucVista****Campo**

id I  
 cod cuenta C(9)  
 num tran I  
 num\_comp I  
 vdebe N(15)  
 vhaber N(15)

### **Tabla: temporal**

Clave principal:

#### **Campo**

codigo\_cue C(15) N  
 nombre\_cue C(20) N

CODIGO\_CUE  
 CODIGO\_CUE  
 CODIGO\_CUE

### **Vista: moditransVista**

#### **camp0**

id I  
 num tran I  
 num comp I  
 cod\_cuenta C(9)  
 vdebe N(15)  
 vhaber N(15)  
 mayor L  
 nombre\_cue C(30)  
 fecha D  
 cod\_comp I

**Tabla: mayor**

Clave principal: id

**campo**

id	N(10)	N
cod cuenta	C(15)	N
s_debe	N(15)	N
s_haber	N(15)	N
periodo	I(4)	N
año	C(4)	N

**Índice**

id	id
cod_cuenta	cod_cuenta

Si bien es cierto que lo que hasta el momento se ha presentado tan solo corresponde a datos obtenidos a partir del mismo lenguaje definido para la implantación del programa a realizar, aquí vale la pena hacer un alto, ya que si afirmamos que por ser un programa de propósito general eso no quiere decir que . estemos contradiciendo, aparentemente a uno de los pasos que por largo tiempo ha sido (y sigue siendo) el pilar de los programas que se utilizan, y es la especificación de requerimientos.

**Es** esta especificación de requerimientos la que podría pensarse ha sido obviada y “casi autoritariamente” se la ha ubicado un puesto detrás de la

lógica del programador. Pero esto que aparenta, se revela en detalle desde esta línea como la verdad sin prueba. Es una verdad porque a pesar de todo siempre fue necesario consultar a entendidos en el área contable principalmente, los cuales lejos de estar seguros de lo que decían, si se mostraban, unos, algo escépticos en cuanto a “otro programa más de contabilidad con las típicas funciones y también las típicas limitaciones”.

Adicionalmente fue necesario explicar que la base del diseño del programa descansaba en un análisis enfocado a las transacciones que son comunes a todo proceso contable, al menos dentro de la contabilidad general, que es el ámbito específico dentro del cual se ha enfocado el presente trabajo.

## CAPÍTULO 5

# **Diseño del Sistema.**

### **5.1 Diseño de la Entrada.**

La idea del diseño de la entrada es para conseguir que sea efectiva, es decir, una entrada de datos que haga que el usuario no se confunda y al mismo tiempo este pueda cometer errores sin que el sistema sufra algún tipo de modificación o daño irreparable. Es así que al momento de diseñar la entrada de datos correspondiente a nuestro sistema de contabilidad, necesitamos asegurar que el usuario pueda con toda confianza trabajar en el programa. Pues bien, entre las técnicas para el desarrollo de entradas efectivas de datos tenemos los siguientes:

- Formas fáciles de llenar.
- Formas que satisfagan el objetivo para el que fueron diseñadas.

- Formas que aseguren el llenado preciso.
- Mantenimiento de las formas atractivas.

Todo esto para cumplir con los objetivos del diseño de la entrada los cuales son:

- efectividad,
- facilidad de uso,
- consistencia,
- simplicidad,
- precision, y
- atractivo

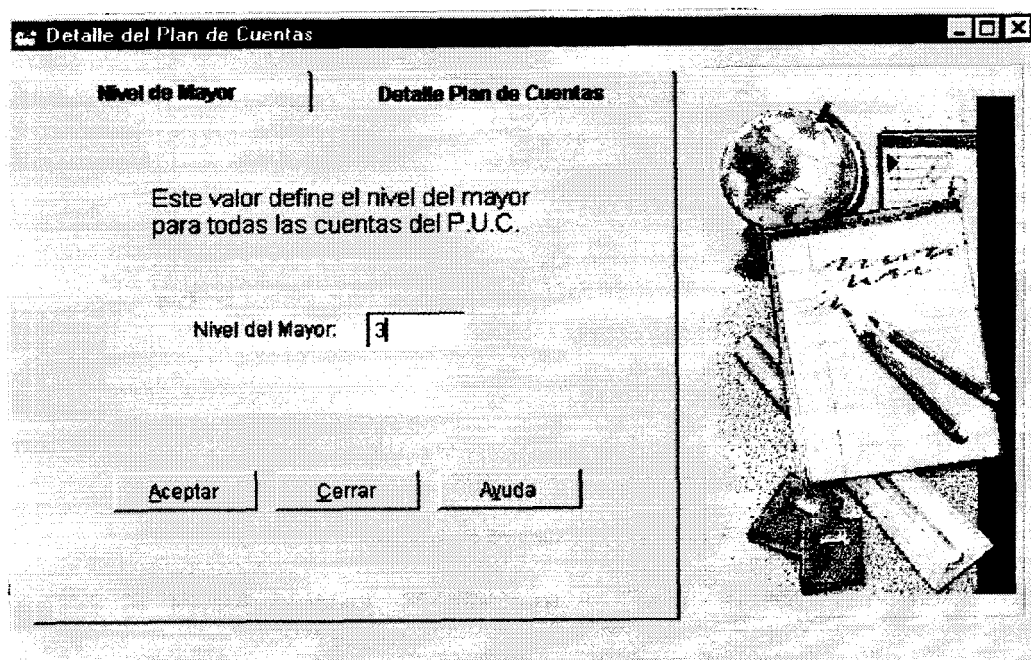
SECRETARÍA  
SUBSECRETARÍA  
CENTRAL

Pues en el caso del programa de contabilidad, hay ocasiones en las que la entrada no solo se vuelve importante, sino que su descuido bien puede ser el origen de una gran catástrofe.

Por ello, el diseño de la entrada empieza con las defmiciones básicas para iniciar un período con Tabla, esto es, un detalle completo del número de niveles en el plan de cuentas, y el nivel, en las cuentas, a partir del cual se defmen las cuentas de mayor.



Pensando en que el usuario puede definir ambas cosas al mismo tiempo, por ello se incluyó ambas entradas en un solo formulario de páginas, como se muestra a continuación:



**Figura 5.1** Diseño de la entrada efectiva: Definición del Plan de Cuentas.

Entonces, podemos apreciar que en este formulario solo tenemos la opción de ingresar un valor, mientras que en el siguiente, que es en el que se define la estructura del plan de cuentas, tenemos que ingresar una serie de números. Estos están orientados a que los usuarios solo se limiten a llenarlos con lo que es útil.

Para tener una idea de la siguiente página dentro de este formulario mostramos la página que está continuaci.

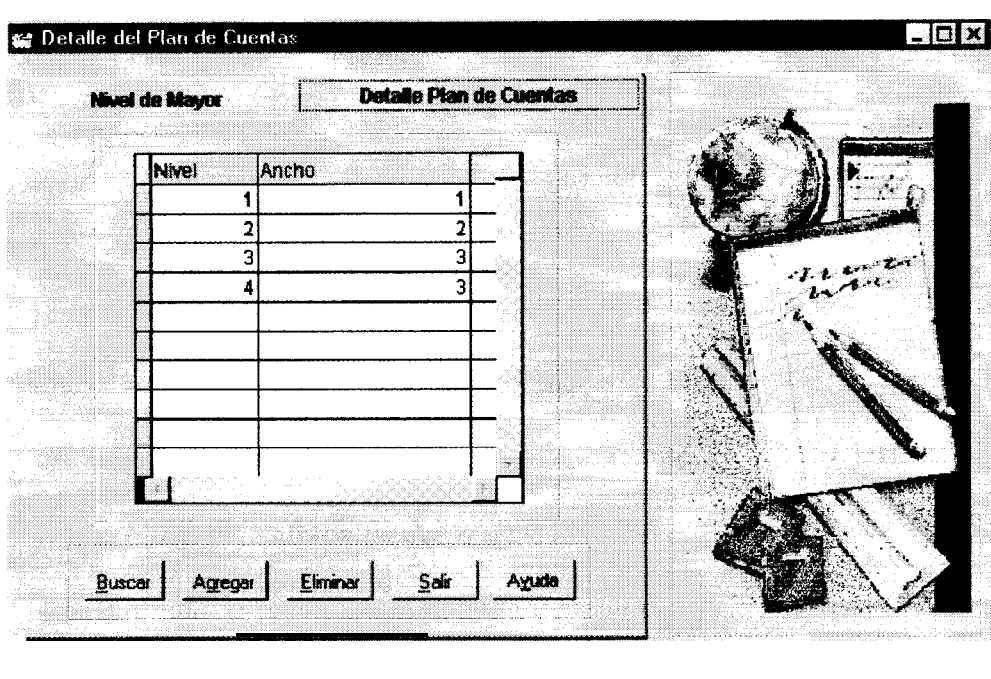


Figura 5.2 Diseiio de la entrada efectiva: Plan de Cuentas

En la figura anterior se muestra el lugar donde el usuario define exclusivamente la forma del plan de cuentas, haciendo que este se limite a

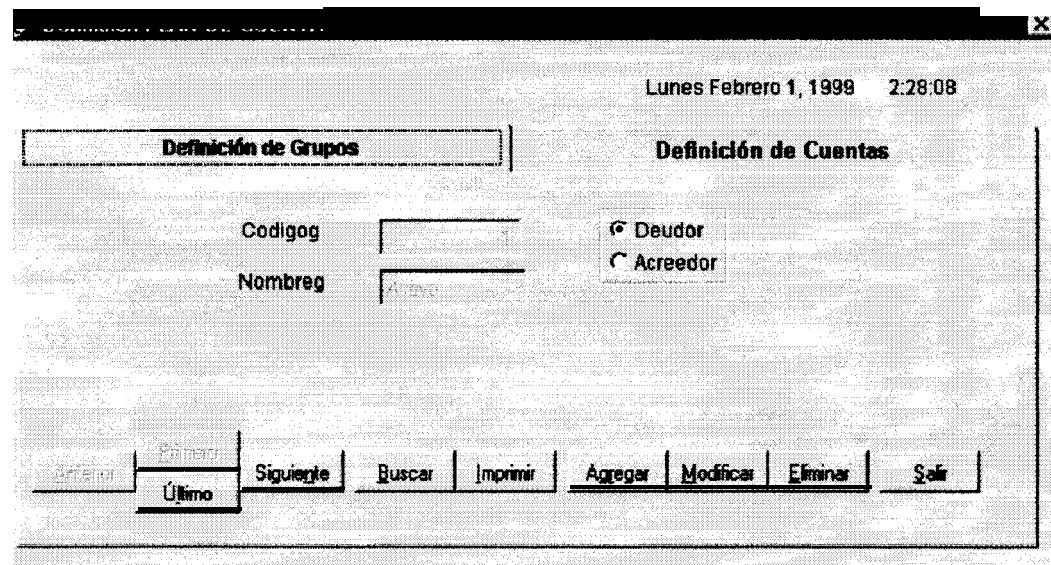


Figura 5.3 Diseiio de la entrada efectiva de datos: Definición de grupos.

solo ingresar parejas de números que claramente están indicados en los encabezados de las cuadrículas se tratan de el número del nivel y del ancho de dicho nivel.

Pero tambien debe garantizarse la entrada efectiva de las cuentas y de los grupos que han de ser definidos para las cuentas, por lo que es necesario una entrada de datos que permita al usuario con facilidad manipular la información antes mencionada. La figura anterior (fTigura5.3) muestra la forma en que la entrada de datos para el ingreso de los grupos de cuentas ha sido definido de una manera sencilla y atractiva.

Seguido en el mismo formulario, podemos darnos cuenta que luego de definir los grupos, se procede a la definición de las cuentas pertenecientes al

Definición PLAN DE CUENTAS

Lunes Febrero 1, 1999 2:29:18

**Definición de Grupos**

Grupo:

Cuenta:

Nombre:

Descripción:

Fecha: 26/01/1999

**Definición de Cuentas**

Código	Cuenta
1-01-001-000	Activo
1-01-000-000	Activo Circulante
1-01-001-000	Caja
1-01-001-001	Caja chica # 1
1-01-001-002	Caja chica # 2
1-01-002-000	Bancos

Figura 5.4 Diseño de una entrada efectiva de datos: Ingreso de cuentas.

Como segundo paso para poder tener la información disponible a ser procesada, hace falta definir lo que va a permitir el ingreso de las transacciones y la modalidad en la que estas van a ser identificadas. Dentro de este proceso contable, dicha modalidad estará dada por comprobantes, los cuales deben ser definidos primero para luego generar transacciones a partir de uno de ellos.

El por qué se van a emplear comprobantes se debe a que de acuerdo al análisis efectuado, es necesario dar facilidades de auditoria en los procesos del departamento de contabilidad. Dichas facilidades deben estar orientadas a facilitar la búsqueda de un documento que haya generado alguna transacción así como debe prestar facilidades para identificar las cuentas afectadas en dicha transacción y el monto respectivo.

Por ello la necesidad de trabajar por medio de comprobantes. Si existe algún problema o si sencillamente se desea hacer alguna clase de seguimiento a los ingresos, por ejemplo, entonces solo nos remitimos a analizar los comprobantes de ingreso.

Una muestra del formulario que ha de emplear el usuario para realizar primero la definición de los comprobantes se muestra a continuación en la figura 5.5.



Como segundo paso para poder tener la información disponible a ser procesada, hace falta definir lo que va a permitir el ingreso de las transacciones y la modalidad en la que estas van a ser identificadas. Dentro de este proceso contable, dicha modalidad estará dada por comprobantes, los cuales deben ser definidos primero para luego generar transacciones a partir de uno de ellos.

El por qué se van a emplear comprobantes se debe a que de acuerdo al análisis efectuado, es necesario dar facilidades de auditoría en los procesos del departamento de contabilidad. Dichas facilidades deben estar orientadas a facilitar la búsqueda de un documento que haya generado alguna transacción así como debe prestar facilidades para identificar las cuentas afectadas en dicha transacción y el monto respectivo.

Por ello la necesidad de trabajar por medio de comprobantes. Si existe algún problema o si sencillamente se desea hacer alguna clase de seguimiento a los ingresos, por ejemplo, entonces solo nos remitimos a analizar los comprobantes de ingreso.

Una muestra del formulario que ha de emplear el usuario para realizar primero la definición de los comprobantes se muestra a continuación en la figura 5.5.

Definición de Comprobantes

Id

Código

Tipo

Fecha creacion

Descripcion

Último Siguiente Buscar Imprimir Agregar Modificar Eliminar Salir

Figura 5.5 Diseño de la entrada efectiva: Definición de Comprobantes.

Una vez ingresados los tipos de comprobantes, entonces se pueden ahora sí ingresar nuevas transacciones para lo cual se emplea el siguiente formulario:

Ingreso de Comprobantes

Fecha 01/26/1999 Número Transacción:

Comprobante Concepto

Guardar Cancelar

Cuenta: Saldo:  Debe  
 Haber

Código	Cuenta	Debe	Haber
Suma			

Agregar Eliminar Modificar

Figura 5.6 Diseño de una entrada efectiva: Ingreso de Transacciones

En CI, podemos darnos cuenta que es el que presenta una mayor complejidad debido al número de controles que incluye. Para evitar que esto sea un impedimento de que la entrada pueda ser efectiva los controles han sido agrupados de acuerdo a su función.

Una vez definida la entrada de datos y luego de habernos asegurado de la eficacia de esta ahora nos preocuparemos de la salida para que, de igual manera, sea efectiva.

## **5.2 Diseño de la Salida.**

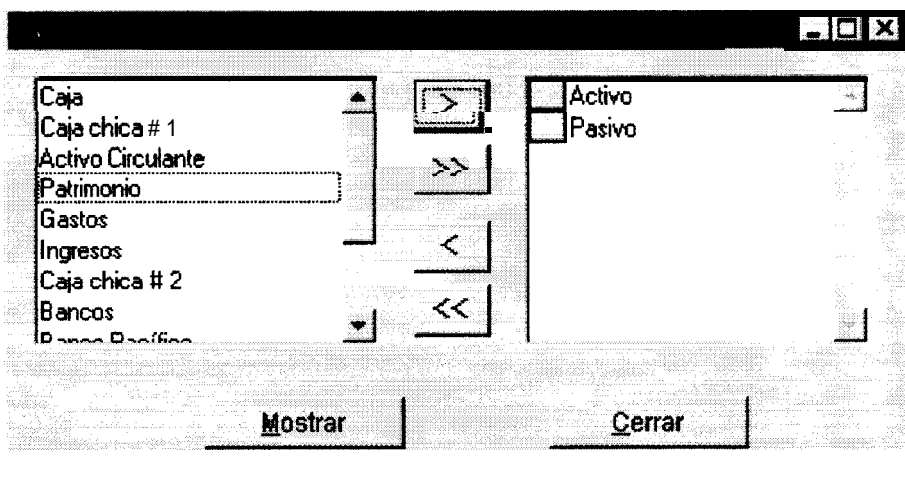
La salida dentro de nuestro programa está básicamente dado por los reportes generados a partir de que las transacciones han sido ingresadas.

Los diferentes reportes son un tema que hemos tratado de abarcar desde el comienzo o al menos desde que fue planteado que una de las principales falencias de los sistemas actuales es el de lo inflexibles que son los reportes. Pues en esta ocasión se ha tratado de que los informes en una parte sean personalizados por el usuario, aunque esto no se puede conseguir en un 100%, si es cierto que una diferencia significativa entre este programa y los existentes es una realidad y el diseño de la salida lo va ilustrar.

En esta ocasión no podemos hacer uso de formularios porque la salida ya no lo constituye un formulario sino un informe. El diseño correspondiente al

informe esta en función directa de los que el programa permite definir y / o ejecutar.

Entre ellos, a manera de ilustración, tenemos al del plan de cuentas, el cual puede definirse para todas las cuentas o para un grupo de ellas.



**Figura 5.7 Diseño de la Salida: Personalización de los reportes.**

La figura anterior muestra cuando se define un grupo de ellas, en la cual, tenemos que el usuario debe escoger las cuentas o grupo de ellas de las que desea el informe.

Así luego de que se ha escogido la o las cuentas ( o grupos) se muestra un informe con el siguiente formato:



**Tabla I. Formato de Reporte de Cuentas**

<b>Código Cuenta</b>	<b>Nombre Cuenta</b>
9-99-999-999	XXXXXXXXXXXXXXXXXXXX
9-99-999-999	XXXXXXXXXXXX
9-99-999-999	XXXXXXXXXX

Cuando se ha hecho la consulta para todas las cuentas el formato cambia:

**Tabla II. Formato Todas las Cuentas.**

<b>Código Cuenta</b>	<b>Nombre Cuenta</b>
<b>Activo</b>	
9-99-999-999	XXXXXXXXXXXX
9-99-999-999	XXXXXXXXXX
<b>Pasivo</b>	
9-99-999-999	XXXXXXXXXXXXXXXXXX
9-99-999-999	XXXXXXXXXXXX
<b>Patrimonio</b>	
9-99-999-999	XXXXXXXXXXXXXXXXXX
9-99-999-999	XXXXXXXXXXXX
<b>Ingresos</b>	
9-99-999-999	XXXXXXXXXXXXXXXXXX
9-99-999-999	XXXXXXXXXXXXXXXXXX
<b>Gastos</b>	
9-99-999-999	XXXXXXXXXXXXXXXXXX

---

9-99-999-999

XXXXXXXXXXXXXXXXXX

---

Esto hace que se obtengan las cuentas agrupadas por su saldo y en orden por su codificación.

El resto de reportes son los correspondientes a los movimientos y los estados financieros.

Un típico reporte de movimiento diario podría ser de la siguiente manera:

**Tabla III. Formato de Reporte de Movimiento de Diario.**

<b>Fecha</b>	<b>Código</b>	<b>Cuenta</b>	<b>Debe</b>	<b>Haber</b>
9919919999	9-99-999-999	x x x x x x x x	99999999	99999999
99/99/9999	9-99-999-999	x x x x x x x x	99999999	99999999
9919919999	9-99-999-999	XXXXXXXXXX	99999999	99999999
9919919999	9-99-999-999	x x x x x x x x	99999999	99999999
9919919999	9-99-999-999	x x x x x x x x	99999999	99999999
99/99/9999	9-99-999-999	x x x x x x x x	99999999	99999999

Si lo que se desea es el movimiento de una cuenta en especial podría mostrarse:

**Tabla IV. Formato de Reporte por Cuenta.**

Cta. : 9-99-999-999

<b>Fecha</b>	<b>Comprobante</b>	<b>Debe</b>	<b>Haber</b>
99/99/9999	<b>99999</b>	<b>99999999</b>	<b>9999999</b>
99/99/9999	<b>99999</b>	<b>99999999</b>	<b>9999999</b>
99/99/9999	<b>99999</b>	<b>99999999</b>	<b>9999999</b>
<b>9919919999</b>	<b>99999</b>	<b>99999999</b>	<b>9999999</b>

Lo anterior se puede también generalizar al caso de que se prefiera no solo de una cuenta sino de un grupo de ellas, en tal caso se procedería a generarse una sucesión de reportes de las mismas características que los anteriores.

Los demás son reportes que pertenecen exclusivamente a la contabilidad por lo que normalmente el formato es el mismo. Entre estos reportes tenemos el balance de sumas y saldos o de comprobación, el de comprobación ajustado, el balance general y el estado de resultados.

El balance de comprobación tiene un formato similar al siguiente:

**Tabla V. Formato de Balance de Comprobación.**

Empresa ABC  
Balance de Comprobación  
Al 99/99/999

<b>Código</b>	<b>Cuenta</b>	<b>Debe</b>	<b>Haber</b>
9-99-999-999	x x x x x x x x	99999999	
9-99-999-999	x x x x x x x x		9999999999
9-99-999-999	x x x x x x x x	99999999	
9-99-999-999	x x x x x x x x		9999999999
9-99-999-999	x x x x x x x x	99999999	
	<b>Suma:</b>	<b>9999999999</b>	<b>9999999999</b>

El estado de resultados posee así mismo un formato, el cual no es único, y se podría esquematizar así.

**Tabla VI. Formato de Estado de Pérdidas y Ganancias.**

Empresa ABC  
Estado de Pérdidas y Ganancias  
Al 9919919999

<b>Código</b>	<b>Parcial</b>	<b>Total</b>
<b>Ingresos</b>		
9-99-999-999	99999999	
9-99-999-999	99999999	

		<b>9999999999</b>
<b>Egresos</b>		
<b>9-99-999-999</b>	<b>999999</b>	
		<b>99999999</b>
<b>Pérdida o Utilidad</b>		<b>999999</b>

Finalmente el balance general se lo puede representar de la siguiente manera:

**Tabla VII. Formato de Balance General.**

Empresa ABC

Balance General

Al 9919919999

<b>Activos</b>		<b>Pasivos</b>	
<b>9-99-999-999</b>	<b>99999999</b>	<b>9-99-999-999</b>	<b>99999999</b>
<b>9-99-999-999</b>	<b>99999999</b>	<b>9-99-999-999</b>	<b>99999999</b>
<b>9-99-999-999</b>	<b>99999999</b>	<b>9-99-999-999</b>	<b>99999999</b>
<b>9-99-999-999</b>	<b>99999999</b>	<b>9-99-999-999</b>	<b>99999999</b>
<b>9-99-999-999</b>	<b>999999</b>	<b>Patrimonio</b>	
<b>9-99-999-999</b>	<b>99999999</b>	<b>9-99-999-999</b>	<b>99999999</b>
<b>9-99-999-999</b>	<b>99999999</b>	<b>9-99-999-999</b>	<b>999999</b>
<b>Total Activos</b>	<b>9999999999</b>	<b>Total Pasivo y Patrimonio</b>	<b>9999999999</b>

Habiendo definido la entrada y la salida entonces podemos pasar a la programación de la aplicación. Hay que aclarar que la intención de ilustrar la

salida de la forma presentada anteriormente es tan solo con fines ilustrativos puesto que tan solo la forma es la que se mantiene.

## CAPÍTULO 6

# Programación y Pruebas.

### 6.1 De la programación en general.

Es necesario recordar que la herramienta a ser utilizada para hacer efectiva la aplicación es Visual FoxPro ® version 5.0. Esta version del Fox es mucho más compleja pero a la vez atractiva porque aparte que el entorno de desarrollo es netamente visual, trae las mismas instrucciones objetos, y librerías que los entendidos en el Fox para D.O.S. conocen.

Durante el desarrollo de este trabajo, los principales inconvenientes, por llamarlos de alguna manera, ha sido con el manejo de eventos, que es otra de las características de este lenguaje.

Si bien es cierto en el capítulo 2 hablamos sobre los lenguajes de programación y sus generaciones; es un poco difícil convencer de primera que éste es una herramienta de cuarta generación como lo debería ser por ser la herramienta elegida que debe ir acorde con el tema de la presente tesis. Mas si profundizamos en el conocimiento del Visual nos daremos cuenta que primeramente sobran motivos para señalarlo como un lenguaje de cuarta; las posibilidades de acceso a tablas de bases de datos por medio de sentencias SQL, lo hacen poderosamente fácil de programar. Lo confuso es que el resultado de estas sentencias se dirigen a objetos que resultan del instanciamiento de clases padres lo cual aparentaría que se trata de una programación orientada a objetos. Pero recordando que la idea de la cuarta generación es que el programador más se preocupe por lo que desea y no por cómo presentarlo mas el hecho de que es reconocido que un lenguaje de programación bien puede ser categorizado en un lenguaje u otro dependiendo de quien lo esté discutiendo y sus puntos de vista, entonces ofkialmente queda establecido el uso de Visual FoxPro® como la herramienta de desarrollo acorde con los delineamientos originales de esta tesis de grado.



## 6.2 Programas y sentencias usadas.

Para poder darle forma a la aplicacion, nos hemos de referir a las principales instrucciones y bloques de programas que han permitido que poco a poco esta aplicacion tome forma.

Primeramente es necesario el uso de un programa que inicie el menu de nuestra aplicacion y antes de ello deberia de establecer ciertos parámetros de arranque para que todo funcione según lo esperado.

Para ello hacemos uso del programa de inicio "INICIO.PRG" el cual se detalla a continuación.

### **Programa de inicio.**

*SET TALK OFF*

*CLEAR*

*title="Solución Contable"*

*CLOSE ALL*

*CLEAR ALL*

*PUBLIC oApp,EjecuTabla*

*EjecuTabla=.F.*

*SET CLASSLIB TO c:\solution\clases\misclas.vcx*

*set path to c:\solution\datos,c:\solution\formularios,;*

*c:\solution\reports*



```
oApp=CREATEOBJECT("entorno")  
  
oApp.Setup  
  
DO c:\solution\menues\menuprin.mpr  
  
READ EVENTS  
  
IF !EjecuTabla  
  
    oApp.Cleanup  
  
ENDIF  
  
CLOSE ALL  
  
RELEASE ALL EXTENDED  
  
CL EAR  
  
CLEAR ALL
```

Pues el código mostrado anteriormente ha sido empleado para, primero eliminar los resultados de los comandos. Seguido para cerrar cualquier base de datos abierta y por ende cualquier tabla, para poder tener un control total de qué tablas y bases se abren y se cierran; se limpia el contenido de la memoria; se ejecuta el programa de menu para que aparezca la ventana principal del programa.

Seguido se declaran las variables objeto que más adelante se van a necesitar y se establecen las rutas de acceso a los directorios que contienen los formularios, las bases de datos los programa de menu, las consultas, etc.

Este programa se mantiene residente hasta cuando ya no se ejecutan más eventos en la aplicación y finaliza la ejecución de la misma borrando todo lo que cargo en memoria y cerrando todas las bases y sus respectivas tablas.

Es de mucha importancia reconocer que existe una diferencia significativa de la programación en esta herramienta y la forma como los entendidos en las versiones anteriores lo hacían. Antes tener muchos y extensos programas como el que acabamos de revisar se usa para el inicio, era esto un símbolo de que el programa era mejor. Ahora el número de programas así aparte queda casi completamente reducido y el código empleado se encuentra en los formularios que se vayan creando, que son con los que el usuario se relaciona durante la ejecución de la aplicación.

Como antes se mencionó el acceso a las bases de datos y sus tablas se puede hacer por medio del SQL ; aunque también se lo podría hacer por medio de la forma tradicional de programación.

Unas muestras de ambos casos se presentan a continuación:

```
SELECT Trancomp. id, Trancomp. cod_cuenta, Trancomp. num_tran, ;
Trancomp. num_comp, Trancomp. vdebe, Trancomp. vhaber;
FROM pcuentas!trancomp INNER JOIN pcuentas!puc ;
ON Trancomp. codcuenta = Puc. cod-cuenta;
ORDER BY Trancomp. num_tran, Trancomp. num_comp
```

En estas líneas de código se especifica el la forma en que se ejecuta una consulta que toma de dos tablas (**FROM, INNER JOIN**) los campos que son de interés (la parte del **SELECT**) sujetos a la condición de que los códigos de las cuentas sean los mismos (**ON**) ordenados por el número de la transacción y el número del comprobante (**ORDER BY**).

El código que a continuación se muestra corresponde al utilizado para generar números de identificación de registro (id) de manera automática:

```

FUNCTION NewID(tcAlias)
LOCAL lcAlias, lnID, lcOldReprocess, lnOldArea

lnOldArea = SELECT()

IF PARAMETERS() < 1
  lcAlias = UPPER(ALIAS())
ELSE
  lcAlias = UPPER(tcAlias)
ENDIF

lcOldReprocess = SET("REPROCESS")

* Bloquea hasta que el usuario presione Esc
*SET REPROCESS TO AUTOMATIC

IF !USED("ids")
  USE pcuentas!ids IN 0
ENDIF
SELECT ids

IF SEEK(lcAlias, "ids", "table")
  IF RLOCK()
    lnID = ids.nextid
    REPLACE ids.nextid WITH ids.nextid + 1
  
```

```

        UNLOCK
    ENDIF
ENDIF

SELECT (lnOldArea)
SET REPROCESS TO lcOldReprocess

RETURN lnID
ENDFUNC

FUNCTION NewtrAN(tcAlias)
LOCAL lcAlias, lnID, lcOldReprocess, lnOldArea

lnOldArea = SELECT()

lcAlias ="TRANSACCION"

lcOldReprocess = SET('REPROCESS')

* Bloquea hasta que el usuario presione Esc
*SET REPROCESS TO AUTOMATIC

IF !USED("ids")
    USE pcuentas!ids IN 0
ENDIF
SELECT ids

IF SEEK(lcAlias, "ids", "table")
    IF RLOCK()
        lnID = ids.nextid
        REPLACE ids.nextid WITH ids.nextid + 1
        UNLOCK
    ENDIF

```

El código anterior es el que corresponde al “NEWID()” que aparece en casi todas las tablas del diccionario de datos en el campo id como su valor predeterminado. Esta es una función definida por el usuario que lo que hace es recibir el nombre (Alias) de la tabla en uso y el valor del campo id correspondiente al nuevo registro que se está agregando se encuentra almacenado en una tabla cuyo objetivo solo es hacer aquello.

El resultado cada inserción de un nuevo registro se obtiene un identificador único y no nos preocupamos por él nunca más en la programación.

Sería demasiado extenso este capítulo si tratásemos de especificar cuáles han sido todos y cada uno de los bloques de programa empleados. Lo importante es saber que ellos se encuentran en los formularios que siempre estarán presentes y el manejo de objetos hace que la herencia de las clases padre nos obvien en ciertos casos el considerar el código que debería usarse.



## CAPÍTULO 7

# Documentación y uso del Sistema.

### 7.1 Del Menú principal.

El que a continuación se muestra es el menú principal del que se realizarán todas las operaciones necesarias para poder efectuar el proceso contable.

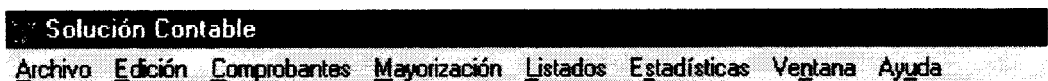


Figura 7.1 Menú Principal.

Este menú como se puede apreciar, cuenta con opciones cuyas funciones las vamos a ir detallando una a una.

### 7.1.1 Menú Archivo.

En él se realizan las principales operaciones de definición del plan de cuentas, re – numeración de los índices de los archivos, etc.

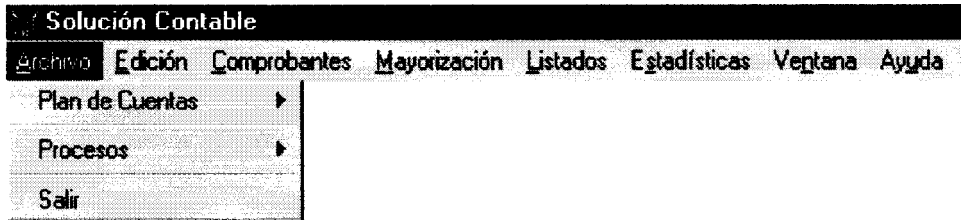


Figura 7.2 Menú Archivo.

Luego de seleccionarlo tenemos las opciones que se muestran en la figura anterior. Dentro de cada una de ellas se tiene:

#### Plan de cuentas:

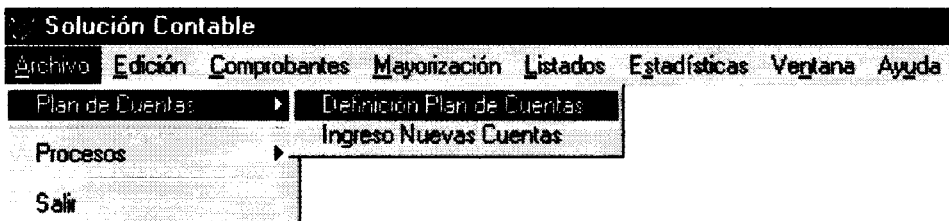


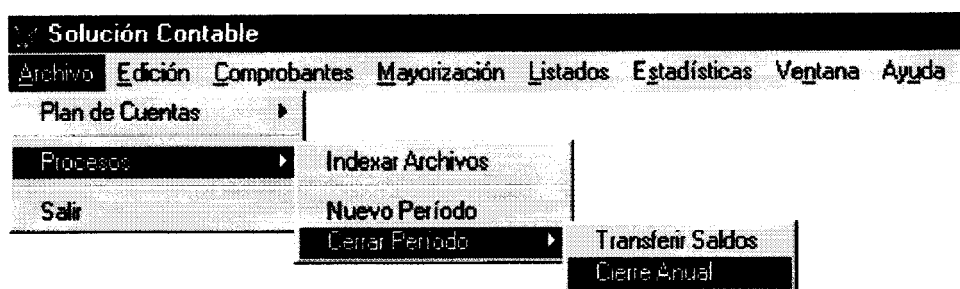
Figura 7.3 Plan de Cuentas.

En la primera opción, Definición del Plan de Cuentas, se establece el nivel del mayor general y se detalla el número de niveles que va a manejar el plan así como el ancho (número de dígitos) que va a tener cada nivel.



En la segunda opción, **Ingreso Nuevas Cuentas**, el programa permite al usuario ingresar nuevas cuentas dentro de los grupos que se hayan definido, o si no, que se definan.

Dentro de la opción de Procesos, tenemos:



**Figura 7.4 Cerrar Período.**

- Indexar Archivos, que es una operación estrictamente técnica (no forma parte del proceso contable) pero es útil en el sentido de que cuando por alguna razón (variación en el voltaje del suministro eléctrico) los índices se han dado, esta operación los reconstruye.
- Nuevo periodo, sirve para crear un nuevo período contable.
- Cerrar periodo, se lo emplea cuando la operación se la cierra en un período menor de un año (un mes por ejemplo), empleando la opción de transferir saldos; y Cierre Anual, que es cuando hemos finalizado el período ejercicio anual.

Finalmente la última opción, **Salir**, es empleada obviamente cuando deseamos finalizar la ejecución del programa.

### 7.1.2 El Menú Edición.

El siguiente menú disponible es el de edición que es similar al que encontramos en cualquier procesador de palabras, por lo que omitiremos su explicación.

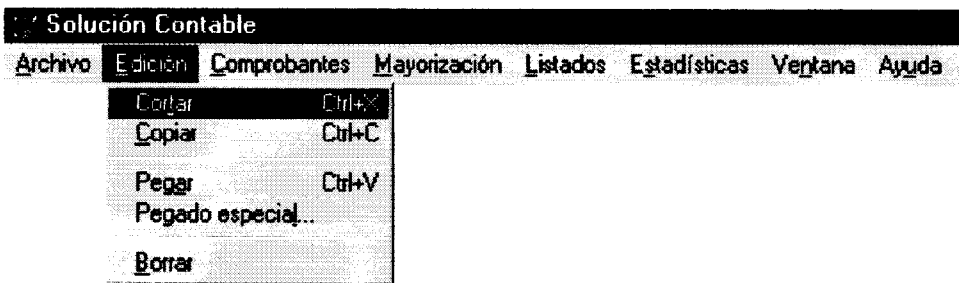


Figura 7.5 Menú Edición.

### 7.1.3 El Menú Comprobantes.

Luego tenemos el menú de **Comprobantes** el cual se muestra y detalla a continuación:

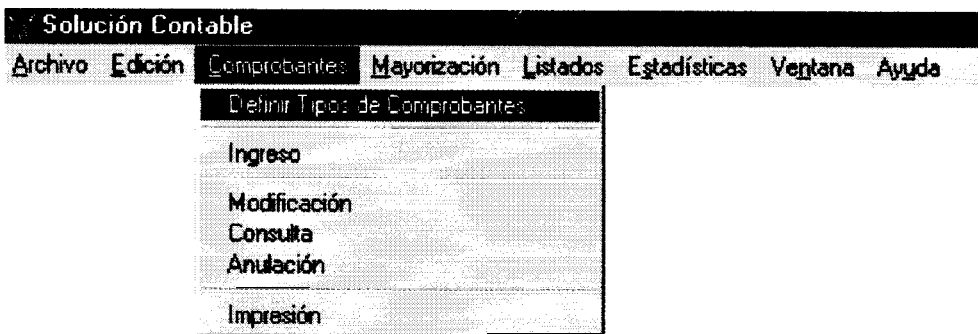


Figura 7.6 Menú Comprobantes.

- Definir Tipos de Comprobante, sirve para detallar la entrada de los nombres y código de los comprobantes con los que vayamos a respaldar nuestra transacción contable.
- Ingreso, sirve para efectuar un ingreso de una transacción nueva como se detalló en el capítulo 5 en el diseño de la entrada efectiva.
- Modificación, Consulta, y Anulación, sirven respectivamente para modificar, consultar y eliminar comprobantes anteriores. La anulación y la modificación solo está disponible para el caso de que el comprobante no haya sido mayorizado, caso en el cual se debe proceder como manda la contabilidad.
- Impresión, esta opción sirve para realizar impresiones de comprobantes que han sido ingresados.

#### **7.1.4 El Menú Mayorización.**

Seguido esta la opción correspondiente a la mayorización. Esta se especifica como sigue:

Todos los comprobantes, es decir, mayoriza todos los comprobantes desde la

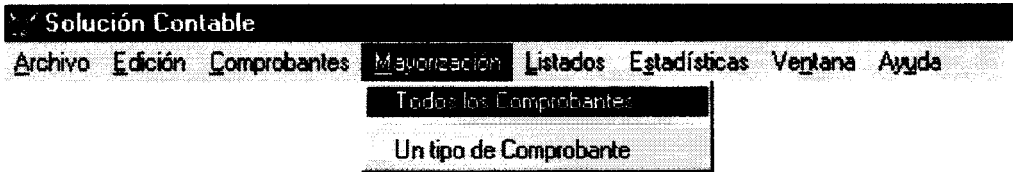


Figura 7.7 Menú Mayorización.

última mayorización efectuada.

- Un tipo de Comprobante, solo realiza la mayorización del tipo de comprobante especificado.

### 7.1.5 El Menú Listados.

A continuación en el menú de listados tenemos:

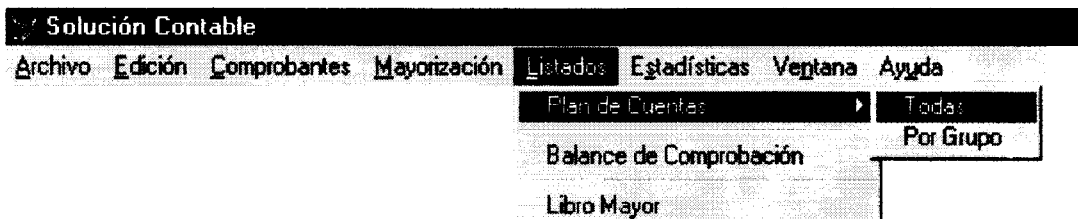


Figura 7.8 Menú Listados.

- Plan de cuentas, para efectuar el listado bien sea de todas cuentas o de las especificadas en la opción Por grupo.
- Balance de Comprobación, lista el balance de su nombre o también conocido como de sumas y saldos.

- Libro Mayor, muestra el contenido del mayor de la cuenta o grupo que se especifique

### 7.1.6 El Menú Estadísticas.

Finalmente tenemos el menú de estadísticas, que se resume así:

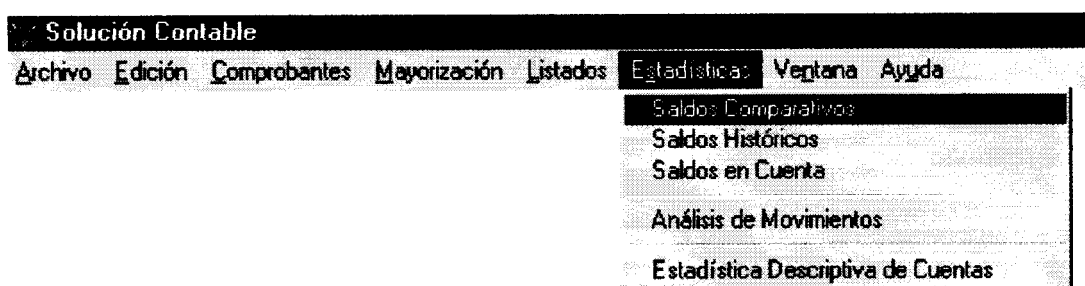


Figura 7.9 Menú Estadísticas

- Saldos comparativos, muestra los saldos de una cuenta o grupo de estas de dos periodos contables.
- Saldos Históricos, muestra los saldos de períodos anteriores de la cuenta o cuentas que se especifiquen.
- Saldos en cuenta, permite obtener el saldo actual de la cuenta que se seleccione.
- Análisis de movimientos, muestra un seguimiento de las cuentas a fin de poder establecer cuál o cuáles son las de mayor actividad contable.

- Estadística Descriptiva de Cuentas, muestra una estadística básica de las cuentas especificadas.

Como una alternativa más se tiene el menú de ventanas, similar al que tienen también los procesadores de palabras. Así mismo en la opción de Ayuda se tiene una descripción acerca de la aplicación desarrollada y una calculadora que es útil para realizar comprobaciones en el momento de hacer los asientos de las transacciones.

## CAPÍTULO 8

# Conclusiones y Recomendaciones.

### 8.1 CONCLUSIONES

- Diseñar sistemas que se adapten fácilmente a las necesidades de los usuarios cuando el producto es elaborado a partir de especificaciones generales hacen que la tarea sea más difícil de concretar, principalmente si nuestra posición es la de penetrar al mercado con una participación representativa y deseamos conseguir clientes que nos sean fieles al producto.
- Ha sido evidente, a partir de los resultados del estudio de mercado de un segmento en particular, que los sistemas que actualmente funcionan, e inclusive los que se ofrecen en el mercado, no satisfacen los requerimientos de todos los clientes en general; aunque la salvedad de esto podría ser que es un tanto dificultoso darle a un solo programa todas las posibilidades de que cubran las expectativas de cualquier cliente, sea lo que fuere que desee obtener

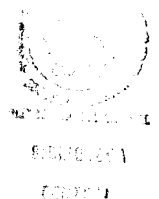
de su contabilidad. Pero no es menos cierto que las herramientas de desarrollo existentes hoy en día permiten hacer cierto tipo de abstracciones a tal medida que el producto resultante es usado o se le es dado el uso dependiendo de quien lo posea.

- Al considerar las posibilidades de introducir como nuevo producto, un sistema de contabilidad al mercado, basados en la estimación realizada a través de la encuesta, podemos afirmar que existe la posibilidad de que quienes sin estar satisfechos con sus programas actuales se sientan muy recelosos de efectuar el cambio por temor a la incertidumbre.
- La investigación de los diferentes productos existentes en el mercado ha permitido observar que hay diferentes maneras de tratar a la contabilidad en su parte externa, pero en sus transacciones internas no se ha notado (ni se debería notar) cambio alguno.



## 8.2 Recomendaciones

- El poco uso de herramientas de desarrollo sofisticadas es un problema que, a pesar que al menos en apariencia se está solucionando, debería darsele más énfasis. El producto nacional, en cuanto a programas se refiere, tiene bastante aceptación, y muy bien podríamos dedicarnos a exportar soluciones hechas en Ecuador. Un indicador de lo lento que ese proceso avanza es la diferencia de programas nuestros que se pueden descargar por Internet para ser evaluados y los disponibles de origen extranjero.
- También en los currículos de materias, aquellas relacionadas con la computación, dejan ver al estudiante en forma muy tardía la amplia gama de herramientas disponibles en el mercado lo cual es perjudicial. Útil sería que no solo programe en ellas sino para que aprenda a evaluarlas, más aún si el elemento clave para la decisión de realizar un proyecto es la rentabilidad ofrecida, en este caso por lo que se emplee para realizar la aplicación.



# **APÉNDICES.**

## APÉNDICE A.

### Formulario de la encuesta realizada a la muestra de los colegios de la ciudad de Guayaquil.



*El estudio siguiente tiene como fin determinar los parámetros para delinear un nuevo sistema de contabilidad en términos de la forma de pago usual, tiempo que tiene el sistema en uso, ambiente de funcionamiento, proporción de aceptación de un sistema con características nuevas y opciones que consideran necesita el sistema en uso.*

#### CUESTIONARIO:

1) La institución maneja la contabilidad mediante algún sistema computacional?

- Si
- No

2) Cómo adquirió el sistema?

- a) De contado \_\_\_\_\_
- b) A crédito \_\_\_\_\_
- c) Otra forma \_\_\_\_\_

3) Hace cuánto tiempo adquirió su sistema de contabilidad?

- Menos de un año. \_\_\_\_\_
- Más de un año pero menos de 5 años. \_\_\_\_\_
- Más de cinco años. \_\_\_\_\_
- Tiene una mejor estimación del tiempo. \_\_\_\_\_

4) Su sistema actual de contabilidad,

- ◆ Fue hecho en particular para la institución \_\_\_\_\_
- ◆ Fue adaptado de uno general y adecuado acorde con los requerimientos. \_\_\_\_\_

5) En qué tipo de ambiente funciona su sistema de contabilidad?

- ❏ Windows 3.x \_\_\_\_\_
- ❖ Windows 95. \_\_\_\_\_
- ❏ Windows NT. \_\_\_\_\_
- ❖ Unix. \_\_\_\_\_
- ❖ Otro: \_\_\_\_\_

6) Si tuviese que considerar la adquisición de un nuevo sistema de contabilidad, le agradaría que fuese un:

- a) Sistema general que pueda ser **adaptado** a los requerimientos de la **institución**. \_\_\_\_\_
- b) Sistema desarrollado exclusivamente **para su institución** \_\_\_\_\_

7) En el **caso** que **decida** cambiar su sistema actual por uno elaborado con una herramienta actual, le **gustaría** que le brinde:

- a) Una **fácil adaptación** a sus necesidades \_\_\_\_\_
- b) **Elaboración** de reportes personalizados \_\_\_\_\_
- c) Las dos anteriores \_\_\_\_\_
- d) Ninguna de las características mencionadas \_\_\_\_\_

8) **Qué componentes** adicionales considera necesita su sistema actual:

- a) Plan de cuentas \_\_\_\_\_
- b) Asientos contables \_\_\_\_\_
- c) Mayorización \_\_\_\_\_
- d) Consultas y/o informes \_\_\_\_\_
- e) Presupuesto \_\_\_\_\_
- f) **Análisis comparativo** por periodo. \_\_\_\_\_
- g) Índices. \_\_\_\_\_
- h) Otro : \_\_\_\_\_

9) Una **estimación del** valor que **fue** pagado por su sistema actual de contabilidad es:

- a) **Menor** que \$1 .000
- b) Entre \$1.000, pero no **más** de \$1.500 \_\_\_\_\_
- c) Entre \$1.500, pero no **más** de \$2.000
- d) Mayor que \$2.000.

10) **¿Cuántas** personas en su **institución** manejan directamente su sistema de contabilidad?

Numero de personas: \_\_\_\_\_

11) Si un sistema en el **cual está** interesado en adquirir ha sido desarrollado en un lenguaje que no puede instalarse en sus equipos por requerimientos de **actualización**:

- a) **Solicita** que lo desarrollen en un lenguaje que pueda ser utilizado por usted. \_\_\_\_\_
- b) **Actualiza** sus equipos. \_\_\_\_\_
- c) **Busca** otra opción. \_\_\_\_\_

12 ) **¿Cuántos** niveles maneja su plan de cuentas?

\_\_\_\_\_

# APÉNDICE B.

## Informe de coleeios

### *Colegio*

Abdón Calderón

ACADEMIA NAVAL ALMIRANTE ILLINGWORTH

ACADEMIA NAVAL GUAYAQUIL

ALEMAN HUMBOLDT DE GUAYAQUIL

AMERICANO DE GUAYAQUIL

ARIEL

ATENAS

BILINGUE JEFFERSON

BRITANICO

CAYETANO TARRUEL

CERVANTES

COLEGIO DE SEÑORITAS REPUBLICA DEL ECUADOR

COPOL

DANTE ALIGHIERI

DOMINGO COMIN

DOMINGO SAVIO

DR. EDMUNDO LOPEZDOMINGUEZ

ECUADOR AMAZONICO, COLEGIO DE SEÑORITAS

ESPIRITU SANTO

FUERZA AEREA ECUATORIANA N° 2

GERMANIA

GUILLERMO ROHDE A.

INDOAMERICA

INSTITUTO COELLO

INSTITUTO INTEGRAL SUDAMERICANO

Instituto San Marcos

INSTITUTO SANTA CATALINA ,

INTERNACIONAL SEK-ECUADOR

INTERNATIONAL SCHOOL

JAVIER

JOHN F. KENNEDY

LA ASUNCION

### *Dirección*

Via Perimetral

Via a Daule Km. 8,5

**Prolongación** Av. Aeropuerto entrada Sauces 3 y 5

Los Ceibos **calle** 1 ra. # 216

Av. Juan **Tanca** Marengo km. 6,5

Jose **Mascote** 3113 y Argentina

Dolores **Sucre** 401

Via a **la** Costa km. 6.5

Urdesa, Victor Emilio Estrada # 713

Los Esteros **calle** Ira. Mz. 715

Chimborazo 2616 y Venezuela

San Martin **407** y Coronel

Campus Prosperina

Febres **Cordero** y Guerrero Martinez

Av. Domingo Comin **# 205**

**Tulcán** y Chambers

6 de **Marzo** 2206 y Brasil

Chambers y Jose **Mascote**

Av. Juan **Tanca** Marengo km. 2,5

Cdla. FAE, **calle** principal Mz. 25

Jose **Mascote** 117 y Piedrahita

Quito y O'Connor

Urdesa, Circunvalacion 216 y Todos Los Santos

Av. Olmedo **# 410**

Urdesa, Costanera del **Salado** # 637

Alborada **10ma etapa**. Mz. **209** V. 22

Alborada 6ta. **Etapa**, Av. 2da. y Herradura

Via a Salinas km. **20,5**

La Saiba **Mz. Q calle** 6ta.

Via a Salinas km. **5,5**

Tungurahua 707 y **Vélez**

Via a Daule km. 6

## ***Colegio***

LA DOLOROSA  
**Sáenz**

LA INMACULADA

LA MODERNA

LA PROVIDENCIA

LA SAGRADA FAMILIA

LETRAS Y VIDA

**LIBERTADOR BOLIVAR**

LICEO DE FRANCIA

LICEO ITALIANO

LICEO MAHARISHI

LICEO NAVAL

LICEO PANAMERICANO

MARIA AUXILIADORA

**MERCANTIL**

MONS. BERNARDINO ECHEVERRIA RUIZ

**NACIONES UNIDAS**

NUESTRA MADRE DE LA MERCED

NUEVOMUNDO

PARTICULAR CAYETANO TARRUEL

PARTICULAR **GUAYAQUIL**

Pedro Oscar **Salas Bajaña**

PROF. **JAIME FABRE JANSEN**

REPUBLICA DE FRANCIA

**ROSARIO SANCHEZ BRUNO**

SAGRADOS CORAZONES

**SALESIANO CRISTOBAL COLON**

**SAN AGUSTIN**

**SAN FRANCISCO DE ASIS**

**SAN JOSE-LA SALLE**

**SAN JUDAS TADEO**  
Carrera A

**SANTA MARIA GORETTI**

**SANTA MARIANA DE JESUS**

**SANTO DOMINGO DE GUZMAN**

**SpeedWriting**

## ***Dirección***

Cdla. Bolivariana, Av. de los Libertadores y **Manuela Sáenz**

Eloy Alfaro 2101 y Camilo Destruge

Via a **Samborondón** km. 2,5

Eloy Alfaro **1003**

Av. Quito y Francisco Segura

Domingo Comin # **804**

Mapasingue. Av. 3ra. s/n y **calle** 1 ra.

Via **Samborondón** KM. **21/2**

Avenida del **Bombero**

Via **Durán-Yaguachi** km. **5,5**

Av. Pedro J. **Menéndez** Gilbert

Dolores **Sucre 302** y **Nicolás** A. Gonzalez

Francisco de P. Ycaza # 434

6 de **Marzo** 1318 y Pedro Pablo **Gómez**

Via a la Costa km. **5,5**

Via a **Samborondón**, La Puntilla km. 1

Av. Kennedy y la F

Via a **Samborondón** km. 2,5

Cdla. Los Esteros, **calle** Ira. Mz. 715

Panama 204 y Juan Montalvo

**Calle** 27 entre Bolivia y Garcia Goyena

Av. de Las Americas

Cdla. Naval Norte Mz. 4 solar 14

Huancavilca 1917

El Oro# 1219

Rosa Borja de Ycaza # **115** y Maracaibo

Pedro Moncayo 525 y Luis Urdaneta

Carchi **2800** y San Martin

**Tomás** Martinez # **500** y Baquerizo **Moreno**

Alborada 3ra. **Etap**a, Alameda A y Herradura 1 de

Febres **Cordero** **2820**

Chimborazo 1819 y Maldonado

Urdesa, Sto. Domingo de **Guzmán** # **608** y **Ficus**

9 de Octubre 441 y Chimborazo

**Colegio**

Teniente Hugo Ortiz

URDESA

Veinticinco de Julio

VICENTE LEON  
Huancavilca

***Dirección***

Samanes VI etapa

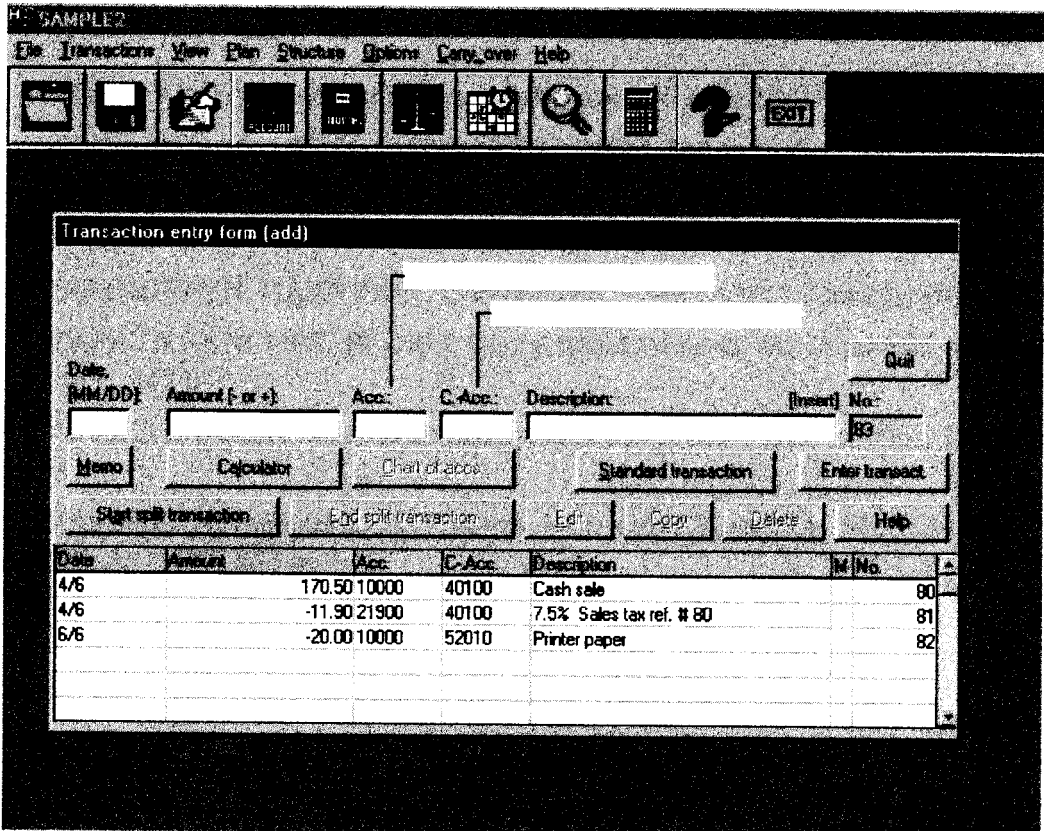
Urdesa calle 4ta. # 409

José de Antepara 2415 y Portete

Pio Montúfar # 1130 entre Capitán Nájera y

# APÉNDICE C.

## Ventana del Programa Account Pro ®





## **BIBLIOGRAFÍA.**

- 1 KENDALL & KENDALL. Análisis y Diseño de Sistemas, Prentice Hall. México, 1997, 9 13 p.
- 2 YOURDON. Análisis Estructurado Modemo, Prentice Hall, México, 1993,735p.
- 3 CARDONA AGUIRRE, JAIRO HERNANDO, Sistemas Contables, MC Graw Hill, Colombia, 1998, 227p.
- 4 IGLESIAS, RUBÉN, Visual FoxPro 5, Alfa Omega, México, 1997, 49 1 p.
- 5 HORNGREEN, HARRISON, Contabilidad, Prentice Hall, México, 1991,641 p.
- 6 ZAPATA SÁNCHEZ, PEDRO, Contabilidad General, MC Graw Hill, Colombia, 1996,347 p.
- 7 PINTER, PINTER, Manual de programación, McGraw Hill, México, 1997, 566 p.
- 8 PRESSMAN, Ingenieria del Software – Un enfoque práctico, McGraw Hill, Espaiia, 1993,823 p.

