

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

45A  
2oj  
j

SISTEMA COTIZADOR DE RAMOS TECNICOS:

INGENIERIA DE SOFTWARE

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A .

ALEJANDRO HERNANDEZ QUEVEDO

MEXICO, D.F. AÑO 1992.

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México

UNAM



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

I N D I C E

<b>CAPITULO I</b>	<b>I N T R O D U C C I O N</b>	<b>5</b>
I.1	Fase de Planeación	6
I.2	Fase de Desarrollo	9
I.3	Fase de Mantenimiento	11
<b>CAPITULO II</b>	<b>P L A N E A C I O N</b>	<b>15</b>
II.1	Planteamiento del Problema	18
II.1.1	Introducción y Objetivos	18
II.1.2	Descripción Funcional	20
II.1.3	Asignación de Funciones	22
II.1.4	Consideraciones y Limitaciones	23
II.2	Análisis de Requerimientos	24
II.2.1	Alcance del Software	24
II.2.2	Recursos	25
II.2.3	Costos	26
II.2.4	Administración	26
II.3	Alternativas de Solución	29
II.3.1	Propuestas	29
II.3.2	Toma de Decisiones	29
II.4	Análisis de Factibilidad	30

<b>CAPITULO III D E S A R R O L L O</b>	<b>31</b>
<b>III.1 Diseño</b>	<b>34</b>
III.1.1 Descripción de Datos	35
III.1.1.1 Flujo de Información	35
III.1.1.2 Estructura de Datos	45
III.1.1.3 Diseño de Bases de Datos	46
III.1.2 Descripción del Proceso	50
III.1.2.1 Estructura del Software	52
III.1.2.2 Estructura de las Bases de Datos	65
III.1.2.3 Lenguaje de Programación	65
<b>III.2 Implementación</b>	<b>68</b>
III.2.1 Bases de Datos	68
III.2.2 Programas y Subprogramas	73
III.2.3 Procesamiento de Datos	103
<b>III.3 Pruebas</b>	<b>122</b>
III.3.1 Unidad	126
III.3.2 Integración	129
III.3.3 Validación	134
<b>CAPITULO IV M A N T E N I M I E N T O</b>	<b>135</b>
IV.1 Introducción	137
IV.2 Bitácora	138
<b>CAPITULO V C O N C L U S I O N E S</b>	<b>141</b>
<b>G L O S A R I O M I N I M O</b>	<b>144</b>
<b>B I B L I O G R A F I A</b>	<b>157</b>

C A P I T U L O   I  
I N T R O D U C C I O N

La década de los 50's y 60's se caracterizan, dentro de la ciencia de la computación, como el desarrollo puro del Hardware.

Durante los 70's ocurre el periodo de transición hacia el desarrollo del Software. En efecto, en los últimos años ha habido un auge considerable en la producción de sistemas para computadora. Paralelamente se han desarrollado un conjunto de técnicas bien definidas que constituyen un método al que se le a dado el nombre de Ingeniería de Sistemas. Este método es un procedimiento sistemático para el desarrollo de cualquier tipo de programa, ya sea que éste tenga un uso específico o general, por lo que nos basaremos en él para el desarrollo del presente trabajo.

En los párrafos siguientes presentamos una breve introducción de lo que es el método de la Ingeniería de Sistemas, comentamos los pasos en la metodología, los resultados que producen cada uno de ellos y el procedimiento para lograrlos.

### I.1 Fase de Planeación

La fase de planeación en la ingeniería de sistemas, figura 1.1, comienza con la planeación del software. Durante este paso nosotros definimos el alcance del paquete, fuentes requeridas para el desarrollo del mismo y, son estimados los costos y su administración. El propósito del paso de la planeación del software es proveernos de una idea preliminar de la viabilidad del proyecto en relación a su costo y su organización. La planeación del software es desarrollada y revisada por la persona encargada del proyecto (ingeniero de sistemas).

# FASE DE PLANEACION

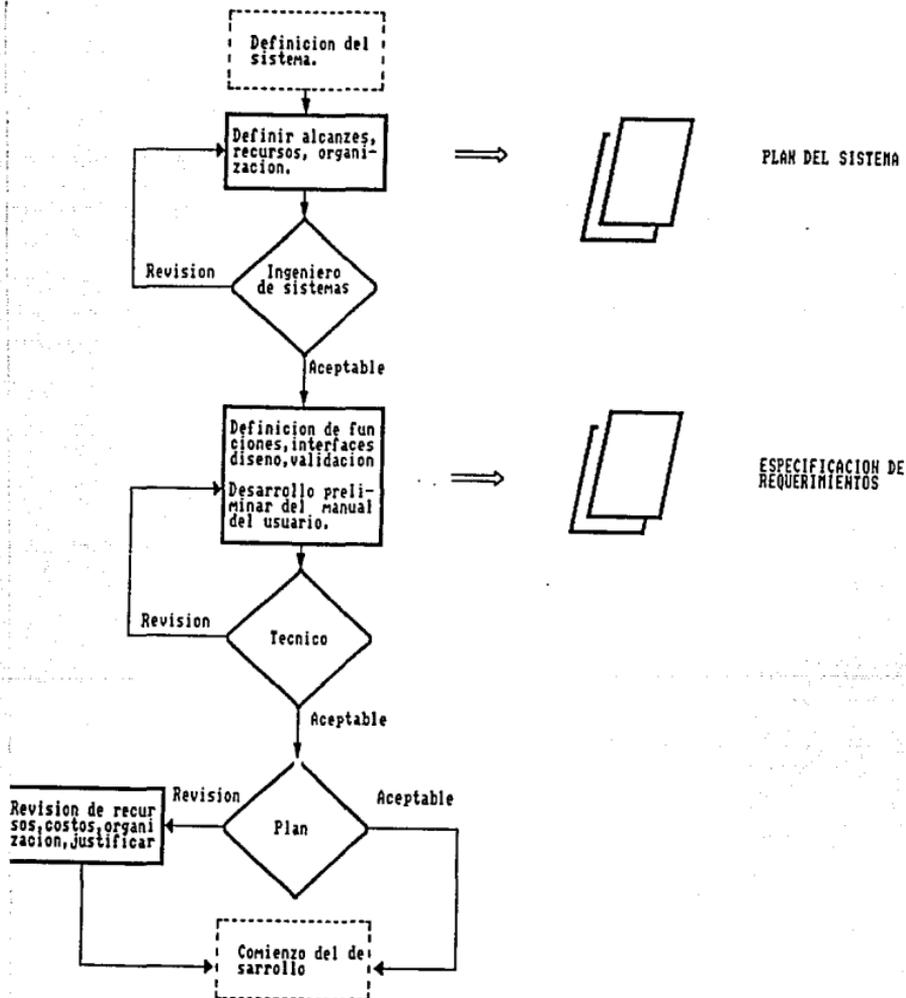


FIGURA 1.1

El próximo paso en la fase de planeación es el análisis y definición de requerimientos. Durante este paso, son definidos en detalle los elementos objetos para el desarrollo del software. El flujo de la información y la estructura definen los elementos de interface y las características funcionales. Los requerimientos y limitaciones son trasladados hacia las características del diseño del paquete. Aquí mismo, es realizado un análisis y una validación de los elementos definidos, criterio que será usado para demostrar que todos los requerimientos son reconocidos y tomados en cuenta.

El análisis y definición de los requerimientos son realizados por el encargado del desarrollo del sistema o el analista (técnico), que es, la persona u organización que cuestiona los requerimientos de un software. Una especificación de los requerimientos del software es una configuración producida como resultado de este paso.

La fase de planeación culmina con una técnica de análisis de la especificación de los requerimientos, efectuada por la persona responsable del desarrollo del mismo, *Ingeniero de Sistemas*, y el analista. Una vez que se han establecido los requerimientos, su alcance, sus fuentes, su costo y se ha identificado su organización en la planeación del software, son reevaluados y corregidos de ser necesario. Información descubierta durante esta revisión puede cambiar radicalmente las consideraciones y estimados definidos previamente. Los resultados de la fase de planeación en la Ingeniería de Sistemas sirven como fundamento para la segunda fase en el desarrollo del software.

## I.2 Fase de Desarrollo

La fase de desarrollo, figura 1.2, traslada un conjunto de requerimientos hacia elementos de un sistema operacional que nosotros llamaremos programa, paquete o software.

El primer paso en la fase de desarrollo se concentra en una aproximación del programa. Este implica el desarrollo de una estructura modular, definición de interfaces y el establecer una estructura de datos. Un diseño heurístico es usado para una evaluación cualitativa del diseño. Este paso de *diseño preliminar* es revisado para completar y satisfacer los requerimientos del sistema. Un primer escrito *Documentación del Diseño* es realizado como parte de la configuración del software.

Los aspectos de procedimiento para cada elemento modular del software son considerados durante el próximo paso en el desarrollo. Las herramientas de diseño son aplicadas para proveer un *Diseño Detallado* de los elementos del programa. Cada procedimiento o módulo es descrito detalladamente y es adicionado a la *Documentación del Diseño* anteriormente mencionado. Es aquí donde obtenemos el diseño base para su posterior codificación.

Finalmente, después de dos pasos de desarrollo del diseño, nosotros tenemos el *código*, que es, la generación de un programa con un apropiado lenguaje de programación. La metodología de la Ingeniería de Sistemas, vé el código como una consecuencia de un buen diseño. Durante el paso de codificación, es realizado un listado del lenguaje fuente para cada elemento modular del software.

# FASE DE DESARROLLO

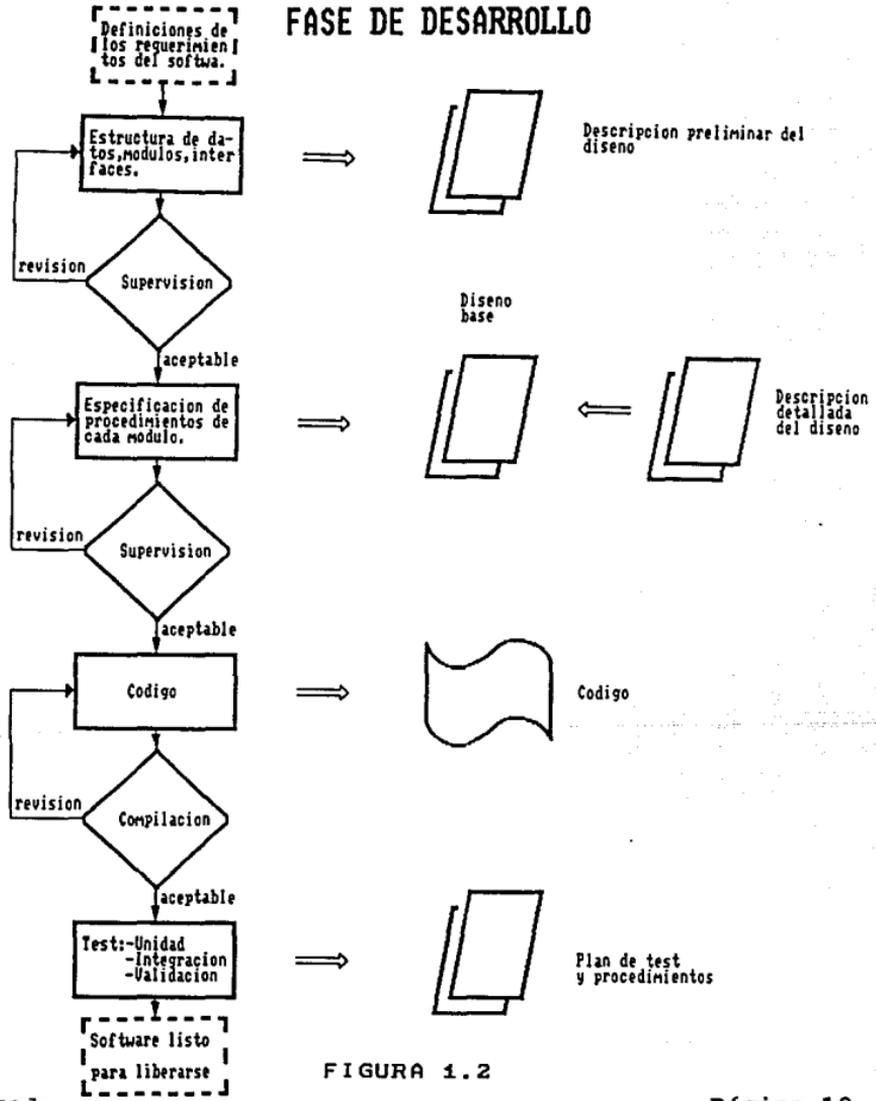


FIGURA 1.2

Al final, los tres pasos de desarrollo son asociados con las pruebas o test del software. Durante estas pruebas se verifican tres aspectos *unidad, integración y validación*. La unidad valida el desarrollo funcional de un componente modular del sistema. La integración nos proporciona información en cuanto a su funcionamiento e interface con cada uno de los módulos que componen el mismo. El proceso de validación verifica que todos los requerimientos sean satisfechos. Un *Plan de Test y Procedimiento* puede ser desarrollado para cada una de las pruebas.

### I.3 Fase de Mantenimiento

Todo software, después de un tiempo de su realización, es modificado, aumentado o corregido, de acuerdo a las necesidades existentes de la persona o empresa que lo utiliza.

La fase de mantenimiento es ilustrada en la figura 1.3, comienza cuando el software es liberado. La revisión de la configuración del paquete es analizada para asegurar que toda la documentación esta disponible y adecuada para la tarea de mantenimiento que continua. La responsabilidad de mantenimiento es establecida y es definido un esquema de reporte por error y sistema de modificación.

La tarea asociada con el mantenimiento del software depende del tipo de mantenimiento que se le realice. En todos los casos la modificación del software incluye la correspondiente configuración, es decir, la modificación de documentos en la fase de planeación y desarrollo, no solamente el del código.

# FASE DE MANTENIMIENTO

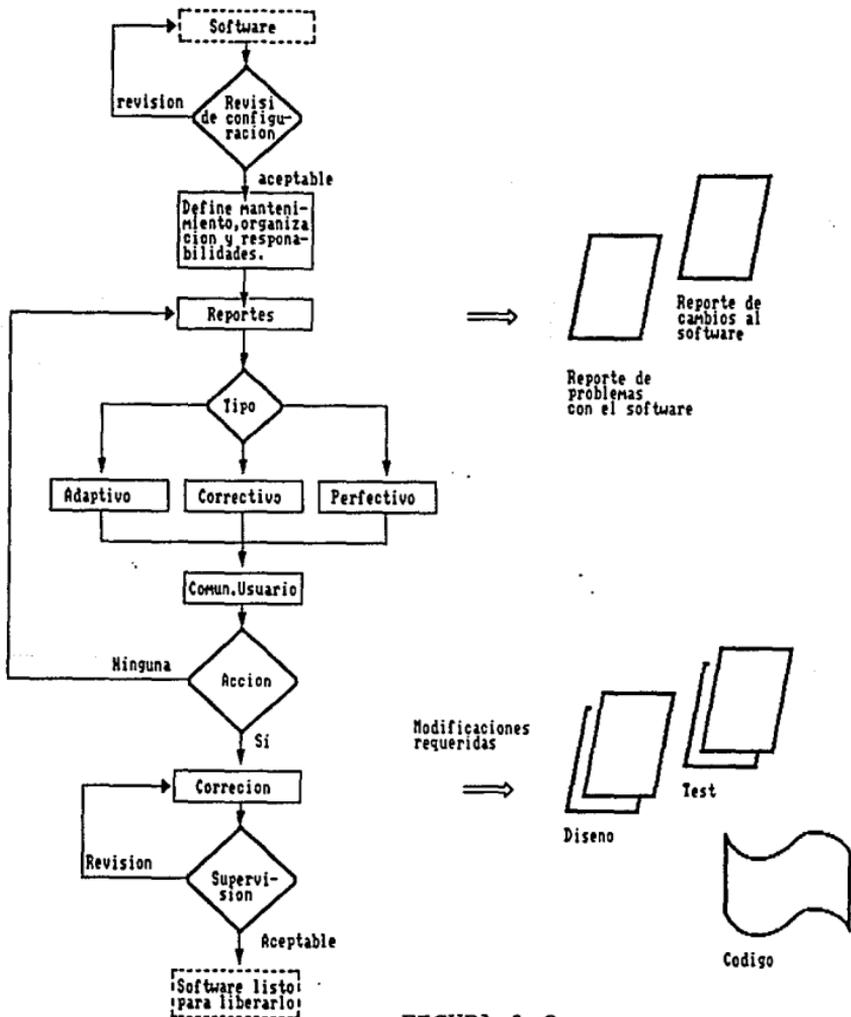


FIGURA 1.3

Como hemos mencionado en los párrafos anteriores, aplicaremos la presente metodología para el desarrollo del software en cuestión, por lo que la organización de la presente tesis se estructura de la siguiente manera.

En el transcurso del segundo capítulo, trataremos la Fase de Planeación, dentro de éste plantearemos la problemática que se desea resolver por medio de un software y consecuentemente, describiremos los objetivos por los cuales se pretende la solución. Posteriormente haremos un análisis de los requerimientos que se deben satisfacer, se propondrán alternativas de solución y finalmente, en cuanto hayamos tomado la mejor solución, efectuaremos un estudio de factibilidad que nos permita saber en cuanto tiempo, de que manera y bajo que condiciones o alcances podremos realizar nuestro software.

Tomada la decisión anterior, comenzaremos el diseño del software. La realización de los diagramas de flujo de datos, su almacenamiento y manejo serán detallados en el tercer capítulo. Aquí mismo y posterior al diseño del sistema, se lista el código fuente de uno de los módulos y sus subprogramas. Al finalizar este capítulo, se ejemplifica la ejecución del software y se verifica mediante una prueba general.

La fase de mantenimiento, expuesta en el capítulo cuarto, es teórica debido a que el software a sido recientemente liberado. Así pues, será necesario que transcurra algún tiempo antes de modificarlo para un uso no considerado en los requerimientos iniciales. Sin embargo, se anexa una hoja técnica que nos permitirá hacer anotaciones sobre las alteraciones que se le deben realizar en cuanto se desida la modificación.

En el quinto capítulo, se exponen las conclusiones sobre los objetivos alcanzados en la presente tesis.

Finalmente, presentamos un glosario mínimo. Este con el fin de definir los conceptos básicos aplicados en los capítulos que lo preceden.

C A P I T U L O    I I

P L A N E A C I O N

La fase de planeación en el ciclo de vida del software es un proceso de definición, análisis, especificación, estimación y revisión. Esta fase la podemos resumir en el diagrama de la figura 2.1.

La definición del sistema, es el primer paso para la fase de planeación y un elemento del proceso de ingeniería de sistemas descrito en el capítulo I. En este punto, la atención es enfocada hacia el sistema como un todo. Las funciones presentes son localizadas como un conjunto de hardware, software y otros elementos basados en un entendimiento preliminar de los requerimientos.

Durante el desarrollo del primer inciso del presente capítulo, definiremos al sistema mediante el reconocimiento de las necesidades de la empresa para la cual vamos a desarrollar el software; descripción de interfaces, funciones y procesos; análisis y diseño del proceso existente; reconocimiento de funciones para hardware, software y elementos suplementarios; costos y organización; darán la pauta para tal definición.

El siguiente paso en la fase de planeación del software, detallado posteriormente en los puntos 2 y 3 de este capítulo, es la realización del plan. El plan del software tiene como objetivo principal estimar costos y desarrollar la organización para los elementos del software de un sistema, mediante la definición de los requerimientos.

En el punto 4, veremos el último paso en la fase de planeación, analizaremos los requerimientos propuestos en el plan del software en cuanto a su factibilidad. Una especificación detallada de los requerimientos del software forma la base para el paso del desarrollo en la ingeniería del sistemas.

Fase de Planeación

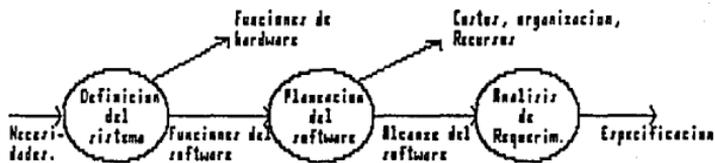


FIGURA 2.1

## II.1 Planteamiento del Problema

### II.1.1 Introducción y objetivos

El proceso actual para la valuación del costo de una póliza, dentro del ambiente de las compañías aseguradoras, se lleva a cabo de una forma manual. Este proceso, definido como proceso de cotización, comienza con la solicitud de la misma en alguna sucursal de la compañía aseguradora, proporcionada previamente por el corredor, broker o agente de seguros.

Existe una variada gama de formas de la solicitud para cotización, el tipo que se elija dependerá de las necesidades del cliente o asegurado para la debida protección de los riesgos eventuales. Entre los tipos de datos requeridos por las solicitudes podemos destacar los más importantes para el efecto de cotización, como son, la relación de los bienes objeto del seguro; cantidad o monto total en monedas, que en adelante haremos alusión como Suma Asegurada; ubicación del riesgo para los diferentes bienes; coberturas adicionales que amparamos dentro de la misma póliza, responsabilidad civil, huelgas y alborotos populares, huracan y granizo, explosión física, terremoto, etc. Dependiendo de la apreciación de riesgo y de las políticas preestablecidas para las oficinas de servicio deciden si debe cotizarse en la misma o si debe enviarse a oficina matriz para su debida autorización.

Si la cotización es por parte de la oficina de servicio, se procede a efectuar el estudio mediante los datos proporcionados en la solicitud y el correspondiente manual de cotización o guía de tarificación. El tiempo empleado para realizar una cotización es variado, depende de la cantidad de equipo por asegurar y de la habilidad del cotizador, sin embargo, consideramos que el tiempo promedio entre cotización y estudio es de aproximadamente cuatro horas. Si las 10 oficinas de servicio en el Distrito Federal atienden alrededor de 50 casos en cada una de ellas diariamente y que, las 18 oficinas foráneas dan servicio a 20 agentes por día, estimamos que son 143 horas hombre por día para efectos de cotización. Por esta razón se requiere de la automatización del proceso.

Dentro de esta enorme estructura debemos definir, en primera instancia, el alcance de nuestro proyecto. Pretendemos que nuestro sistema realice la cotización del equipo, maquinaria y bienes que sean considerados como producto o auxilio de la ciencia de la ingeniería, ya sea química, civil, mecánica, eléctrica, electrónica, etc. En el ambiente de los seguros los bienes o equipos relacionados con la ingeniería se subdivide en seis ramos, equipo electrónico, calderas y recipientes sujetos a presión, maquinaria, obras civiles, obras de montaje y equipo de contratistas.

De una forma clara, listaremos los objetivos implícitos en el desarrollo del software especificado, es decir, los objetivos del software como un subsistema dentro de la estructura:

- a) Mejorar el servicio de las oficinas filiales en cuanto a tiempo de respuesta y presentación de estudios.
- b) Reducir errores de cotización.
- c) Facilitar la cotización de los ramos técnicos en las oficinas de servicio.
- d) Reducir costo de comunicación (teléfono, correo electrónico, fax, mensajería) por consultas a oficina matriz.
- e) Homogenizar las cotizaciones entregadas por oficinas filiales.

### II.1.2 Descripción Funcional

Como se mencionó anteriormente, nuestro software contendrá las seis funciones básicas de cotización para los ramos de ingeniería. La información mínima para cotizar, el proceso y su resultado final, depende del ramo seleccionado.

En cuanto al ramo de calderas y aparatos sujetos a presión se consideran riesgos cubiertos: Explosión Física, Quemaduras por insuficiencia de agua, Deformación súbita y Agrietamiento de maquinaria entre otras. La información mínima para cotizar contendría el giro del negocio, ubicación, marca, capacidad, suma asegurada por equipo, tipo, edad, coberturas y vigencia.

La póliza de equipo electrónico nos ampara los riesgos de incendio, rayo, explosión, implosión, humo, gases, inundación, impericia, granizo y otros accidentes no excluidos. La información necesaria comprende el giro del negocio, ubicación, descripción del equipo, suma asegurada, vigencia, cúmulo máximo por ubicación y límites.

Rotura de maquinaria ampara la impericia, negligencia, descuido, montaje incorrecto, introducción de cuerpos extraños y otros accidentes no excluidos. La información para cotizar consiste en el giro del negocio, ubicación, descripción del equipo, suma asegurada, equipo de mayor valor y edad.

Por otra parte, equipo de contratistas avala incendio, huracán, explosión, colisión, hundimiento, inundación, terremoto, caída, derrumbes, robo y huelgas. En cuanto a información se refiere es necesario la razón social, la ubicación del riesgo, descripción del equipo, suma asegurada, cúmulo máximo, edad y vigencia.

En cuanto a la póliza de montaje cubre incendio, helada, hundimiento, robo y otros accidentes no excluidos. La información mínima que se requiere para su estudio es la suma asegurada, ubicación, descripción del equipo, duración del montaje, duración del período de pruebas, descripción del edificio, colindates y fuentes posibles de inundación.

Finalmente, obra civil ampara incendio, rayo, explosión, robo, negligencia, impericia, materiales defectuosos y otros accidentes. La información para su estudio consiste en suma asegurada, avance de la obra, ubicación, programa de obra, programa de inversión, altura, sótanos, subsuelo, capa freática, duración, colindates y posible fuentes de inundación.

Dependiendo de la anterior información y del ramo que el cliente desea contratar nos dirigimos a la tabla de tarificación correspondiente, se asignan los diferentes recargo y descuentos que apliquen de acuerdo a una inspección o experiencia del cotizador. Este mismo proceso se realiza para cada uno de los equipos. Posteriormente, mediante la ayuda de una microcomputadora, se calcula la cuota promedio, y se aplica sobre la suma asegurada total, dandonos como resultado la prima o costo del seguro para ese equipo en especial.

### II.1.3 Asignación de Funciones

Como hemos podido advertir en el inciso anterior, son seis las funciones básicas para la cotización de los seguros de ingeniería, cada una de estas funciones nos representará un elemento de nuestro software. Las tablas de tarificación, auxiliares en el estudio de la cotización, las manejaremos como bases de datos, una por cada función y, una por cada cobertura solicitada. Más adelante detallaremos cada uno de estos elementos, ahora simplemente bastará con tenerlos en mente.

#### II.1.4 Consideraciones y limitaciones

Como es de nuestro conocimiento, se pretende que el software descrito, sea utilizado en todas las oficinas, tanto locales como foráneas. Pero además, debido a concesiones o por promoción, se facilitará a los agentes, los cuales tienen un contacto estrecho con el asegurado. Debido a esta consideración, nuestro software debe ser flexible, es decir, ser compatible y transportable a varias computadoras, por esta simple razón estaremos limitados al uso de la computadora personal que tenga mayor auge en el mercado (IBM PC y compatibles). El modelo más utilizado dentro de la empresa es la Olivetti M220 y M240, es importante destacarlo ya que el modelo nos representa la configuración, características y potencial de la máquina.

Por otro lado y debido a que un gran número de personas utilizaran este software, deberemos diseñarlo de tal forma que sea fácilmente entedible, inclusive para aquella población que no tiene los conocimientos básicos de la computadora y su uso.

Una consideración final, nuestro software debe de estar estructurado de tal forma que lleve una lógica y secuencia similar a las de las formas de la solicitud o cuestionarios relacionados, esto es con el fin de facilitar el reconocimiento en cuanto a la captura de la información.

## II.2 Análisis de Requerimientos

### II.2.1 Alcance del Software

Dentro de este ámbito, dejaremos claros los objetivos de nuestro software, como una forma de representar los requerimientos del mismo y que nos dan la pauta para el análisis de factibilidad, el último paso en la fase de planeación.

- a) Al término de su realización, el software será capaz de efectuar la cotización de los seguros de ingeniería.
- b) Los estudios presentados por el software deberán estar basados en las tarifas autorizadas para la empresa que será desarrollado.
- c) El software deberá cumplir y reconocer las políticas particulares de cada ramo preestablecidas para las oficinas de servicio.
- d) Los reportes de cotización, deberán ser estructurados de tal forma que se facilite su entendimiento y la expedición de la póliza correspondiente.
- e) La organización de la información mínima para cotizar deberá ser de tal forma que se facilite su captura.
- f) El software contará con ayudas para facilitar su manejo.
- g) El software será capaz de validar las entradas de datos en todo momento.
- h) El reporte deberá de tener la facilidad de presentarlo con diferentes opciones de deducible.

- i) El software será capaz de calcular las coberturas adicionales de cada uno de los ramos de ingeniería.
- j) El software tendrá la capacidad de calcular las diferentes modalidades (equipos individuales o por grupo) de los ramos de ingeniería.

### II.2.2 Recursos

#### a) Humanos:

Debido a que la magnitud del proyecto es relativamente pequeño, hemos considerado que todos los pasos de la vida del software pueden ser realizados por una persona y que sea asesorada por especialistas para el ramo que así lo requiera.

#### b) Hardware:

Para el desarrollo del presente programa nos basaremos, como ya hemos mencionado, en una computadora personal, en específico en una Olivetti M240, bajo la siguiente configuración.

- CPU 8086, Coprocesador 8087, ROM 32 KB, RAM 640 KB, diskette de 5.25", HDU 3.5" (20 MB), Controlador OGC, interface paralela Centronics y serie RS232C.
- Video color normal, pantalla 14", resolución 640x400, con fuente de alimentación incorporada.
- Teclado, 12 teclas de función, teclado numérico, alfanumérico y de desplazamiento del cursor.
- Impresora matricial, modelo ATI3000A.

## c) Software:

Mencionaremos aquí el software necesario para el desarrollo de nuestro programa:

- Sistema operativo MS-DOS versión 3.20.
- Clipper Compiler versión 1.0 por Nantucket Corp.
- Helpsys versión 1.0 por FSP Enterprises.
- Plink86 Plus versión 2.21 por Nantucket.
- Popcalc versión 1.5 por Popular Programs Inc.
- SPF/PC1 versión 02.00 por Command Techn. Corp.
- Flow Charting II+ ver.2.40B Patton Soft. Corp.
- Side Kick Ver. 1.56A Borland Inc.
- Quattro Ver. 1.0 Borland Inc.
- Dbase III+ Ver. 1.0 Ashton-Tate.

## II.2.3 Costos

Existe una gran variedad de métodos, desde manuales hasta automáticos, para la estimación de los costos del software. Para la estimación de costos de nuestro paquete, nos basaremos en la *Técnica de Esfuerzo por Tarea*, esta técnica es la más común en el desarrollo de algún proyecto de ingeniería. Aplicando la técnica mencionada, obtenemos los resultados presentados en la figura 2.2.

## II.2.4 Administración

Siguiendo la misma filosofía del punto anterior, consideramos y en primera instancia aplicamos los principios de la regla 40-20-40 para la administración o distribución del tiempo a cada función del software. Finalmente obtenemos el programa de administración del tiempo representado en la figura 2.3.

SISTEMA COTIZADOR DE RAMOS TECNICOS GRUPO NACIONAL PROVINCIAL

ESTUDIO DE COSTOS

MARZO 1989

FUNCIÓN	PLANEACION	DISEÑO	CODIGO	TEST	TOTALES(*)
Pantallas	0.4000	0.2000	0.1000	0.3000	1.00
Eq. Electronico	0.1000	0.0500	0.0250	0.0750	0.25
Elec. Adicionales	0.6000	0.3000	0.1500	0.4500	1.50
Calderas	0.1000	0.0500	0.0250	0.0750	0.25
Cald. Adicionales	0.4000	0.2000	0.1000	0.3000	1.00
Montaje	0.2000	0.1000	0.0500	0.1500	0.50
Mont. Adicionales	1.2000	0.6000	0.3000	0.9000	3.00
Obra Civil	0.1000	0.0500	0.0250	0.0750	0.25
O.C. Adicionales	0.8000	0.4000	0.2000	0.6000	2.00
Eq. Contratistas	0.1000	0.0500	0.0250	0.0750	0.25
Contr. Adicionales	0.3000	0.1500	0.0750	0.2250	0.75
Maquinaria	0.4000	0.2000	0.1000	0.3000	1.00
Maq. Adicionales	1.2000	0.6000	0.3000	0.9000	3.00
Reporte	0.2000	0.1000	0.0500	0.1500	0.50
Ayudas	1.3500	0.6750	0.3375	1.0125	3.38
SUBTOTALES (*)	7.4500	3.7250	1.8625	5.5875	18.63
COSTO (+)	666.67	333.33	166.67	500.00	
T O T A L E S:	4,966.69	1,241.65	310.42	2,793.75	9,312.52

NOTAS : (\*) Estimados en meses por persona.

(+) En USCy por mes.

FIGURA 2.2

## TABLA DE ADMINISTRACION

FUNCION	MAY				JUN				JUL				AGO				SEP				OCT				NOV				DIC				ENE							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1. Pantallas	■																																							
2. Eq. Electrónico					■																																			
3. Elec. Adicionale									■																															
4. Calderas					■																																			
5. Cald. Adicionale						■																																		
6. Montaje					■																																			
7. Mont. Adicionale													■																											
8. Obra Civil					■																																			
9. O.C. Adicionales																	■																							
10. Eq. Contratista					■																																			
11. Cont. Adicional									■																															
12. Maquinaria					■																																			
13. Maq. Adicionale																					■																			
14. Reporte																																	■							
15. Ayudas									■	■	■										■												■							
TOTALES (*)	1	1	1	1	6	3	2	2	4	2	3	1	2	2	1	1	1	1	1	1	2	2	3	3	3	3	4	4	1	1	1	1	1	1	1	1	2	2	3	3
ACUMULADO (*)	1	3	10	15	21	26	29	32	34	37	42	48	55	60	62	64	67	72																						
	2	4	13	17	23	27	31	33	35	39	45	51	59	61	63	65	69	75																						

Figura 2.3

(\*) NOTA: En semanas por hombre.

### II.3 Alternativas de solución

#### II.3.1 Propuestas

Las alternativas de solución para nuestro problema son un tanto obvias, pero vale la pena nombrarlas para tenerlas como referencia y apoyo a la solución óptima.

- a) Continuar con el sistema tradicional, cotización manual.
- b) Desarrollar e implementar el sistema propuesto durante el presente plan.
- c) Adquirir el software existente de los reaseguradores.

#### II.3.2 Toma de Decisiones

Al igual que las propuestas de solución, intuimos de antemano la solución óptima de las expuestas anteriormente.

Dejar el proceso en un sistema tradicional, implicaría la pérdida prematura de beneficios, contra lo cual se ha venido luchando. Por otro lado, al adquirir el producto de los reaseguradores, nos daría una base para la cotización, pero no sería correcto distribuirlo a las oficinas de servicio y a los agentes, puesto que este paquete no cumple con las normas y políticas establecidas por la empresa.

La solución es implementar un software planeado para cumplir con las políticas y requerimientos específicos de la empresa. La decisión final es de la compañía, basándose en los costos y tiempo de solución de su problema.

## II.4 Análisis de Factibilidad

Durante el desarrollo de la fase de planeación de nuestro software, hemos obtenido las tablas o programas de costos o inversión y el programa de administración o de avance, analizando estas tablas podemos discernir si es factible el desarrollo del sistema en cuanto a costo y tiempo se refiere.

Del estudio de costos, figura 2.2, podemos observar que nuestro esfuerzo teórico para el desarrollo del sistema es de aproximadamente 19 meses, con un costo total de 9,313 dolares. Esta misma tabla nos representa por función el tiempo empleado para cada fase de desarrollo en la Ingeniería de Sistemas y el tiempo total empleado en cada una de ellas para el desarrollo de nuestro software.

El estudio de administración del tiempo, figura 2.3, nos representa el tiempo real empleado en el desarrollo del paquete, 9 meses, obtenidos en base al tiempo teórico total empleado (75 semanas) y la similitud de los procesos realizados o por su facilidad de combinarlos. La cuarta semana de junio, por ejemplo, observamos que estamos desarrollando dos subprogramas, el que realiza la función de cotizar el ramo de rotura de maquinaria y el que cotiza los riesgos adicionales de calderas. En el primero se esta verificando la prueba mientras que el segundo esta en la etapa de codificación.

Con la información anterior y habiendo definido el alcance y requerimientos de nuestro software, concluimos que podemos proseguir con la siguiente fase en la Ingeniería de Sistemas, diseño e implementación del software.

C A P I T U L O    I I I

D E S A R R O L L O

La fase central en el ciclo de vida del software es el desarrollo. La fase de desarrollo comienza cuando se han establecido los requerimientos del software y se resume en cuatro pasos básicos.

- Diseño Preliminar.
- Diseño detallado.
- Código.
- Pruebas

Cada paso transforma la información, dando como resultado final el software validado.

El flujo de información durante la fase de desarrollo es ilustrado en la figura 3.1. Los requerimientos del software y su estructura son analizados en el primer paso, diseño preliminar. Con el uso de un número de metodologías de diseño la *Estructura del Software* es desarrollada. La estructura del software, también llamada arquitectura, define las relaciones entre los elementos del programa. El diseño detallado transforma los elementos estructurales en una descripción del procedimiento del software. El código fuente es generado, y una prueba preliminar es el conducto entre el código y el paso de la prueba de unidad. El paso final en la fase de desarrollo es el detallado de las pruebas de integración y validación.

## Fase de Desarrollo

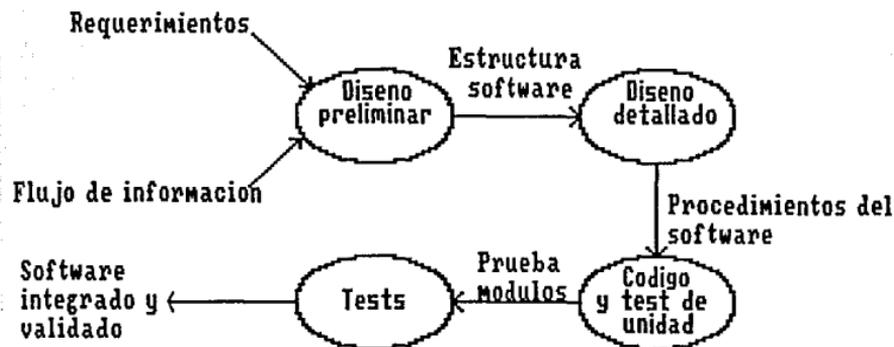


FIGURA 3.1

En el transcurso del primer inciso del presente capítulo, realizaremos los dos primeros pasos en la fase de desarrollo, definiremos, en primera instancia y de forma global, el proceso del sistema, posteriormente detallaremos cada función del sistema hasta llegar a un entendimiento accesible del mismo.

En el segundo inciso presentamos un ejemplo del código fuente como resultado del paso anterior. A manera de ilustración presentamos una de las subrutinas utilizadas por el software, no incluimos las demás debido a que no consideramos necesario hacerlo. Al final del mismo realizamos un ejemplo completo de cotización para verificar la prueba de unidad.

Como último procedimiento, inciso final de este capítulo, en la fase de desarrollo, efectuamos las pruebas de integración y validación del software para su posterior liberación.

### III.1 Diseño

Continuando con nuestro diagrama de flujo para la fase de desarrollo en la ingeniería de sistemas, contemplaremos primeramente el paso del diseño. El diseño del software es un proceso a través del cual los requerimientos son trasladados a representaciones del software. Inicialmente, tenemos una vista de lo que será nuestro software, subsecuentemente, mediante técnicas de refinamiento para la representación del diseño llegamos a la definición de la arquitectura del software. El flujo de eventos durante el diseño del software lo podemos dividir en dos aspectos, el *diseño preliminar*, que será desarrollado en el primer inciso de este subcapítulo y el *diseño detallado*, que será analizado en el segundo inciso.

## III.1.1 Descripción de Datos

Durante el presente capítulo mostraremos los diagramas del diseño preliminar del software, el flujo de información es representado en varios esquemas, comenzando con uno muy básico hasta llegar al detalle del sistema.

El segundo punto, toca el entendimiento preliminar de las estructuras de datos que serán utilizadas en el software. Como veremos en este subcapítulo, debido a la sencillez de las estructuras de datos utilizadas en el software, no será necesario un desarrollo más exhaustivo del mismo.

En el último subcapítulo veremos el desarrollo de la estructura lógica de las bases de datos. En capítulos posteriores, la representación de su estructura física y su implementación para su utilización dentro del software.

## III.1.1.1 Flujo de Información.

Una técnica para representar el *modelo fundamental del sistema* es ilustrado en la figura 3.2. Toda función del sistema es representada como una sencilla caja negra, *transformador de información* que es representado por un círculo en la figura. Una o varias entradas son ilustradas como flechas etiquetadas, la información es manejada y transformada para producir salidas. Como se puede intuir, el modelo puede ser aplicado al sistema en su totalidad o solamente al software. El *diagrama de flujo de datos* tiene tres atributos que son importantes destacar:

## Modelo fundamental del sistema



Figura 3.2

- El flujo de información en cualquier sistema (manual, automático o híbrido) puede ser representado.
- Cada círculo, en el diagrama, puede requerir de un refinamiento para la comprensión del mismo.
- El flujo de datos, que representa el flujo de control, es enfatizado.

La información es representada por una flecha etiquetada. El proceso es representado por un círculo propiamente descrito. Fuentes y destinos son ilustrados como cuadros etiquetados y el almacenamiento de información es representado por una doble línea horizontal.

Con fundamento en los párrafos anteriores comenzaremos representando nuestro software como una caja negra, ilustrada en la figura 3.3. El suscriptor de los seguros de ingeniería o la persona a cargo de las cotizaciones para este ramo nos representa la fuente de la información. La información proporcionada por esta persona es digerida y transformada por el sistema cotizador, la comunicación entre la persona y el sistema será auxiliada mediante pantallas para la representación de opciones y ventanas de ayuda. Es importante recordar que el sistema cuenta con un archivo que contiene las tablas de cotización para cada uno de los ramos considerados. Al término del proceso de la información, obtendremos un reporte el cual será proporcionado al agente de seguros o propiamente al asegurado.

### Flujo de Información

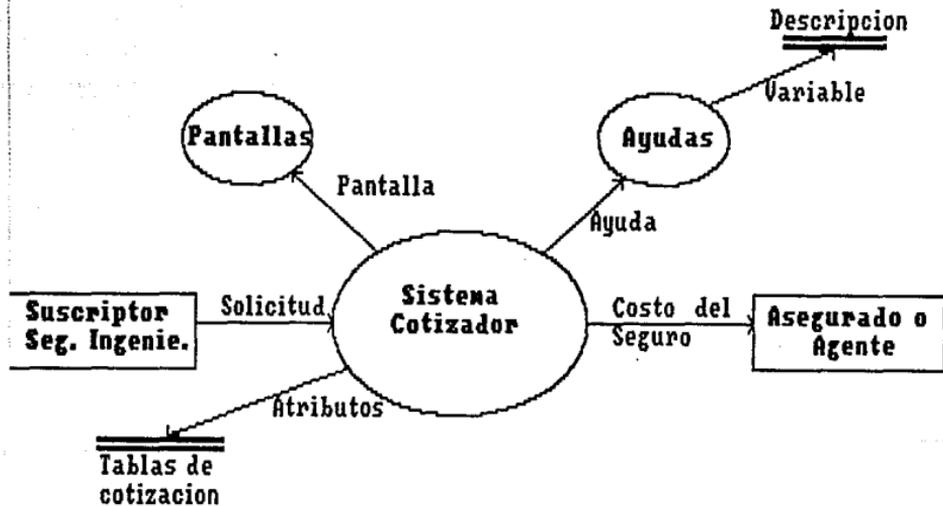


Figura 3.3

Una vez definido nuestro modelo fundamental del sistema y teniendo en mente los requerimientos previamente establecidos procedemos a su detalle. En primer instancia nuestro modelo debe contar con un subsistema de recepción, que como su nombre lo sugiere prepara al usuario e inicializa funciones. Posteriormente, como recordaremos, el sistema se divide en seis tipos de cotizaciones, dependiendo de las necesidades del usuario se realizará el estudio correspondiente. Para discernir entre el tipo de cotización solicitado podemos definir un nuevo subsistema entre el de recepción y el de la cotización correspondiente.

Cada una de las cotizaciones son diferentes e independientes entre sí, sin embargo, el proceso de cotización es similar para cada uno de los ramos de ingeniería. Es decir, el subsistema que continua en nuestro modelo debe ser capaz de identificar el tipo de maquinaria para el cálculo de la cuota básica, a ésta última deben aplicarse recargos y descuentos que apliquen, según sea el caso, lo que podemos representar mediante otro subsistema. Posteriormente se calculan las coberturas adicionales y finalmente se genera el reporte especificado, estos dos últimos procesos pueden ser representados independientes para su mejor entendimiento.

Realizando ésta serie de transformaciones al modelo de la figura 3.3, obtenemos el esquema ilustrado en la figura 3.4. Esta nos muestra los detalles del software, cada una de las funciones por las que la información proporcionada por el usuario debe realizar para obtener el reporte final. El proceso comienza en la determinación del ramo de ingeniería que se desea cotizar, posteriormente se identifica el tipo de giro del negocio, esto es en los ramos de equipo electrónico y rotura de maquinaria; en los ramos de equipo de contratistas y calderas el proceso es identificar el tipo de aparato; y, en el de montaje es identificar si el montaje es de una sola máquina o de un conjunto de ellas.

La siguiente función en el ramo de equipo de contratistas, es el cotizar la maquinaria. Dependiendo de la máquina se obtiene una cifra, la que nos representa la cuota básica, la siguiente función efectúa el cálculo de los riesgos adicionales, posteriormente se agregan los descuentos que procedan y finalmente se obtiene el reporte. De una forma similar se tiene el procedimiento para el cálculo de la cuota básica de calderas, con la excepción de que los descuentos son aplicados antes de los riesgos adicionales y solamente para las calderas. El cálculo de la cuota básica de montaje es similar a las dos anteriores, pero no se aplican descuentos.

El proceso de las funciones para equipo electrónico, obra civil y rotura de maquinaria, son similares en cuanto a su secuencia. Identificando la obra o el giro del negocio, según el ramo, se agregan los descuentos y se calculan los riesgos adicionales, finalmente se obtiene el reporte para cada ramo.

## Modelo Refinado del Flujo de Informacion

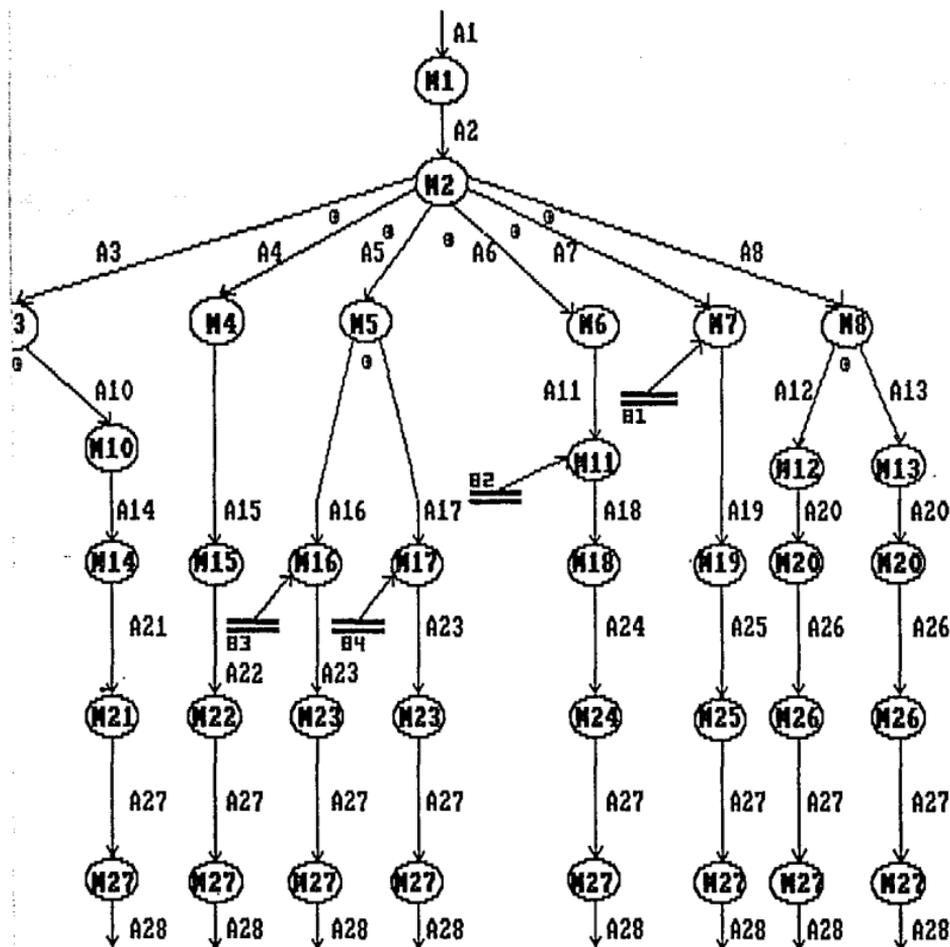
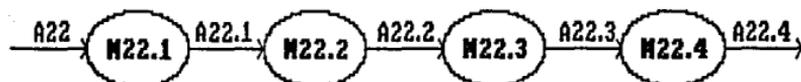
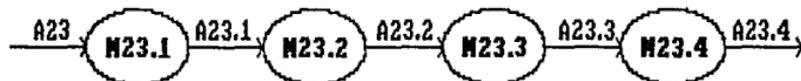


Figura 3.4

## Refinado de Funciones



a) Coberturas Adicionales para el ramo de Obra Civil en Construcción



b) Coberturas Adicionales para el ramo de Montaje



c) Coberturas Adicionales para el ramo de Equipo Electronico

Figura 3.5

**Descripción del Etiquetado para el  
Modelo Refinado del Flujo de Información**

A1: Solicitud	M1: Resepción al Sistema
A2: Solicita Cotización	M2: Define Cotización
A3: Ramo Calderas	M3: Tipo de Equipo
A4: Ramo de Obra Civil	M4: Identifica Obra
A5: Ramo de Montaje	M5: Define Tipo Montaje
A6: Ramo de Maquinaria	M6: Identifica Giro Mqui
A7: Ramo de Eq. Electrónico	M7: Identifica Giro Elec
A8: Ramo de Eq. Contratista	M8: Define Equipo
A9: Recipientes	M9: Identifica Recip.
A10: Calderas	M10: Identifica Caldera
A11: Giro	M11: Identifica Maquina
A12: Uso Específico	M12: Identifica Especific.
A13: Uso General	M13: Identifica General
A14: Cta. Básica Calderas	M14: Descuento Calderas
A15: Cta. Básica Obra Civil	M15: Descuentos Obr. Civ.
A16: Montaje Plantas	M16: Iden. Giro Mon. Planta
A17: Montaje Maquinaria	M17: Iden. Giro Mon. Maq.
A18: Cta. Básica Maquinaria	M18: Descuentos Máquinas
A19: Cta. Básica Eq. Elec.	M19: Descuentos Eq. Elec.
A20: Cta. Básica Contratistas	M20: Descuentos Contrat.
A21: Cta. Básica Calderas	M21: Ries. Adic. Calderas
A22: Cta y Desc Obra Civil	M22: Ries. Adic. Obra Civil
A23: Cta. Básica Montaje	M23: Ries. Adic. Montaje
A24: Cta y Desc Maquinaria	M24: Ries. Adic. Maquinaria
A25: Cta y Desc Eq. Elec.	M25: Ries. Adic. Eq. Elec.
A26: Cta y Desc Contratistas	M26: Ries. Adic. Eq. Cont.
A27: Cta, Desc, Adici. por Ramo	M27: Reporte por Ramo
A28: Costo Seguro por Ramo	

**Figuras 3.4 y 3.5  
(Anexo)**

**Descripción del Etiquetado para el  
Modelo Refinado del Flujo de Información  
( Continua )**

A22.1:Cta.Terr.Obra Civil	M22.1:Calc.Terr.Obra Civ
A22.2:Cta.Huracán Ob.Civ.	M22.2:Calc.Hur.Obra Civ.
A22.3:Cta.R.C.Obra Civil	M22.3:Calc.R.C.Obra Civ.
A22.4:Cta.Remoción Ob.Civ.	M22.4:Calc.Rem.Obra Civ.

A23.1:Cta.Terr.Montaje	M23.1:Calc.Terr.Montaje
A23.2:Cta.Huracán Montaje	M23.2:Calc.Hura.Montaje
A23.3:Cta.R.C.Montaje	M23.3:Calc.R.C.Montaje
A23.4:Cta.Remoción Montaje	M23.4:Calc.Rem.Montaje

A25.1:Cta.Terr.Eq.Elec.	M25.1:Calc.Terr.Eq.Elec.
A25.2:Cta.Huracán Eq.Elec.	M25.2:Calc.Hura.Eq.Elec.
A25.3:Cta.Eq.Móvil	M25.3:Calc.Eq.Móvil
A25.4:Cta.Huelgas	M25.4:Calc.Huelgas
A25.5:Cta.Hurto	M25.5:Calc.Hurto

B1:Tarifa de Equipo Elect.	B2:Tarifa de Rot. de Maq
B3:Tarifa de Montaje Plantas	B4:Tarifa de Mont.Maquin

**Figuras 3.4 y 3.5  
(Anexo)**

El detalle mostrado en la figura 3.4, es suficiente para el entendimiento del software a desarrollar, sin embargo, es necesario hacer un detalle más exhaustivo de las coberturas adicionales de obra civil, montaje y equipo electrónico. Estos detalles son ilustrados en la figura 3.5

Como se observa en esta figura, el cálculo de las coberturas adicionales son auxiliadas por las funciones que realizan el cómputo de cada una de ellas, cosa que no ocurría en otros ramos.

#### III.1.1.2 Estructura de Datos

Una estructura de datos en su forma general consiste de una colección de nodos o registros que mantienen una relación importante entre sí. El nodo es el elemento básico para mantener la información en una estructura de datos. Para representar la información contenida en un nodo se pueden usar una o más palabras de computadora dependiendo de las características de los datos. Un nodo puede subdividirse en campos de tal manera que sea fácil la manipulación de la información.

Las estructuras de datos utilizadas en la ciencia de la computación se subdividen en dos ramos para su estudio: *Estructuras de datos elementales*, aquellos datos (números enteros, números reales, caracteres, arreglos y otros) cuya manipulación y representación se han estandarizado en los lenguajes de programación, y *Estructuras de datos compuestas* (pilas, colas, gráficas y otros) cuya manipulación y representación requieren del ingenio de los usuarios.

Debido a que la representación de la estructura de datos más complicada es de tipo matricial; el cálculo de las coberturas adicionales, así como el de la cuota básica extraída de tablas, es realizado por números reales; el manejo de los descuentos, que implica la utilización de números enteros; la definición de varias cuotas, que podrían ser representadas por diferentes vectores; en fin, la Estructura de Datos de nuestro software es representativa de los tipos de datos elementales.

Como es de nuestro conocimiento, las localidades de memoria básicamente pueden organizarse de dos formas, de forma contigua y de forma ligada, un arreglo o matriz de datos, también puede organizarse de cualquiera de estas dos forma. Para nuestro propósito, tomaremos en cuenta la primera, debido a su facilidad en el manejo.

### III.1.1.3 Diseño de Bases de Datos

El análisis y diseño de las bases de datos utilizadas por el software es un punto importante a considerar en la fase de desarrollo, ya que el diseño de las mismas redundará en el del software.

La función de las bases de datos no es meramente la de almacenar datos, si eso fuese, su organización resultaría por demás simple. En efecto, la mayoría de sus complejidades surgen por el hecho de que también tiene que indicar las relaciones que existen entre diversos ítems de datos que almacenan.

Antes de intentar la descripción de la representación física de la relación entre los datos debemos considerar su representación lógica, para ello nos auxiliaremos del esquema. El esquema es un diagrama de los tipos de datos que se usan, proporciona los nombres de las entidades y atributos, especifica también las relaciones que existen entre ellos.

Los esquemas se presentan a menudo a modo de diagramas de bloque, las relaciones se representan como líneas continuas y las referencias cruzadas como líneas discontinuas. Las relaciones agregan una información que no es inherente a los ítems de datos indicados en el esquema, mientras que las referencias cruzadas, no aportan nueva información, es decir, sólomente indican vinculaciones entre archivos, que permiten localizar los datos más fácilmente.

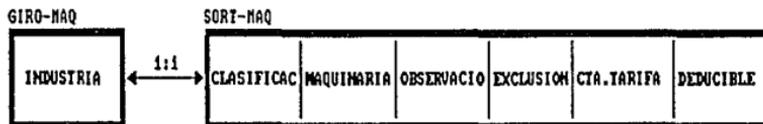
La relación que existe entre dos tipos de datos puede ser simple o compleja. Por simple queremos decir que hay una correspondencia biunívoca (de uno a uno) entre los ítems, una relación compleja representa la interacción de un ítem a varios o de varios a varios. Un estilo de representar gráficamente estas relaciones consiste en indicar las relaciones simples por medio de una flecha simple y las relaciones complejas mediante una doble flecha.

En cuanto a la representación gráfica de las bases aludiremos a las 8 reglas para el dibujo de esquemas que nos indica Jhon Martin:

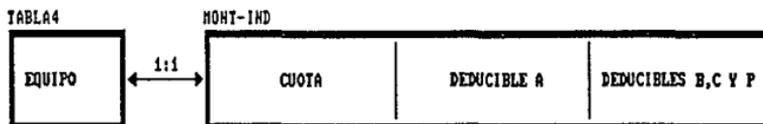
- 1.-El diagrama debe distinguir claramente los nombres de registro, los ítems de datos y otros.
- 2.-Debe quedar clara la reunión de ítems de datos en agregados de datos.
- 3.-Debe aclararse la distinción entre agregados de datos y registros.
- 4.-Deben mostrarse los identificadores de registro.
- 5.-El diagrama debe indicar claramente qué relaciones son simples y cuáles complejas.
- 6.-Las relaciones deben diferenciarse de las referencias cruzadas.
- 7.-Las relaciones entre registros deben nominarse o numerarse.
- 8.-No deben utilizarse nombres duplicados.

Tomando como base las reglas anteriores y teniendo en mente los ítems o campos que componen a los archivos representados con una doble línea en las figuras 3.4 y 3.5, ilustramos la estructura lógica o esquema de estas bases en la figura 3.6.

Esquema de las Bases de Datos



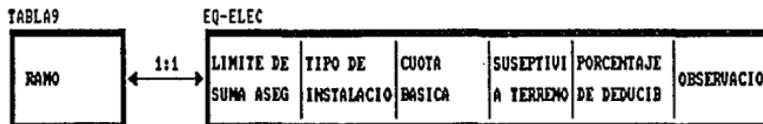
a) Rotura de Maquinaria



b) Montaje de Maquinas Individuales



c) Montaje de Plantas Completas



d) Equipo Electronico

Figura 3.6

Como se observa en la figura 3.6 y como es lógico suponerlo, no existe algún tipo de relación entre los esquemas, ya que las tablas de cotización están compuestas por elementos fijos e independientes entre sí, por esta misma razón, corroboramos que las bases de datos utilizadas por nuestro software serán accesadas sólo para su lectura.

De la misma manera, por la razón expuesta anteriormente, es decir, independencia de datos, no se describe el *diccionario de datos* correspondiente a las mismas, sin embargo podemos describir brevemente el significado de cada ítem por medio de la tabla 3.1.

### III.1.2 Descripción del Proceso

Durante el desarrollo del presente subcapítulo detallaremos el diseño preliminar planteado anteriormente. Propondremos la estructura del software y la detallaremos hasta llegar a su arquitectura, se definirán módulos, relaciones, narrativas del proceso, interfaces, etc.

En el siguiente punto efectuaremos la representación física de las bases de datos utilizadas por el sistema.

Finalmente, como tercer y último punto definiremos el lenguaje de programación.

## Descripción de las bases de datos

BASE DE DATOS	ITEM	DESCRIPCION
Rotura Maquinaria	Industria Clasificac Maquinaria Observacio Exclusion Cta.Tarifa Deducible	Giro de la empresa Grupo de la máquina Tipo de máquina Consideraciones Elementos excluidos Cuota básica Porcentaje de dedu.
Montaje Individual	Equipo Cuota Dedu.A Dedu.BCYP	Tipo de máquina Cuota básica Dedu. Cober. básica Deducible de terrem moto, huracán y prue.
Montaje Plantas	Industria Planta Mes 1 Mes 2 a 12 Mes 13 a 24 Mes restan. Mes 1 prueb Mes restant Deduci_A Deduci_BCPM	Giro industrial Tipo de planta Cuota ler. mes Hasta 12 meses Hasta 24 meses Meses siguientes Cuota 1 mes pruebas Meses restantes Dedu. Cober. básica Deducible de terremoto huracán, pruebas y man- tenimiento
Equipo Electrónico	Ramo Limite Instalacio Cta. básica Suseptivili Deducible Observacio	Giro Industrial Suma Aseg. límite Tipo de equipo Cuota básica Susceptibilidad a terrem. Porcentaje dedu. Observaciones sobre ins- pección o consultas a GP.

Tabla 3.1

## III.1.2.1 Estructura del Software

A través del desarrollo de los capítulos anteriores hemos ilustrado la metodología para la implementación del diagrama de flujo de datos. El diagrama de flujo de datos es usado como una herramienta gráfica para indicar el flujo de la información, sin embargo, necesitamos de otra herramienta para el análisis y la implementación de la arquitectura del software. El *Diseño Orientado al flujo de datos* define una serie de pasos para transformar el flujo de la información hacia la estructura del software.

La transición del flujo de información a la estructura es realizada como parte del proceso del diseño del software resumido en 5 pasos: 1) La categoría del flujo de la información es establecida; 2) Es indicado el flujo de la información; 3) El diagrama de flujo de datos es mapeado hacia la arquitectura del software; 4) Son definidos los controles jerárquicos mediante la factorización; 5) La estructura resultante es refinada por el uso del *Diseño Heurístico*.

En los capítulos anteriores, nuestra metodología a cumplido con los dos primeros pasos, para el desarrollo de los pasos siguientes es necesario reconocer que existen dos tipos característicos para el flujo de la información, la transformación y la transacción.

La transformación del flujo es ilustrada mediante el gráfico representado en la figura 3.7. La información entra al sistema a través de elementos que transforman el dato exterior a una forma interna. El flujo de estos elementos de comunicación es llamado *afecto*. Dentro del software existe una transición de los anteriores datos, que son pasados a través del *centro de transformación* y son movidos hacia otros elementos de salida que los transforman nuevamente para su entendimiento. El flujo de estos elementos de salida es llamado *efecto*. Cuando un segmento del diagrama de flujo de datos exhibe estas características está presente la transformación del flujo.

El modelo fundamental de un sistema implica transformación del flujo; así, es posible caracterizar todo flujo de información en esta categoría. Sin embargo, el flujo de la información es caracterizado frecuentemente por un dato llamado una *transacción*, que acciona otros flujo de datos a través de uno o varios elementos. Cuando un diagrama de flujo de datos toma la forma mostrada en la figura 3.8, la transacción del flujo está presente.

La transacción del flujo está caracterizado por un movimiento de los datos a través de un elemento de recepción que convierte la información del exterior hacia una transacción. La transacción es evaluada, y en base a este valor, es iniciado el flujo de uno o varios elementos de acción. El elemento del flujo de la información del cual emanan los elementos de acción es llamado el *centro de transacción*.

## Flujo de Información

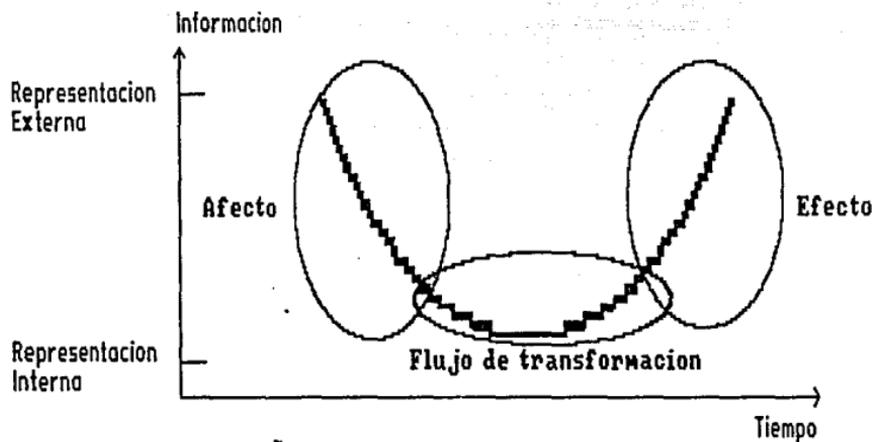


Figura 3.7

## Flujo de Transaccion

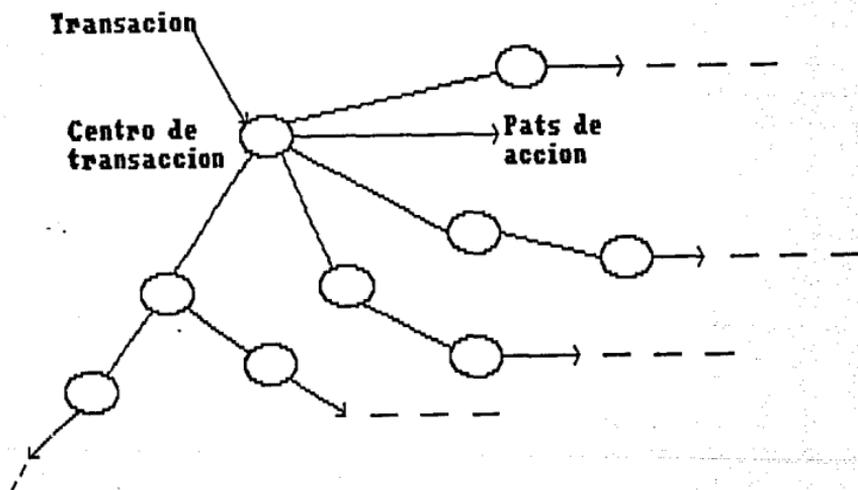


Figura 3.8

En base a la información anterior podemos dividir nuestro diagrama de flujo mostrado en la figura 3.4 y reconocer una transacción de flujo, en donde el pat de recepción al sistema es el elemento de transacción, el pat etiquetado como definición de cotización nos representa el centro de transacción y los pats a su derecha son los elementos de acción.

Los pats de acción a su vez los podemos dividir nuevamente. Podemos reconocer que estos pats nos representan una figura característica de transformación del flujo. Tomemos como ejemplo el ramo de equipo electrónico. El pat de identificación del giro nos representa el elemento de afecto, el elemento de transformación es realizado por los pats etiquetados como descuentos y coberturas adicionales, finalmente, el último pat nos representa el elemento de efecto.

De la misma forma al párrafo anterior, podemos subdividir cada uno de los ramos accionados por el centro de transacción. La figura resultante de esta serie de procesos la podemos resumir mediante el esquema de la figura 3.9 en donde son representadas las divisiones a través de líneas punteadas.

### Analisis de Transaccion

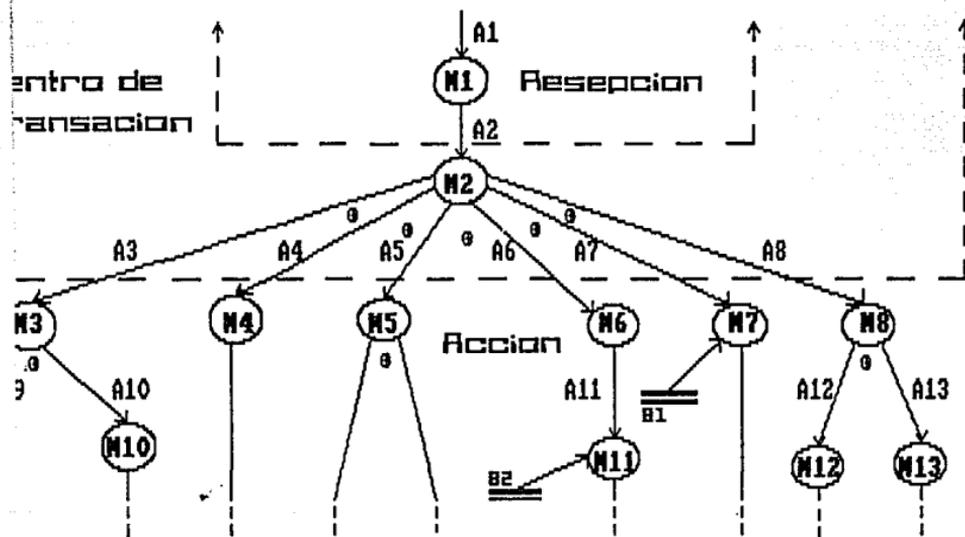


Figura 3.9

Primer Nivel de Factorización

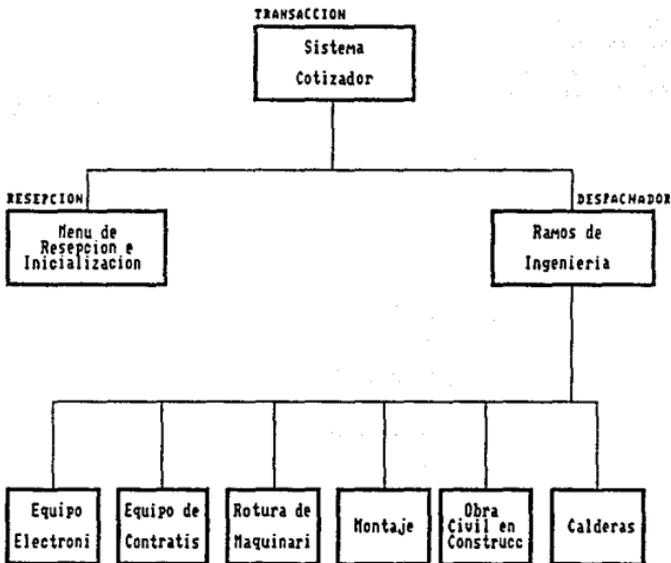


Figura 3.10

Nuestro diagrama de flujo de información puede ser ahora mapeado hacia la estructura del software. Efectuemos, así pues, el "primer nivel de factorización". La primer forma de flujo, la de transacción, nos representa en la estructura del software al centro de transacción como el nodo de primer nivel en la jerarquía, de éste nodo sobresalen dos ramas, una de ellas nos representa los nodos de recepción y la otra los de selección. Redefiniendo el centro de transacción, pat etiquetado como "Definición de cotización", como "Sistema Cotizador" dentro de la estructura del software, el primer pat en el flujo de datos nos representará el nodo de recepción y los pats de acción nos representarán los nodos de selección. La figura 3.10 nos ilustra el "primer nivel de factorización" en la estructura del software.

Continuando con el ejemplo del flujo de equipo electrónico, cada pat en el diagrama de flujo puede ser representado como un sólo nodo en la estructura del software. De la figura 3.9 podemos deducir y como ya se mencionó anteriormente, una transformación de flujo para este ramo. El primer y segundo pat nos representa un controlador de recepción de información, afecto. Los siguientes pats nos representan el centro de transformación y finalmente, el pat etiquetado como "reporte" nos representará el efecto, el cual coordina las salidas. Con el reconocimiento anterior y mediante las recomendaciones del diseño heurístico obtenemos el "segundo nivel de factorización y refinado" mostrado en la figura 3.11.

Segundo Nivel de Factorizacion  
y Refinacion mediante el Diseno  
Heuristico aplicado a cada pat

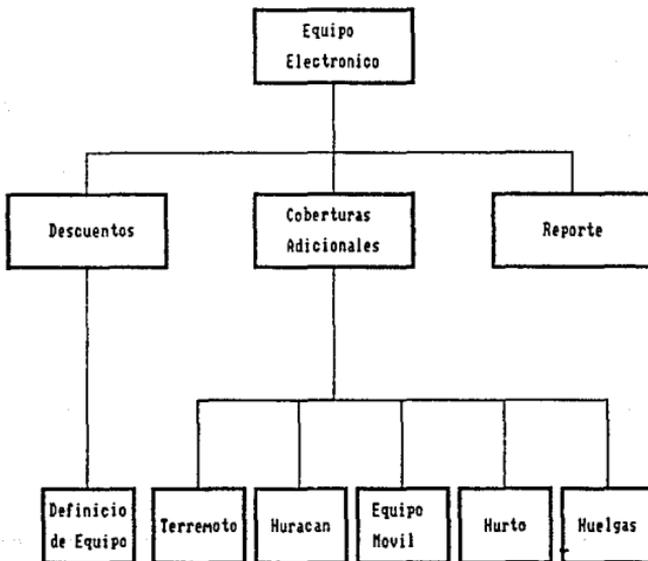


Figura 3.11

Así, si el usuario decide una cotización sobre el ramo de equipo electrónico lo primero que hará será definir al equipo, posteriormente, como la arquitectura del software nos lo sugiere, aplicará descuentos de acuerdo a una inspección realizada previamente. Si así lo desea, el usuario podrá calcular y agregar las cuotas de terremoto, huracán, equipo móvil, hurto y huelgas; dichas cuotas son independientes unas de otras, por lo que se encuentran separadas y dentro del mismo nivel jerárquico, finalmente se realizará el reporte correspondiente.

Un proceso semejante, descrito en los párrafos anteriores, lo podríamos aplicar en cada uno de los ramos de la figura 3.9, junto con una breve explicación o narrativa del proceso que realiza el ramo para su mejor entendimiento. La arquitectura de software mostrada en la figura 3.12 es la representación final de la estructura de nuestro software realizada mediante éste proceso.

Es importante destacar los elementos de interfase entre cada uno de los módulos, por ello, y para un mejor entendimiento en el desarrollo de nuestro sistema presentamos la tabla 3.2 en donde se muestra, como un ejemplo, las interfases para cada uno de los módulos más sobresalientes.

Refinado del Primer Corte de la Estructura del Software Bajo las Medidas del Diseño y la Heurística

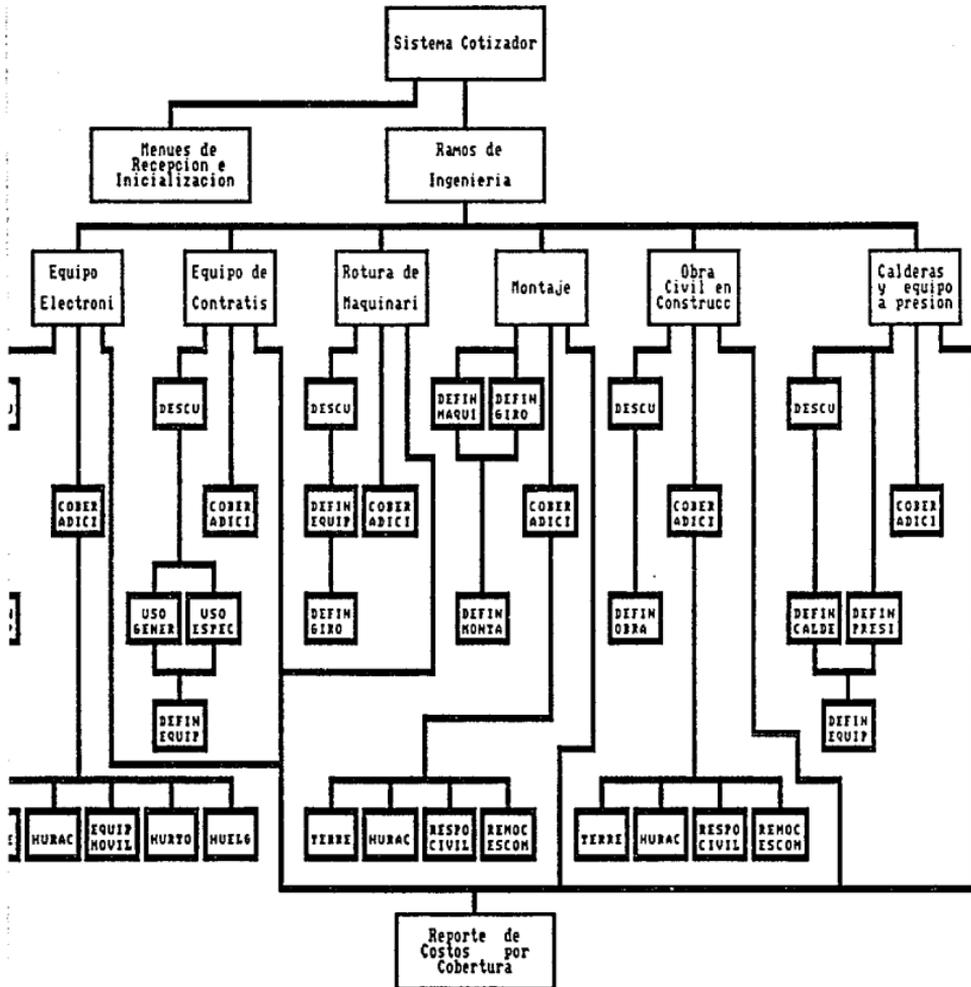


Figura 3.12

**Elementos de Interfase entre Módulos**

RAMO	MODULO	VARIABLE INTERFASE	TIPO	LONG.
CALDERAS	DEFINICION	CTABAS	Real	6
	RESP.PRESION	CTABAS	Real	6
	DESCUENTOS	DESC	Entero	2
OBRA CIVIL	DEFINICION	CTABAS	Real	6
	DESCUENTOS	DESC	Entero	2
	TERREMOTO	CTATERR	Real	6
	HURACAN	CTAHUR	Real	6
	R.C.	CTARC	Real	6
		CTARCCZ	Real	6
		CTADOPA	Real	6
REM. ESCOMBR.	CTAREM	Real	6	
CONTRATISTA	MAQ.GENERAL	CTABAS	Real	6
	MAQ.ESPECIF.	CTABAS	Real	6
	DESCUENTOS	DESC	Entero	2

**Tabla 3.2**

**Elementos de Interfase entre Módulos  
( continua )**

EQ.ELECTRON	INSTALACION	CTABAS	Real	6
	TERREMOTO	CTATERR	Real	6
	HURACAN	CTAHUR	Real	6
	PORTATILES	CTAPORTA	Real	6
	HUELGAS	CTAHUEL	Real	6
	HURTO	CTAHUR	Real	6
	DESCUENTOS	DESC	Entero	2
ROT.MAQUINA	DEFIN. GIRO	CPROME	Real	6
	DEFIN. EQUIPO	RAMO	Caracter	30
	DESCUENTOS	DESC	Entero	2
MONTAJE	MAQUINARIA	CTABAS	Real	6
	PLANTAS	CTABAS	Real	6
	TERREMOTO	CTATERR	Real	6
	HURACAN	CTAHUR	Real	6
	R.C.	CTARC	Real	6
		CTARCCZ	Real	6
		CTADOPA	Real	6
REM. ESCOMBR.	CTAREM	Real	6	

Tabla 3.2

## III.1.2.2 Estructura de las Bases de Datos

En capítulos anteriores hemos descrito las bases de datos y las relaciones existentes entre ellas. Una vez realizada esta función podemos describir con mayor detalle los ítems utilizados para cada una de ellas. De acuerdo a lo anterior, presentamos una relación en donde se muestra cada uno de los campos, su tipo y longitud. En algunas ocasiones, cuando el sistema así lo requiere, se puede complementar esta información mediante el *diccionario de datos*. El diccionario de datos define toda información cuantitativa en términos de información que es usada para el desarrollo de la cantidad. En nuestro caso, los campos entre sí y entre otras bases son completamente independientes, por lo que no haremos uso de esta herramienta. En la tabla 3.3 presentamos cada una de las bases de datos, sus campos utilizados y su tipo.

## III.1.2.3 Lenguaje de Programación

Los lenguajes de programación son un vehículo para la comunicación entre los humanos y las computadoras. El proceso de codificación, es una actividad humana. Así pues, las características psicológicas de un lenguaje tienen un importante impacto en la calidad de la comunicación. El proceso de codificación también puede ser visto como un paso en la metodología de la ingeniería del software. Las características ingenieriles también tienen un impacto importante en el suceso del desarrollo de un proyecto de software. Finalmente, las características técnicas de un lenguaje pueden influenciar la calidad del diseño.

## Estructura lógica de las Bases de Datos

BASE DE DATOS	NOMBRE CAMPO	TIPO	LONGUITUD
EQ-ELEC	RAMO	Caracter	30
	LIMITE	Numerico	7
	INSTALACIO	Caracter	60
	CTA BASICA	Numerico	6
	SUSEPTIVIL	Caracter	2
	DEDUCIBLE	Numerico	2
	OBSERVACIO	Caracter	50
MONT-IND	EQUIPO	Caracter	34
	CUOTA	Numerico	4
	DEDU_A	Numerico	4
	DEDU_BCYP	Numerico	4
MONT-PLA	INDUSTRIA	Caracter	30
	PLANTA	Caracter	67
	MES00A01MO	Numerico	4
	MES02A12MO	Numerico	5
	MES13A24MO	Numerico	5
	MES25A_MO	Numerico	5
	MES00A01PB	Numerico	4
	MES01A_PB	Numerico	4
	DEDUCI_A	Numerico	4
	DEDU_BCPYM	Numerico	4
ROT-MAQ	INDUSTRIA	Caracter	30
	CLASIFICAC	Caracter	30
	MAQUINARIA	Caracter	67
	OBSERVACIO	Caracter	80
	EXCLUSION	Caracter	50
	CTA_TARIFA	Numerico	5
	DEDUCIBLE	Numerico	3

Tabla 3.3

Así, el escoger un lenguaje de programación para un proyecto específico debe de considerar ambas características, ingenieril y psicológica. Como lo recomienda Pressman, los criterios que son aplicados para la evaluación de la disponibilidad de un lenguaje son: 1) Area de aplicación general, 2) algoritmos y complejidad computacional, 3) ambiente en el cual el software será ejecutado, 4) consideraciones de funcionamiento, 5) complejidad de la estructura de datos, y 6) conocimiento del lenguaje.

Atendiendo las anteriores características y recomendaciones, sugerimos un lenguaje de programación, que sea fácil en su entendimiento, que sea dirigido hacia los lenguajes de programación estructurada, que sea común en el mercado, de fácil adquisición y que maneje estructuras de datos elementales. El Dbase III+ nos ofrece las observaciones anteriores, además de la ventaja del manejo de las bases de datos y de sus pantallas. El Dbase III+ no es solamente un manejador de bases de datos, sino que incluye un lenguaje de programación. Este lenguaje de programación es tan fácil de aprender como cualquier lenguaje estructurado, ya que las características de sus funciones y comandos fueron desarrollados con tal motivo. Así pues, tomaremos éste lenguaje como la herramienta de comunicación y codificaremos nuestro software teniendo en mente sus características técnicas y funcionales.

### III.2 Implementación

Los pasos de ingeniería de software que han sido presentados hasta este punto están dirigidos hacia un objetivo final: traducir las representaciones de los elementos del software hacia una forma que pueda ser entendida por la computadora. Nosotros hemos llegado finalmente al paso de *programación y codificación* del software, un proceso que transforma el diseño hacia un *lenguaje de programación*.

Durante el desarrollo del primer inciso del presente subcapítulo crearemos las bases de datos bajo la especificación descrita en los capítulos previos. En el segundo inciso efectuaremos la implementación del software y como último punto realizaremos un ejemplo de su ejecución.

#### III.2.1 Bases de Datos

Las relaciones existentes y la descripción lógica de las bases de datos, descritas en los anteriores capítulos, ahora pueden ser implementadas sin mayor dificultad con el sistema manejador definido previamente.

Bajo los comandos del Dbase III+, creamos las bases de datos para cada una de las tablas de cotización, cumpliendo con los requerimientos especificados para cada una de ellas. Posteriormente, procedemos a la captura de cada registro y finalmente revisamos que los datos sean correctos.

Como un ejemplo, en la figura 3.13, presentamos la creación y captura de algunos campos de la base de datos para la tabla de cotización de equipo electrónico. Los registros de esta base cuentan con una relación simple con el archivo indexado "tabla9.ndx". Los registros de este archivo están conformados por el primer campo de la base de datos, los cuales nos representan la llave para su localización. El Dbase III+ requiere de este archivo para su proceso de búsqueda.

Mediante la definición del equipo electrónico obtenemos, de la base de datos, su cuota básica anual; el deducible que aplicaría en caso de reclamación; el grado de susceptibilidad a terremoto, marcado por una letra "E" o "F" dependiendo de la zona sísmica; y, encontramos las observaciones que las oficinas de servicio deben tomar en cuenta para una correcta suscripción del negocio.

De la misma forma, podemos crear cada una de las bases de datos requeridas por el software y así tomarlas en cuenta durante la implementación del mismo.

Base de Datos para Equipo Electrónico  
(Listado de algunos campos)

INSTALACION	CTA_BASICA	S	O	OBSERVACION
ESTUDIOS DE RADIO	5.000	E	5	Sujeta a inspección
EMISORAS Y REPETIDORAS DE RADIO	7.000	E	5	Sujeta a inspección
ANTENAS RECEPTORAS A LA INTemperIE	9.000	F	2	No requiere de inspección
GRABADORES DE SONIDO PARA SALONES DE FIESTA	8.500	F	5	Sujeta a inspección
INSTRUMENTOS ELEC PARA CONJUNTOS MUSICALES.	0.000	F	0	Consultar con Gerencia
INSTALACIONES DE CABLEVISION	0.000	F	0	Sujeta a inspección.
EQUIPO AUDIOVOSUAL EN ESCUELAS Y HOTELES	11.000	E	10	Sujeta a inspección
EQUIPO DE PROYECCION EN SALAS PUBLICAS DE CINE	7.000	E	5	Sujeta a inspección
ESTUDIOS DE TELEVISION	0.000	F	0	Sujeta a inspección
VEHICULOS TRANSMISORES DE RADIO Y T. V.	8.000	F	5	Sujeta a inspección
ANALIZADORES DE SUSTANCIAS, CROMATOGRAFO, ETC.	7.000	F	1	Sujeta a inspección
RAYOS X FIJOS PARA ANALISIS DE ESTRUCTURAS	11.000	F	2	Sujeta a inspección
RAYOS X MOVIBLES PARA ANALISIS DE ESTRUCTURA	25.000	F	2	Sujeta a inspección
ULTRASONIDO FIJO PARA ENSAYO DE MATERIALES	9.000	F	2	Sujeta a inspección
ULTRASONIDO MOVIBLES PARA ENSAYO DE MATERIALES	13.500	F	2	Sujeta a inspección
EQUIPO DE MEDICION Y ENSAYO EN LABORATORIOS	7.500	F	2	Sujeta a inspección
EQUIPO DE MEDICION Y ENSAYO EN CENTROS EDUCATIV.	10.500	F	10	Sujeta a inspección
SISMOGRAFOS ESTACIONARIOS	5.500	F	1	No requiere inspección
MICROSCOPIO ELECTRONICO	4.000	F	1	No requiere inspección
CONSOLAS FIJAS DE DIAGNOSTICO PARA AUTOMOVILES	17.500	E	10	Sujeta a inspección
CONSOLAS MOVILES DE DIAGNOSTICO PARA AUTOMOVILES	15.000	E	10	Incluir EQUIPO MOVIL
TABLERO DE REGISTRO PARA MANDO DE MAQUINAS	0.000	E	0	Consultar a Gerencia
REGISTRADORES O MEDIDORES FIJOS	8.500	F	2	No requiere inspección
REGISTRADORES O MEDIDORES MOVILES EN EL PREDIO	10.000	F	2	No requiere inspección
REGISTRADORES O MEDIDORES MOVILES FUERA PREDIO	0.000	F	0	Consultar a la Gerencia
CONMUTADORES TELEFONICOS MENORES A 20,000 UScy	6.500	F	1	No requiere de inspección
CONMUTADORES TELEFONICOS MAYORES A 20,000 UScy	5.500	F	1	Sujeta a inspección
TELEFONOS CELULARES	0.000	F	0	Consultar a la Gerencia
TELEX, TELEFAX, TELECOPIADORAS	6.500	F	1	No requiere de inspección
EMISORES-RECEPTORES DE RADIO FIJOS	10.000	E	2	No requiere de inspección
EMISORES-RECEPTORES DE RADIO MOVILES	17.500	F	1	Incluir EQUIPO MOVIL
INSTALACIONES DE ALTAVOCES	6.500	E	2	No requiere de inspección
EQUIPO FIJO PARA TRADUCCION SIMULTANEA	8.000	E	2	No requiere de inspección

Figura 3.13

Base de datos para Equipo Electrónico  
(Continúa)

EQUIPO MOVIL PARA TRADUCCION SIMULTANEA	18.000	F 5 Se EXLUYE HURTO
INSTALACION FIJA DE BUSCAPERSONAS	10.000	E 2 No requiere de inspeccion
INSTALACION MOVIL DE BUSCAPERSONAS	0.000	F 0 Consultar con Gerencia
AMPLIFICADOR DE RADIOGRAFICA, TECNICA TELEVISIVA	9.000	F 1 No requiere de inspeccion
EQUIPO DE RAYOS X TERAPEUTICO	9.000	F 1 No requiere de inspeccion
OMOGRAFOS COMPUTARIZADOS	5.500	F 1 Sujeta a inspeccion
BOMBA DE COBALTO (GAMETRON)	6.000	F 1 Sujeta a inspeccion
INSTALACIONES POR SEPARADO PARA MEDICINA NUCLEAR	17.000	F 2 Sujeta a inspeccion
CENTRALES DE MEDICINA NUCLEAR MENOR A 20,000USC	13.000	F 2 Sujeta a inspeccion
CENTRALES DE MEDICINA NUCLEAR MAYOR A 20,000 USC	0.000	F 0 Consultar con Gerencia
EQUIPO DE MEDICION COMO ELECTROCARDIOGRAFOS, ETC.	16.000	F 2 No requiere de inspeccion
CENTRALES DE MEDICION MENOR A 20,000 USC	13.000	F 2 Sujeta a inspeccion
CENTRALES DE MEDICION MAYORES A 20,000 USC	0.000	F 0 Sujeta a inspeccion
EQUIPOS DENTALES COMPLETOS	6.000	F 5 No requiere de inspeccion
EQUIPOS DE DIALISIS DOMESTICOS	6.000	F 5 No requiere de inspeccion
STERILIZADORES, EQUIPOS DE ANESTESIA	11.000	F 2 No requiere de inspeccion
EQUIPO MEDICINA DE ENSAYO CLINICO DE LABORATORIO	0.000	0 Consultar con Gerencia
EQUIPOS INDIVIDUALES DE FISICA MEDICINAL	16.000	F 2 No requiere de inspeccion
CENTRALES COMPLETAS MENORES A 20,000 USC	13.000	F 2 Sujeta a inspeccion
CENTRALES COMPLETAS MAYORES A 20,000 USC	0.000	F 0 Consultar Gcia. Prod.
OMOGRAFOS Y EQUIPO PARA OSCULTAR	15.000	F 2 No requiere de inspeccion
EQUIPO DE DETECCION DE HUMO, ROBO, INCENDIO	7.500	E 2 No requiere de inspeccion
CIRCUITO CERRADO DE TELEVISION	13.000	E 5 No requiere de inspeccion
ABLEROS DE CONTROL Y SEÑALIZACION AUTOMATICOS	11.000	E 2 Sujeta a inspeccion
ABLEROS DE CONTROL Y SEÑALIZACION NORMALES	8.500	E 2 Sujeta a inspeccion
ELOJES REGISTRADORES DE HASTA 20,000 USC	5.000	E 2 No requiere de inspeccion
ELOJES REGISTRADORES MAYORES A 20,000 USC	0.000	E 0 Consultar a Gcia. Prod.
ELOJES EXTERIORES PARA EDIFICIOS	15.000	E 5 Sujeta a inspeccion
ELOJES INTERIORES PARA EDIFICIOS	7.000	E 5 No requiere inspeccion
ABLEROS LUMINOSOS EN RELOJES	49.000	E 5 Consultar a Gcia. Prod.
ABLEROS LUMINOSOS DE PUBLICIDAD	56.000	E 5 Consultar con Gcia. Prod.
INSTALACIONES DE SEMAFOROS MENORES A 35,000USC	7.700	E 2 Sujeta a inspeccion

Figura 3.13

Base de Datos para Equipo Electrónico  
(Continua)

INSTALACIONES DE SEMAFOROS MAYORES A 35,000USCy	0.000	E 0 Consultar con Gcia. Prod.
INSTALACIONES DE SEÑALES CON MANDO DE RADIO	25.500	E 2 No requiere de inspeccion
CLISADORA STANDARD, COLORIMETROS, CROMATOGRAFOS	5.000	E 1 No requiere de inspeccion
VARIOCLISADORAS	4.500	E 1 No requiere de inspeccion
HELIOCLISADORAS	3.000	E 1 No requiere de inspeccion
EQUIPOS DE FOTOCOMPOSICION MENORES A 20,000 USCy	4.500	E 1 No requiere de inspeccion
EQUIPOS DE FOTOCOMPOSICION MAYORES A 20,000 USCy	0.000	E 0 Sujeta a inspeccion
PORTADORES TIPOGRAFICOS PARA FOTOCOMPOSICION	50.000	E 15 No requiere de inspeccion
EQUIPO FOTOGRAFICO PARA MENOR A 20,000 USCy	4.500	E 1 No requiere de inspeccion
EQUIPO FOTOGRAFICO PARA MAYOR A 20,000 USCy	0.000	E 0 Consultar Gcia. Prod.
EQUIPO DE MICROFILMACION MENOR A 20,000 USCy	6.500	F 2 No requiere de inspeccion
EQUIPO DE MICROFILMACION MAYOR A 20,000 USCy	0.000	F 0 Consultar Gcia. Prod
TRANSPORTACION INTERNA (CORREO NEUMATICO)	6.500	F 2 No requiere de inspeccion
EQUIPO DE OFICINA MENOR A 20,000 USCy	5.500	F 2 No requiere inspeccion
EQUIPO DE OFICINA MAYOR A 20,000 USCy	3.000	F 1 Sujeta a inspeccion
EQUIPO DE COMPUTO SEPARADO MENOR A 35,000 USCy	6.500	F 2 No requiere de inspeccion
EQUIPO DE COMPUTO SEPARADO MAYOR A 35,000 USCy	3.500	F 1 Sujeta a inspeccion
INSTALACIONES DE COMPUTO MENORES A 35,000 USCy.	3.500	F 1 No requiere de inspeccion
INSTALACIONES DE COMPUTO MENORES A 5,000,000 USCy	2.800	F 1 Sujeta a inspeccion
INSTALACIONES DE COMPUTO MAS DE 10,000,000 USCy	0.000	F 1 Consultar Gcia. Prod.

Figura 3.13

## III.2.2 Programas y subprogramas

Antes de continuar con el desarrollo de nuestro software, debemos denotar las diferencias entre los conceptos de programación y codificación, ya que estas dos actividades son con frecuencia confundidas. Por "programa" entendemos un proceso mental complejo, dividido en varias etapas, que describiremos posteriormente. La finalidad de la programación, así entendida, es comprender con claridad el problema que va a resolverse o simularse por medio de la computadora, y entender también con detalle cuál va a ser el procedimiento mediante el cual la máquina llegará a la solución deseada.

En su libro, Ingeniería de Software, Pressman nos habla del *Diseño Detallado* como un nombre opcional al de programación. Nos indica que las herramientas para la representación del diseño detallado se dividen en tres grandes categorías: gráficas, tabulares y de lenguaje. Las herramientas gráficas y de lenguaje aplican sobre la construcción lógica de la programación estructurada, proveyendo una base sobre la cual el código puede ser generado directamente. Las herramientas tabulares, como las tablas de decisión, proveen una conciencia, una especificación inambigua de acciones que ocurren como resultado de combinaciones complejas de condiciones.

Por lo que, "codificar" constituye una etapa necesariamente posterior a la de programar, y consiste en describir, en términos de algún lenguaje de programación adecuado, la solución ya encontrada, o sugerida, por medio de la programación. Es decir, uno primero programa la solución de un problema y después se preocupa por "traducirla" para la computadora.

La actividad de la programación es ante todo conceptual, y su campo de acción está centrado en tratar de definir, cada vez con mayor precisión, acercamientos que resuelvan el problema de manera "virtual", es decir, que efectuen una especie de "experimentos mentales" sobre el problema a resolver o similar. El resultado de tales experimentos constituirá una descripción de los pasos necesarios para encontrar la solución.

Esta descripción, como cualquier otra, estará expresada en un lenguaje determinado. La importancia de la programación consiste en que este lenguaje funcione a la vez como vehículo descriptor y como modelo de la representación dada a la solución. Las principales características de este lenguaje consiste en que sea "neutro" y "completo". El primer concepto denota su independencia respecto de alguna máquina en particular, y el segundo se refiere al poder del mismo para expresar cualquier idea computacional. Denotaremos este tipo de lenguaje como *seudocódigo* y sobre el cual nos basaremos para la programación de nuestro software.

Tomemos cualquiera de los módulos de la arquitectura del software y detallémoslo como ejemplo del proceso descrito anteriormente. Refiriéndonos a la figura 3.12, observamos que el módulo de Cotización para Equipo Electrónico tiene entre sus subordinados al módulo que identifica al equipo para el cálculo de la cuota básica, para ello, deberá hacer algunos accesos a la base de datos correspondiente. Así pues, tomemos este último módulo para el desarrollo de nuestro ejemplo.

La función básica de este módulo, como lo indicamos anteriormente, es de obtener la cuota básica del equipo electrónico. Debemos realizar las siguientes funciones básicas: a) Definir entre las opciones ya existentes el giro del negocio, b) identificar el equipo, c) obtener de la base de datos correspondiente la cuota, el deducible, la sensibilidad a terremoto y los comentarios para la oficina de servicio. Finalmente, si el equipo en cuestión corresponde a Procesadores de Datos, deberán darse las opciones para el d) cálculo de la cuota para la sección II (Portadores Externos de Datos), y sección III (Incremento en el Costo de la Operación).

Definidas nuestras funciones, debemos ahora averiguar las relaciones estructurales entre ellas, o sea, integrar un modelo armónico de funcionamiento de nuestro problema con base a las partes mencionadas. Llega el momento de hacer una descripción de las interrelaciones entre estas funciones, y es cuando debemos hacerlas explícitas por medio del pseudocódigo. Proponemos lo siguiente:

!PRIMERA DESCRIPCION DEL MODULO DE IDENTIFICACION  
!PARA INSTALACIONES DE EQUIPO ELECTRONICO.

```

proc instalación
  parámetros real ctabas           !Cuota básica

** Estructura de Datos **
declaración de la estructura de
datos utilizada en este módulo.

** Inicia Programa **
mientras no se haya obtenido registro
  proporciona opciones de giros      !Función a)
  identifica el equipo                !Función b)
fin mientras
obten los campos del registro         !Función c)
si ctabas<>0                          !Si existe cuota
  obten suma asegurada                !Dada por usuario
  si registro es EPD                    !Eq. Procesadores
                                         !de Datos
      calcula cuota sección II         !Función d)
      calcula cuota sección III       !Función e)
fin si
otro
  informa fuera de políticas
fin si
regresa

```

El entendimiento del seudocódigo anterior, es realmente sencillo, sin embargo lo comentaremos para aclarar las posibles dudas. Las palabras seguidas del signo de admiración "!" nos indican simples comentarios, las palabras subrayadas indican algún tipo de *estructura de control*, ya sea secuenciación, selección o iteración condicional. Estas estructuras de control, son las herramientas para la construcción de programas por medio del seudocódigo.

Las palabras "proc" y "regresa" implican que se trata de un módulo o función al servicio de un programa maestro. La palabra "parámetros" y "real" nos definen las variables y su tipo como elementos de comunicación entre este módulo y el programa maestro.

Como podemos observar, las funciones "a" y "b)" se encuentran en una iteración hasta que se haya localizado algún registro válido y, en seguida, obtenemos los campos del registro localizado (función "c)"). Las funciones "d)" y "e)" son definidas en dos estructuras de control de tipo "selección". La primer selección, verifica que el equipo se encuentre dentro de las políticas establecidas para las oficinas de servicio, de lo contrario desplegará un mensaje de error. La segunda selección define si se trata de un equipo de Procesamiento de Datos, si así lo es, realiza las funciones ahí indicadas, de lo contrario no realiza cambios.

Ya tenemos un modelo de solución para el problema planteado, que puede ser "probado" para determinar si es correcto o no. Existen varios tipos de pruebas de programas, pero al nivel que estamos ahora trabajando nos conformaremos con una prueba "mental", que consiste en examinar el flujo de acciones propuesto por el programa, a la vez que haremos razonamientos sencillos sobre el mismo. Esta facilidad de "avanzar sobre seguro" en la programación es tal vez la ventaja más importante del método propuesto, ya que es ahora, a tiempo de diseño, cuando podemos hacer los cambios necesarios para asegurar el funcionamiento correcto del modelo.

Si ya no hay ninguna duda acerca de nuestra primera versión, procedamos a detallar cada una de las funciones.

!SEGUNDA DESCRIPCION DEL MODULO DE IDENTIFICACION  
!PARA INSTALACIONES DE EQUIPO ELECTRONICO.

proc instalación

parámetros real ctabas

!Cuota básica

\*\* Estructura de Datos \*\*

declaración de la estructura de datos utilizada en este módulo.

\*\* Inicia Programa \*\*

mientras no se haya obtenido registro

!Detalles de la función a)

limpia pantalla

crea pantalla con las opciones de giro

obten la opción escogida

!Dada por usuario

caso

opcion escogida = 1

tipo de giro para "RADIO"

opcion escogida = 2

tipo de giro para "MEDICION"

opcion escogida = 3

tipo de giro para "COMUNICACION"

opcion escogida = 4

tipo de giro para "MEDICINA"

opcion escogida = 5

tipo de giro para "ALARMA"

opcion escogida = 6

tipo de giro para "IMPRESAS"

opcion escogida = 7

tipo de giro para "PROCES. DE DATOS"

fin caso

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

!Detalles de la función b)

limpia pantalla  
 crea nueva pantalla  
 obtén los registros del giro seleccionado  
 crea menú de opciones con los registros obtenidos  
 requisita la opción !Dada por usuario  
 localiza el registro !En Base de Datos

fin mientras

obten los campos del registro !Función c)  
si ctabas<>0 !Si existe cuota  
 obtén suma asegurada !Dada por usuario  
si registro es EPD !Eq. Procesadores  
 !de Datos

!Detalles de la función d)

requisita S.A. para sección II !Dada por usuario

caso

S.A. sección II = 0

cuota sección II = 0

S.A. sección II < 170,000 dolares

cuota sección II = 4.8%

otro

cuota sección II = 3.2%

fin caso

deducible sección II igual a 10%

observaciones sección II: deducible sobre pérdida

!Detalles de la función e)

requisita S.A. para secciónIII!Dada por usuario  
caso

S.A. secciónIII = 0

cuota secciónIII = 0

S.A. secciónIII < 350,000 dólares

cuota secciónIII = 2.63%

S.A. secciónIII < 1'000,000 dólares

cuota secciónIII = 3.27%

otro

cuota secciónIII = 4.63%

fin caso

si cuota secciónIII <> 0

requisita meses del periodo de indemnización

caso

periodo < 2 meses

descuento del 50% a la cuota secciónIII

periodo < 3 meses

descuento del 35% a la cuota secciónIII

periodo < 6 meses

descuento del 25% a la cuota secciónIII

periodo < 9 meses

descuento del 10% a la cuota secciónIII

periodo < 12 meses

descuento del 5% a la cuota secciónIII

fin caso

deducible secciónIII:2

observaciones secciónIII: en días laborables

fin si

fin si

otro

informa: fuera de políticas

fin si

regresa

La segunda versión del algoritmo para el módulo de identificación de las instalaciones de equipo electrónico, es esencialmente correcta, podemos convencernos de que sí lo es, por medio de algunos ejemplos mentales, más no es aún ejecutable por una computadora. Para que esto sea posible hace falta detallar con mayor precisión y "traducirlo" a un programa escrito en un lenguaje de programación, que luego será compilado, ensamblado, cargado y ejecutado.

La idea fundamental de esta traducción es convertir la "carga semántica" del programa en instrucciones que la ejecuten. Con ese término queremos sintetizar nuestro "conocimiento acerca del mundo", que empleamos para escribir nuestro programa en pseudocódigo y que ahora será necesario bajar el nivel gnoseológico. Aquí es donde el pseudocódigo cumple con su doble papel de servir como intermediario entre nosotros por un lado y la máquina por el otro, y de ayudar a la descripción conceptual de un problema en términos de estructuras primitivas.

El proceso de esta traducción entre pseudocódigo y un lenguaje de programación ocupa la frontera entre programa y codificar. Aunque no entraremos en los detalles específicos de un lenguaje en particular, sí llevaremos más adelante nuestra traducción extrayéndole más detalles al pseudocódigo, hasta convertirlo en algo parecido a la codificación de un programa. El proceso que se emplea para esto, recibe el nombre de *refinamientos progresivos* porque consiste precisamente en ir obteniendo cada vez más detalles (dentro de la misma estructura ya definida) hasta llegar a un lenguaje de programación específico.

La diferencia entre el seudocódigo ya planteado y lo que queremos obtener está determinada por la estructura de datos; es decir, debemos especificar cuidadosamente la "forma" que van a tener los datos en nuestro programa, para poder codificarlo.

Diremos entonces que otra manera de ver la función de las estructuras de datos es precisamente la de descomponer la carga semántica de un programa en elementos más cercanos a la naturaleza operativa de una computadora.

Para nuestros fines utilizaremos las estructuras de datos que ya han sido mencionadas en capíulos anteriores y que ahora detallaremos. El presente módulo utiliza algunas variables externas que son definidas en el programa maestro, estas variables son manejadas para el paso de parámetros globales y que a continuación listaremos en la tabla 3.4.

**Variables Externas**

VARIABLE	DIMENSION	TIPO	DESCRIPCION
sumase	8	entero	sumas aseguradas
cuota	8	real	cuotas
dedu	8	entero	deducibles
obser	8	cadena	observaciones
condi	3	cadena	condiciones para la oficina
sen_tev	1	caracter	sensibilidad a terremoto.
opcion	1	entero	opciones

**Tabla 3.4**

Además de éstas estructuras de datos externas debemos mencionar que haremos utilización de los campos definidos anteriormente para la base de datos de Equipo Electrónico. En cuanto a las variables locales definiremos "registro" como una variable entera en donde se almacenará el valor actual del registro localizado; "tipo" como una cadena de caracteres que guardará el tipo de giro; "cadena" almacenará la descripción del equipo, por lo que deberá ser una cadena de caracteres; "conta", variable entera que la utilizaremos como un contador de renglones; "reg\_pdt" nos indica el valor del registro en donde, después de éste, comienzan los registros referidos a equipos de procesamiento de datos, por lo que su valor será una constante; y finalmente, "periodo" guardará los meses de indemnización para la sección III, por lo que nos representa un valor entero. Tomando todo esto en cuenta tenemos ahora

!TERCERA DESCRIPCION DEL MODULO DE IDENTIFICACION  
!PARA INSTALACIONES DE EQUIPO ELECTRONICO.

```

proc instalación
  parámetros real ctabas                !Cuota básica

** Estructura de Datos **
entero registro                            !registro obtenido
caracter tipo(30)                          !tipo de giro
caracter cadena(70)                        !Descr. del equipo
entero conta                               !contador renglon
constante reg_pdt = 80                    !registros EPD.
entero período                             !meses indemnizac.

```

**\*\* Inicia Programa \*\***  
mientras no se haya obtenido registro

!Detalles de la función a)

```

llama clr_scr                !limpia pantalla
!creación de la pantalla de giros
llama titulo('SEGURO DE EQUIPO ELECTRONICO')
g_menu(1)='RADIO, T.V. Y CINEMATOGRAFIA'
g_menu(2)='MEDICION, REGISTRO Y CONTROL'
g_menu(3)='EQUIPO DE COMUNICACION'
g_menu(4)='MEDICINA Y FISICA MEDICINAL'
g_menu(5)='ALARMA, SEGURIDAD Y SEÑALIZACION'
g_menu(6)='EQUIPO PARA IMPRENTAS'
g_menu(7)='PROCESAMIENTO DE DATOS Y OFICINAS'
!obtención de la opción escogida
llama menu(7,14,7,opcion,1,'RAMO INDUSTRIAL')
caso
    opcion=1
        tipo="RADIO"
    opcion=2
        tipo="MEDICION"
    opcion=3
        tipo="COMUNICACION"
    opcion=4
        tipo="MEDICINA"
    opcion=5
        tipo="ALARMA"
    opcion=6
        tipo="IMPRENTAS"
    opcion=7
        tipo="PROCES. DE DATOS"
fin caso

```

!Detalles de la función b)

```
llama clr_scr                !limpia pantalla
!creación de nueva pantalla para registros
llama screen                !nueva pantalla
llama titulo('INSTALACIONES PARA'+tipo)
!obtención de los registros del giro seleccionado
abre eq-elec                !abre base
localiza tipo                !localiza regis.
crea menú de opciones con los registros obtenidos
conta=0
mientras no sea EOF                !No fin archivo
    conta= conta+1
    en (3+conta),2 escribe conta    !despliega reng.
    en (3+conta),4 escribe '.-'    !despliega .-
    cadena=instalacio
    en (3+conta),6 escribe cadena  !despliega equipo
    continua                        !próximo registro
fin mientras
!requisitando la opción, dada por usuario
llama opcion('NUMERO DE RAMO',opcion)
!localizando el registro en Base de Datos
si opcion<=0
    registro=0                !No hay registro
    otro
        registro=registro+opcion    !actualiza regis.
    fin si
fin mientras
```

!Detalles de la función c)

```

!obtención de los campos del registro
posicionate registro                !Trae registro
llama msg_nw(instalacio)            !Despliega equipo.
cta_bas=cta-basica                    !Se asignan los
sen_tev=suseptivil                    !campos de la ba-
dedu(0)=deducible                      !se de datos
condi(1)=observacio
si ctabas<>0                          !Si existe cuota
    !obtencion de la suma asegurada, dada por usuario
    en 23,56 escribe 'Valor en USCy?'
    lee sumase(0)
    si registro>reg_pdt                !Eq. Procesadores
                                        !de Datos

```

!Detalles de la función d)

```

!obtención de la S.A. para sección II
en 23,0 escribe 'SECCION II: PED en USCy'
lee sumase(2)
caso
    sumase(2)=0
    cuota(2)=0
    sumase(2)<170000
    cuota(2)=4.8
    otro
    cuota(2)=3.2
fin caso
dedu(2)=10
obser(2)=' % s/pérdida

```

!Detalles de la función e)

!obtención de la S.A para sección III  
en 23,0 escribe 'SECCIONIII:INC. COST. en USCy'  
lee sumase(3)

caso

sumase(3)=0  
cuota(3)=0  
sumase(3)<350000  
cuota(3)=2.63  
sumase(3)<1000000  
cuota(3)=3.27

otro

cuota(3)=4.63

fin caso

si cuota(3)<>0

!obtención período de indemnización

en 23,0 escribe 'PERIODO EN MESES'

lee periodo

caso

período<2  
cuota(3)=cuota(3)\*0.50  
período<3  
cuota(3)=cuota(3)\*0.65  
período<6  
cuota(3)=cuota(3)\*0.75  
período<9  
cuota(3)=cuota(3)\*0.90  
período<12  
cuota(3)=cuota(3)\*0.95

fin caso

dedu(3)=2

obser(3)='días laborables'

fin si

fin si

otro

en 23,0 escribe 'FUERA DE COBERTURA'

fin si

regresa

Los detalles nuevos saltan a la vista, se hace referencia a la palabra "llama", esta palabra nos indica que se ejecutará una subrutina. Dentro de este módulo se han utilizado varias de ellas y que principalmente son funciones de pantalla como "clr\_scr" que limpia la pantalla; "titulo" centra en la parte superior de la pantalla una cadena de caracteres; "menú" despliega las cadenas de caracteres previamente asignadas al vector "g\_menu" y obtiene la opción del usuario a manera de un número; "screen" es una función que despliega una pantalla; y finalmente, "opcion" requisita al usuario sobre un número entero. Otras palabras que se han introducido son "localiza", "abre", "continua" y "posicionate", estas palabras son comandos generales para el manejo de las bases de datos. Las palabras "lee", "escribe" y "en" nos sirven para el manejo de las pantallas.

Lo que hemos ganado en detalle y estructura de datos lo hemos perdido en la claridad "humana" que tenía la segunda versión. Esto es inevitable, puesto que estamos ahora en un proceso tendiente a "hacer claro a la máquina" lo que ya antes era claro para nosotros. En virtud de la enorme distancia gnoseológica entre una computadora y el ser humano, mientras más detalles explícitos tenga un programa será menos claro, en general, para nosotros. Esta brecha puede acortarse con buenos hábitos de estructuración en los programas y con una adecuada documentación.

En las páginas que continúan, presentamos la codificación del seudocódigo de la "tercer descripción del módulo de identificación para las instalaciones de equipo electrónico" en el lenguaje seleccionado en el anterior capítulo. También presentamos la codificación de los programas subordinados y superordinarios, como una ilustración de las características de un programa totalmente terminado en este lenguaje.

De la misma forma en que hemos llevado acabo este proceso, el detalle de diseño para cada uno de los módulos presentados en la arquitectura del software, puede hacerse factible. La codificación, captura, compilado y ligado de los programas, dependen de las características del hardware y software de utilería.

```

*****
**                                     **
** Este programa realiza la cotizacion de equipo electronico **
**                                     **
** AUTOR: Eduardo Alejandro Hernández Q. **
**                                     **
** FECHA DE ULTIMA MODIFICACION: 18 de marzo de 1991 **
**                                     **
*****
inciso = 8                                && indica el numero de coberturas
cober = ' '                                && vector de coberturas
cober1 = 'BASICA'
cober2 = 'SECCION II'
cober3 = 'SECCION III'
cober4 = 'TERREMOTO'
cober5 = 'HURACAN'
cober6 = 'PORTATILES'
cober7 = 'HUELGAS'
cober8 = 'HURTO'
sumase = 0                                && vector de suma asegurada por cobertura
sumase1= 0
sumase2= 0
sumase3= 0
sumase4= 0
sumase5= 0
sumase6= 0
sumase7= 0
sumase8= 0
cuota = 0.0000                            && vector de cuotas
cuota1 = 0.0000
cuota2 = 0.0000
cuota3 = 0.0000
cuota4 = 0.0000
cuota5 = 0.0000
cuota6 = 0.0000
cuota7 = 0.0000
cuota8 = 0.0000

```

```

desc = 0                                && vector de descuentos
desc1 = 0
desc2 = 0
desc3 = 0
desc4 = 0
desc5 = 0
desc6 = 0
desc7 = 0
desc8 = 0
dedu = 0                                && vector de deducibles
dedu1 = 0
dedu2 = 0
dedu3 = 0
dedu4 = 0
dedu5 = 0
dedu6 = 0
dedu7 = 0
dedu8 = 0
obser = ' '                             && vector de observaciones por cobertura
obser1 = ' '
obser2 = ' '
obser3 = ' '
obser4 = ' '
obser5 = ' '
obser6 = ' '
obser7 = ' '
obser8 = ' '
riesgo1= .f.                            && vector de riesgos, indica si la cobertura fue dada
riesgo2= .f.
riesgo3= .f.
riesgo4= .f.
riesgo5= .f.
condi = 3                               && apuntador al texto de condiciones
condi1 = space(30)                       && texto de condiciones
condi2 = space(30)
condi3 = space(30)

```

```
sen_tev = ' '                && sensibilidad a terremoto
deseo = ' '
opcion = 0                   && variable con la opcion deseada
close databases
set scoreboard off
set escape off
set procedure to rut_ele
```

```
***** inicia programa principal *****
```

```
do while .t.
  do instal with cuota
  if cuota <> 0
    obser = '% S/EQUIPO'
    deseo = ' '
    do while deseo <> 'S' .and. deseo <> 'N'
      do opc_p with 'SE EFECTUO INSPECCION (S/N)? ', deseo
    enddo
    if deseo = 'S'
      do descuento with desc
    endif
    deseo = ' '
    do while deseo <> 'S' .and. deseo <> 'N'
      do opc_p with 'DESEA RIESGOS ADICIONALES (S/N)? ', deseo
    enddo
    if deseo = 'S'
      do adi_ele
    endif
    sumase1 = sumase
    cuota1 = cuota
    desc1 = desc
    dedu1 = dedu
    obser1 = obser
    do reporte with inciso,cober,sumase,cuota,desc,dedu,obser,condi
  else
    exit
  endif
enddo
close databases
return
```

```

*****
**                               **
**   La siguiente rutina regresa al programa principal la cuota   **
**   basica, suceptibilidad a TEV, deducible y observaciones     **
**   los parámetros de entrada son los que propor-               **
**   ciona el usuario en esta misma subrutina.                   **
**                               **
**   AUTOR: Eduardo Alejandro Hernández Q.                       **
**                               **
**   ULTIMA MODIFICACION: 18 de marzo de 1991                     **
**                               **
*****

```

```

procedure instal
  parameters ctabas

```

```

** Estructura de Datos **

```

```

registro= 0           && indica el equipo escogido
tipo   = space(30)    && tipo de instalacion
cadena = space(71)    && descripcion del equipo
conta  = 0            && numero de renglon en pantalla
reg_pdt = 80          && comienzo de los registros EPD
periodo = 0           && periodo de indemnizacion para la seccion III

```

```

** Inicia Programa **
do while registro=0

```

```

FUNCTION a)

```

```

do clr_scr
do titulo with ' SEGURO DE EQUIPO ELECTRONICO '
g_menu1='      RADIO, T.V. Y CINEMATOGRAFIA      '
g_menu2='      MEDICION, REGISTRO Y CONTROL      '
g_menu3='      EQUIPO DE COMUNICACION            '
g_menu4='      MEDICINA Y FISICA MEDICINAL       '
g_menu5='      ALARMA, SEGURIDAD Y SEÑALIZACION  '
g_menu6='      EQUIPO PARA IMPRENTAS             '
g_menu7='      PROCESAMIENTO DE DATOS Y OFICINAS '
do menu with 7,14,7,opcion,1,' RAMO INDUSTRIAL '

```

```
do case
  case opcion=1
    tipo='RADIO'
  case opcion=2
    tipo='MEDICION'
  case opcion=3
    tipo='COMUNICACION'
  case opcion=4
    tipo='MEDICINA'
  case opcion=5
    tipo='ALARMA'
  case opcion=6
    tipo='IMPRESNTAS'
  case opcion=7
    tipo='PROCES. DE DATOS'
endcase
```

I F U N C I O N   b)

```
clear
do screen
do titulo with 'INSTALACIONES PARA '+tipo
use eq-elec index tabla9
locate for ramo=tipo
conta = 0
do while .not. EOF()
  conta=conta+1
  @ 3+conta,2 say conta picture,'99'
  @ 3+conta,4 say '.-'
  cadena=trim(instalacio)
  @ 3+conta,6 say cadena
  continue
enddo
do opcion with 'NUMERO DEL RAMO LOCALIZADO? ',opcion
if opcion <= 0
  registro = 0
else
  registro = registro + opcion
endif
enddo
```

```
! FUNCION c)
```

```
goto registro
do msg_nw with trim(instalacio)
ctabas =cta_basica
sen_tev =trim(suseptivil)
dedu =deducible
condil =(observacio)
if ctabas <> 0
  @ 23,56 say 'VALOR EN USCy.' get sumase picture 'aB 9,999,999'
  read
  if registro > reg_pdt
```

```
! FUNCION d)
```

```
@ 23,0 say 'SECCION II: PED VALOR EN USCy.' get sumase2 picture 'aB 9,999,999'
read
do case
  case sumase2 = 0
    cuota2 = 0.0000
  case sumase2 < 170000
    cuota2 = 4.8000
  otherwise
    cuota2 = 3.2000
endcase
dedu2 = 10
obser2 = '% S/PERDIDA'
```

```
IFUNCIÓN e)
@ 23,0 say 'SECCION III: INC.COST.VALOR USCy.' get sumase3 picture '08 9,999,999'
read
do case
  case sumase3 = 0
    cuota3 = 0.0000
  case sumase3 < 35000
    cuota3 = 2.6363
  case sumase3 < 1000000
    cuota3 = 3.2727
  otherwise
    cuota3 = 4.6363
endcase
if cuota3 <> 0
  @ 23,0 say 'PERIDO DE INDEMNIZACION, EN MESES? ' get periodo picture '08 99'
  read
  do case
    case periodo < 2
      cuota3 = cuota3 * 0.5000
    case periodo < 3
      cuota3 = cuota3 * 0.6500
    case periodo < 6
      cuota3 = cuota3 * 0.7500
    case periodo < 9
      cuota3 = cuota3 * 0.9000
    case periodo < 12
      cuota3 = cuota3 * 0.9500
  endcase
  dedu3 = 2
  obser3 = 'DIAS LABORABLES'
endif
endif
else
  @ 23,0 say 'FUERA DE COBERTURA,CONSULTAR CON GERENCIA'
endif
return
```

```
*****
* Las siguientes subrutinas calculan la cuota para los riesgos      *
* adicionales de equipo electronico                                 *
*                                                                    *
* 28/03/91                                                         Eduardo Quevedo *
*****
```

```
procedure adi_ele
  clear
  do titulo with ' SEGURO DE EQUIPO ELECTRONICO '
  do while .t.
    do clr_wrk
    g_menu1=' TERREMOTO Y ERUPCION VOLCANICA '
    g_menu2=' HURACAN, CICLON Y TIFON '
    g_menu3=' EQUIPOS MOVILES O PORTATILES '
    g_menu4=' HUELGAS, MOTIN Y CONMOCCION CIVIL '
    g_menu5=' HURTO, ROBO SIN VIOLENCIA '
    do asignada with 5
    do menu with 8,19,5,opcion,1,' RIESGOS ADICIONALES '
    do case
      case opcion=1
        do terremoto with cuota4
      case opcion=2
        do huracan with cuota5
      case opcion=3
        do moviles with cuota6
      case opcion=4
        do huelgas with cuota7
      case opcion=5
        do hurto with cuota8
      case opcion=0
        exit
    endcase
  enddo
return
```

```

**
** terremoto
**
procedure terremoto
parameters ctatev
zona = 0
do msg_nw with '
g_menu1=' A '
g_menu2=' Brm '
g_menu3=' Bdf '
g_menu4=' C '
g_menu5=' D '
do hmenu with 5,zona,'HTEV'
if sen_tev='E'
do case
case zona=1
ctatev=0.2000
case zona=2
ctatev=0.4000
case zona=3
ctatev=0.5500
case zona=4
ctatev=0.6000
case zona=5
ctatev=0.8000
case zona=0
ctatev=0.0000
return
endcase
else
do case
case zona=1
ctatev=0.3500
case zona=2
ctatev=0.6000
case zona=3
ctatev=0.9000
case zona=4
ctatev=1.0500
case zona=5

```

INDICA LA ZONA SISMICA'

```
        ctatev=1.4500
        case zona=0
            ctatev=0.0000
        endcase
    endif
    if ctatev <> 0
        sumase4 = sumase+sumase2+sumase3
        dedu4   = dedu
        obser4  = obser
        riesgo1 = .t.
    endif
return
**
** huracan
**
procedure huracan
    parameters ctacic
    deseo=' '
    do while deseo<>'S' .and. deseo<>'N'
        do opc_p with 'EL EQUIPO ESTA A LA INTEMPERIE (S/N)? ',deseo
        enddo
        if deseo='S'
            deseo=' '
            do while deseo<>'S' .and. deseo<>'N'
                do opc_p with 'ESTA LOCALIZADO CERCA DE LA COSTA (S/N)? ',deseo
                enddo
                if deseo='S'
                    ctacic=3.2000
                else
                    ctacic=1.6000
                endif
            endif
        else
            deseo=' '
            do while deseo<>'S' .and. deseo<>'N'
                do opc_p with 'ESTA LOCALIZADO CERCA DE LA COSTA (S/N)? ',deseo
                enddo
                if deseo='S'
                    ctacic=0.8000
                else
                    ctacic=0.0000
                endif
            endif
        endif
    enddo
enddo
```

```

endif
sumase5 = sumase+sumase2+sumase3
dedu5 = dedu
obser5 = obser
riesgo2 = .t.
return
**
** portatiles
**
procedure moviles
parameters ctapor
do msg_nw with ' INDICA TIPO DE RIESGO'
g_menu1=' BAJO '
g_menu2=' MEDIO '
g_menu3=' ALTO '
do hmenu with 3,opcion,'HMOVIL'
do case
case opcion=3
ctapor=25.0000
case opcion=2
ctapor=12.5000
case opcion=1
ctapor=10.0000
case opcion=0
ctapor=00.0000
return
endcase
if ctapor <> 0
sumase6 = sumase+sumase2+sumase3
dedu6 = dedu
obser6 = obser
riesgo3 = .t.
endif
return
**
** hulegas
**
procedure huelgas
parameters ctahuel
sumase7 = sumase+sumase2+sumase3
ctahuel = (((cuota2*sumase2/1000) + (cuota3*sumase3/1000)) / (sumase7/1000)) * 0.1000

```

```
dedu7 = dedu
obser7 = obser
riesgo4 = .t.

return
**
** hurto
**

procedure hurto
parameters ctahurt
ctahurt = (((cuota2*sumase2/1000) + (cuota3*sumase3/1000)) / (sumase7/1000)) * 0.0500
sumase8 = sumase+sumase2+sumase3
dedu8 = 25
obser8 = '% S/PERDIDA'
riesgo5 = .t.

return
**
** descuento

procedure descuento
parameters desc
do msg_nw with '
g_menu1=' BAJA '
g_menu2=' MEDIA '
g_menu3=' ALTA '
do hmenu with 3,opcion,'HORDYLIM'
case opcion = 1
desc = desc + 0
case opcion = 2
desc = desc + 5
case opcion = 3
desc = desc + 10
endcase
do msg_nw with '
do hmenu with 3,opcion,'HVIGI'
do case
case opcion = 1
desc = desc + 0
case opcion = 2
desc = desc + 2
case opcion = 3
desc = desc + 4
endcase
```

ORDEN Y LIMPIEZA'

VIGILANCIA'

```
do msg_nw with '  
do hmenu with 3,opcion,'HEQAUX'  
do case  
  case opcion = 1  
    desc = desc + 0  
  case opcion = 2  
    desc = desc + 2  
  case opcion = 3  
    desc = desc + 4  
endcase  
do msg_nw with '  
do hmenu with 3,opcion,'HOPER'  
do case  
  case opcion = 1  
    desc = desc + 0  
  case opcion = 2  
    desc = desc + 1.5  
  case opcion = 3  
    desc = desc + 3  
endcase  
do msg_nw with '  
do hmenu with 3,opcion,'HMANT'  
do case  
  case opcion = 1  
    desc = desc + 0  
  case opcion = 2  
    desc = desc + 2  
  case opcion = 3  
    desc = desc + 6  
endcase  
do msg_nw with '  
do hmenu with 3,opcion,'HPROTEC'  
do case  
  case opcion = 1  
    desc = desc + 0  
  case opcion = 2  
    desc = desc + 1.5  
  case opcion = 3  
    desc = desc + 3  
endcase  
return
```

INSTALACIONES Y EQUIPOS AUXILIARES'

CONFIABILIDAD DE LOS OPERARIOS'

MANTENIMIENTO'

PROTECCIONES CONTRA INCENDIO'

## II.2.3 Procesamiento de Datos

Como mencionamos antes, el concepto de prueba de programas se ha modificado, enfocándose cada vez más a corregir detalles operativos que a determinar si el sistema ya codificado cumple o no las funciones para las que fue hecho.

Esto implica que los analistas y programadores están seguros de la funcionalidad de los programas aún antes de terminar con la codificación, por haberlos diseñado de manera estructurada.

Se prevee que el programa omita ciertos detalles o haga mal algunas cosas cuando se "corren" por primera vez en la computadora, más esto no implica que estén mal estructuralmente.

Viene entonces una etapa de corrección de detalles, hasta dejar al producto terminado con las características de elegancia y funcionalidad pactadas.

Puede también suceder, sin embargo, que se detecten errores estructurales cuando se ha terminado con la codificación de un sistema, en cuyo caso habrá que planear con cuidado las acciones a tomar. Puede ser que no se haya comprendido bien lo que se deseaba (lo cual es a todas luces catastrófico a estas alturas, y que se espera ya no suceda), o que no se haya hecho correctamente el análisis preliminar. El que ocurra cualquiera de las dos cosas es grave y habla de una mala planeación en las etapas del desarrollo. El único antídoto contra este mal consiste en una buena dosis de cuidado y moderación en el momento de la planeación (junto con una sólida experiencia que permita ofrecer proyectos robustos y maduros), por un lado, y un buen entendimiento de la metodología científica de programación y diseño por el otro.

Una vez obtenido el programa ejecutable por la computadora, es decir, libre de errores de sintaxis, podemos realizarle alguna serie de pruebas básicas, detectar y corregir aquellos errores de semántica que salten a la vista, mas sin embargo, ésto no es suficiente para garantizar que nuestro programa funciona correctamente. En el presente capítulo, ilustramos la ejecución del programa mediante el desarrollo de un ejemplo, posteriormente, en los capítulos que continúan, formalizaremos las técnicas de pruebas que se les realizan al software para detectar aquellos errores "ocultos". La manera de localizarlos y corregirlos, dependerán en gran parte de la habilidad y experiencia del analista o programador del sistema, según sea el caso.

Supongamos que se desea contratar el seguro de equipo electrónico para la Compañía INFONAVIT. El agente o asesor de seguros nos proporciona una relación del equipo existente, dicha relación se muestra en la figura 3.14.

Ahora bien, para efectos del desarrollo de nuestro ejemplo, supongamos que sólomente se desea esta contratación para el equipo Serie A-93 de las oficinas ubicadas en el Distrito Federal, la relación proporcionada por parte del asesor de seguros para éste equipo es mostrada en la figura 3.15.

Relación General del Equipo de Procesamiento  
de Datos utilizado por INFONAVIT.



EQUIPO DE PROCESAMIENTO DE DATOS Y EQUIPOS PERIFERICOS INSTALADOS EN OFICINAS CENTRALES  
Y DELEGACIONES REGIONALES QUE AMPARAN LA POLIZA 900710

<u>EQUIPO</u>	<u>TOTAL DLLS.</u>
1.- EQUIPO A-33 (HERMOSILLO SOV.)	501,263.44
2.- EQUIPO A-32 (MONTERREY N.L.)	503,426.33
3.- EQUIPO A-31 (GUADALAJARA, JAL)	525,158.60
4.- EQUIPO A-93 (OFICINAS CENTRALES)	1'266,694.26
5.- EQUIPO A-92 (OFICINAS CENTRALES)	1'183,020.18
6.- EQUIPO A-91 (OFICINAS CENTRALES)	1'176,771.66
7.- EQUIPO A-34 (OFICINAS CENTRALES)	496,747.08
8.- EQUIPO DE SOPORTE OPERACION (OFICINAS CENTRALES)	265,168.74
9.- DISCOS 207 (OFICINAS CENTRALES)	177,890.13
10.- DISCOS 24 (OFICINAS CENTRALES)	76,746.11
11.- UNIDADES DE CINTA BT (OFICINAS CENTRALES)	554,744.00
12.- MICROCOMPUTADORAS	2'886,779.00
13.- EQUIPO ELECTRONICO DE SOPORTE A COMPUTO	140,650.00
TOTAL	9'756,059.57

Figura 3.14

Coordinación de Información y Sistemas  
Departamento de Procesamiento de Datos  
Oficina de Operación

INVENTARIO DEL EQUIPO A-93

Inventario	Descripción	Serie	Modelo	Código	Factura	Costo
52051	Procesador Central	333947802	A9-FCP	CPU	24746	155,677.75
	Gastos de Instalación				24730	22,024.00
	Subtotal					183,701.75
	Subtotal M.N. (Tipo de cambio 288.7)					
	Derechos de Importación				13992	55,372.89
	TOTAL					239,074.64
52052	Gabinete de Memoria 24 MB	333947830	A9-MEM	MEMO	24746	88,217.75
	Memoria base 6 MB	NS 2000210-2	A9-MB		24746	67,460.00
	Memoria base 6 MB	NS 2000212-2	A9-MB		24746	67,460.00
	Incremento de Memoria 6 MB	NS 2000212-4	A9-MI		24746	62,271.00
	Expansión de Memoria 256 KB	NS 2000212-16	A378-2		24746	10,378.00
	Gastos de Instalación				24730	3,388.00
	Subtotal					299,174.75
	Subtotal M.N. (Tipo de cambio 288.7)					
	Derechos de Importación				13992 13994	61,468.36
	TOTAL					360,643.11
52055	ODT con Procesador Sist. de Ctrl	333947810	A9-ODT	SL5	24746	10,000.00
	Subtotal M.N. (Tipo de cambio 289.7)				13992	
	Derechos de Importación					1,278.04
	TOTAL					11,278.04
52056	ODT con Procesador Sist. de Ctrl	333947828	A9-ODT	SL6	24746	10,000.00
	Subtotal M.N. (Tipo de cambio 288.7)				13992	
	Derechos de Importación					1,278.04
	TOTAL					11,278.04
52057	Impresora 2000 L P M	303115865	B9246-20	LP15	24766	64,248.00
	TOTAL M.N. (Tipo de cambio 287)					64,248.00
52058	Impresora 2000 L P M	303315071	B9246-20	LP16	24766	64,248.00
	TOTAL M.N. (Tipo de cambio 288.7)					64,248.00
52059	No incluido en el Contrato (*)	100841105	B9499-2	TCU-B		
52060	No incluido en el Contrato (*)	100849561	B3499-2	TCU-D		
52061	No incluido en el Contrato (*)	100829729	B9495-32	MT21		
52062	No incluido en el Contrato (*)	100820661	B9495-32	MT22		
52063	No incluido en el Contrato (*)	100820604	B9495-32	MT23		
52064	No incluido en el Contrato (*)	100820653	B9495-32	MT24		

TOTAL: 1'266,694.26 dls.

Figura 3.15

Además de ello, consideremos que el asesor de seguros nos solicita que este equipo quede amparado contra los riesgos adicionales de huelgas, motín y/o, conmoción civil; terremoto y/o erupción volcánica; huracán, ciclón y/o tifón; y, hurto y/o robo sin violencia. No desea la cobertura de equipo móvil por razones obvias. Adicionalmente, desea contratar la Sección II (Portadores Externos de Datos) de la póliza con una Suma Asegurada de 200,000USCy y la Sección III (Incremento en el Costo de la Operación) con una responsabilidad para la Compañía Aseguradora de 1'250,000USCy.

De acuerdo a las políticas de la Compañía Aseguradora, debe efectuarse una visita de inspección al prospecto antes de aceptar el riesgo. El reporte del ingeniero inspector, entre aspectos de recomendaciones, fallas y conclusiones, nos destaca las siguientes observaciones:

- a) Excelente limpieza y orden en general.
- b) Vigilancia las 24 horas con control de acceso.
- c) Sistema ininterrumpido de energía eléctrica y sistema exclusivo de aire acondicionado.
- d) Experiencia de operarios de cinco años en el uso de estos sistemas.
- e) Mantenimiento preventivo dos veces al año por una compañía reconocida y equipo de mantenimiento disponible las 24 horas del día.
- f) Sistemas de alarmas en base a detección de calor y sistema contra incendios de Halón 1211 por medio de Blitz.

De acuerdo al procedimiento normal de cotización, el primer paso corresponde a la localización de éste equipo. Así pues, en la sección VII de la Tarifa de Seguro de Equipo Electrónico observamos un subcapítulo correspondiente a los sistemas de procesamiento de datos, de acuerdo a las características de este tipo de instalaciones y de su suma asegurada, obtenemos de la tabla de cotización la cuota de 2.8%, si continuamos sobre el renglón, a la derecha observamos que el deducible al equipo es del 1% de la suma asegurada, así mismo, obtenemos que la susceptibilidad al riesgo de terremoto es de grado "F", y finalmente, el comentario es que necesita ser inspeccionada como ya lo habíamos denotado.

Al observar los comentarios extraídos del reporte de inspección concluimos que podemos otorgarle a esta cuota un descuento del 30%, ya que de acuerdo a la tabla de descuentos a las características de seguridad del equipo así lo amerita. La cuota final resultaría en 1.96%.

De acuerdo a la petición del asesor de seguros, nos avocaremos al cálculo de la cuota de los riesgos adicionales. En la tabla para el cálculo de la cuota de terremoto, observamos que necesitamos la localización geográfica del equipo y su susceptibilidad a terremoto para obtener la cuota y deducible para este riesgo. Sabemos que su susceptibilidad es "F", como ya lo habíamos comentado, y por otro lado sabemos que se encuentra en la zona del Distrito Federal. El mapa de localización de zonas sísmicas nos indica que la zona del Distrito Federal es "Bdf", por lo tanto la cuota para dicho riesgo es de 0.9% y el deducible es del 1% de la suma asegurada.

Para el caso de Huracán, primeramente y de una forma similar al cálculo anterior, localizamos en el mapa la zona de huracán correspondiente al Distrito Federal. La tabla de cálculo correspondiente nos indica que ya está incluida en la cuota básica y que el deducible es el mismo.

En cuanto a la cuota para el riesgo de Huelgas, la tarifa nos indica que es el 10% de la cuota básica, por lo que el costo de la contratación de éste riesgo es de 0.28% con el mismo deducible.

Por otro lado, en el riesgo de Hurto, la tarifa nos indica que se trata del 5% de la cuota básica, es decir, el costo de la contratación del riesgo es de 0.14% de la suma asegurada y el deducible por causas imputables a este riesgo será del 25% sobre la pérdida.

Continuando con la petición del asesor de seguros, la cotización para la segunda sección de la póliza de equipo electrónico es de 3.2%, de acuerdo a la suma asegura. La susceptibilidad al riesgo de terremoto para equipo de almacenamiento secundario es de grado "F" y el deducible a aplicar en caso de siniestro es del 10% de la pérdida. Los riesgos adicionales para este equipo son calculados de una forma similar al de la sección I (Daños Materiales).

La tercer sección, está basada en un período de indemnización de 12 meses. De acuerdo a la suma asegurada solicitada por el asesor de seguros para esta sección, obtenemos de la tabla correspondiente la cuota de 4%. Nuevamente, el asesor de seguros nos informó que el tiempo máximo en que el asegurado repondría el total de su equipo en caso de siniestro es de 2 meses, por lo que se debe reconsiderar la cuota y así aplicar un factor reductor. Haciendo referencia a la tabla de factores reductores de esta sección, el factor es 0.65.

La cuota resultante y el costo del seguro para esta sección es de 2.6% aplicado a la suma asegurada de la sección. La misma tabla, para el cálculo de la sección II, nos indica que la cuota básica tiene incluido el riesgo de Huelgas y el costo del riesgo de terremoto es de 0.90% para la zona del Distrito Federal. En caso de siniestro el deducible que correspondería es de 2 días laborables. El cálculo de los demás riesgos adicionales, es similar al de la sección de Daños Materiales. En resumen obtenemos la memoria de cálculo mostrada en la tabla 3.5

Como hemos observado, es un tanto problemático estar haciendo esta serie de cálculos y búsquedas con tan solo la ayuda de unos manuales y calculadora. Se dificulta más este aspecto si consideramos que no es la única cotización de este equipo en una oficina, que se tienen que atender más cotizaciones de ramos de ingeniería, que son varias oficinas de servicio en las que no se tiene experiencia para el cálculo de estos ramos y que constantemente se cometen errores.

Instalemos nuestro software cotizador de ramos técnicos y alimentémoslo con los datos del ejemplo anterior.

El software se encuentra localizado en dos diskettes de 5 1/4", doble lado y doble densidad. El primero de ellos contiene al programa ejecutable EDDY.EXE, el cual consta de 210,368 bytes, y el autoejecutable HERATEC.BAT, el cual ejecuta un programa residente en memoria que simula una calculadora.

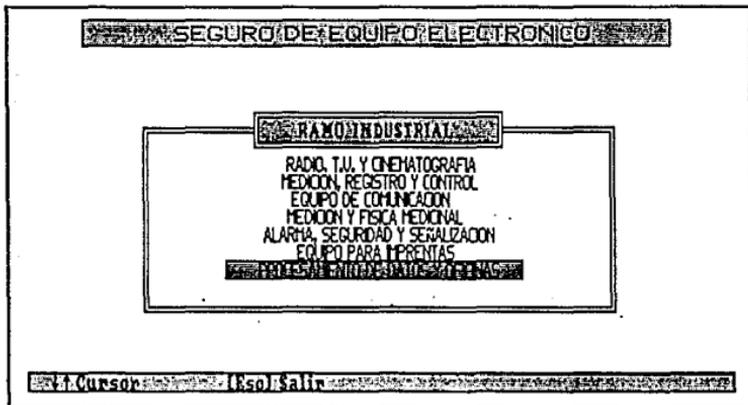


El segundo de ellos contiene las bases de datos EQ-ELEC.DBF, FLDHELP.DBF, HUR-CIV.DBF, HUR-MON.DBF, INUN-MON.DBF, MONT-IND.DBF, MONT-PLA.DBF, SORT-PLA.DBF, SORT-MAQ.DBF, TERR-CIV.DBF y TERR-MON.DBF. Además de ello contiene los archivos indexados FLDHINX.NTX, TABLA1.NTX, TABLA2.NTX, TABLA3.NTX, TABLA4.NTX, TABLA5.NTX, TABLA6.NTX, TABLA7.NTX, TABLA8.NTX y TABLA9.NTX. Una vez instalados estos archivos en el disco duro de una computadora P.C. lo ejecutaremos al teclear el pseudónimo HERATEC.

En la primer pantalla se observan los seis ramos de ingeniería. Precaución, si presionamos la tecla ESC saldremos del software.



Al posesionar el cursor en el ramo de Equipo electrónico obtenemos los giros que se manejan en las tarifas.



Si nos dirigimos al apartado de Procesamiento de Datos y Oficinas, obtenemos el siguiente desplegado.

INSTALACIONES PARA PROCESOS DE DATOS	
1.-	EQUIPO DE MICROFILMACION MENOR A 20,000 USCy.
2.-	EQUIPO DE MICROFILMACION MAYOR A 20,000 USCy.
3.-	TRANSPORTACION INTERNA (CORREO NEUMATICO).
4.-	EQUIPO DE OFICINA MICROS, COPIADORAS, ETC) MENOR A 20,000 USCy.
5.-	EQUIPO DE OFICINA MICROS, COPIADORAS, ETC) MAYOR A 20,000 USCy.
6.-	EQUIPO DE COMPUTO POR SEPARADO MENOR A 35,000 USCy.
7.-	EQUIPO DE COMPUTO POR SEPARADO MAYOR A 35,000 USCy.
8.-	INSTALACIONES DE COMPUTO COMPLETAS MENORES A 5,000,000 USCy.
9.-	INSTALACIONES DE COMPUTO COMPLETAS MENORES A 5,000,000 USCy.
10.-	INSTALACIONES DE COMPUTO COMPLETAS DE MAS DE MAS DE 10,000,000 USCy.

NUMERO DE RAMO LOCALIZADO? (Etc) Salvo

En este punto nosotros debemos indicar a la máquina, de acuerdo a las características, cuál es la instalación que corresponde. Para nuestro caso es la número 9. A continuación nos solicita la Suma Asegurada para la sección I de Daños Materiales.

**INSTALACIONES PARA ARROCES DE DATOS**

- 1.- EQUIPO DE MICROFILACION MENOR A 20,000 USCY.
- 2.- EQUIPO DE MICROFILACION MAYOR A 20,000 USCY.
- 3.- TRANSPORTACION INTERNA (CORREO NEUMATICO).
- 4.- EQUIPO DE OFICINA MICROS, COPIADORAS, ETC) MENOR A 20,000 USCY.
- 5.- EQUIPO DE OFICINA MICROS, COPIADORAS, ETC) MAYOR A 20,000 USCY.
- 6.- EQUIPO DE COMPUTO POR SEPARADO MENOR A 35,000 USCY.
- 7.- EQUIPO DE COMPUTO POR SEPARADO MAYOR A 35,000 USCY.
- 8.- INSTALACIONES DE COMPUTO COMPLETAS MENORES A 5,000,000 USCY.
- 9.- INSTALACIONES DE COMPUTO COMPLETAS MENORES A 5,000,000 USCY.
- 10.- INSTALACIONES DE COMPUTO COMPLETAS DE MAS DE MAS DE 10,000,000 USCY.

**INSTALACIONES DE COMPUTO MENORES A 5,000,000 VALORES EN USCY**

De la misma forma nos solicita la Suma Asegurada para la sección II, Portadores Externos de Datos , y sección III, Incremento en el Costo de la Operación. El software propone un porcentaje de la suma asegurada de la sección I para las secciones II y III, sin embargo, estas sumas pueden ser modificadas. En la sección III, el software requisita al usuario el período de indemnización, el período base, como ya lo hemos mencionado es de 12 meses.

A continuación el software pregunta si se efectuó inspección a las instalaciones. Si la respuesta es afirmativa, aparecen una serie de preguntas, que el usuario contesta de acuerdo al reporte de inspección, en cuanto a orden, limpieza, vigilancia, equipos auxiliares, experiencia de los operarios, mantenimiento y protecciones contra incendio.

INSTALACIONES PARA PROGRAMAS DE DATOS	
1.- EQUIPO DE MICROFILMACION MENOR A 20,000 USCy.	
2.- EQUIPO DE MICROFILMACION MAYOR A 20,000 USCy.	
3.- TRANSPORTACION INTERNA (CORREO NEUMATICO).	
4.- EQUIPO DE OFICINA (MICROS, COPIADORAS, ETC) MENOR A 20,000 USCy.	
5.- EQUIPO DE OFICINA (MICROS, COPIADORAS, ETC) MAYOR A 20,000 USCy.	
6.- EQUIPO DE COMPUTO POR SEPARADO MENOR A 35,000 USCy.	
7.- EQUIPO DE COMPUTO POR SEPARADO MAYOR A 35,000 USCy.	
8.- INSTALACIONES DE COMPUTO COMPLETAS MENORES A 50,000 USCy.	
9.- INSTALACIONES DE COMPUTO COMPLETAS MENORES A 50,000,000 USCy.	
10.- INSTALACIONES DE COMPUTO COMPLETAS DE MAS DE MAS DE 10,000,000 USCy.	

BAJA MEDIA **ALTA** **ORDEN Y LIMPIEZA**

(←) Seleccionar (←) Cursor (Esq) Salir

Es importante destacar que si se tiene duda sobre cuál opción escoger, el software proporcionará ayuda al usuario mediante la presión de la tecla F1. Al término de esta serie de preguntas referentes a la inspección, el software solicitará si se desean los riesgos adicionales. De nueva cuenta, si la respuesta es afirmativa, se despliega un menú que contiene los cinco riesgos que se manejan para este tipo de seguro.



Una vez que se ha tomado la opción de alguno de los riesgos, el software formulará alguna preguntas para el cálculo de la cuota correspondiente. Por ejemplo, tomemos Terremoto y Erupción Volcánica. En seguida el software nos solicita la Zona Sísmica y ello es suficiente para el cálculo.

SEGURO DE EQUIPO ELECTRONICO

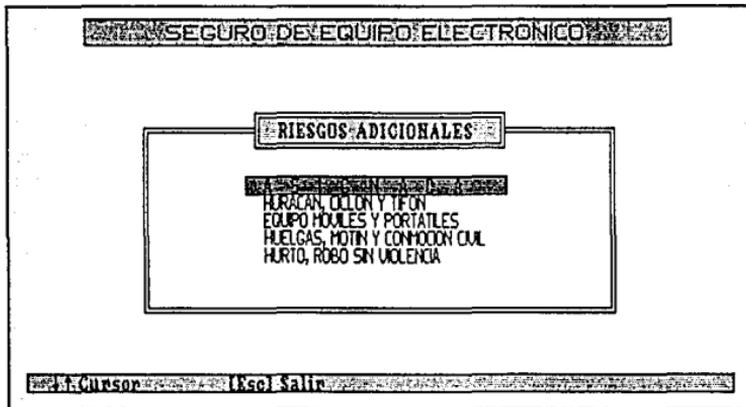
RIESGOS ADICIONALES

TERREMOTO Y ERUPCION VOLCANICA  
HURACAN, CICLON Y TIFON  
EQUIPO MOVILES Y PORTATILES  
HUELGA, MOTIN Y CONMOCION CIVIL  
HURTO, ROBO SIN VIOLENCIA

A Bm C D INDICA LA ZONA SISMICA

Seleccionar Cursor Esc/Salir

Una vez que se termine de asignar el riego, aparecerá en pantalla un mensaje de confirmación.



De una forma semejante son asignados cada uno de los riesgos adicionales. Al término de asignar los riesgos adicionales solicitados por el asesor de seguros, presionamos la tecla ESC e inmediatamente el software nos presenta el resultado de la cotización. En éste se puede observar los costos por cobertura, las sumas aseguradas, las cuotas, los deducibles correspondientes y las condiciones a que queda sujeto, en nuestro caso está sujeto a la inspección. Finalmente, el software nos proporciona la opción de aumentar el deducible para abatir el costo del seguro.

PREPORA 22/10/91				
COBERTURAS	S. A. (US Cy)	CUOTA (%)	PRIMA (US Cy)	DEDUCIBLE
BASICA	1266,694	1960	2483	100% S/EQUIPO
SECCION III	200,000	3,200	640	10,00% S/PERDIDA
SECCION IIII	1250,000	2,500	3,250	2,00% DIAS LABORABLES
TERREMOTO	2,716,694	0,900	2,445	SEGUN SECCION
HURACAN	2,716,694	0,000	0	SEGUN SECCION
HUELGA	2,716,694	0,154	419	SEGUN SECCION
HURTO	2,716,694	0,137	372	SEGUN SECCION
PRIMA NETA			9,609	

CONDICIONES : Sujeto a inspección por oficinas

~~DESEAJUMENTO DE DEDUCIBLE (S/N)?~~

Como hemos observado el resultado por medio del software es mucho más rápido que el procesamiento de datos manual.

El procesamiento de datos para los demás ramos de ingeniería, es similar al descrito anteriormente. El software, nos solicitará la información necesaria para el cálculo de cada uno de los ramos y si se tiene algún tipo de duda, no hay que olvidar que con tan sólo presionar la tecla F1 recibiremos ayuda, por lo que ahorraremos tiempo en consultas de tarifa.

## III.3 Pruebas

Las Pruebas del software presenta una antagonía interesante para la ingeniería. Durante los pasos de planeación y desarrollo, la ingeniería es aplicada en la construcción del software, desde un concepto abstracto hasta una tangible implementación. Con la aplicación de Pruebas, la ingeniería crea casos particulares para la "destrucción" del software que ha sido construido. En efecto, la Prueba es un paso en la Ingeniería del Software que podría ser visto como un elemento destructivo.

En su libro Pressman, nos menciona una serie de objetivos que son enfocados a las pruebas del software:

- 1.-La Prueba es un proceso de ejecución de un programa con la intención de obtener un error.
- 2.-Un buen caso de Prueba es aquel que tiene una alta probabilidad de obtener en un instante un error escondido.
- 3.-Una Prueba suficiente es aquella que descubre en un instante un error escondido.

Así mismo, tomando como base los puntos anteriores, Pressman nos indica que el objetivo primordial es el diseño de una Prueba que sistemáticamente descubra diferentes clases de errores.

Antes de continuar con el proceso del diseño de la Prueba, debemos diferenciar los caminos por los cuales se efectúan pruebas a cualquier tipo de producto de la ingeniería. El primero de ellos está basado en el conocimiento de la función específica para el cual fué diseñado el producto, la Prueba está enfocada a demostrar que cada una de las funciones es completamente operativa. El segundo camino está basado en el conocimiento de los trabajos internos del producto, la Prueba es manejada para asegurar que "todas las piezas encajen", es decir, que la operación interna se ajuste a las especificaciones (que los componentes internos sean comprobados de una forma adecuada). El primer camino es nombrado como *prueba de la caja negra* y el segundo de ellos como *prueba de la caja blanca*.

Teniendo en cuenta los dos conceptos anteriores, continuemos con nuestro análisis del proceso de diseño de la Prueba. La Prueba, dentro del contexto de la Ingeniería de Sistemas, está constituido por una serie de cuatro pasos, que son implementados secuencialmente, figura 3.16.

Inicialmente, la Prueba está enfocada a la prueba de cada uno de los módulos por individual, asegurándose de que sus funciones propias actúen como una unidad; a este paso se le nombra *Prueba de unidad*. A continuación, los módulos deben ser ensamblados o integrados para formar el paquete completo del software. La *Prueba de integración* verifica que no haya errores de ensamblado. Finalmente, son validados los requerimientos del software (establecidos durante la fase de planeación). La *Prueba de validación* comprueba que los requerimientos sean satisfechos por el software.

Una vez validado, el software debe ser combinado con otros elementos del sistema (hardware e información). La Prueba del sistema confirma que todos los elementos alrededor y sus funciones sean compatibles.

Durante el desarrollo del presente capitulo analizaremos los modelos para el diseño de cada uno de los tres primeros pasos del desarrollo de la Prueba. El cuarto de ellos, la Prueba del sistema, solamente comentaremos que, el software aquí desarrollado es un elemento de un sistema, el cual debe de cumplir con los requerimientos solicitados en la fase de planeación para que el sistema cumpla con su objetivo funcional.

## Pasos para Tests

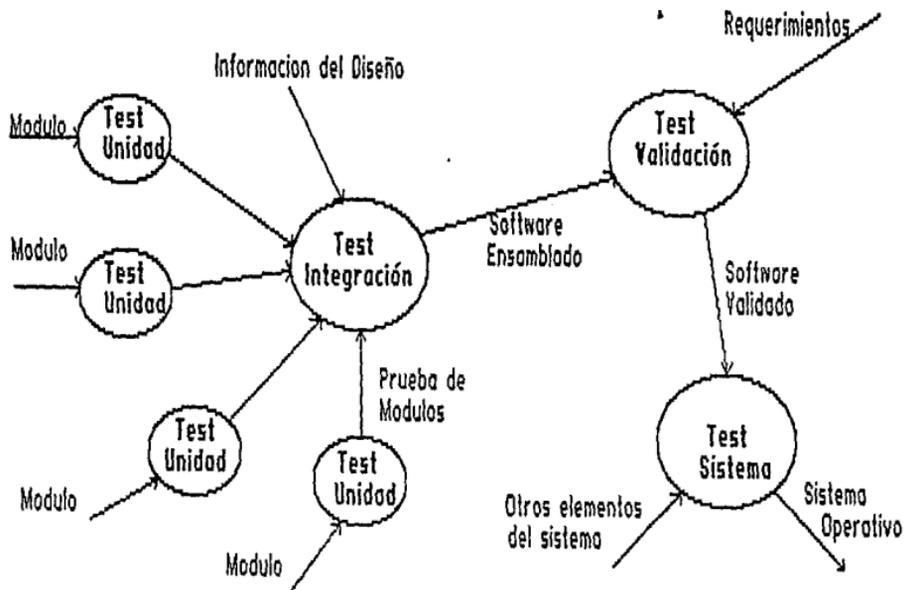


Figura 3.16

## III.3.1 La Prueba de Unidad

La Prueba de unidad enfoca los esfuerzos de verificación a la unidad más pequeña del diseño del software, el módulo. Con la descripción del diseño detallado usado como una guía son probados los pats de control importantes. La Prueba de .unidad es una prueba de caja blanca y este paso puede efectuarse en paralelo a varios módulos.

Las cinco características básicas que son evaluadas durante la Prueba de unidad son:

- a) Las interfaces del módulo: Es frecuente que surgan errores durante la comunicación entre módulos, por ello es importante verificar que el paso de parámetros sea correcto.

De una forma general y como ejemplo, en el programa de equipo electrónico podemos observar las características siguientes, tanto en los argumetos transmitidos por el programa principal como en los parámetros recibidos por el módulo que calcula la cuota básica de las tres secciones:

- el número de argumentos es igual al número de parámetros, es decir, uno;
- la definición del tipo para los argumentos transmitidos es idéntico al de los parámetros, de tipo real;

- la longitud de las variables transmitidas y recibidas son las mismas, longitud de la palabra de 16 bits;
- el orden en que se envian los argumentos corresponde al de los parámetros recibidos, en este caso es tan sólo una variable;
- y finalmente, debe relacionarse el nombre del argumento con el del parametro, "cuota" es el nombre del argumeto y "ctabas" el del parámetro.

Además de verificarse las características anteriores hay que asegurarnos de que las variables globales sean utilizadas en forma correcta dentro del módulo analizado.

- b) Estructura de datos local: Debemos verificar que nuestras estructuras de datos sean manejadas correctamente en nuestro programa. Así pues, la variable "registro" está definida como un tipo entero y es utilizada para tal objeto. Es recomendable que el nombre de las variables sugiera su uso, como al ver la variable "registro", sabremos de entrada que su utilización es almacenar el número del registro y por tanto debe ser de tipo entero.

- c) Ejecución de pats importantes: Aquí es donde se prueban y se descubren los errores de manipulación de las variables y/o control del flujo del programa. En nuestro caso, el módulo que calcula la cuota básica está conformado por dos ciclos anidados, el primero compara y efectua el proceso hasta que la variable "registro" sea modificada. El segundo se encuentra dentro del primer ciclo y efectua un proceso de búsqueda hasta terminar con el archivo. Los otros pats importantes y que vale la pena verificar su funcionamiento son las comparaciones tales como los "if" y "case"; dentro de nuestro módulo existen cuatro "case" y dos "if" que nos sirven para la asignación de cuotas y comparación de sumas aseguradas respectivamente.
- d) Manejo de errores: En nuestro software, debemos asegurar que el manejo de los mensajes de error sea legible y que corresponda al error. Como un ejemplo, tenemos el mensaje "fuera de cobertura, consultar con oficina matriz" cuando se verifica el error de cotizar un producto fuera de políticas.
- e) Condiciones de frontera que afecte a las características anteriores: Los errores tienden a darse más en los límites que en el "centro". Un análisis de valor límite nos lleva a la elección de casos de prueba en los "bordes".

## III.3.2 Prueba de Integración

La Prueba de integración es una técnica sistemática en el ensamblado del software mientras que al mismo tiempo las pruebas estan enfocadas a descubrir errores asociados con las interfaces. El objetivo es tomar los módulos ya verificados por la Prueba de unidad y construir la estructura del software que ha sido dictada por el diseño.

Existen dos técnicas complementarias, utilizadas por la Prueba de integración, la Integración Alta (*top-down*) y la Integración Baja (*bottom-up*). La selección de cualquiera de las dos técnicas depende de las características del software y a veces de la administración del proyecto.

El uso de la aproximación Alta es frecuente para los niveles altos de la estructura del software, así mismo, la aproximación Baja, es utilizada con frecuencia para los niveles subordinados.

La integración Alta es un proceso efectuado por una serie de cinco pasos, como así lo sugiere Pressman:

- 1.-El módulo de control principal es usado como un *Manejador de Prueba (test driver)*, y los *Módulos Prueba (stubs)* son sustituidos por todos los módulos directamente subordinados.
- 2.-Los Módulos Prueba subordinados son reemplazados uno a uno por los módulos reales.

- 3.-Las pruebas son efectuadas para cada nuevo módulo que es integrado.
- 4.-Al completar el conjunto de pruebas, otro Módulo Prueba es reemplazado por el módulo real.
- 5.-Regresión de Prueba (Inicialización de algunas o todas las pruebas), esto es para asegurar de que nuevos errores no hayan sido introducidos.

El proceso continua desde el segundo paso hasta que la estructura del software así lo requiera.

Durante el detalle del proceso Pressman nos hace alusión al Manejador de Prueba y/o al Módulo Prueba. En muchas ocasiones un Manejador de Prueba no es más que un "programa principal" que acepta los datos para las pruebas, pasa estos datos al módulo a ser probado e imprime los resultados relevantes. Los Módulos Prueba sirven para reemplazar módulos que son subordinados al modulo a ser probado. Un Módulo Prueba o "programa fantasma (dummy)" usa las interfaces de los módulos subordinados, efectua una manipulación mínima de los datos, imprime verificaciones de entrada, y regresa el control al módulo superordinario.

Para ilustrar el proceso, hagamos referencia a nuestra arquitectura de software, figura 3.12, el módulo etiquetado como "sistema cotizador" nos representaría el Manejador de Prueba y los módulos subordinados nos representan los Módulos Prueba. Una vez efectuadas las pruebas al programa principal, se procede a reemplazar a S2 por el módulo real M2 (Menús de recepción e inicialización). Se efectúan pruebas. Se reemplaza a S3 por M3. Se efectúan nuevamente las pruebas. Se continúa con este proceso hasta integrar los niveles altos del paquete, figura 3.17.

Por otro lado la integración Baja puede ser implementada por medio de los siguientes pasos:

- 1.- Los módulos de niveles bajos son combinados y forman una *Sección de Prueba (cluster)* la cual efectúa una subfunción específica del software.
- 2.- Un Manejador de Prueba es escrito para coordinar las pruebas de entrada y salida.
- 3.- Las Secciones de Prueba son comprobadas.
- 4.- Los Manejadores de Prueba son removidos y las Secciones de Prueba son combinadas moviéndose hacia arriba y conectándose a la estructura del software.

En nuestro software lo manejamos de la siguiente forma: Las Secciones de Prueba en la figura 3.18 son combinadas y es escrito un Manejador de Prueba que les conduce, son efectuadas las pruebas correspondientes, y finalmente son removidos D1, D2 y D3 para conectar las Secciones de Prueba a la estructura del software. Este proceso continúa hasta integrar los niveles bajos del paquete.

Integracion Top-Down

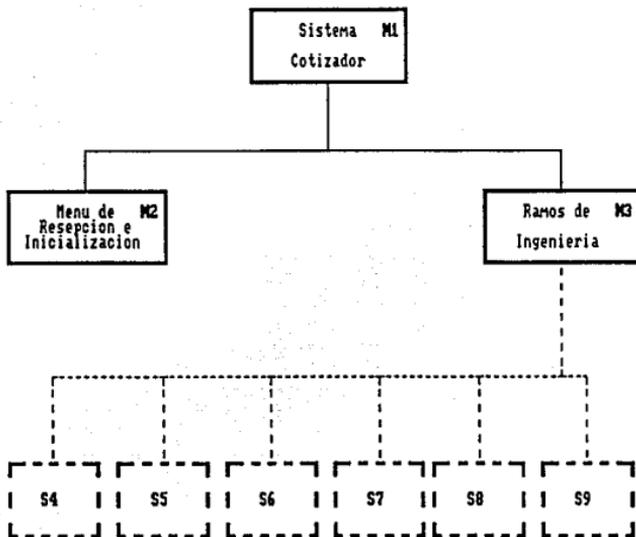


Figura 3.17

Integracion Bottom-Up

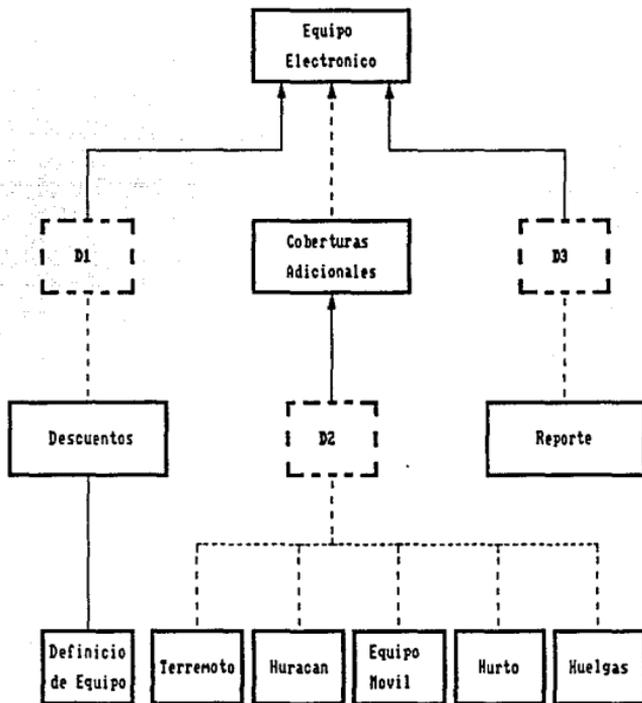


Figura 3.18

### III.3.3 Prueba de Validación

La Prueba de validación es la culminación de una serie de pruebas. Al término de la Prueba de integración el software ha sido ensamblado, sin errores de interface, es aquí en donde comienza la validación del software y verifica que los requerimientos establecidos en la fase de planeación correspondan a los resultados del software. La validación del software es una serie de pruebas de caja negra que demuestran y confirman el cumplimiento de los requerimientos.

En el capítulo II.2.1 hemos expuesto los requerimientos del usuario a manera de objetivos para el software, en base a este punto de vista, podemos discernir que se han cubierto satisfactoriamente. Nosotros podemos intuir tales conclusiones al comparar el manejo y resultados de nuestro software con tales requerimientos.

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

Si nos hemos tomado el trabajo de planear cuidadosamente un software y de transformarlo en un conjunto bien estructurado de programas y módulos, es por que seguramente va a tener una vida útil prolongada y no va a ser utilizado tan sólo una o dos veces. Este simple hecho nos obliga a considerar un esquema de mantenimiento que asegure que el modelo ya sistematizado evolucione a un ritmo parecido al que lo haga la realidad que está siendo simulada. Tal vez llegue el momento en que ese proceso o aspecto de la realidad para el que se construyó el software haya cambiado cualitativamente, en cuyo caso se habla del término de la vida útil del software.

Debido a su reciente creación, no hemos practicado el mantenimiento al software, mientras tanto, sin embargo, hay que ser capaces de hacer alteraciones no estructurales al software con un mínimo de costo en recursos de análisis y programación; lo cual de alguna manera tenemos asegurado si el software ha sido construido de manera modular y estructurada y si se dispone de la documentación adecuada que lo describa tanto en su diseño como en su uso. Diremos que si un software sólo es entendido por su creador, es un mal software.

En los capítulos precedentes introducimos los tipos de mantenimiento realizados en la vida útil de un software y finalmente proponemos una hoja técnica para su mantenimiento.

### IV.1 Introducción

El mantenimiento del software es, por supuesto, algo más que la "corrección de los errores". Debido a que no es razonable asumir que las pruebas efectuadas al software hayan descubierto todos los errores latentes en el software mismo. Al proceso que incluye el diagnóstico de uno o más errores se le denomina *mantenimiento correctivo*.

Por el rápido cambio inherente a todo aspecto de la informática, las nuevas generaciones de hardware, los nuevos sistemas operativos o nuevas versiones, la frecuente mejora o modificación de los equipos periféricos y otros elementos de sistemas, y tomando en consideración además que la vida útil del software de aplicación puede fácilmente sobrepasar los diez años, haciéndose obsoleto para el entorno del sistema para el que fue originalmente desarrollado. Por tanto, el modificar el software para que interaccione adecuadamente para su entorno cambiante es tanto necesaria como corriente, a esto se le denomina *mantenimiento adaptivo*.

A medida que el software es usado, se reciben recomendaciones por parte de los usuarios sobre nuevas posibilidades, algunas modificaciones de funciones ya existentes y mejoras en general. Para satisfacer estas peticiones se lleva a cabo el *mantenimiento perfectivo*.

## IV.2 Bitácora

Como hemos mencionado con anterioridad, los mantenimientos antes definidos son practicados en la vida útil del software. Es importante llevar una relación de los mantenimientos efectuados al software para, en un momento dado, apreciar su rentabilidad y tener un control del historial del mismo. Nos auxiliamos de una hoja técnica denominada *Bitácora de mantenimiento*.

Para el presente software proponemos la bitácora mostrada en la figura 4.1. Esta hoja técnica es un modelo representativo del mantenimiento de algún software en general. Cada módulo o función de la arquitectura de nuestro software es representado mediante la bitácora y la persona responsable del mantenimiento se hace cargo de las modificaciones necesarias.

En la hoja técnica, podemos observar que se lleva un control de folio el cual es asignado por cada uno de los nuevos requerimientos del software. En la columna siguiente se define el tipo de mantenimiento, ya sea correctivo, adaptivo o perfectivo, en seguida se escriben las fechas de cuando el requerimiento fue solicitado, cuando debe iniciarse el proceso y la propuesta de terminación.

El efectuar modificaciones al software no solamente implica el modificar el código, sino más bien, aplicar nuevamente los pasos de la ingeniería de sistemas y documentarlos, la hoja técnica nos representa una columna para el proceso de la modificación del software, este proceso se divide en la fase de planeación y la fase de desarrollo e implementación.

La fase de planeación a su vez la podemos subdividir en el planteamiento del problema, análisis del mismo, las soluciones propuestas y su factibilidad. Por otro lado, la fase de desarrollo e implementación la podemos subdividir en diseño, codificación, Pruebas y documentación. Dentro de los diseños podemos diferenciar los preliminares y los detallados. En cuanto a las Pruebas, podemos diferenciarlas en unidad, integración y validación.

En cada uno de los rectángulos presentados por la hoja técnica se anota la fecha o es simplemente sombreado al momento de haber cumplido con ese paso.

Finalmente, se dispone de un espacio en la última columna para las observaciones o comentarios de la persona responsable sobre la satisfacción del requerimiento.

Mediante la implementación y el buen hábito de uso de la hoja técnica llevamos un control sobre el mantenimiento de nuestro software. Así se cumple con una doble función al considerarla como un registro estadístico.

# Bitácora de Mantenimiento

SISTEMA COTIZADOR DE RAMOS TECNICOS		
BITACORA DE MANTENIMIENTO		

MODULO O FUNCION	RESPONSABLE	HOJA No.
------------------	-------------	----------

No. FOLIO	TIPO DE MENTENIMIE	FECHAS (ENTRADA INICIO, FIN)	P R O C E S O										
			P L A N E A C I O N				D E S A R R O L L O E I M P L E M E N T A C I O N						
			PROBLE	ANALIS	SOLUCI	FACTIB	DISEÑOS		CODIFI	TEST			DOCUMENT
							PRELIM	DETAJL		UNIDAD	INTEGR	VALIDA	

OBSERVACIONES GENERALES:

Figura 4.1

---

C A P I T U L O    V  
C O N C L U S I O N E S

La computadora representa, de alguna forma, el genio encerrado en la botella, capaz de cumplir con los deseos de rapidez y eficiencia en el cálculo y la organización de grandes masas de datos, nos indica Levine en su libro, y continua: Y esto es precisamente la idea original, un genio mecánico capaz de liberar al hombre de la pesada tarea de tener que calcular a mano.

Nosotros pensamos que la computadora es algo más que un conjunto de circuitos integrados para realizar supercálculos. La computadora es mucho más que una calculadora compleja, puesto que esta basada en una idea mucho más potente, el modelaje por medio de programas. Tal vez la idea central de la computadora consista en posibilitar el hallazgo de la representación adecuada para la descripción de un problema o de un fenómeno.

Es evidente que siempre será posible describir algún aspecto de la realidad por medio de cierto lenguaje; basta con encontrar las combinaciones adecuadas de símbolos para representar lo que tenemos en mente.

Escribir programas para una computadora es una actividad que requiere de una buena cantidad de esfuerzo mental y dedicación de tiempo. Empezar un camino de volver realidad la construcción de modelos para representar descripciones de fenómenos o procesos del mundo real implica una metodología científica, repetible y comprobable, para llegar a su fin. De otra forma, agudizaremos el problema de la crisis del software.

La ingeniería de sistemas nos marca la pauta científica para la solución de la crisis del software y por ende el método científico para la representación de dichos modelos.

Mediante la implementación del software "HERATEC" hemos definido, desarrollado y verificado las fases del ciclo de vida de un software. La aplicación de las técnicas y herramientas sugeridas por Pressman han facilitado nuestro entendimiento para con un problema íntegro y como consecuencia han hecho viable la realización de un modelo capaz de solucionar nuestro problema real. Hemos cumplido con un objetivo: desarrollar un sistema cotizador de ramos técnicos para las compañías aseguradoras.

Finalmente, cabe destacar, que hemos cumplido satisfactoriamente con los objetivos listados en los requerimientos del programa durante la fase de planeación y que gracias a ello podemos evaluar a nuestro software como:

- Util: Satisfacción plena de lo deseado por las personas vinculadas con la utilización del software.
- Bajo costo de operación: Solamente requiere de una microcomputadora PC compatible.
- Seguro y confiable: La usuarios no necesitan de un gran conocimiento sobre computación. El manejo de pantallas es por medio de opciones y ventana de ayudas.
- Modular: El software esta constituido por subrutinas con funciones específicas que facilitan su mantenimiento.
- Compatible: Puede utilizarse en diferentes marcas de microcomputadoras PC compatibles y por ende es flexible.

---

G L O S A R I O

M I N I M O

## Algoritmo

Desde un punto de vista elemental, un algoritmo no es más que la especificación, detallada y libre de ambigüedad, de un proceso, es decir, un conjunto de pasos que hay que seguir para llegar a cierto fin medible o comprobable.

Los algoritmos son la base sobre la cual se escriben los programas, que una computadora ejecutará para resolver los problemas que se plantean; un medio para el diseño de algoritmos es el pseudocódigo.

## Análisis de Sistemas

Es una metodología para especificar funcionalmente un sistema. De hecho, es una actividad de solución de problemas que conlleva a una relación intrínseca entre el que requiere y el que desarrolla el software, y comprende los puntos: 1) Seleccionar un objetivo y 2) Producir una documentación del objetivo de tal forma que se pueda evaluar la implementación.

## Banco de Datos, Bases de Datos

El término Base de Datos puede usarse como sinónimo de Banco de Datos o Banco de Información, y se refiere a un conjunto de archivos organizados de tal forma que permitan guardar y extraer información útil por medio de programas especiales.

Puede también emplearse de forma más técnica y referirse entonces a un sistema de uso general que sirve para crear y mantener banco de datos sin necesidad de escribir programas específicos para manejarlos, sino usando las facilidades integradas del manejador de las bases de datos. Un manejador de bases de datos es entonces un sistema complejo que se encarga de relacionar entre sí los diversos archivos de un banco de información para que éste se comporte como si estuviera dotado de cierta inteligencia que le permita responder preguntas acerca de sus contenidos.

**Codificación**

Con esta palabra se designan varias cosas, en distintos niveles de complejidad. Al nivel más sencillo, codificar es realizar un proceso mediante el cual se traduce información de un lenguaje a otro, usando un diccionario de equivalencias. Tratándose de programación de computadoras en general, la codificación es la última etapa del complejo conjunto de acciones requeridas para resolver un problema por medios computacionales.

**Código**

Se llama así a un programa que está escrito en lenguaje máquina. Los programadores escriben el programa fuente en algún lenguaje de programación y es entonces el ensamblador o compilador el encargado de generar el código, es decir, de traducir el lenguaje fuente a lenguaje máquina.

**Compilador**

Programa que sirve como traductor entre un lenguaje de programación y el lenguaje máquina de una computadora.

**Diagrama de Flujo de Datos**

Herramienta gráfica del análisis de sistemas, y tiene como objetivo mostrar las transformaciones de los datos a medida que éstos fluyen a través de los procesos del programa, es decir, ayuda a analizar los cambios que ocurren a los datos de entrada a fin de lograr las salidas deseadas.

## **Diagrama de Flujo de Programa**

Método gráfico para describir algoritmos. En principio, todo algoritmo puede representarse mediante un diagrama de flujo de programa, pero en la práctica este acercamiento tiene muchas desventajas, mismas que han conducido a su casi total abandono, en favor de la técnica del pseudocódigo.

## **Diccionario de Datos**

En el aspecto de la programación nos referimos a la lista de rútilos o claves que se utilizan en un programa y descripción de sus significados lógicos con fines de documentación.

En el ambiente de las Bases de Datos el diccionario de datos define toda información de cantidad en términos de información que es usada para desarrollar la cantidad.

## **Diseño Detallado**

El diseño detallado provee un camino para el código. Con el uso de una técnica de diseño que puede ser gráfica, tabular o textual una especificación a detalle de los procedimientos para el software es creado. El diseño detallado provee suficiente información para algún otro que el diseñador a desarrollado resultando el código fuente.

## **Diseño Estructurado**

Conjunto de métodos para construir sistemas de información o, en general, para resolver problemas por medio de una computadora. El diseño estructurado parte de la base de que los programas se escriben estructuradamente, pero no se limita a consideraciones locales al programa, sino que estudia las relaciones entre todos los programas que constituyen un sistema.

### **Diseño Heurístico**

Arte de inventar. Método de ensayo y error que se vale de reglas empíricas para encontrar la solución a un problema evaluando por etapas los procesos hechos a lo largo de su curso.

### **Diseño Preliminar**

El paso del diseño preliminar es un antecedente para el desarrollo del software. El flujo de información o estructura, determinado por los requerimientos, proporcionan una herramienta para una representación general del software. Esta representación, llamada estructura del software o arquitectura, puede ser evaluada, redefinida, y/o modificada fácilmente en la fase de desarrollo.

En adición a la estructura del software, el diseño preliminar define las interfaces de elementos internos del software y elementos externos del sistema, descripción funcional de cada elemento del software, detalles de la estructura de datos y consideraciones especiales.

### **Documentación de Sistemas**

Colección de descripciones y procedimientos escritos que proporcionan información y orientación sobre un programa o sobre todo un sistema de computación o parte de él para que pueda ser utilizado y mantenido correctamente.

### **Ensamblador**

En su primera acepción, un ensamblador es una programa que recibe programas escritos en un lenguaje (que también se llama ensamblador) y los traduce al lenguaje de máquina.

La segunda acepción se refiere precisamente al lenguaje que reconoce un programa ensamblador, y que constituye el segundo nivel (en complejidad creciente) de los lenguajes con que se puede programar una computadora.

### **Entidades y Atributos**

Una entidad es una persona, un lugar, una cosa, un evento o un concepto acerca del cual se registra información. Ejemplo: En un almacén las entidades podrían ser **PROVEDORES, PARTES, ENVIOS** y otras similares.

Toda entidad tiene algunos atributos básicos que la caracterizan, el número de identificación del cliente, el nombre del cliente y la dirección del cliente son tres atributos que describen a la entidad **CLIENTE** en un banco. Al atributo frecuentemente se le llama elementos de datos, campo de datos, campo ítem de datos o ítem elemental.

### **Estructura del Software, Arquitectura del Software**

Conjunto de componentes específicos que constituyen a un sistema y el modo como interactúan.

### **Estructuras de Control**

Construcciones mediante las cuales se escriben programas. Cuando las estructuras de control se definen y emplean de manera ordenada y sistemática se habla de programación estructurada. No todos los lenguajes de programación disponen de estructuras de control completas o complejas, por lo que la tendencia es abandonar los que son deficientes en este punto.

## **Estructuras de Datos**

Métodos que se emplean en programación para organizar y representar la información en una computadora. En términos generales, un programa está formado por el algoritmo y las estructuras de datos. Todos los lenguajes de programación disponen de al menos de un tipo de estructura de datos, y mientras más estructuras diferentes de datos ofrezca un lenguaje, más fácil y versátil se convierte la tarea de escribir algoritmos por medio de él.

## **Estructura Física y Lógica de la Base de Datos**

La estructura conceptual nos representa los requerimientos de los usuarios individuales como una unidad íntegra.

La versión de la estructura conceptual que se adapta al sistema de manejo de base de datos es llamada estructura lógica.

Finalmente, la estructura física, toma en cuenta la distribución de los datos, los métodos de acceso, etc.

## **Hardware**

Conjunto de elementos y sistema electrónicos que forman un sistema de cómputo.

## **Informática**

Conjunto de conocimientos, métodos y sistemas para el manejo computarizado de la información en las organizaciones. Se ha hecho un uso abusivo y ligero de este término, confundiéndolo con campos que sólo lo influyen de manera marginal. La parte del problema estriba en la holgura con la que se emplea este término, donde unos entienden "administración moderna", otros "derechos a la información" y otros "sistemas de información por medio de computadoras".

## Ingeniería de Sistemas

La ingeniería de sistemas es una disciplina que se dedica a la aplicación de técnicas para la definición, análisis y mantenimiento de proyectos de cómputo. Todos estos relacionados específicamente con la implementación de programas o sistemas, esta disciplina ha sido el producto de tres décadas de enormes cambios en el campo de la computación.

## Lenguaje de Alto Nivel

Llámesese así a los lenguajes e programación que están por encima (en poder expresivo) del lenguaje de máquina y del lenguaje ensamblador.

## Lenguaje de Máquina

La única manera de comunicarse con el procesador de una computadora es por medio de un programa directamente ejecutable, mismo que debe estar escrito forzosamente en este lenguaje. Sin embargo, el lenguaje de máquina no es propiamente un lenguaje, porque carece de estructura; podría describirse más apropiadamente como "conjunto de signos aislados ejecutables".

## Lenguaje de Programación

Nombre genérico que se aplica a cualquier lenguaje (fuera de la máquina) disponible para escribir programas para una computadora. Si le quitamos el ensamblador, entonces este concepto es idéntico al de lenguaje de alto nivel". De forma aparentemente paradójica, escribir un programa en un lenguaje de programación es la última etapa del proceso conocido comunmente como programar.

**Llave, elementos clave de datos, idetificadores de entidad**

Algunos campos de datos tienen la propiedad de que, al conocer el valor tomado por un campo particular de una entidad, es posible identificar los valores tomados por otros campos de datos de la misma entidad. Por ejemplo, una vez conocido el número de identificación de un cliente en un banco, podemos determinar la persona en cuestión y tipo de cuenta.

**Mantenimiento**

Plan o servicio para preveer, aislar y corregir problemas de mal funcionamiento en un sistema por medio de revisiones frecuentes, pruebas y remplazos.

**Módulo**

Nombre que denota un conjunto o secuencia de instrucciones para realizar una tarea determinada. Este nombre implica una relación entre la teoría y la práctica de la programación.

**Nodo**

En una red, dispositivo o grupo de dispositivos del equipo que une dos o más unidades adicionales a la red. En una estructura de datos, un ítem de datos que proporcione un vínculo con otros ítems. En teoría de grafos, punto de unión.

En la gráfica de la estructura del software le hemos representado físicamente como un rectángulo, el cual nos define una función específica.

**Paquetes de Programación**

Con este nombre se conocen diversos sistemas disponibles en el mercado de computadoras, diseñados para resolver problemas específicos que aparecen normalmente en la organizaciones, tales como cálculos y pago de nóminas, control de inventarios, clientes, etc.

**Pat**

Nombre genérico para la representación de un proceso. En el diagrama de flujo de datos se representa graficamente como un círculo o "burbuja" etiquetada

**Programa Fuente**

Conjunto de instrucciones escritas en algún lenguaje de programación, y que sirven para modelar o solucionar un problema.

**Programa Objeto**

Conjunto de instrucciones escritas en lenguaje de máquina.

**Programación de Sistemas**

Nombre genérico con el que se conoce el enorme conjunto de programas y sistemas que se encargan de la operación automática de las computadoras. Los componentes más importantes de la programación de sistemas son los ensambladores, los compiladores, y los sistemas operativos. El objetivo a que aspira la programación de sistemas es convertir a la computadora en un compañero natural del hombre, cosa que está aún muy lejos de suceder.

## **Programación Estructurada**

Conjunto de métodos para diseñar y escribir programas empleando el método científico y no tan sólo el método de ensayo y error.

La programación estructurada dicta componentes primitivos y reglas de composición que permiten construir programas que cumplan con su objetivo a la vez de que sean legibles y fáciles de entender y modificar.

## **Pruebas**

Ejecución de una rutina de diagnóstico que verifica todas o determinadas partes de un dispositivo a fin de detectar fallas comparando los resultados con resultados correctos conocidos.

## **Registro de Datos**

Un registro de datos es una colección de valores tomados por campos de datos relacionados. Por ejemplo, en los campos de datos com "nombre del cliente", "número de identificación del cliente" y "dirección". Uno de los registros de datos es "Prof. Higgins", "123456789", "Vietnam 66". Los valores tomados por los campos de datos son para las entidad CLIENTE.

## **Regla 40-20-40**

Una buena distribución de esfuerzos, algunas veces llamada la regla 40-20-40, sugiere el porcentaje o proporcióna el esfuerzo utilizado en la planeación, el diseño y la prueba. Podemos inferir que el código (20% del esfuerzo) es desproporcionado, pero la regla 40-20-40 debe ser tomada tan solo como una guía. Las características de cada proyecto deben dictar la distribución del esfuerzo.

**Seudocódigo**

Mezcla de lenguaje de programación y español que se emplea, dentro de la programación estructurada, para realizar el diseño de un programa. La idea del pseudocódigo consiste en aprovechar la flexibilidad y poder expresivo del lenguaje natural, por un lado, y el poder estructurante de las construcciones formales (o reglas de composición), por otro lado.

**Sistema**

Red de procedimientos interrelacionados que son enfocados a realizar un objetivo específico.

**Sistemas de Información**

Complejo de elementos que interactúan entre sí para manipular, crear y consultar información proveniente de un banco de datos. Para diseñar un sistema de información se requieren conocimientos de computación y de informática, y de la participación de las áreas demandantes de la información.

**Software**

Nombre genérico que se da a los programas de una computadora.

**Subrutina**

Secuencia de instrucciones que realizan una tarea específica, generalmente utilizada más de una vez en un programa; puede estar escrita para realizar una tarea que se necesita en forma repetida para un programa específico o una tarea comúnmente necesitada para muchos programas.

**Técnica de Esfuerzo por Tarea**

La técnica de costos de esfuerzo por tarea es la técnica más común para delimitar costos en el desarrollo de algún proyecto ingenieril. Un número de persona-día, -mes, o -año es estimada para la solución de cada una de las tareas del proyecto.

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

**Diccionario de Computación**

Laura Darcy y Louis  
Boston. Argentina, 23  
de Junio de 1987. pp.  
226. Primer Edición.

**Diccionario de Informática**

Georges A. Nania.  
Madrid 1990. pp. 785.

**Introduccion a los Sistema de Bases de Datos.**

C.J. Date. México  
1986. pp. 640. Tercer  
Edición.

**Software Engineering**

Richard E. Fairley.  
Estados Unidos 1976.  
pp. 365.

**Software Engineering: A Practitioner's Approach**

Pressman S. Roger.  
Estados Unidos 1982.  
pp. 360.

Introducción a la Computación

Guillermo Levine G.  
México 1984. pp. 284  
Primer Edición.

Aplique el Dbase III Plus

Edward Jones. México  
1988. pp. 390.

ESTE LIBRO SE ACABO DE IMPRIMIR EL DIA  
12 DE DICIEMBRE DE 1991 EN LOS TALLERES DE

**TIPOGRAFICA ALEX**  
CENTENARIO 91, COL. DEL CARMEN, COYOACAN.

LA EDICION CONSTA DE 25 EJEMPLARES  
MAS SOBRANTES DE REPOSICION.