

115
Zy



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**SISTEMA PARA EL REGISTRO Y CONTROL DE LA
INFORMACION DE REAFIANZAMIENTO TOMADO
NACIONAL.**

**TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
RAMSES TERRAZAS SOLIS**



DIRECTOR DE TESIS: ING. ADOLFO MILLAN NAJERA

MEXICO D. F.

1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mi mamá:

Por tu amistad, seguridad y apoyo. Gracias por el sacrificio que has hecho por nosotros.

A mi papá:

Por tu ejemplo y sacrificio. Gracias por la formación que me has dado.

A mis hermanos:

Su ejemplo ha sido muy importante. Gracias Mabi, Vladi e Hiram.

A Adriana:

Por tu apoyo y confianza. Este logro es de los dos.

A la memoria de Ezequiel:

Este éxito es nuestro. Gracias por haber sido un hermano.

A mi familia y amigos.

A la Universidad.

Indice :

•	PREFACIO	3
	Capítulo 1 : INTRODUCCION.	6
	1.1 ¿Qué es un sistema?.	7
	1.2 Sistemas basados en computadoras.	8
	1.3 Tipos de sistemas automatizados.	9
	Capítulo 2 : INGENIERIA DE SOFTWARE.	14
	2.1 Historia del software.	15
	2.2 Paradigmas de la Ingeniería de Software.	17
	2.3 Metodologías de Análisis de Programación.	26
	Capítulo 3 : CARACTERISTICAS DE LA EMPRESA.	38
	3.1 Compañía afianzadora.	38
	3.2 Relación con otras compañías afianzadoras.	41
	3.3 Situación actual de sistemas.	44
	3.4 Planteamiento de un sistema de reafianzamiento tomado.	45
	Capítulo 4 : ANALISIS DEL SISTEMA.	47
	4.1 Objetivo del sistema.	48
	4.2 Definición de requerimientos.	48
	4.2.1 Requerimientos generales.	49
	4.3 Modelado.	52
	4.3.1 Diagramas de flujo de datos.	52
	4.3.2 Diccionario de requisitos.	53
	4.3.3 Diagramas de flujo de datos del sistema de reafianzamiento tomado y sus diccionario de requisitos.	54
	4.3.4 Diagrama de entidad-relación del sistema.	74

Capítulo 5 : DISEÑO DEL SISTEMA.	76
5.1 Diseño de datos.	77
5.1.1 Diccionario de datos.	78
5.2 Normalización de datos.	84
5.3 Diseño estructural y de interfase.	93
5.3.1 Diagramas de arquitectura.	94
Capítulo 6 : DESARROLLO.	101
6.1 Construcción de la base de datos.	101
6.2 Programación.	106
6.3 Pruebas.	124
Capítulo 7 : PUESTA EN MARCHA.	128
7.1 Implantación.	128
7.2 Capacitación.	129
• CONCLUSIONES.	132
• BIBLIOGRAFIA.	134
• APENDICES.	
I Manual de usuario.	136

Prefacio

Actualmente, el software ha superado al hardware como la clave del éxito de muchos sistemas basados en computadoras.

Lo que diferencia a una compañía con su competidora es la suficiencia y oportunidad de la información dada por el software.

Durante las primeras décadas de la informática el principal desafío era el desarrollo del hardware, de manera que se redujera el costo de procesamiento y almacenamiento. Ahora el problema es mejorar la calidad del software.

Las empresas se enfrentan a un profundo dilema. Cada vez más se están convirtiendo en organizaciones basadas en información, dependiendo de un flujo continuo de datos para cualquier aspecto de sus operaciones. Sin embargo, su habilidad para manejar esos datos se ve disminuida porque el volumen de información se expande más rápido que su capacidad de procesarla. El resultado de ello es que las empresas prácticamente se están ahogando en sus propios datos, y el problema no está en el hardware; las computadoras continúan incrementando su capacidad y potencia a una velocidad impresionante. La falla se debe al software. Desarrollar software que iguale el potencial de las computadoras resulta ser un reto mucho mayor que el construir máquinas más rápidas.

La diferencia de potencial entre el hardware y el rendimiento del software se está ampliando continuamente. Este potencial gastado afecta a cualquiera que

usa una computadora, sin embargo, lo hace con especial énfasis en las grandes organizaciones, las que dependen ampliamente de su habilidad para construir sistemas de información confiables y de gran escala. Hoy en día son contados los proyectos de integración que se terminan a tiempo y muchos menos aquellos que lo hacen dentro de lo presupuestado. Peor aún, es típico que los sistemas creados con estos esfuerzos estén plagados de defectos, y han sido tan rígidamente estructurados que es casi imposible realizar cambios significativos sin tener que rediseñarlos.

La mayoría del software empresarial corporativo por el tiempo que se utiliza para su desarrollo es obsoleto antes de liberarse y frecuentemente es incapaz de evolucionar para satisfacer necesidades futuras.

El objetivo principal de este trabajo es el desarrollo de un producto de software para el registro y control de la información de reafianzamiento tomado nacional de una compañía afianzadora y definir una metodología propia para crear sistemas.

El trabajo está organizado de manera que alguien ajeno a la computación pueda entenderlo, ya que sigue de manera genérica el ciclo de vida del proyecto, que esta compuesta de tres fases : definición, desarrollo y mantenimiento.

El primer capítulo nos muestra la teoría general de sistemas y este una buena base para el entendimiento de las necesidades y alcances de la ingeniería de software. Se explican los principales tipos de sistemas y que características se deben tomar en cuenta para automatizarlos.

El capítulo dos nos presenta una visión general de la teoría de la ingeniería de software; se explica la importancia de la ingeniería de software en la competencia empresarial actual y sus componentes más importantes. Se define al software como una integración de datos y programas. También nos explica las principales metodologías para el ciclo de vida del proyecto y las principales metodologías de análisis de programación que existen.

El capítulo tres nos introduce a lo que es una compañía afianzadora y las relaciones con otras compañías de su mismo ramo. Este tema es fundamental en el trabajo ya que muestra una visión general de las actividades y el objetivo de una compañía afianzadora, así como también la necesidad de un sistema automatizado para el control de información de reafianzamiento tomado.

En el capítulo cuatro se desarrolla el análisis del proyecto, tanto su objetivo, como los requerimientos generales del sistema hasta llegar a un modelo tangible de datos y relaciones entre ellos. Hay que tomar en cuenta que el análisis es la base de todo sistema y entre mejor se haga este paso menores recursos se

consumirán en el desarrollo del sistema. Los resultados del análisis son el diccionario de requerimientos, los diagramas de flujo de datos y los diagramas de entidad-relación.

El capítulo cinco que se ocupa del diseño del sistema, toma como base los productos del análisis para el diseño de datos, el diseño arquitectónico y el diseño procedimental. En esta etapa el usuario participa en el sistema, se toman en cuenta sus opiniones y se siente más involucrado en el proyecto. Al terminar el diseño se obtienen algunos productos como el diccionario de datos, la relación estructural de la base de datos, la definición estructural del sistema, la interfase con el usuario y el prototipo del sistema.

El capítulo seis se dedica al desarrollo del sistema que abarca tres actividades primordiales, la construcción de la base de datos, la programación y las pruebas. Estas actividades se basan en los resultados de las etapas anteriores que fueron el análisis y el diseño. Para el desarrollo de la construcción de la base de datos se analiza de manera general la teoría de base de datos. En la sección de programación analizamos la herramienta de programación (PowerBuilder) usada en este proyecto y por último se menciona la etapa de pruebas del sistema.

El capítulo siete se explican las actividades derivadas de la migración y puesta en marcha, se muestran los procesos de migración de datos, la implantación del sistema en el área usuaria y la capacitación del usuario.

Las conclusiones son una parte fundamental del trabajo, ya que en este momento se dirá si los objetivos planteados se cumplieron, y además se mencionan los logros obtenidos con el sistema.

Por último se presenta en el Anexo 1, el manual de usuario. En este manual de usuario se presenta toda la navegación del sistema, así como el uso del mismo.

1 Introducción

El presente trabajo se ocupará del desarrollo del sistema de información de reafianzamiento tomado de una compañía afianzadora, para el cual se cuenta con herramientas de desarrollo de cuarta generación y un ambiente de base de datos. Es necesario antes de pasar con cuestiones técnicas avanzadas dar una pequeña introducción a la teoría de sistemas y la ingeniería de software.

No se puede decir mucho sobre el análisis y desarrollo de sistemas sin antes tener una idea clara de lo que es un sistema. Existe una definición oficial en el diccionario pero puede parecer un tanto abstracta. Es importante conocer diferentes tipos de sistemas debido a que generalmente los sistemas forman parte de un sistema mayor, por esto para que un sistema tenga éxito, se deben entender todos los sistemas con los que interactúa.

La segunda razón importante por la cual se debe entender que es un sistema es que aunque los sistemas parezcan diferentes, resulta ser que se comportan generalmente de la misma manera: existen principios, teorías y filosofías comunes que se aplican, prácticamente a todos los tipos de sistemas. Precisamente por esto, se puede aplicar lo aprendido en otros sistemas, a los sistemas basados en computadoras.

De aquí que si entendemos la teoría general de sistemas nos ayudará a entender mejor los sistemas de información automatizados.

1.1 ¿ Qué es un sistema ?

Un sistema es un grupo de elementos interdependientes o que interactúan regularmente formando un todo.

Algunos ejemplos de sistemas son los siguientes:

- Un sistema termodinámico
- Un sistema taxonómico
- El sistema numérico
- El sistema digestivo
- Un sistema de proceso de información

Existen principalmente dos tipos de sistemas: los sistemas naturales y los sistemas hechos por el hombre.

Entre los sistemas naturales están los sistemas físicos y los sistemas vivientes. Entre los sistemas físicos se encuentran, los sistemas geológicos, los sistemas estelares, etc.; y entre los sistemas vivientes se encuentran todos aquellos que comprenden a los animales y las plantas.

Es importante el entendimiento de los sistemas naturales ya que se desarrollan gran variedad de sistemas hechos por el hombre que se relacionan entre si, incluyendo sistemas basados en computadoras.

Los sistemas hechos por el hombre son aquellos que fueron contruidos, organizados y mantenidos por humanos y pueden ser:

- **Sistemas sociales.**- organización de partidos políticos, organización familiar, leyes, costumbres, etc.
- **Sistemas de comunicación.**- señales de humo, mímica, señales de radio, telex, telégrafo, correo, televisión, etc.
- **Sistemas financieros.**- contabilidad, inventarios, libro mayor, etc.

1.2 Sistemas basados en computadoras

La mayoría de los sistemas hechos por el hombre existían antes que las computadoras, es por esto que no todos los sistemas deben automatizarse y para esto debemos tener en cuenta muchas variables antes de intentarlo, algunas de las más importantes son las siguientes:

- **Costo.**- Es probable que sea menos costoso llevar a cabo el trabajo en forma manual.
- **Conveniencia.**- Un sistema automatizado puede ocupar mucho espacio, gastar mucha energía eléctrica, generar mucho calor etc.
- **Seguridad.**- El sistema debe ser lo suficientemente seguro como para guardar datos confidenciales.
- **Facilidad de mantenimiento.**- Es necesario utilizar tecnología de fácil mantenimiento, ya que en caso de una avería nadie podría arreglarla.

Los sistemas automatizados de control de información tienen algunos componentes en común que son:

El hardware.- La parte física de la computadora, como procesadores, discos, impresoras, monitores, etc.

El software.- La parte lógica de la computadora, como sistemas operativos, bases de datos, programas de aplicación, etc.

Las personas.- Los que operan el sistema, son aquellos que alimentan a la computadora y utilizan o consumen las salidas de la misma.

Los datos.- La información que el sistema necesita para que este sea útil.

Los procedimientos.- Las políticas formales e instrucciones de operación.

La documentación.- Los manuales, los impresos y otra información que describe el uso y operación del sistema.

1.3 Tipos de sistemas automatizados

Una división general de los sistemas automatizados es la siguiente

- Sistemas en línea.
- Sistemas de tiempo real.
- Sistemas de apoyo a decisiones.
- Sistemas basados en el conocimiento.

1.3.1 Sistemas en línea.

Los sistemas en línea son todos aquellos que aceptan entradas directamente del lugar en donde se creó y regresa información directamente a quien lo requiera, véase la figura 1.1 .

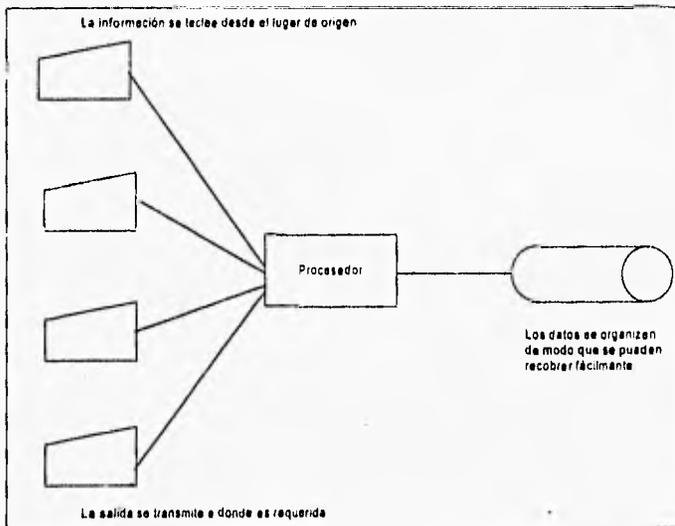


Figura 1.1 Sistemas en línea

La principal característica de un sistema en línea es que se puede acceder a los datos almacenados rápidamente y sin necesidad de recurrir a otros componentes de información del sistema. Esto contrasta con los sistemas batch o en lotes ya que la información suele recuperarse de una manera secuencial.

Dado que un sistema en línea interactúa directamente con personas es necesario prestar especial atención a la interfaz humano-computadora. Esto usualmente se logra identificando todos los estados en los que la computadora y el usuario pudieran encontrarse, e identificando todos los cambios de estado.

1.3.2 Sistemas en tiempo real

Los sistemas en tiempo real son considerados como una variación de los sistemas en línea, pero se definen de la siguiente manera:

Son los sistemas computacionales que controlan un ambiente recibiendo datos, procesándolos y devolviéndolos con la suficiente rapidez como para influir en dicho ambiente en ese preciso momento, véase figura 1.2 .

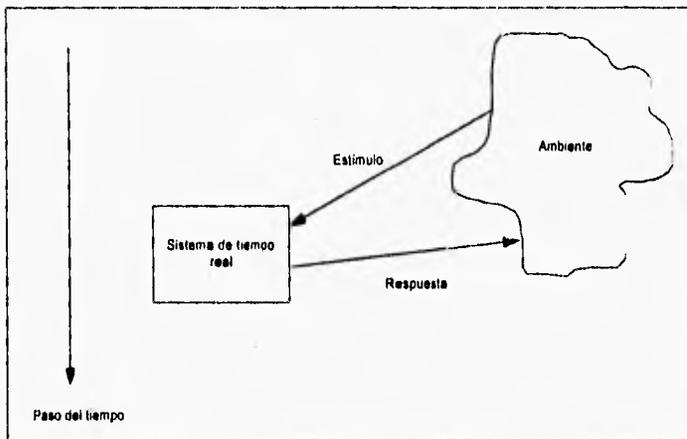


Figura 1.2 Sistemas de tiempo real.

Algunos sistemas que son de tiempo real son los siguientes:

- Sistemas de control de procesos.
- Sistemas de cajeros automáticos.
- Sistemas de conmutación telefónica.
- Sistemas de vigilancia de pacientes en hospitales.

Los sistemas de tiempo real se caracterizan por lo siguiente:

- ◆ Simultáneamente se lleva a cabo el proceso de muchas actividades.
- ◆ Se asignan prioridades diferentes a los diferentes procesos.
- ◆ Se interrumpe alguna tarea antes de concluirla, para procesar otra de mayor prioridad.
- ◆ Existe acceso simultáneo a datos comunes, por eso es necesario un elaborado proceso de sincronización.
- ◆ Existe un uso y asignación dinámico de memoria RAM (Memoria de Acceso Aleatorio).

1.3.3 Sistemas de apoyo a decisiones

Este tipo de sistemas no toman decisiones por si mismo, sino ayudan a los administradores y a otros profesionistas a tomar decisiones inteligentes y documentadas acerca de la operación. Una característica común entre este tipo de sistemas es que no sólo despliegan los datos sino que además realizan varios tipos de análisis matemáticos y estadísticos de los mismos. La mayoría de estos sistemas presentan la información en diferentes formas como gráficas o reportes convencionales, véanse las figuras 1.3 y 1.4 .

Presupuesto de caja o flujo de efectivo

Concepto	Enero	Febrero	Marzo	Total
Saldo Inicial	80,000	261,500	80,000	
a) Entradas de efectivo				
Cobro de ventas	475,000	556,000	655,000	1,686,000
Otros ingresos			350,000	
Total disponible	555,000	817,500	1,085,000	2,457,500
b) Salida efectivo				
Pago materia prima	60,000	626,592	378,624	1,065,216
Pago gastos	103,500	115,500	124,000	343,000
Compra Activo fijo		50,000		50,000
Otras erogaciones				
Acreedor Hipotecario	50,000	50,000	50,000	150,000
c) Total salidas	213,500	842,092	552,624	1,608,216
Saldo minimo deseado	80,000	80,000	80,000	240,000
Total necesario	293,500	922,092	632,624	1,848,216
d) Sobrante (faltante)	261,500	(104,592)	452,376	609,284
e) financiamiento				
Monto prestamo		116,213		116,213
Intereses 10%		11,621	11,621	23,243
Pago c/prestamo			116,213	116,213
Saldo neto final		104,592	127,834	232,426
	261,500	80,000	404,542	746,042

Figura 1.3 Reporte tabulado de una hoja de cálculo

Algunos ejemplos de este tipo de sistemas son (Lotus 1,2,3 , Excel de Microsoft, Quatro Pro de Borland, etc.)

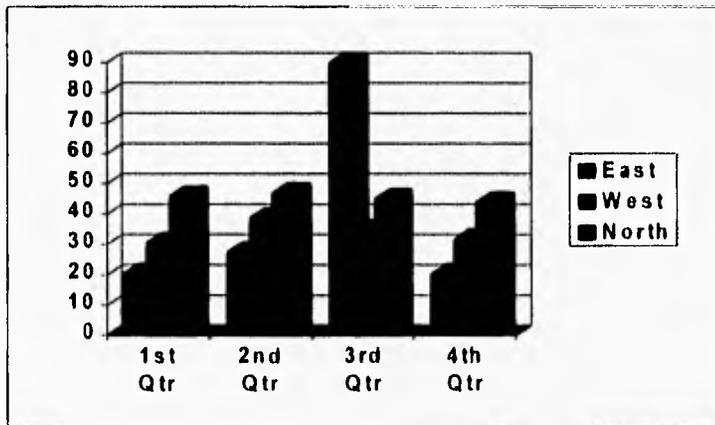


Figura 1.4 Gráfica de un sistema de apoyo a la toma de decisiones

1.3.4 Sistemas basados en el conocimiento

Los sistemas basados en el conocimiento contienen grandes cantidades de conocimiento que emplean en el desempeño de una tarea dada. Este tipo de sistemas es un apoyo de alto nivel intelectual para el experto humano, por lo cual se le denomina asistente inteligente.

Los sistemas expertos o sistemas basados en el conocimiento por lo general se construyen de tal manera que sean capaces de explicar las líneas de razonamiento que llevaron a las decisiones que tomaron, en contra posición con los sistemas de apoyo a decisiones este tipo de sistemas si toma sus propias decisiones.

Después de haber entendido lo que es un sistema y los tipos de sistemas automatizados estamos preparados para el siguiente capítulo que desarrollará la ingeniería de software, que es el nombre que recibe el conjunto de metodologías que se usan para la producción de sistemas como un producto de software.

2 Ingeniería de software

La ingeniería de software es el establecimiento y uso de principios de ingeniería, orientados a obtener un producto que sea confiable, económico y funcione de manera eficiente.

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. En las pasadas cuatro décadas, el software ha pasado de ser una resolución de problemas y una herramienta de análisis de información a ser una industria por sí misma.

La ingeniería de software es una disciplina que integra métodos, herramientas y procedimientos para el desarrollo del software de computadora.

Los métodos de la ingeniería de software introducen una notación generalmente gráfica y un conjunto de criterios para la calidad del software, y sus principales tareas son: planificación del proyecto, análisis de requerimientos, diseño de estructuras de datos, arquitectura de programas, codificación, pruebas y mantenimiento.

Las herramientas de la ingeniería de software suministran un soporte automático para el desarrollo de los métodos. Existen herramientas para el desarrollo de cada uno de los métodos antes mencionados, y también existen productos integrados de modo que la información creada por una herramienta

puede ser utilizada por otra, entonces se establece un sistema para el soporte del desarrollo del software llamado CASE (Computer Aided Software Engineering).

Los procedimientos de la ingeniería de software definen la secuencia en que se aplican los métodos, las entregas y las directrices que ayudan a evaluar el progreso, en otras palabras los procedimientos sirven para la conjunción de métodos y herramientas y facilita un desarrollo racional y oportuno del software de computadora.

La ingeniería de software está compuesta por una serie de pasos que abarcan los métodos, las herramientas y los procedimientos. Estos pasos se denominan paradigmas de la ingeniería de software.

La elección de un paradigma se lleva a cabo de acuerdo con el tipo de proyecto, los métodos, las herramientas, los controles y las entregas requeridas.

2.1 Historia del software.

El desarrollo del software se puede decir que esta fuertemente ligado con las últimas cinco décadas, la figura 2.1 nos muestra la evolución del software dentro de las áreas de aplicación de los sistemas automatizados. Los primeros años de desarrollo se le dio mucha importancia al hardware mientras que al software se le veía como algo secundario, el desarrollo del software se hizo sin ninguna planificación, la mayoría de los sistemas tenían una orientación por lotes. En general el hardware era de propósito general mientras que el software se diseñaba a medida para cada aplicación y no tenía una distribución adecuada, No existía el software como producto.

Debido a este entorno personalizado del software, el diseño de programas era totalmente implícito y la documentación no existía.

La segunda era abarca desde mediados de la década de los sesentas a finales de los setentas. Los sistemas multiusuario y la multiprogramación modificaron los conceptos de la ingeniería de software, los sistemas de tiempo real obtenían, analizaban y transformaban datos de manera, que se controlaban los procesos y se producían salidas en milisegundos en lugar de en minutos como en la primera era.

Esta era también se caracterizó por establecer el software como un producto y aparecieron las primeras casas de software. Los programas se distribuían a cientos e incluso a miles de usuarios.

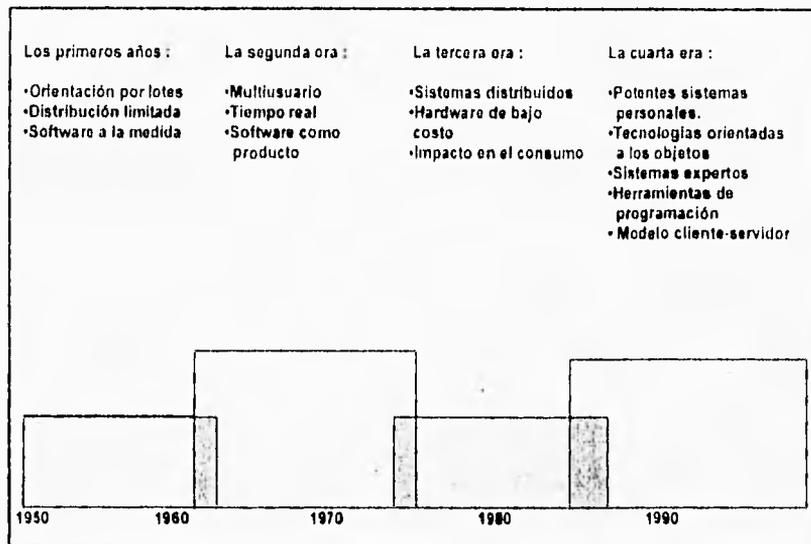


Figura 2.1 Evolución del software

Conforme crecía el número de usuarios y de productos de software aumentaban cada vez más las bibliotecas de software. Las casas de software incorporaban cientos de miles de nuevas sentencias y éstas se distribuían por todo el mundo usuario. Apareció un gran problema, todos esos programas ya distribuidos empezaron a tener problemas y tenían que ser corregidos. Estas actividades se llamaron colectivamente mantenimiento del software. El esfuerzo gastado comenzó a gastar recursos de una manera alarmante, con este suceso se dice que empezó la crisis del software.

La tercera era va de mediados de los setentas y llega hasta el momento actual, -- aunque en algunas compañías ya se haya superado esta etapa -- nacieron las redes de área local y de área extendida, el procesamiento distribuido de computadoras, las comunicaciones digitales y la creciente demanda de acceso instantáneo de información, crearon una gran presión sobre los desarrolladores, "había que crear todo de nuevo".

Otra característica de esta era fue la aparición de las computadoras personales y el amplio uso de los microprocesadores, gracias a la difusión de las computadoras personales algunas compañías de software lograron crecer de una manera impresionante logrando vender millones de copias de sus productos.

El hardware en las computadoras se ha vuelto de cierta manera en un producto estándar y el software es el que marca las diferencias entre la eficiencia de los sistemas.

La cuarta era empezó a partir de principios de los noventas, las tecnologías orientadas a objetos están desplazando los enfoques convencionales de desarrollo de software. Las técnicas de cuarta generación están cambiando el enfoque de los desarrolladores de sistemas. Los sistemas expertos y la inteligencia artificial se han trasladado de los laboratorios a las áreas de aplicación.

Conforme nos movemos en la cuarta era, los problemas de la crisis del software siguen intensificándose y muchas industrias están adoptando prácticas de ingeniería de software.

2.2 Paradigmas de la ingeniería del software.

Para poder ser un buen analista de sistemas es necesario tener métodos que seguir. En las pequeñas empresas o sociedades los proyectos nacen de manera informal, solamente de conversaciones entre el usuario y el analista; en cambio en las corporaciones grandes, las cosas se llevan a cabo de manera más formal, la comunicación entre el usuario y el analista se hace normalmente por escrito, aún así no se logra tener una metodología bien definida ya que se deja a discreción del líder del proyecto las fases y actividades del mismo.

Por lo anterior es importante tener bien definido el ciclo de vida del proyecto y sus objetivos principales son:

- Definir las actividades a llevarse a cabo durante el desarrollo del proyecto.
- Lograr una congruencia entre el proyecto y los sistemas en uso de la organización.
- Dar tiempos de control y revisión administrativos.

Debe de quedar muy claro que el ciclo de vida de un proyecto no evitará que el líder del proyecto realice sus actividades habituales. En lo que ayudará el ciclo de vida será en que las tareas del líder estarán bien organizadas de manera que podrá tomar las decisiones correctas en el momento adecuado.

2.2.1 Ciclo de vida clásico

La figura 2.2 muestra el ciclo de vida del proyecto clásico y se compone de las siguientes actividades :

Requerimientos del sistema.- Este planteamiento es esencial cuando el software debe relacionarse con otros elementos, tales como el hardware, otros sistemas, personas y bases de datos.

Requerimientos del software.- Debido a que el software es siempre parte de un sistema mayor, hay que establecer los requisitos, así como la función, el rendimiento y las interfaces requeridas.

Análisis.- Este proceso se basa en los requerimientos del sistema y del software, ya que para comprender la naturaleza de los sistemas hay que construirlos y analizarlos.

Diseño del programa.- El diseño del software se divide en cuatro etapas, la estructura de los datos, la arquitectura del software, el detalle procedimental y la interfaz. Este proceso traduce los requerimientos en una representación del software que obtenga la calidad requerida antes de la codificación.

Codificación.- Este paso se realiza cuando el diseño se traduce en una forma legible o entendible para la máquina.

Pruebas.- Las pruebas se centran en la lógica interna de los programas, asegurando que las entradas definidas produzcan los resultados esperados.

Operaciones y mantenimiento.- El software sufrirá cambios una vez que se entregue al cliente. Esto puede deberse a cambios del entorno del sistema o que el cliente requiera ampliaciones funcionales del rendimiento del software. A los requerimientos de cambios se les conoce como mantenimiento del sistema y este mantenimiento aplica cada uno de los pasos del ciclo de vida.

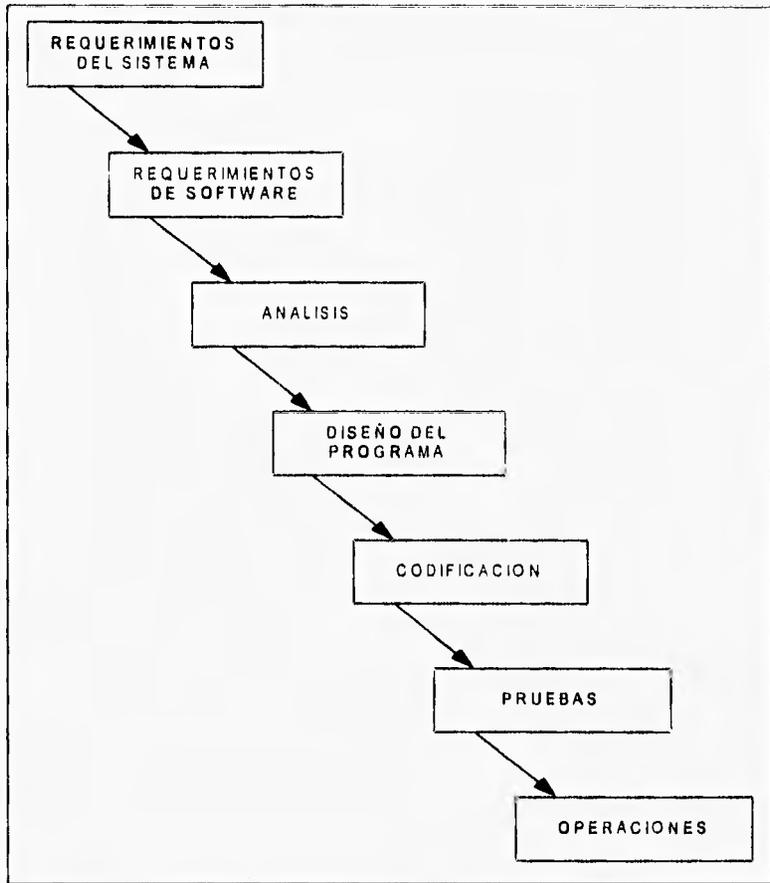


Figura 2.2 Ciclo de vida clásico

El ciclo de vida clásico se caracteriza por tener una implantación ascendente y una progresión secuencial.

El enfoque ascendente presenta un gran número de dificultades

- Nada está hecho hasta que todo esté terminado.

- Las fallas más triviales se encuentran al comienzo del período de prueba y las más graves al final.
- La eliminación de fallas suele ser extremadamente difícil durante las últimas etapas de prueba.
- La necesidad de prueba con la computadora es más necesaria en las últimas etapas.

La segunda debilidad más importante del ciclo de vida de un proyecto clásico es su insistencia en que las fases se sucedan secuencialmente. Este concepto de progreso ordenado es ideal, por que está muy alejado de la realidad; el enfoque secuencial no permite el tratamiento de fenómenos reales como las políticas de la compañía, los problemas económicos y personales.

2.2.2 Ciclo de vida estructurado

El ciclo de vida estructurado se muestra en la figura 2.3, explicaré brevemente cada uno de los pasos a seguir.

La encuesta .- Esta etapa comienza cuando el usuario solicita que una o más partes de su sistema se automaticen. Los principales objetivos de la encuesta son los siguientes:

- Identificar a los usuarios responsables y crear un grupo de trabajo inicial del sistema.
- Identificar las deficiencias actuales en el ambiente usuario, por ejemplo: el hardware del sistema actual no es confiable, el software actual no se puede mantener, el tiempo de respuesta es malo, el sistema actual no proporciona la información requerida.
- Establecer metas y objetivos para un sistema nuevo.
- Determinar si es factible automatizar el sistema. Hacer estimaciones del costo y tiempo necesarios.

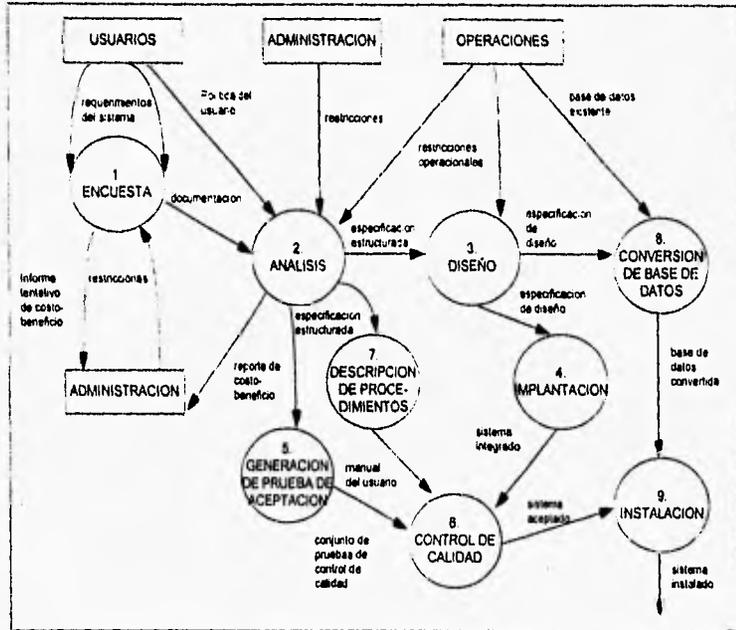


Figura 2.3 Ciclo de vida estructurado

- Preparar un documento guía del proyecto, con los objetivos y alcances del mismo.

Análisis del sistema.- El propósito principal es transformar las dos entradas principales, que son las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente usuario con diagramas de flujo de datos, diagramas de entidad- relación, diagramas de transición de estado, etc.

Como producto final se tendrá una descripción formal de lo que el nuevo sistema debe hacer independientemente de la naturaleza de la tecnología que se utilice para su realización.

Diseño.- Esta etapa se dedica a asignar porciones de la especificación a procesadores adecuados y a labores apropiadas. Además se ocupa de la transformación de modelos de datos de entidad-relación en un diseño de base de datos.

Implementación.- Esta actividad incluye la codificación y la integración de módulos en un esquema progresivamente más completo del sistema final.

Generación de pruebas de aceptación.- La especificación debe contener toda la información necesaria para definir un sistema que sea aceptable.

Las pruebas de aceptación pueden suceder al mismo tiempo que el diseño e implementación.

Garantía de calidad.- Esta actividad la lleva a cabo la organización usuaria, y es simplemente la prueba final de la calidad del sistema.

Descripción del procedimiento.- Una de las actividades más importantes es la generación de una descripción formal de las partes del sistema. El resultado es un manual para el usuario.

Conversión de bases de datos.- Esta actividad requiere como entrada la base de datos actual del usuario, al igual que la especificación del diseño. Hay casos en que no existe una base de datos que convertir.

Instalación.- La actividad final es la instalación; sus entradas son el manual de usuario, la base de datos convertida y el sistema aceptado.

2.2.3 Construcción de prototipos

Una alternativa de enfoque para la definición de los requerimientos consiste en capturar un conjunto inicial de necesidades e implantarlas rápidamente con la intención declarada de expandirlas y refinarlas iterativamente al ir aumentando la comprensión que del sistema tiene el usuario y quien lo desarrolla. La definición del sistema se realiza mediante el descubrimiento evolutivo y gradual. Este tipo de enfoque se llama de prototipos.

Esta construcción de prototipos se apoya en los siguientes tipos de herramientas de software

- Diccionario de datos integrado.
- Generador de pantallas.
- Generador de reportes no procedimental.
- Lenguaje de programación de cuarta generación.
- Lenguaje de consultas y búsquedas no procedimental.
- Poderosa administración de base de datos.

El ciclo de vida de prototipos se muestra en la figura 2.4.

Es necesario evaluar si el proyecto es un buen candidato para utilizar este enfoque de prototipos, para ello es necesario que tengan las siguientes características:

- El usuario no puede o no esta dispuesto a examinar modelos abstractos.
- El usuario no puede o no esta dispuesto para especificar sus requerimientos y solo se pueden determinar por medio de un proceso de tanteo o prueba y error.
- El sistema no requiere la especificación de grandes cantidades de detalles algorítmicos, ni de muchas especificaciones de proceso.

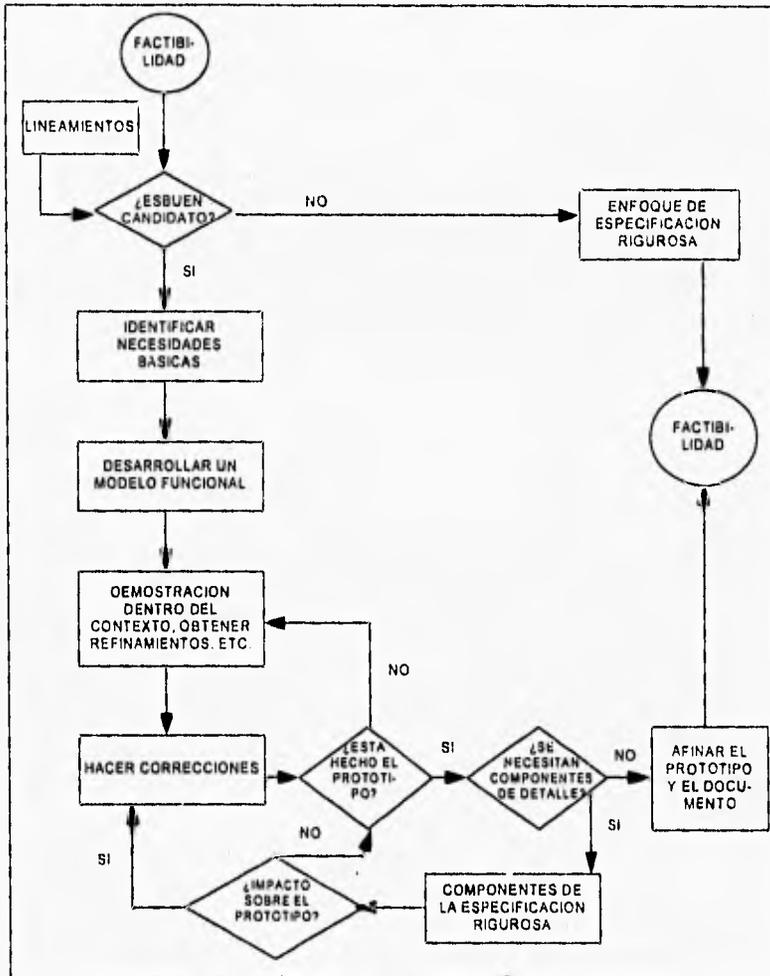


Figura 2.4 Ciclo de vida de prototipos

2.2.4 Técnicas de cuarta generación

Este tipo de técnica abarca un amplio espectro de herramientas de software que tienen algo en común, todas facilitan el trabajo al que desarrolla el software. se especifican de algunas características del software a un alto nivel, y después la herramienta genera automáticamente el código fuente y la estructura de la base de datos a partir de las especificaciones.

Actualmente, un entorno para el desarrollo de software que soporte el modelo de técnicas de cuarta generación (T4G) puede incluir todas o algunas de las siguientes herramientas:

- Lenguajes no procedurales para creación y consultas de bases de datos.
- Interacción y definición de pantallas.
- Generación de formas o pantallas.
- Manipulación de datos.
- Generación de código.
- Gráficas de alto nivel y facilidades de hoja de cálculo.

En la figura 2.5 se describe el modelo de técnicas de cuarta generación para la ingeniería de software. Al igual que los otros modelos, se comienza con la recolección de requisitos. La construcción se apoya principalmente en lenguajes de cuarta generación (L4G) o generadores de código, para la codificación de los programas y en el sistema administrador de la base de datos (DBMS) para la creación de la base de datos.

La gran flexibilidad y rapidez de programación otorgada con el uso de esta técnica permite la posibilidad de utilizar un esquema de manejo de prototipos, iterando las etapas del modelo tantas veces como sea necesario.

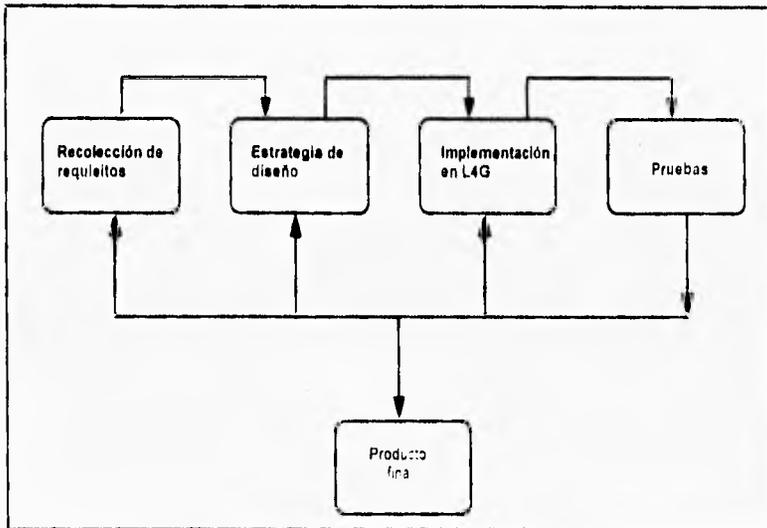


Figura 2.5 Ciclo de vida con técnicas de cuarta generación

2.3 Metodologías de análisis de sistemas

Existen diferentes tecnologías para el análisis de sistemas, como en este trabajo uno de los objetivos es el de proponer una metodología para el desarrollo de sistemas, veremos a continuación las técnicas más usadas actualmente para el análisis de sistemas, estas metodologías son las siguientes:

- Análisis estructurado.
- Ingeniería de software asistida por computadora.
- Análisis orientado a objetos.

2.3.1 Análisis Estructurado.

Hasta fines de los años setentas, los proyectos de desarrollo de sistemas empezaban con la creación de una "novela" de requerimientos del usuario. Esta pesada documentación se veía afectada por diversos problemas

- Eran monolíticos.- había que leer completamente la especificación de principio a fin, para poder entenderla.
- Eran redundantes.- a menudo se repetía la misma información en diversas partes del documento.
- Eran ambiguas.- el reporte detallado de los requerimientos podía ser interpretado de diferente manera por el usuario, el analista, el diseñador y el programador.
- Eran imposibles de mantener.- la especificación funcional era casi obsoleta para cuando llegaba el final del proceso de desarrollo del sistema.

Mientras se debatían todos estos problemas ya se estaba adoptando un conjunto de ideas conocidas como diseño y programación estructurados. Estas ideas prometían grandes mejoras en la organización, codificación, prueba y mantenimiento; pero había un problema, no tenía caso escribir programas brillantes y diseñar sistemas altamente modulares si nadie sabía realmente que era lo que se suponía que el sistema debería hacer.

Como resultado se ha procurado hacer especificaciones funcionales que reúnan los siguientes requisitos

- Gráficas.- compuesto por diagramas y apoyadas con material textual detallado.
- Particionadas.- que se puedan leer independientemente porciones individuales.
- Mínimamente redundantes.- de manera que los cambios necesarios se hagan en solo una parte de la especificación.

Algunos años de experiencia mostraron que el análisis estructurado clásico necesitaba algunas mejoras

- La construcción de modelos de estado físico y lógico actuales podrían ser dentro de la empresa políticamente peligrosos.
- El análisis clásico hacía una distinción difusa y poco definida entre los modelos físicos y los modelos lógicos. Ahora nos referimos a modelos esenciales en lugar de modelos lógicos y modelos de implantación en lugar de modelos físicos.
- El análisis estructurado clásico se concentraba casi totalmente en modelar las funciones que deberían llevarse a cabo en un sistema; el modelado de datos se hacía de manera primitiva.
- El análisis empieza con un diagrama de flujo de datos con una sola burbuja que representa a todo el sistema, y luego se dividirá en varias funciones y almacenes de datos en una forma estrictamente descendente. En consecuencia, se ha añadido un nuevo enfoque, conocido como división de eventos.

El análisis estructurado se compone de los siguientes elementos

Diagrama de flujo de datos.- El diagrama de flujo de datos es una herramienta de modelado que únicamente proporciona un punto de vista de un sistema, el orientado a las funciones. Un diagrama típico de flujo de datos esta compuesto por el proceso, el flujo, el almacén y el terminador, véase la figura 2.6.

Diccionario de datos.- El diccionario de datos es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas y salidas. El diccionario describe el significado de los flujos de los diagramas de datos.

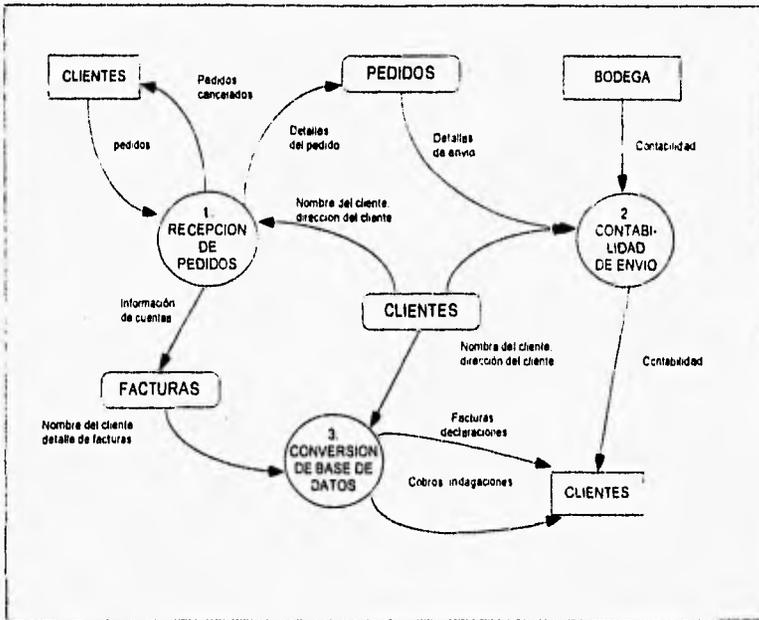


Figura 2.6 Diagrama de flujo de datos típico

Diagrama de entidad-relación.- El diagrama de entidad-relación es un modelo de red que describe con un alto nivel de abstracción, la distribución de datos almacenados en un sistema. El diagrama de entidad relación esta compuesto por tipos de objetos, relaciones, indicadores asociativos de tipo de objetos, indicadores de supertipo/subtipo, véase figura 2.7.

Diagrama de transición de estados.- El diagrama de transición de estados, enfatiza el comportamiento dependiente del tiempo del sistema. Los principales componentes del diagrama son estados y flechas que representan los cambios de estado, véase figura 2.8.

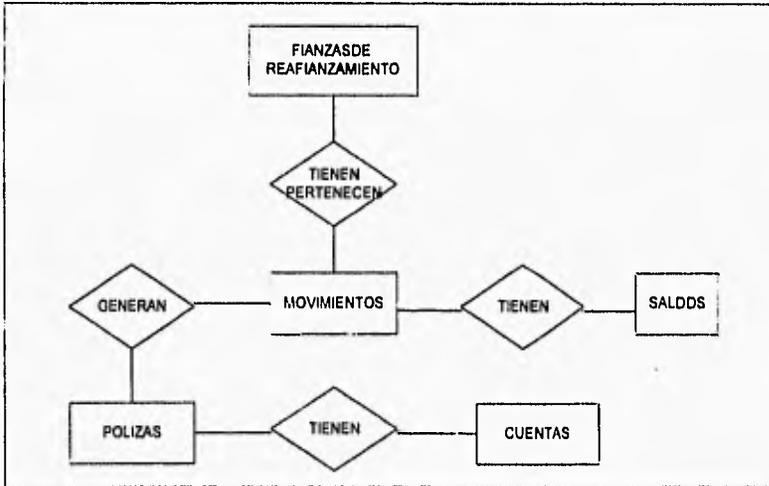


Figura 2.7 Diagrama de entidad-relación típico.

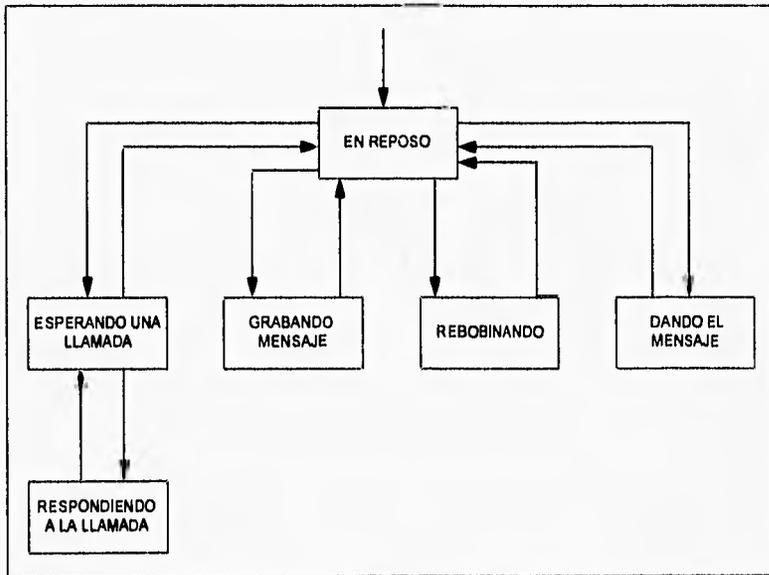


Figura 2.8 Diagrama de transición de estados típico.

2.3.2 Ingeniería de software asistida por computadora (CASE)

La ingeniería de software asistida por computadora puede ser tan simple como una única herramienta que permita desarrollar una actividad específica, o tan compleja como un entorno que integre distintas herramientas, una base de datos, gente, hardware, una red, sistemas operativos, estándares, etcétera.

La figura 2.9 representa los bloques que componen el CASE. Es interesante ver que el fundamento para un CASE efectivo tiene poco que ver con las herramientas de ingeniería de software en si mismas. Los buenos entornos de ingeniería de software se construyen sobre una arquitectura de entorno que engloba los correspondientes sistemas de software y hardware.



Figura 2.8 Bloques del CASE

La arquitectura de entorno, compuesta por la plataforma de hardware y el soporte del sistema operativo constituye la base del CASE. Pero el entorno CASE necesita de servicios de portabilidad que constituyen un puente entre las herramientas CASE y su marco de integración y la arquitectura de entorno. El marco de integración es un conjunto de programas que permiten a una herramienta la comunicación con otras, para crear una base de datos y mostrar una apariencia homogénea al usuario final. Los servicios de portabilidad permiten la migración de las herramientas CASE a diferentes plataformas de hardware y sistemas operativos.

Las herramientas CASE se pueden clasificar de la siguiente manera:

- Planificación de sistemas de gestión.
- Gestión de proyectos.
- Soporte.
- Análisis y diseño.
- Programación.
- Integración y prueba.
- Prototipos.
- Mantenimiento.
- Estructura.

2.3.3 Análisis orientado a objetos.

A pesar de que la tecnología orientada a objetos ha recibido especial atención tiene más de veinte años de edad.

En ese tiempo toda la programación era modular pero había quienes definían sus módulos basados en los objetos que querían simular. Los objetos en una simulación ofrecen una forma natural de descomponer el problema a resolver. Cada objeto tiene un cierto comportamiento a ser modelado, y cada uno tiene que mantener información sobre su propio status.

Un objeto se define entonces como una unidad de software que contiene una colección de datos y procedimientos interrelacionados. Los datos se definen como variables porque definen el estado del objeto en cualquier momento. Los procedimientos reciben el nombre de métodos y ellos definen todo el comportamiento de un objeto.

El concepto de un objeto es simple y a la vez poderoso. Los objetos forman módulos de software ideales debido a que pueden definirse y mantenerse independientemente. Todo lo que un objeto conoce está expresado en sus variables y todo lo que puede hacer está expresado en sus métodos. Véase figura 2.9.

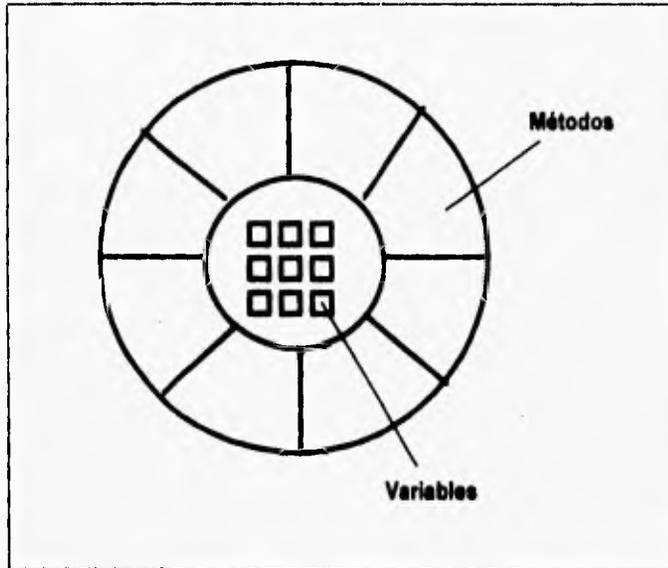


Figura 2.9 Diagrama general de un objeto.

La forma en que los objetos interactúan unos con otros es enviándose mensajes pidiendo que se ejecute un método específico. Un mensaje consiste simplemente del nombre del objeto a quien va dirigido seguido del nombre del método que el objeto receptor sabe como ejecutar. Un sistema orientado a objetos consiste en varios objetos interactuando unos con otros enviándose mensajes

entre sí. Debido a que todo lo que un objeto puede hacer está expresado en sus métodos, este simple mecanismo soporta todas las posibles interacciones entre objetos. Véase figura 2.10

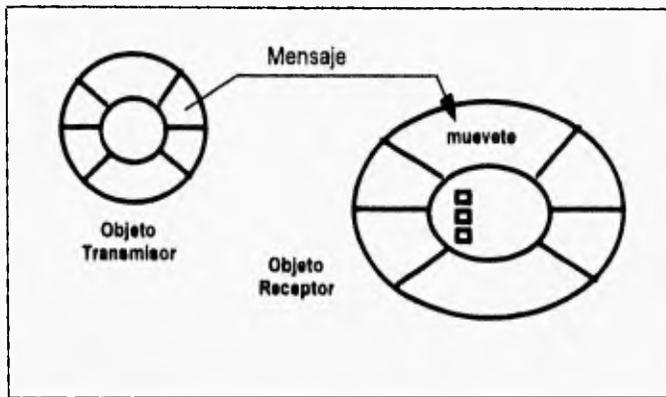


Figura 2.10 Diagrama de comunicación entre objetos.

Se llegó a tener alguna dificultad por que es raro que en un sistema se involucre a un solo objeto de cada tipo, para esto se llegó a la solución de crear clases de objetos. Una clase es un prototipo que define los métodos y variables que serán incluidas en un tipo de objeto particular. Las descripciones de los métodos y variables que los soportan se describen solo una vez, en la definición de la clase.

Los objetos que pertenecen a una clase se denominan instancias de la clase, y contienen tan solo los valores particulares para las variables, compartiendo el código de los métodos. Véase figura 2.11.

El concepto de definir clases a partir de otras se llama herencia y es un elemento central de la tecnología de la programación orientada a objetos.

La herencia es un mecanismo por el cual una clase de objetos puede definirse como un caso especial de una clase más general, con lo cual automáticamente se incluye la definición de métodos y variables de la clase general. Casos especiales de una clase se conocen como subclase de esa clase y la clase más general de las subclases se conoce como superclase.

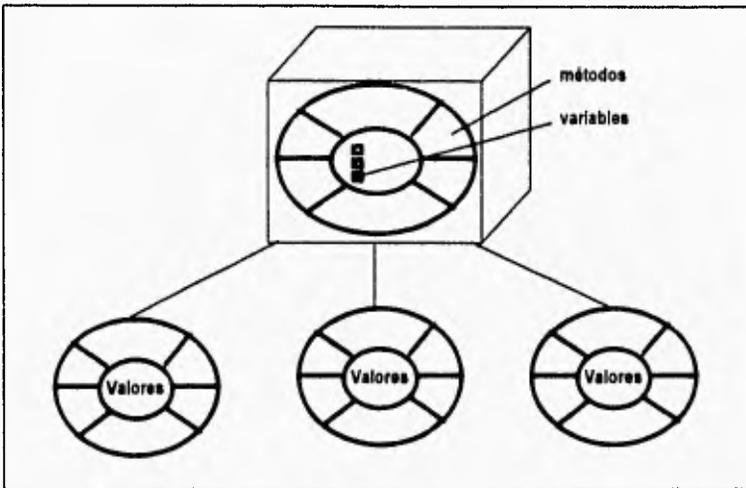


Figura 2.12 Definición de clase

La herencia entre clases puede extenderse a cualquier grado. El resultado es una estructura arborescente conocida como jerarquía de clases. Véase figura 2.12.

La programación orientada a objetos frecuentemente se considera más natural que la programación tradicional, y esto es cierto en varios niveles. En un primer nivel es más natural porque permite organizar la información en formas que nos resultan más familiares. En un nivel más profundo, es más natural porque refleja las mismas técnicas que la naturaleza usa para manejar la complejidad.

Como ejemplo podemos tener una célula, una célula es la unidad de vida y esta formada por información y comportamientos. Las células tienen una membrana que permite la comunicación con ciertos tipos de otras células. Esta membrana oculta la complejidad de la célula y ofrece una interfaz amigable para su comunicación. Todas las interacciones entre células toman lugar por medio de mensajes químicos reconocidos por la membrana de la célula y pasados a su interior.

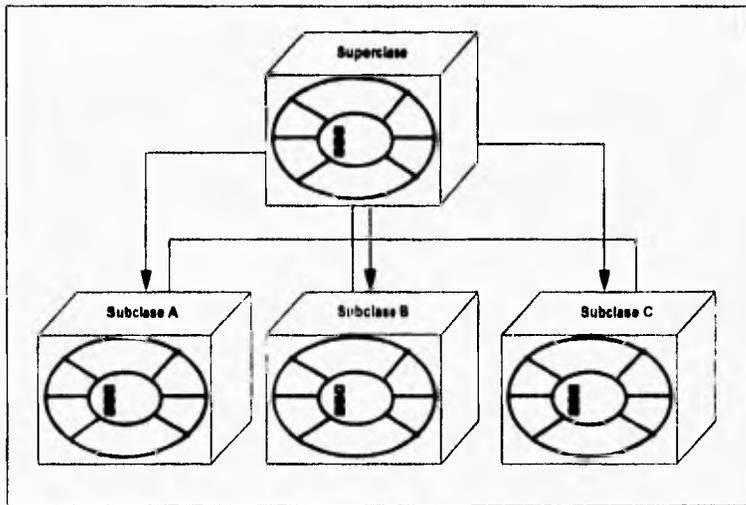


Figura 2.12 Herencia.

A partir de esta estructura básica hay una infinita variedad la cual no es caótica sino que está perfectamente organizada, o clasificada, en un jerarquía especializada de clases y subclases. Véase figura 2.13.

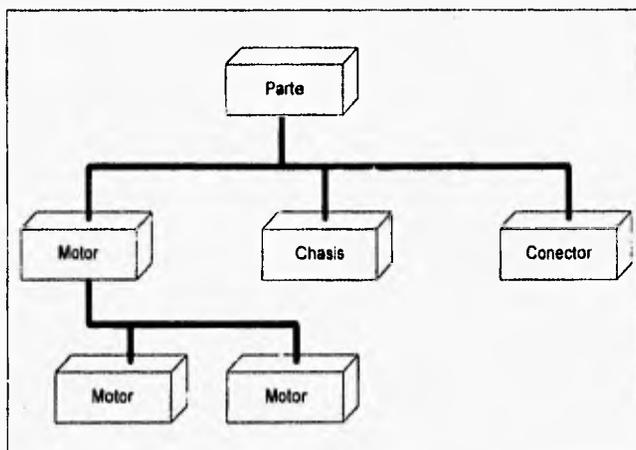


Figura 2.13 Jerarquía de clases.

El proceso por el cual el software se construye es muy diferente en el enfoque orientado a objetos. Los objetos, al ser bloques generales que modelan entidades reales, más que realizar tareas específicas permiten su reutilización en proyectos subsecuentes aún si los objetivos del nuevo proyecto son totalmente diferentes.

Ahora estamos preparados para entrar de lleno al problema, que es la creación del sistema de reafianzamiento tomado. tenemos las bases teóricas mínimas para el análisis y desarrollo de sistemas. es por esto que el siguiente capítulo se dedica a revisar las características de una compañía afianzadora.

3 Características de la Empresa

El participante más importante en el juego de los sistemas es alguien que se conoce como usuario; de manera característica el usuario es aquel que formalmente solicita un sistema. El usuario puede ser una persona o una empresa en general, es por esto que el inicio de todo desarrollo de software debe estar respaldado por la información del tipo de negocio, sus objetivos, su historia, su operación y su gente.

Las características antes mencionadas forman el ambiente al cual ingresará nuestro producto de software y por esto que es necesario conocerlo y entenderlo.

3.1 Compañía afianzadora.

Una compañía afianzadora es una compañía autorizada por el gobierno para expedir fianzas a título oneroso, es decir, cobrando una prima por su emisión. El producto que la afianzadora ofrece es la fianza de empresa, la cual es un contrato accesorio en el que la afianzadora garantiza el cumplimiento de una obligación, estipulada en un contrato principal, por parte de un fiado a favor de un beneficiario. Véase cuadro 3.1

Tipo de Fianza	Definición
<ul style="list-style-type: none"> • Fianza Civil 	<ul style="list-style-type: none"> • Es la garantía otorgada por una persona física que tiene capacidad de uso y goce sobre su patrimonio. Esta garantía se da en forma excepcional; esto es, que no puede darse en forma reiterativa, y siempre es gratuita ya que se da en razón de la persona por la que se fía, por lo que no es susceptible de comercialización.
<ul style="list-style-type: none"> • Fianza Mercantil 	<ul style="list-style-type: none"> • Es la garantía otorgada por un comerciante ante otro, derivadas de operaciones mercantiles. Esta garantía es igualmente excepcional, es gratuita; se otorga por relaciones de comercio y no puede comercializarse.
<ul style="list-style-type: none"> • Fianza de Empresa 	<ul style="list-style-type: none"> • Es la garantía otorgada por una sociedad anónima, autorizada por el Gobierno para otorgar todo tipo de fianzas y cobrar por este servicio, pudiendo comercializar sus ventas por agentes comisionistas, operando bajo normas técnicas y apoyadas en leyes y reglamentos, y teniendo una capacidad de asumir responsabilidades por su solvencia económica y su técnica operacional.

Cuadro 3.1 Clases de fianzas

La afianzadora es una empresa de servicios por lo que su estructura organizacional, sus estrategias de ventas, su operación y su gente deben orientarse a obtener la excelencia en el servicio al cliente; de la misma manera, los sistemas se deben concebir para apoyar a la organización en el logro de este objetivo primario. Para estar más cerca de sus clientes, la compañía ofrece sus servicios a través de una extensa red de sucursales, oficinas de servicio y agentes comisionistas en toda la república mexicana.

La afianzadora, por medio de la fianza, garantiza que el fiado cumplirá su responsabilidad (fijada en un contrato) ante el beneficiario, por lo que antes de otorgarla investiga al fiado y se asegura de la solvencia moral y económica. La afianzadora emite una póliza en donde especifica lo que garantiza y su valor (monto de la fianza) para que el fiado se la entregue al beneficiario. Por este servicio le cobra al fiado un porcentaje del monto llamado prima, que está en función del riesgo de la operación y su monto afianzado. La vigencia de la fianza es por un año, pudiendo renovarse anualmente hasta que el beneficiario le indique a la afianzadora que el fiado cumplió con su obligación y puede proceder a la cancelación.

Dentro de la vigencia de la fianza es probable que, a petición del beneficiario, el fiado tramite con la afianzadora algunas modificaciones a las características originales de la fianza; estos accidentes de la fianza se les llama movimientos. Véase cuadro 3.2

Tipo de movimiento	Definición
• Expedición	• Es el inicio de toda fianza; la afianzadora emite una póliza de fianza garantizando el cumplimiento de una obligación por el fiado ante el beneficiario hasta un monto determinado.
• Renovación	• Al expirar la vigencia de la fianza (normalmente un año) y no haberse cumplido totalmente la obligación, el beneficiario le exige al fiado que se renueve la fianza por lo que la afianzadora emite una póliza de renovación en donde se especifica que la fianza continúa vigente con el mismo monto y cierta vigencia. La afianzadora cobra una prima por la renovación en función del monto y del período que se renueva.
• Aumento (Aumento del monto)	• Dentro de la vigencia de la fianza es posible que se incremente la obligación del fiado ante el beneficiario por anexos al contrato original. En este caso, el fiado solicita a la afianzadora un aumento al monto de la fianza por el período de vigencia que falta por transcurrir. La afianzadora cobra una prima por el aumento en función del incremento al monto y el período de vigencia que se cubre.
• Disminución (Disminución del monto)	• Es el caso contrario del punto anterior, y por lo tanto la afianzadora emite un endoso de disminución del monto sin cobro ni devolución de prima.
• Anulación del movimiento	• El fiado puede solicitar la anulación de un movimiento por error tipográfico o porque la fianza ya no se necesite. La afianzadora devuelve la prima que cobró por el movimiento original.
• Cancelación de la fianza	• Al cumplir el fiado la obligación ante el beneficiario, la fianza puede ser cancelada, extinguiéndose la responsabilidad de la afianzadora.
• Levantamiento de cancelación	• Movimiento administrativo que regresa a la fianza al estado en que se encontraba antes de que se cancelara por error.

Cuadro 3.2 Movimientos de la fianza.

En todos los movimientos de fianza que se cobra prima al fiado, la afianzadora le paga comisión al agente colocador; esta comisión es variable y depende del agente, el tipo de fianza y el movimiento que se está emitiendo.

Por el tipo de obligación que se garantiza, las fianzas se clasifican por ramos. Véase cuadro 3.3.

Ramo	Obligación garantizada
• I Fianzas de Fidelidad.	Garantizan el resarcimiento del daño patrimonial que cause un empleado por la comisión de un delito en contra de los bienes de la empresa beneficiaria o de los que ésta sea jurídicamente responsable.
• II Fianzas Judiciales.	Garantizan el cumplimiento de los deberes y obligaciones o actos de particulares o del sector público dentro de un procedimiento judicial o derivado de resoluciones judiciales
• III Fianzas Administrativas.	• Garantizan el cumplimiento de obligaciones generales entre dos partes, que no se pueden englobar en las fianzas de fidelidad y judiciales.
• IV Fianzas de Crédito.	• Es el instrumento operativo y legal, que garantiza la obligación de pago de recursos monetarios de compromisos crediticios entre empresas.

Cuadro 3.3 Ramos de la fianza

Cada ramo o producto tiene características propias en lo referente a vigencia, tarifa de la prima, forma de cobro, comisiones a los agentes colocadores y movimientos permitidos.

3.2 Relación con otras compañías afianzadoras.

La relación primordial entre las compañías afianzadoras surge cuando se excede el margen de operación de alguna compañía. El margen de operación es el monto máximo autorizado de las obligaciones que una afianzadora puede garantizar por sí sola en una póliza. Si el monto de la fianza es mayor que el margen de operación, la responsabilidad de la fianza deberá de compartirse con otras compañías de fianzas. A la aceptación de tal responsabilidad se le denomina reafianzamiento. En dicho reafianzamiento también se debe tomar en cuenta los márgenes de operación de cada afianzadora. En el momento del reafianzamiento se identifican dos procesos diferentes :

1) La afianzadora cede parte del monto de su póliza. Se le conoce como reafianzamiento cedido.

2) La afianzadora toma la parte cedida de una póliza de otra compañía. Se le conoce como reafianzamiento tomado.

El margen de operación es igual al 15% de los recursos de capital, considerados éstos de acuerdo con la forma que fija la Ley Federal de Instituciones de Fianzas.

La Secretaría de Hacienda y Crédito Público, a través de la Comisión Nacional de Seguros y Fianzas, fija trimestralmente los márgenes de operación de las Instituciones de Fianzas. Estos márgenes aparecen publicados en el Diario Oficial de la Federación.

El reafianzamiento es la operación de repartir la responsabilidad o la obligación garantizada entre varias instituciones de Fianzas. El reafianzamiento puede deberse por dos actividades principalmente la primera es por haber excedido el margen de operación y la segunda es por la siniestralidad de la fianza. En el cuadro 3.4 se muestran los tipos de reafianzamiento que se manejan.

Tipo	Definición
<ul style="list-style-type: none">• Reafianzamiento Automático	<ul style="list-style-type: none">• Se presenta con empresas con las cuales ya se tienen acuerdos de aceptación de compartir e determinadas proporciones, la responsabilidad de las obligaciones garantizadas por ésta. El reafianzamiento deberá realizarse de preferencia con instituciones mexicanas.
<ul style="list-style-type: none">• Reafianzamiento Cedido	<ul style="list-style-type: none">• Consiste en designar la empresa que compartirá la responsabilidad de garantizar el cumplimiento de la obligación y ésta lo acepta.
<ul style="list-style-type: none">• Reafianzamiento Tomado	<ul style="list-style-type: none">• Es la respuesta de aceptación del reafianzamiento que otra institución de fianzas propone.

Cuadro 3.4 Tipos de reafianzamiento

A continuación y para comprender mejor se explica con un ejemplo típico el ciclo de emisión de una fianza con reafianzamiento tomado.

El gobierno (Beneficiario) necesita construir un edificio, por lo tanto contrata a una compañía constructora (Fiado). La compañía constructora para empezar el proyecto solicita un anticipo, pero el gobierno requiere que se le garantice que ese dinero que se va dar de anticipo sea invertido en la obra y por lo tanto pide una fianza. El fiado acude a la afianzadora y esta le evalúa la operación y la solvencia del fiado, como el monto por garantizar excede el margen de operación de la afianzadora, ésta se ve obligada a pasar parte de la responsabilidad a otras compañías afianzadoras, después de esto la afianzadora inicial emite la póliza en la que garantiza al beneficiario que el fiado utilizará el anticipo en la obra, de lo contrario la afianzadora le pagará al beneficiario el monto de la fianza. La afianzadora cobra la prima debida y reparte en partes proporcionales a las demás afianzadoras, y las demás afianzadoras le pagan una comisión por haber colocado una fianza a la afianzadora original. Véase la figura 3.1.

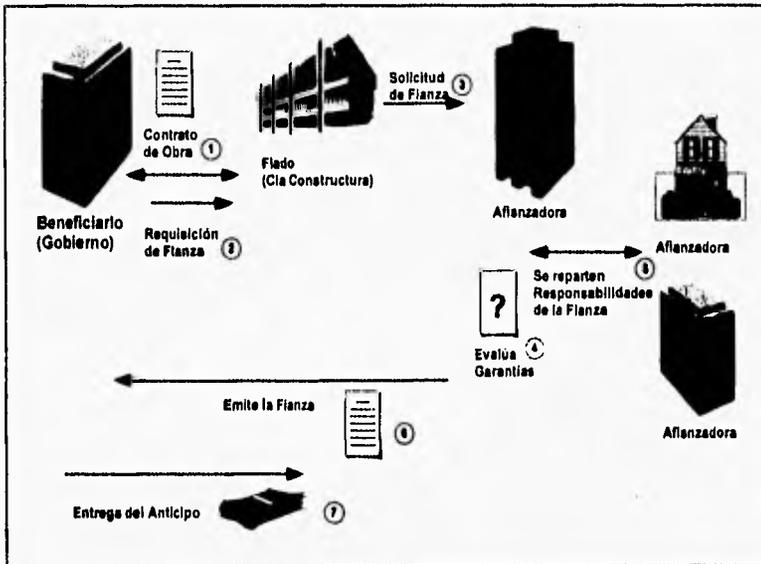


Figura 3.1 Ciclo de emisión de fianza con reafianzamiento

3.3 Situación actual de sistemas

La evolución permanente de los sistemas de información de la afianzadora ha transformado la base de su plataforma tecnológica de sistemas de un IBM S/36 a un equipo DEC VAX, en el cual se encuentra prácticamente el 100% de la operación de la empresa: el único sistema que aún se encuentra en operación en el IBM S/36 es el sistema de registro y control de fianzas de reaflanzamiento tomado nacional.

En 1990 la compañía adquiere un sistema micro VAX-3900 como equipo central y toda una infraestructura de comunicaciones es instalada para soportar el esquema que es mostrado en la figura 3.2.

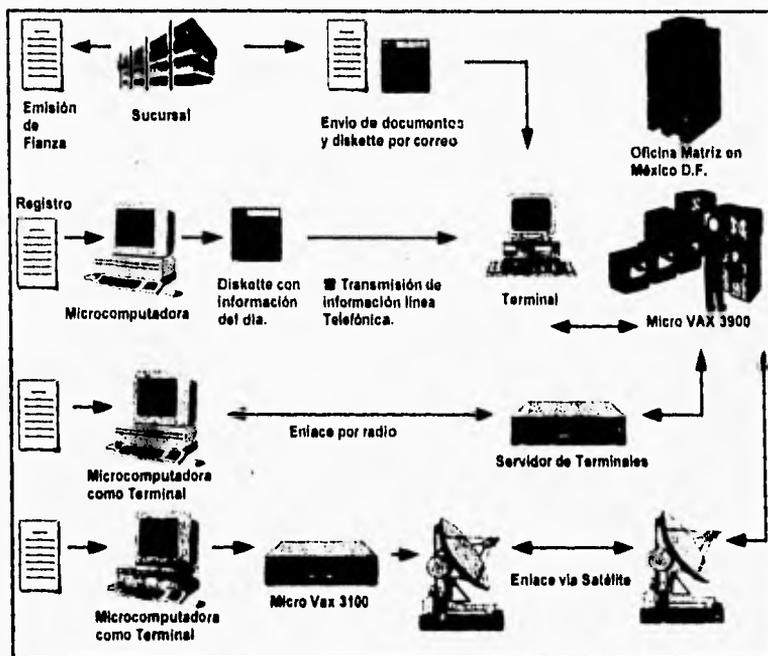


Figura 3.2 Esquema de operación.

El servidor central es una máquina VAX-3900 con 5.4 Gigabytes de almacenamiento y 64 Megabytes de memoria principal, se tienen servidores secundarios micro VAX-3100 en Guadalajara, Monterrey, Hermosillo y Tijuana conectados vía satélite a la ciudad de México. Se cuenta con un servidor de 50 terminales para el área metropolitana, se utilizan las computadoras personales que ya se tienen en los departamentos emulando terminales, estas microcomputadoras están conectadas en una red DECNET.

Para las sucursales locales del área metropolitana el enlace es vía radio y las sucursales pequeñas en el interior del país envían su información por diskette al centro divisional correspondiente para procesar información.

Actualmente, la afianzadora se encuentra iniciando un nuevo e importante proceso de reingeniería de sus procesos operativos el que deberá traducirse en nuevos sistemas de información que exploten eficientemente la plataforma tecnológica que se está creando y herramientas para desarrollo de sistemas de vanguardia.

El nuevo esquema de los sistemas de información cliente/servidor requiere definir tanto las características del cliente como el servidor. Para el servidor SYBASE ha sido definido como el servidor de datos trabajando en equipos DEC ALPHA y COMPAQ con Windows NT. Para el caso de los clientes, éstos son computadoras personales con Windows, con procesadores mayores al 80386 de Intel, ejecutando aplicaciones cliente/servidor.

Para poder iniciar formalmente la reconstrucción de los procesos y los sistemas operativos de la compañía, la afianzadora requiere terminar con el ciclo de operación del S/36 para lo cual es necesario desarrollar un nuevo sistema de registro y control de reafianzamiento tomado nacional (RT nacional), sistema del cual se ocupa esta tesis.

3.4 Planteamiento de sistema de reafianzamiento tomado

El sistema de reafianzamiento tomado se lleva a cabo en una máquina adquirida en 1984 que es un sistema IBM S/36, este sistema fue programado en RPG-II con un manejador de archivos nativo del S/36.

La compañía en su proceso de reingeniería quiere deshacerse del S/36, pero para hacerlo es necesario volver a realizar el sistema de reafianzamiento tomado que es el único que se maneja en el S/36.

Definiremos los procesos módulos o actividades principales que integran la operación del sistema de reafianzamiento tomado nacional:

- **Mantenimiento a los catálogos.**- Este módulo maneja la información general de la fianza, y se compone del catálogo de las compañías con que se trabaja, catálogo de monedas, de ramos y tipos, usuarios y cuentas. Es necesario que si hay algún cambio de los catálogos se haga de inmediato porque la información general de la fianza puede estar errónea.
- **Registro y control de las fianzas cedidas y sus movimientos.**- Este proceso es el más importante ya que aquí se centra el objetivo del sistema. hay que recordar que las fianzas cedidas deben de ser controladas y para eso es necesario el registro de la expedición, y sus posibles movimientos que son renovación, aumento, disminución, cancelación, levantamiento de cancelación y anulación, siempre bajo previo aviso de la compañía cedente.

Otro punto importante es cuando se necesitan hacer cambio de los datos de las fianzas o cambios sobre los movimientos registrados y también tener bien definido la divisa, ya que se diferencian si es moneda nacional o moneda extranjera.

- **Contabilización de la producción.**- Dentro del objetivo del sistema es el de contabilizar todos los movimientos de reafianzamiento tomado, para esto es necesario dar la información al sistema central contable que esta funcionando en una VAX-3100 del área contable, para que esto sea posible se necesita un esquema de migración de información en los que se incluyen las pólizas contables de expedición, renovación, aumento, cancelación, levantamiento de cancelación, anulación, disminución y liberación de futuros.
- **Control de cobranza.**- Este proceso requiere de estados de cuenta para la información a las compañías, también un control automático de cobranza desde el registro de producción y la creación de saldos por cobrar. El control de pagos de hechos por las compañías, aplicación de saldos a favor y cambio en ajustes de dinero.
- **Reportes y consultas de reafianzamiento tomado.**- Este módulo es necesario para tener de manera oportuna la información del registro de los movimientos, y también para los reportes que solicita la Comisión Nacional de Seguros y Fianzas.

Después de haber visto lo que es una compañía afianzadora, su manejo y los procesos principales para el sistema de reafianzamiento tomado podemos pasar a nuestra siguiente actividad que es la del análisis del sistema.

4 Análisis del Sistema

El análisis de sistemas es un proceso de descubrimiento, refinamiento, modelado y especificación en donde el analista y el usuario juegan un papel esencial. Esta etapa tiene como objetivo el conocimiento de situaciones y no la solución de problemas.

El análisis se compone principalmente de cuatro actividades principales: 1) reconocimiento del problema, 2) evaluación y síntesis, 3) modelado y 4) revisión.

El modelo propuesto de análisis del sistema tendrá como resultado los siguientes productos:

- Definición de requerimientos generales.
- Diagramas de flujo de datos del sistema y su diccionario de requisitos.
- Diagramas de entidad-relación

4.1 Objetivo del sistema.

El objetivo principal del Sistema de registro y control de reafianzamiento tomado es registrar los movimientos de fianzas que la afianzadora toma de otras compañías afianzadoras o aseguradoras nacionales, mantener el control de estas fianzas y asegurar su correcta aplicación en la contabilidad de la compañía.

Por lo anterior se concluye que el objetivo general del sistema de registro y control de reafianzamiento tomado (RT) nacional es el registrar los movimientos de fianzas que una afianzadora toma de otras compañías afianzadoras y aseguradoras nacionales, mantener el control de estas fianzas y asegurar su correcta aplicación en la contabilidad de la empresa.

4.2 Definición de Requerimientos.

Todo elemento desarrollado por el hombre es una idea en su mente. Los sistemas automatizados, como otros productos de la tecnología se desarrollan en respuesta a requerimientos detectados. Las fuentes que deben proporcionar las ideas deben incluir las necesidades del cliente generadas externamente, las necesidades internas de la organización y planes de mercadotecnia.

Las técnicas para obtener este conocimiento, por parte del analista, son entrevistas con el cliente, observación de las tareas problemáticas y desarrollo de las reales.

Un requerimiento es una característica que debe incluirse en el nuevo sistema y puede ser una forma de captar o generar información. En el capítulo anterior se presentaron las características de la empresa y su relación con otras de su mismo ramo.

4.2.1 Requerimientos generales

Para el desarrollo de la definición de requerimientos generales es necesario adentrarse a detalle a la operación del sistema de reafianzamiento tomado nacional, y de acuerdo a lo visto en el capítulo dos los requerimientos generales son los siguientes.

1. Registro y control de las fianzas cedidas y sus movimientos de compañías nacionales.

- Registro de expediciones de fianzas.
- Registro de renovación a fianzas.
- Registro de aumentos a fianzas.
- Registro de disminuciones a fianzas.
- Registro de cancelaciones de fianzas.
- Registro de levantamientos de cancelación de fianzas.
- Registro de anulaciones de movimientos de fianzas.
- Cambios a datos de las fianzas.
- Cambios a movimientos registrados.
- Registro de fianzas en moneda nacional y extranjera.
- Registro de movimientos de fianzas fuera de secuencia de control de emisión.

2. Mantenimiento a los catálogos utilizados en RT nacional.

- Ramos - tipos.
- Monedas.

- Compañías afianzadoras.
- Operaciones.

3. Contabilización de la producción.

- Póliza de contabilización de expedición.
- Póliza de contabilización de renovación.
- Póliza de contabilización de aumento.
- Póliza de contabilización de cancelación.
- Póliza de contabilización de levantamiento de cancelación.
- Póliza de contabilización de anulación.
- Póliza de contabilización de disminución.
- Póliza de contabilización de liberación de futuros.
- Transferencia de las pólizas de contabilización de RT al sistema central de contabilización.
- Manejo flexible de las cuentas de referencia en las pólizas contables, es decir, poder cambiar las cuentas de cada póliza contable.
- Impresión de las pólizas de contabilización.

4. Control de la cobranza a las compañías.

- Impresión de estados de cuenta de cobranza.
- Control automático de cobranza desde el registro de producción, creación de saldos por cobrar.
- Manejo de saldos por cobrar, saldos en rojos, saldos a favor de la compañía por anulada pagada y ajustes.

- Registro de pagos hechos por las compañías. aplicación de saldos a favor y cambio en ajustes.

5. Reportes de reafianzamiento tomado.

- Inventarios de fianzas en vigor detallados y en resumen.
- Estados de cuenta de cobranza.
- Estadísticos de producción.
- Pólizas de contabilización.
- Reportes de los catálogos actualizados.
- Detallados de producción y validación de registro.
- Inventario de primas futuras.
- Futuros liberados.

6. Consultas de reafianzamiento tomado.

- Fianzas.
- Movimientos de fianzas.
- Fianzas en moneda extranjera y sus movimientos.
- Saldos por cobrar, rojos, anuladas y ajustes.
- Futuros creados y liberados.
- Catálogos del sistema.

7. Parámetros del sistema.

- Control de usuarios y su acceso.
- Datos generales como son el iva para todas la fianzas, etc.

4.3 Modelado

Gran parte del trabajo desarrollado consiste en el modelado del sistema. Los modelos de análisis de sistemas son en su gran mayoría modelos en papel, es decir, representaciones abstractas de lo que al final será una combinación de software y hardware de computadora.

El ¿por qué? de construir modelos, es para enfatizar ciertas propiedades críticas del sistema, mientras que simultáneamente desacentuamos otros de sus aspectos. Esto nos permite comunicarnos con el usuario de una manera enfocada, sin distraernos con asuntos y características ajenas al sistema; esto nos permite percatarnos si la comprensión de los requerimientos fue la correcta.

En resumen se usan las herramientas de modelado para:

- Concentrarse en las propiedades importantes del sistema y al mismo tiempo restar atención a otras menos importantes.
- Discutir cambios y correcciones de los requerimientos del usuario, a bajo costo y con riesgo mínimo.
- Verificar si se comprendió correctamente el ambiente del usuario y se ha respaldado con suficiente información documental tanto para el diseño como para la programación.

Las tres herramientas de modelado de sistemas importantes son el diagrama de flujo de datos, el diagrama de entidad-relación, y el diagrama de transición de estados. El diagrama de flujo de datos ilustra las funciones que el sistema debe realizar; los diagramas de entidad-relación hacen énfasis en las relaciones entre los datos y el diagrama de transición de estados se enfoca al comportamiento dependiente del tiempo del sistema.

4.3.1 Diagramas de flujo de datos

El diagrama de flujo de datos (DFD) es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y "tanques de almacenamiento" de datos.

Antes de examinar los componentes de un DFD con detalle, es necesario hacer notar que:

- Un DFD prácticamente no necesita explicación; al mirarlo se debe entender fácilmente. La notación es sencilla y clara.
- El diagrama debe caber fácilmente en una página. Esto indica dos cosas: 1) alguien puede mirarlo sin ofuscarse y 2) el sistema que se está modelando aparenta no ser muy complejo.
- Si el diagrama es hecho por computadora indica que se pueden hacer cambios y producir nuevas versiones en cuestión de minutos.

El primer componente de un DFD se conoce como proceso. El proceso muestra una parte del sistema que transforma entradas en salidas; es decir, muestra cómo es que una o más entradas se transforman en salidas. El proceso se representa gráficamente como un bloque.

Un flujo se representa gráficamente por medio de una flecha que entra o sale de un proceso. El flujo se usa para describir el movimiento de paquetes de información de una parte a otra. Por ello los flujos representan datos en movimiento, mientras que los almacenes representan datos en reposo. Existen tres diferentes tipos de flujo que son: el flujo de entrada, el flujo de salida y el flujo de diálogo.

El almacén se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por un rectángulo parcial, este almacén en realidad son archivos o bases de datos.

4.3.2 Diccionario de requisitos

Un análisis del sistema de información estaría incompleto si sólo se considera el flujo de información. Cada elemento del diagrama de flujo de datos debe quedar definido totalmente, así como las relaciones entre ellos.

El diccionario de requisitos es una relación detallada de todos los elementos que conforman el diagrama de flujo de datos: la definición precisa de

los almacenes de información, la descripción detallada de los procesos y la explicación de los elementos externos integran este diccionario de requisitos.

4.3.3 Diagramas de flujo de datos del sistema de reafianzamiento tomado nacional y su diccionario de requisitos.

El análisis de las entradas y salidas del sistema en general definió las entidades externas al sistema que tienen relación con él. El DFD inicial muestra al sistema como un solo proceso y las entidades externas con las que se relaciona son:

- **Otras afianzadoras:** esto se refiere a compañías afianzadoras que envían fianzas para ser registradas y controladas por el sistema.
- **Contabilidad:** esta área interna de la afianzadora requiere información para la contabilización de los inventarios de la fianzas en vigor y las primas futuras, el registro de las ventas, sus ingresos y la cobranza.

Un análisis más detallado nos mostró siete procesos principales en el sistema de reafianzamiento tomado: 1) Catálogos del sistema, 2) Registro y control de fianzas tomadas, 3) Registro de cobranza a las compañías, 4) Contabilización de la producción, 5) Consultas a reafianzamiento tomado, 6) Reportes de reafianzamiento tomado y 7) Seguridad del sistema.

El proceso de modelado por DFD debe ir progresando de lo general a lo particular, detallando separadamente cada uno de los procesos hasta el nivel requerido, es recomendable no usar un nivel de abstracción mayor a los 3 niveles, en este sistema fueron solo necesario 2 niveles de abstracción y por esto que no es muy complicado para el usuario el entendimiento del sistema.

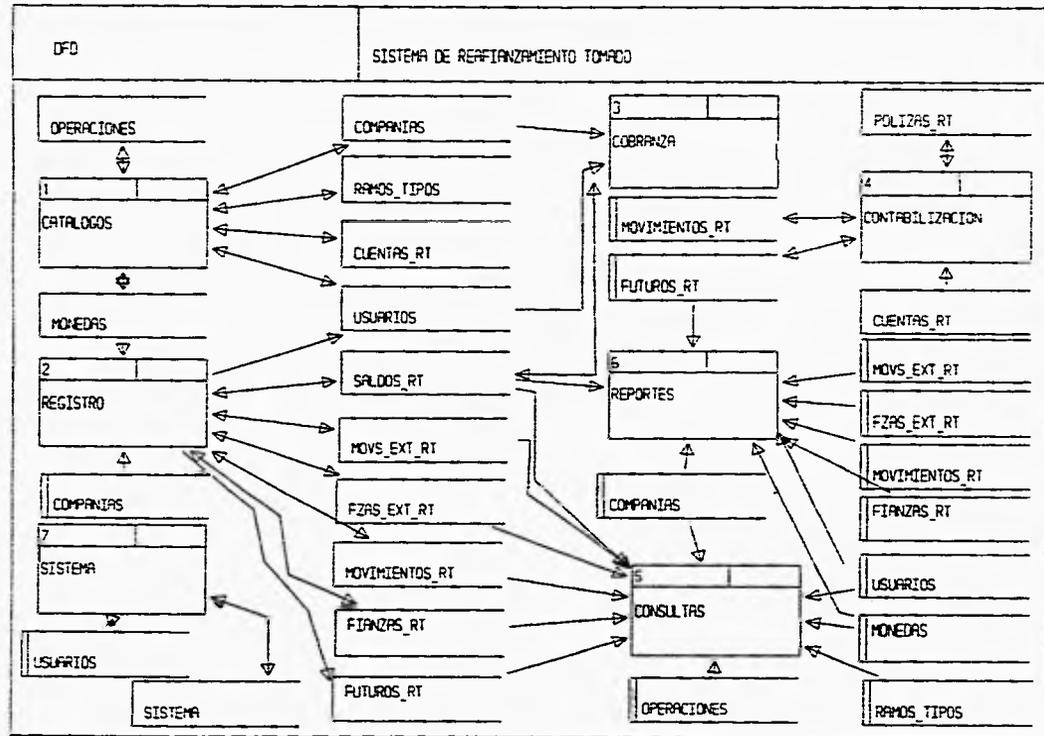


Figura 4.1 Proceso principal del sistema de reafianzamiento tomado

Proceso principal: 1.- Mantenimiento a los catálogos del sistema.

Descripción: Módulo utilizado por el área de administración de reafianzamiento para mantener actualizados los catálogos utilizados por el sistema.

Subprocesos que integran el proceso principal:

1.1. Catálogo de monedas: mantenimiento a la información referente a las monedas en las que se pueden registrar fianzas en reafianzamiento tomado.

1.2. Catálogo de operaciones: mantenimiento a la información referente a las operaciones o movimientos que se le pueden hacer a las fianzas en RT.

1.3. Catálogo de compañías: mantenimiento a la información referente a las compañías afianzadoras y aseguradoras que ceden fianzas.

1.4. Catálogo de ramos - tipos: mantenimiento a la información de los diferentes tipos de fianzas que se toman de la cesión de otras compañías.

1.5. Catálogo de usuarios: mantenimiento a la información referente a los usuarios registrados en el sistema y su nivel de acceso dentro de éste.

1.6. Catálogo de cuentas para las pólizas: mantenimiento a las cuentas que son afectadas por cada tipo de póliza de producción y liberación de futuros. Mediante este proceso la referencia de las cuentas para las pólizas de producción debe ser totalmente flexible ya que sólo es necesario modificar este catálogo para cambiar la contabilización en las pólizas.

Estructuras de datos asociadas al proceso principal:

Monedas (monedas): catálogo de las monedas en que se puedan registrar fianzas en RT. Incluye la clave de la moneda, su nombre y su paridad de ajuste mensual.

Operaciones (operaciones): catálogo de las operaciones que se le pueden realizar a las fianzas en RT.

Compañías (companias): catálogo de compañías afianzadoras y aseguradoras que ceden fianzas. Incluye información general como su clave, nombre, RFC, contacto, etc.

Ramos - Tipos (ramos_tipos): catálogo de los diferentes tipos de fianzas que se pueden registrar en el sistema de RT.

Usuarios (usuarios): catálogo de los usuarios registrados con acceso al sistema.

Cuentas de las pólizas RT (cuentas_rt): catálogo donde se mantiene la información de las cuentas que afecta cada póliza de contabilización generada por el sistema de RT.

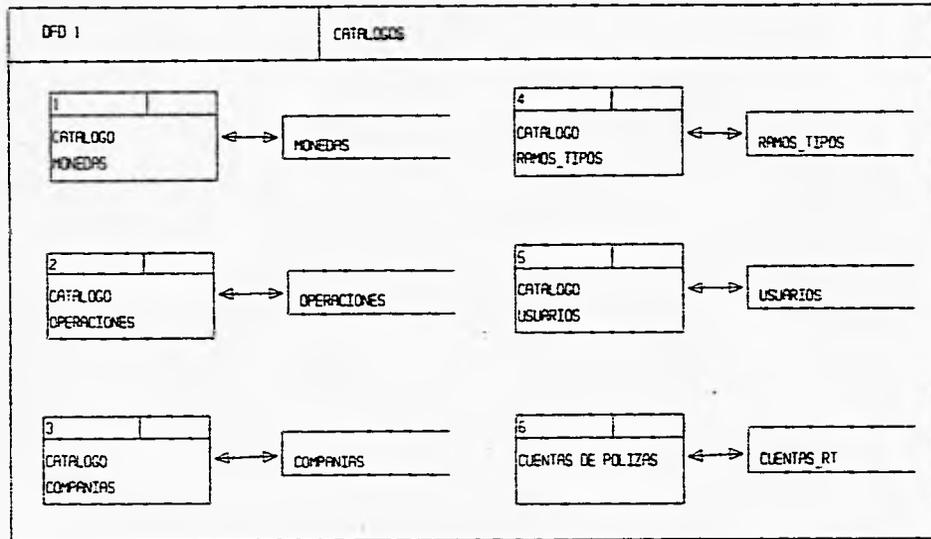


Figura 4.2 Mantenimiento de catálogos

Proceso principal: 2.- Registro de fianzas en RT.

Descripción: Módulo utilizado por el área de administración de reafianzamientos para registrar los movimientos de fianzas cedidas a la afianzadora en RT.

Subprocesos que integran el proceso principal:

2.1. Registro de expedición: proceso de carga de expediciones de fianzas tomadas en RT .

2.2. Registro de renovación: proceso de carga de renovaciones a fianzas tomadas en RT .

2.3. Registro de aumento: proceso de carga de aumentos a fianzas tomadas en RT .

2.4. Registro de anulación: proceso de identificación de anulación de movimientos registrados de fianzas tomadas en RT .

2.5. Registro de disminución: proceso de carga de disminución de fianzas tomadas en RT .

2.6. Registro de cancelación: proceso de cancelación de fianzas tomadas en RT .

2.7. Registro de levantamiento de cancelación: proceso de levantamiento de cancelación de fianzas tomadas en RT .

Estructuras de datos asociadas al proceso principal:

Fianzas en reafianzamiento tomado (fianzas_rt): información de las fianzas en RT con otras compañías.

Movimientos de fianzas en RT (movimientos_rt): información de los movimientos realizados a las fianzas en RT.

Fianzas en RT en moneda extranjera (fianzas_ext_rt): información en moneda extranjera de las fianzas en RT con esta particularidad.

Movimientos en RT de fianzas en moneda extranjera (movimientos_ext_rt): información de los movimientos realizados a las fianzas en RT en moneda extranjera.

Proceso principal: 3.- Registro de cobranza a las compañías.

Descripción: Módulo utilizado por el área de administración de reafianzamientos para controlar los saldos por cobrar de las fianzas en RT de las compañías cedentes y registrar sus pagos.

Subprocesos que integran el proceso principal:

3.1. Pagar saldos por cobrar de las compañías: proceso para registrar los pagos que hacen las compañías cedentes por concepto de RT.

3.2. Creación de saldos en rojo: proceso para registrar pagos como saldos en rojo a favor de las compañías.

3.3. Ajustes al saldo: este proceso crea registros para ajustar los saldos totales por compañía a los saldos contables por diferencias en pagos.

3.4. Modifica ajuste de saldos: cambio a los ajustes creados por compañía para comparar contra saldos en contabilidad.

3.5. Aplicación de anulada pagada: proceso para utilizar el saldo a favor de las compañías debido a una anulación de fianza pagada en el pago de otras primas por cobrar.

3.6. Aplicación de saldo en rojo: proceso que aplica saldos en rojo a favor de las compañías para pagar otras primas por cobrar.

Estructuras de datos asociadas al proceso principal:

Primas por cobrar a las compañías (saldos_rt): información de los saldos por cobrar a las compañías de las fianzas tomadas. Incluye además información del pago de comisión.

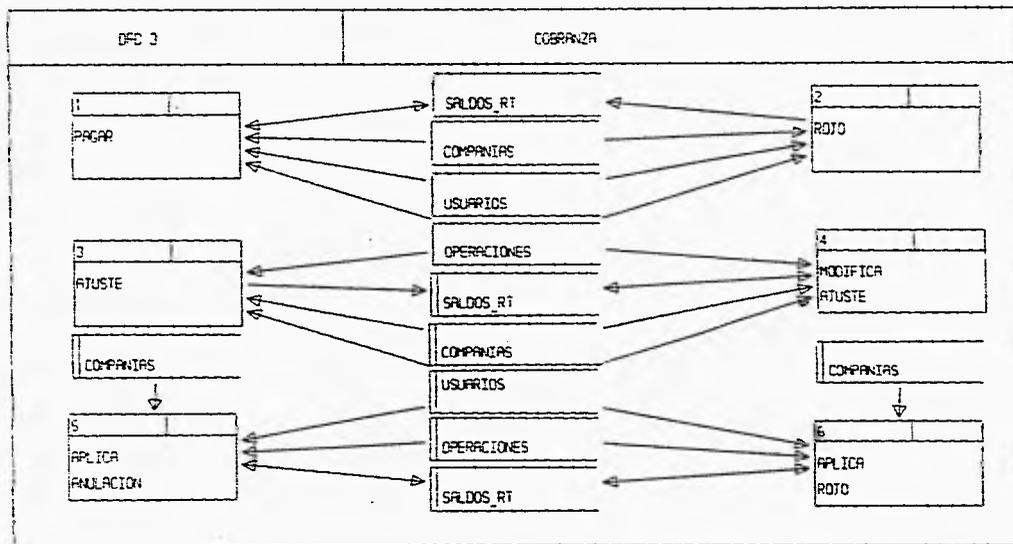


Figura 4.4 Registro de cobranza

Proceso principal: 4.- Contabilización de la producción.

Descripción: Módulo utilizado por el área de administración de reafianzamientos para crear los registros contables en las correspondientes pólizas tanto de producción como de liberación de futuros. El módulo incluye la interfase con el sistema central de contabilidad.

Subprocesos que integran el proceso principal:

4.1. Pólizas de expedición: proceso que genera los registros contables de la póliza de expedición de la producción en un período. Este proceso utiliza la información del catálogo de cuentas de contabilización para crear los registros contables correspondientes.

4.2. Póliza de renovación: proceso que genera los registros contables de la póliza de renovación de la producción en un período. Este proceso utiliza la información del catálogo de cuentas de contabilización para crear los registros contables correspondientes.

4.3. Póliza de aumento: proceso que genera los registros contables de la póliza de aumento de la producción en un período. Este proceso utiliza la información del catálogo de cuentas de contabilización para crear los registros contables correspondientes.

4.4. Póliza de disminución: proceso que genera los registros contables de la póliza de disminución de la producción en un período. Este proceso utiliza la información del catálogo de cuentas de contabilización para crear los registros contables correspondientes.

4.5. Póliza de anulación: proceso que genera los registros contables de la póliza de anulación de la producción en un período. Este proceso utiliza la información del catálogo de cuentas de contabilización para crear los registros contables correspondientes.

4.6. Póliza de cancelación: proceso que genera los registros contables de la póliza de cancelación de la producción en un período. Este proceso utiliza la información del catálogo de cuentas de contabilización para crear los registros contables correspondientes.

4.7.- Póliza de levantamiento de cancelación: proceso que genera los registros contables de la póliza de levantamiento de cancelación de la producción en un período. Este proceso utiliza la información del catálogo de cuentas para contabilización de crear los registros contables correspondientes.

4.8.- Póliza de liberación de futuros: proceso donde se generan los registros para contabilizar los futuros que se liberan en el período creando la póliza contable correspondiente.

4.9.- Inicializa registros de pólizas: proceso para limpiar la estructura de transferencia para la interfase hacia el sistema central contable.

4.10.- Transferencia de pólizas: proceso que genera los archivos de interfase para transferir los registros de las pólizas al sistema central contable.

Proceso principal: 5.- Consultas a reafianzamiento tomado.

Descripción: Módulo utilizado por el área de administración de reafianzamientos para consultar a cualquier nivel la información generada por el reafianzamiento tomado.

Subprocesos que integran el proceso principal:

5.1. Consulta de fianzas en RT: este proceso presenta la información de cada fianza en RT; datos de la fianza y sus movimientos.

5.2. Consulta de fianza RT en moneda extranjera: este proceso presenta la información de las fianzas en RT en moneda extranjera y sus movimientos.

5.3. Consulta de saldos de RT: este proceso presenta la información del estado de las primas por cobrar a las compañías por el RT de fianzas.

5.4. Consulta de futuros en RT: proceso que presenta los datos de los futuros creados y liberados por las fianzas tomadas en RT.

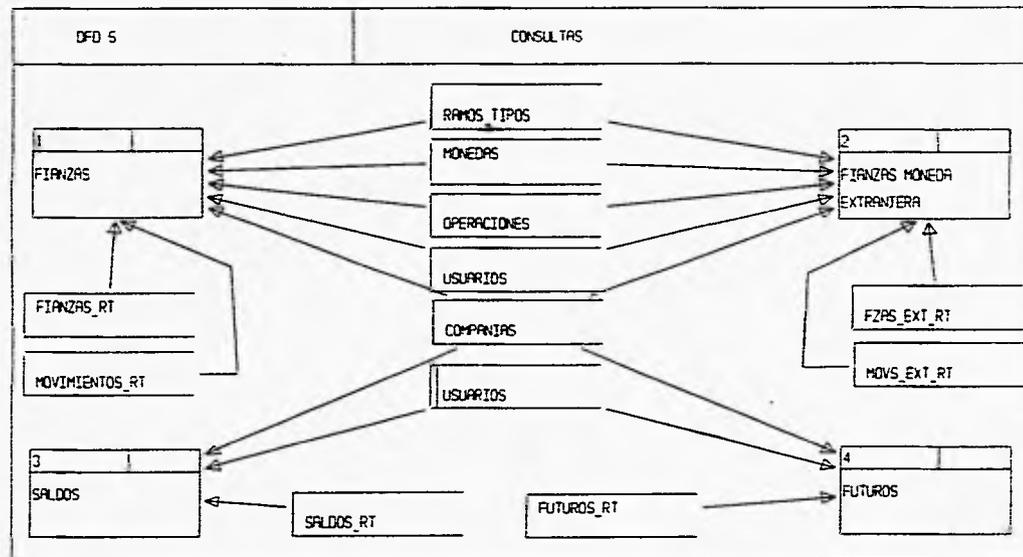


Figura 4.6 Consultas

Proceso principal: 6.- Reportes de reafianzamiento tomado.

Descripción: Módulo utilizado por el área de administración de reafianzamientos para obtener información impresa de las fianzas en RT.

Subprocesos que integran el proceso principal:

6.1. Reportes de inventario de RT: proceso que integra reportes varios de inventarios de fianzas en vigor que presentan Información de reafianzamiento tomado.

6.2. Reportes de producción de RT: proceso que incluye reportes varios de la producción registrada.

6.3. Reportes de saldos de RT: incluye reportes varios de la información sobre las primas por cobrar de las compañías por reafianzamiento tomado.

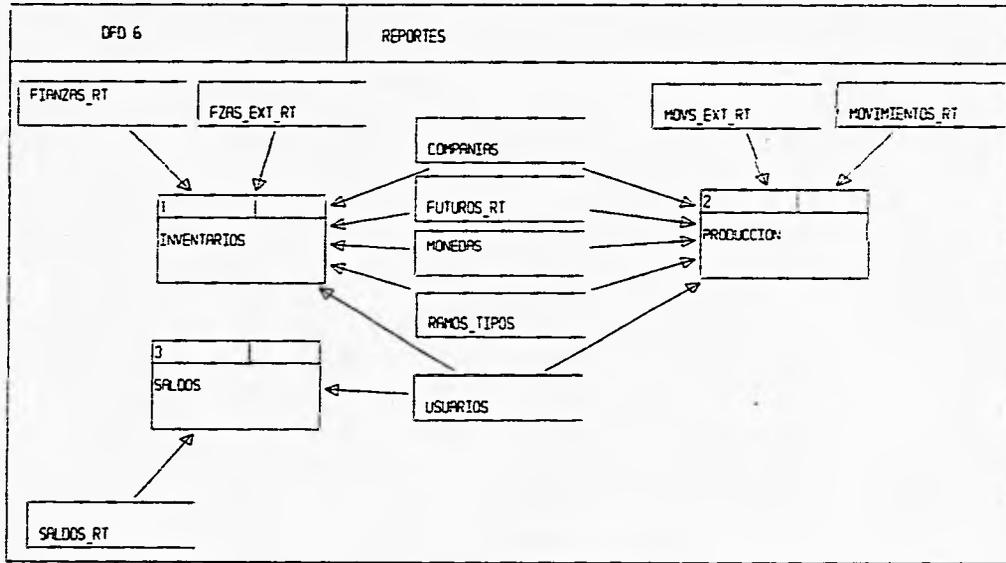


Figura 4.7 Reportes

Proceso principal: 7.- Control y seguridad del sistema.

Descripción: Módulo utilizado para la seguridad del sistema. Principalmente restringe la entrada a usuarios no autorizados y también maneja los parámetros por omisión del sistema.

Subprocesos que integran el proceso principal:

7.1. Control de usuarios: proceso que integra tanto el acceso de usuarios, como el nivel al que pueden utilizar el sistema.

7.2. Parámetros del sistema: proceso que incluye todos los parámetros que son necesarios para la ejecución del sistema.

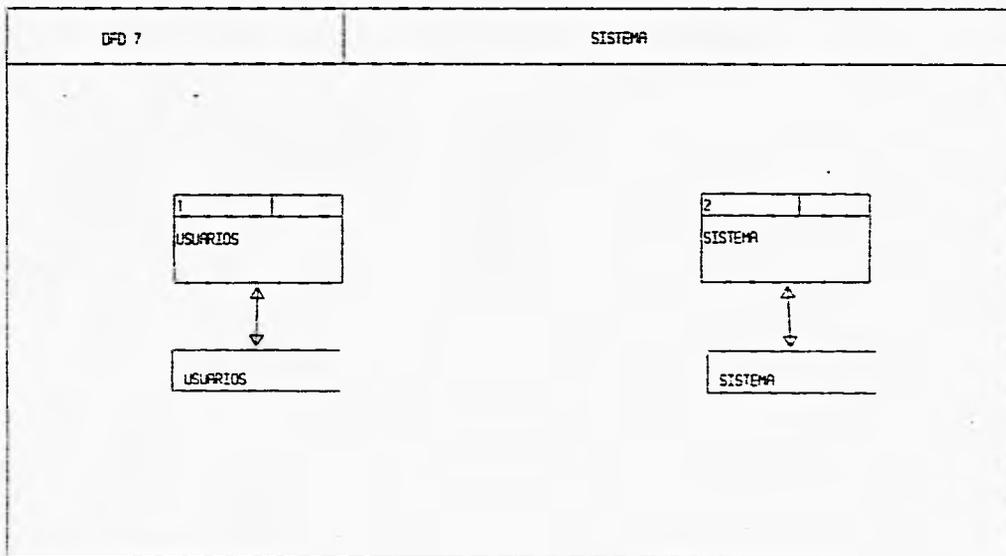


Figura 4.8 Control y seguridad del sistema

4.3.4 Diagrama de entidad-relación del sistema de reafianzamiento tomado

La notación de los diagramas de entidad relación es muy sencilla, los objetos de datos se representan mediante rectángulos, las relaciones mediante rombos, y las conexiones entre objetos de datos y relaciones mediante líneas de conexión.

En el diagrama de flujo de datos se reconocieron las entidades que integran el sistema. Por la manera en que se usan y la información que deben contener se definen las relaciones entre ellas.

La creación del diagrama de entidad relación del sistema de reafianzamiento tomado se inicia considerando todas las tablas (entidades) que genera el diccionario de requisitos. se analizan las relaciones entre ellas y se clasifican de manera que arrojen si las relaciones son uno a uno o uno a muchos. De todo esto se crea el diagrama de entidad-relación que se muestra a continuación.

5 Diseño del sistema

Una vez finalizado el análisis del sistema, todo se vuelve cuestión de desarrollo. La parte tangible de esta labor es la programación y la prueba, que se mostrarán en el capítulo 6. Sin embargo, la programación debe ir precedida por una actividad de nivel superior: el diseño.

El diseño es el primer paso de la fase de creación de cualquier sistema de ingeniería y su objetivo principal es producir un modelo de una entidad que se construirá más adelante. El diseño normalmente es creado en base a la experiencia de construir entidades similares, un conjunto de principios que guían la forma de desarrollo de un modelo y un proceso de iteración que va relacionado con la calidad del modelo, que conduce finalmente a una representación del diseño final.

En la figura 5.1 se muestran las entradas y salidas del proceso de diseño. Los requisitos del software, establecidos mediante la relación de requerimientos, el modelo de flujo de datos y el modelo de entidad relación, alimentan esta etapa, y la salida esta compuesta por el diseño de datos, el diseño estructural y el de interfase.

El diseño de datos transforma el modelo del campo de información, creado durante el análisis, en el diccionario y las estructuras de datos. El diseño estructural define la estructura del producto del software y la interfase del usuario.

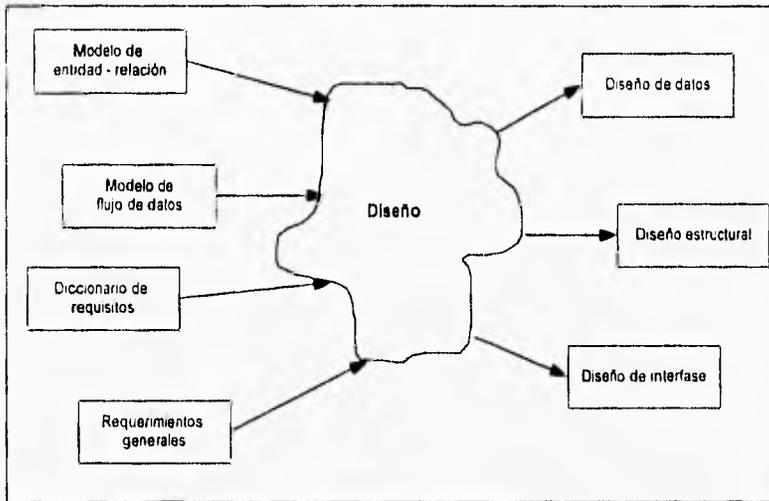


Figura 5.1 Entradas y salidas del diseño

5.1 Diseño de datos

El diseño de datos es la primera de las actividades de diseño realizadas durante la ingeniería de software. El impacto de la estructura de datos sobre la estructura del programa y la complejidad procedimental, hace que el diseño de datos tenga una gran influencia en la calidad del software. El diseño de datos da como resultado los siguientes productos: el diccionario de datos y las estructuras de datos.

5.1.1 Diccionario de datos

El proceso de diseño de datos debe iniciar con la creación del diccionario de datos. El diccionario de datos contiene la nomenclatura de cada campo, el tipo y la descripción mínima necesaria. La base de todo sistema de información utilizando herramientas de cuarta generación es precisamente el diccionario de datos.

Para la obtención del diccionario de datos es necesario recurrir a los modelos de análisis como lo son los diagramas de flujo de datos. Para cada elemento se deben definir sus características como dato, es decir, el tipo de dato (numérico, carácter, fecha, etc.), su formato o máscara y su tamaño o longitud en base a la información que debe contener.

La manera en la que se crea el diccionario de datos es utilizando los diagramas de flujo de datos de lo general a lo particular; por otro lado el diccionario de requisitos lo utilizamos para la redacción de la descripción de cada uno de los elementos que integran los diagramas de flujo de datos.

Para la creación del diccionario de datos del sistema, se analizó primero el diagrama de flujo general del sistema y así llegando a particularizar cada uno de los procesos y de esta manera se pudieron obtener los elementos dato necesarios sin que hubiera repeticiones. Estos elementos se pueden obtener por medio de los flujos entre los procesos.

El análisis de flujo de datos del sistema de reafianzamiento tomado de una afianzadora arroja una relación de la información que deba manejarse en el sistema. Para cada campo dato debe definirse un nombre, sus características y su descripción obteniendo finalmente el diccionario de datos. La tabla 5.1 constituye el diccionario de datos completo del sistema para el registro y control de la información de reafianzamiento tomado de una compañía afianzadora.

CAMPO	DEFINICION	DESCRIPCION
bis_fianza	entero (2)	Bis de la fianza tomada.
codigo_oper	entero (1)	Código de operación para histórico de movimientos.
comentarios	texto	Comentarios referentes al uso de la cuenta.
comis_mov	numérico (9.2) moneda	Importe de la comisión por el movimiento tomado.
concepto	caracter (40)	Descripción del concepto de la cuenta.
consecutivo_cta	entero (1)	Secuencia para identificar cada cuenta de la póliza.
contacto_cia	caracter (40)	Nombre del contacto con la compañía afianzadora.
cve_aplicacion	caracter (2)	Clave que identifica la operación que contabiliza.
cve_cia	entero (2)	Clave de la compañía afianzadora.
cve_colectiva	entero (2)	Clave de identificación de la colectiva en la contabilidad.
cve_concepto	entero (2)	Clave de identificación del concepto en la contabilidad.
cve_cta	entero (4)	Clave de la cuenta contable.
cve_div	entero (1)	Clave de división de la póliza contable.
cve_mone	entero (2)	Clave de la moneda original de la fianza.
cve_oper_mov	caracter (2)	Clave de la operación del movimiento.
cve_poliza	caracter(2)	Clave de identificación de la póliza correspondiente a la operación.
cve_ramo	entero(1)	Clave del ramo de la fianza.
cve_scta1	entero (2)	Clave de la primera subcuenta de la contabilización de producción.

CAMPO	DEFINICION	DESCRIPCION
cve_scta2	entero (2)	Clave de la segunda subcuenta de la contabilización de producción.
cve_scta3	entero (2)	Clave de la tercera subcuenta de la contabilización de producción.
cve_sistema	caracter (8)	Clave del sistema para acceder los datos generales.
cve_suc	entero (2)	Clave de la gerencia de la póliza contable.
cve_tipo	entero (2)	Clave tipo de la fianza.
cve_usuario	caracter (10)	Clave del usuario del sistema.
cve_zona	entero (2)	Clave de la zona a la que pertenece la gerencia de contabilización.
desc_ramo_tipo	caracter (30)	Descripción del ramo y tipo de la fianza.
direccion_cia	caracter (60)	Dirección de la compañía afianzadora.
edo_fianza	caracter (1)	Estado de la fianza (V, A, C).
fec_anul_mov	fecha	Fecha de anulación del movimiento.
fec_emis_mov	fecha	Fecha de emisión del movimiento.
fec_fin_vig_fut	fecha	Fecha de fin de vigencia del futuro.
fec_fin_vig_mov	fecha	Fecha de vigencia del movimiento.
fec_ini_vig_fut	fecha	Fecha de inicio de vigencia del futuro.
fec_fin_vig_fza	fecha	Fecha de fin de vigencia de la fianza.
fec_ini_vig_fza	fecha	Fecha de inicio de vigencia de la fianza.
fec_ini_vig_mov	fecha	Fecha de inicio de vigencia del movimiento.

CAMPO	DEFINICION	DESCRIPCION
fec_liber_fut	fecha	Fecha de liberación del futuro.
fec_pago	fecha	Fecha de pago.
fec_reg_mov	fecha	Fecha de registro del movimiento.
importe_debe	numérico (13.2) moneda	Importe en debe del registro contable.
importe_haber	numérico (13.2) moneda	Importe en haber del registro contable.
iva_comis	numérico (9.2) moneda	Importe del iva de la comisión.
iva_mov	numérico (9.2) moneda	Importe del iva del movimiento.
monto_fianza	numérico (13.2) moneda	Importe del monto de la fianza.
monto_fza_ext	numérico (13.2) moneda	Importe en moneda extranjera de la fianza.
monto_mov	numérico (13.2) moneda	Monto del movimiento.
monto_mov_ext	numérico (13.2) moneda	Monto del movimiento en moneda extranjera.
nivel_usuario	entero (1)	Nivel de acceso del usuario.
nom_cia	caracter (40)	Nombre de la compañía.
nom_fiado	caracter (40)	Nombre del fiado.
nom_mone	caracter (30)	Nombre de la moneda de la fianza.
nom_oper	caracter (30)	Nombre de la operación.
nom_usuario	caracter (30)	Nombre del usuario del sistema.
num_fianza	entero (6)	Número de fianza tomada.

CAMPO	DEFINICION	DESCRIPCION
pagado	numérico (13.2) moneda	Importe pagado.
paridad_ajuste	numérico (2.2) moneda	Importe del ajuste a la paridad.
password	caracter (10)	Clave del usuario para acceso.
porc_comis_r1	numérico (1.2)	Porcentaje de comisión para ramo 1.
porc_comis_r2	numérico (1.2)	Porcentaje de comisión para ramo 2.
porc_comis_r3	numérico (1.2)	Porcentaje de comisión para ramo 3.
porc_comis_r4	numérico (1.2)	Porcentaje de comisión para ramo 4.
porc_iva	numérico (1.2)	Porcentaje de iva.
porc_ret_reserva	numérico (1.2)	Porcentaje de retención de reserva por la compañía.
pri_fut_mov	numérico (13.2) moneda	Prima futuro de movimiento.
prima_fianza	numérico (13.2) moneda	Prima de fianza.
prima_fut	numérico (13.2) moneda	Prima por futuros.
prima_fza_ext	numérico (10.2) moneda	Prima de la fianza en moneda extranjera.
prima_mov	numérico (13.2) moneda	Prima del movimiento.
prima_mov_ext	numérico (13.2) moneda	Prima del movimiento en moneda extranjera.
prima_neta	numérico (13.2) moneda	Prima neta.
puesto_contacto_cia	caracter (30)	Puesto del contacto de la compañía.
rango_colectiva	entero (2)	Clave del rango de la colectiva para contabilización.

CAMPO	DEFINICION	DESCRIPCION
rango_concepto	entero (2)	Clave del rango del concepto para contabilización.
referencia_cia	caracter (13)	Referencia de la compañía (su fianza).
reserva_fianza	numérico (13.2) moneda	Importe de reserva de la fianza.
reserva_fza_ext	numérico (13.2) moneda	Importe de reserva de la fianza en moneda extranjera.
reserva_mov	numérico (13.2) moneda	Importe de reserva del movimiento.
reserva_mov_ext	numérico (13.2) moneda	Reserva del movimiento en moneda extranjera.
reserva_ret_fza	numérico (13.2) moneda	Reserva retenida por la compañía.
reserva_ret_mov	numérico (13.2) moneda	Reserva retenida del movimiento por la compañía.
rfc_cia	caracter (13)	Clave de registro federal de contribuyentes de la compañía.
saldo	numérico (13.2) moneda	Importe del saldo.
signo	caracter (1)	Clave de debe o haber del registro contable.
tipo_saldo	caracter (1)	Tipo de saldo para las compañías (Por cobrar, rojo, anulado).
total	numérico (13.2) moneda	Importe total por cobrar.

5.2 Normalización de datos

El proceso de cristalización de las entidades y su relación en formatos de tabla de dos dimensiones usando los conceptos relacionales se llama proceso de normalización. Para esto debemos apoyarnos por el modelo de entidad-relación obtenido durante el análisis, y enfocarnos en las relaciones existentes en el entorno de base de datos.

Una de las principales ventajas del planteamiento relacional es su simplicidad, es decir, la facilidad de comprensión por parte del usuario final.

El proceso de normalización de datos descompone una relación inicial universal, conteniendo todos los datos del diseño de la base adquiridos del diccionario de datos, en varias relaciones más pequeñas. Cada una de las relaciones resultantes se examina para ver si es necesaria una nueva normalización.

Un modelo de datos no normalizado no quiere decir que no funciona, pero este paso es necesario para la optimización de recursos como lo son memoria, tiempo de acceso y duplicidad de información.

La primera forma normal es cuando en una relación los campos en cada registro contienen un solo valor tomado de sus dominios respectivos.

El segundo paso de la normalización es establecer las llaves y relacionarlas con los campos de datos. En la primera forma normalizada, el renglón entero de la tabla depende de todo los campos llave. La división de la primera tabla normalizada en una serie de tablas en las que cada campo sólo depende de la llave completa, se llama la segunda forma normalizada.

Una relación es o pertenece a la segunda forma normal si es de la primera forma normal y cada atributo no llave de la relación es total y funcionalmente dependiente de su llave principal.

El tercer paso en la normalización consiste en separar los campos de las segundas relaciones normales que, aunque dependan sólo de una llave, deben tener una existencia independiente en la base de datos.

Una relación de la tercera forma normal es si es de la segunda y ningún atributo no llave en la relación es funcionalmente dependiente de algún otro atributo no llave.

El objetivo del proceso de normalización es minimizar las anomalías de inserción, borrado y actualización, así como la reducción de almacenamiento. Sin embargo, para minimizar estas anomalías hay que pagar con el desempeño del sistema.

Un último paso en el diseño de datos es importantísimo para el desempeño del sistema y este es el proceso de desnormalización. Para este proceso es necesario considerar los tipos de acceso a la base y la información que será solicitada por los usuarios. Este proceso de desnormalización se conjuga con la definición de métodos de acceso para obtener el desempeño de la base de datos esperado, y estipulado como requisito del producto.

El proceso de normalización de datos debe realizarse para todo el diccionario de datos del sistema.

Para el caso del sistema y como ejemplo explicaré el proceso principal de normalización para el registro de fianzas RT.

El diccionario de datos antes descrito es la relación universal de los campos del sistema de aquí sacaremos los campos de datos que se utilizan en el registro. Al obtener esta relación es necesario para obtener la 1era forma normal (1FN), definir la llave primaria de la relación de fianzas_rt. Para lo cual es suficiente identificar la compañía de emisión, el número de fianza y su bis.

Considerando que la relación de fianzas_rt se encuentra en 1FN por no tener grupos repetitivos se aplica el mismo proceso a la relación movimientos_rt a fianzas_rt. De la dependencia entre las dos relaciones se concluye que la llave de movimientos_rt es la llave de fianzas, más la identificación de la operación y su fecha de emisión. El siguiente paso es buscar los grupos repetitivos en la relación no normalizada. Ya que existen grupos repetitivos al aplicar la 1FN se derivan 2 relaciones movimientos_rt a fianzas_rt que ya está en 1FN y primas a futuros que hay que normalizar.

A la relación de primas por futuros se le encuentra su llave primaria y se considera en 1FN.

Para la segunda forma normal después de tener las relaciones anteriores en 1FN, se analiza la llave de fianzas_rt y se aprecia que la compañía de emisión (cve_cia) depende de otros atributos que nos llevan a generar una nueva relación llamada compañías. Con esto la relación de fianzas_rt queda en 2FN .

Aplicando la regla de la tercera foema normal a la relación de fianzas, se derivan las nuevas relaciones de ramos y tipos de la fianza, así como el de monedas. Por lo tanto la relación base de fianzas queda en 3FN, por lo que se concluye el proceso de normalización.

El siguiente grupo de tablas 5.2 muestra la estructura de cada una de las relaciones después de haber pasado por las formas de normalización y finalmente por la desnormalización.

**Tabla: Fianzas en
reafianzamiento tomado
(fianzas_rt)**

K	cve_cia
K	num_fianza
K	bis_fianza
	referencia_cia
	cve_ramo
	cve_tipo
	cve_mone
	fec_ini_vig_fza
	fec_fin_vig_fza
	monto_fianza
	prima_fianza
	nom_fiado
	edo_fianza
	reserva_fianza
	reserva_ret_fza

**Tabla: Fianzas RT
en moneda extranjera
(fianzas_ext_rt)**

K	cve_cia
K	num_fianza
K	bis_fianza
	monto_fza_ext
	prima_fza_ext
	edo_fianza
	cve_mone
	reserva_fza_ext
	reserva_ret_fza

**Tabla: Movimientos de RT
en moneda extranjera
(movs_ext_rt)**

K	cve_cia
K	num_fianza
K	bis_fianza
K	cve_oper_mov
K	fec_emis_mov
	monto_mov_ext
	prima_mov_ext
	reserva_mov_ext
	fec_anul_mov
	codigo_oper

**Tabla: Movimientos de fianzas
en reafianzamiento tomado
(movimientos_rt)**

K	cve_cia
K	num_fianza
K	bis_fianza
K	cve_oper_mov
K	fec_emis_mov
	fec_ini_vig_mov
	fec_fin_vig_mov
	monto_mov
	prima_mov
	pri_fut_mov
	reserva_mov
	codigo_oper
	fec_anul_mov
	reserva_ret_mov
	iva_mov
	comis_mov
	iva_comis
	fec_reg_mov
	cve_ramo
	cve_tipo

**Tabla: Catálogo de monedas
(monedas)**

K	cve_mone
	nom_mone
	paridad_ajuste

Tabla: Catálogo de operaciones
(operaciones)

K	cve_oper
	nom_oper
	cve_poliza

Tabla: futuros creados por RT
(futuros)

K	cve_cia
K	num_fianza
K	bis_fianza
K	cve_oper_mov
K	fec_emis_mov
	fec_ini_vig_fut
	fec_fin_vig_fut
	prima_fut
	fec_liber_fut

Tabla: Catálogo de
compañías afianzadoras
(compañías)

K	cve_cia	
	nom_cia	
	contacto_cia	
	puesto_contacto_cia	
	direccion_cia	
	rfc_cia	
	porc_comis_r1	
	porc_comis_r2	
	porc_comis_r3	
	porc_comis_r4	
	porc_ret_reserva	

Tabla: catálogo de ramos-tipos
(ramos_tipos)

K	cve_ramo
K	cve_tipo
	desc_ramo_tipo

**Tabla: Catálogo de referencias
para la creación de
pólizas de contabilidad**
(cuentas_rt)

K	cve_aplicacion
K	consecutivo_cta
	cve_cta
	cve_scta1
	cve_scta2
	cve_scta3
	cve_div
	cve_zona
	cve_suc
	rango_concepto
	cve_concepto
	rango_colectiva
	cve_colectiva
	concepto
	signo
	comentarios

**Tabla: Pólizas para contabilización
de producción de RT
(pólizas_rt)**

K	cve_cta
K	cve_scta1
K	cve_scta2
K	cve_scta3
K	cve_div
K	cve_zona
K	cve_suc
K	rango_concepto
K	cve_concepto
K	rango_colectiva
K	cve_colectiva
	importe_debe
	importe_haber
	signo
	cve_poliza

**Tabla: Catálogo de
usuarios del sistema
(usuarios)**

K	cve_usuario
	password
	nivel_usuario
	nom_usuario

**Tabla: Saldos de producción
por cobrar a las
compañías cedentes
(saldos_rt)**

K	cve_cia
K	referencia_cia
K	tipo_saldo
K	cve_oper_mov
K	fec_emis_mov
	num_fianza
	bis_fianza
	prima_neta
	comis_mov
	iva_mov
	iva_comis
	fec_reg_mov
	fec_pago
	fec_anul_mov
	total
	pagado
	saldo

**Tabla: Datos generales
del sistema RT
(sistema)**

K	cve_sistema
	porc_iva

5.3 Diseño estructural y de interfase

El objetivo del diseño estructural es crear una estructura de sistema modular de fácil entendimiento al usuario y definir las interfases que faciliten la interacción usuario-computadora.

La estructura de los menús del sistema, surge de la explotación al máximo de los diagramas de flujo de datos creados durante la etapa de análisis.

La interfase con el usuario se une a la arquitectura del sistema como dos de los elementos más trascendentes para que el usuario pueda entender tanto el objetivo del sistema como su manejo. Es por este motivo que durante el diseño se presentó como un estilo de interacción usuario-computadora de menú simple, aunque en realidad lo que se utiliza es de iconos, pero se presenta así es para facilitarle el entendimiento al usuario que ya tenía cierta experiencia con menús y no fuera muy confuso para él.

Los modelos de interfase usuario-computadora más utilizados son: la interfase de preguntas y órdenes, la interfase de menú simple, la interfase con ventanas y la interfase con iconos. Es importante señalar que se pueden utilizar combinaciones de éstas y que la implementación del sistema puede ser realizada independientemente del estilo de interfase seleccionado durante el diseño.

En las siguientes páginas se muestran los diagramas de arquitectura donde se muestra la estructura definida para el sistema y su prototipo.

Con este punto terminamos con la etapa del diseño y ya sólo nos queda por cristalizar el trabajo desarrollado durante las etapas de análisis y diseño tanto en la creación de la base de datos como el de la programación del sistema, puntos que se desarrollaran en el siguiente capítulo.

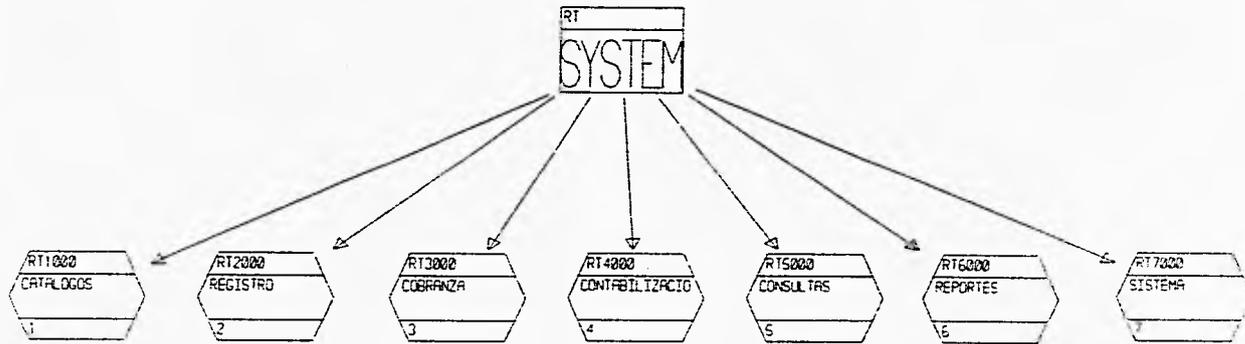


Figura 5.3 Menú principal

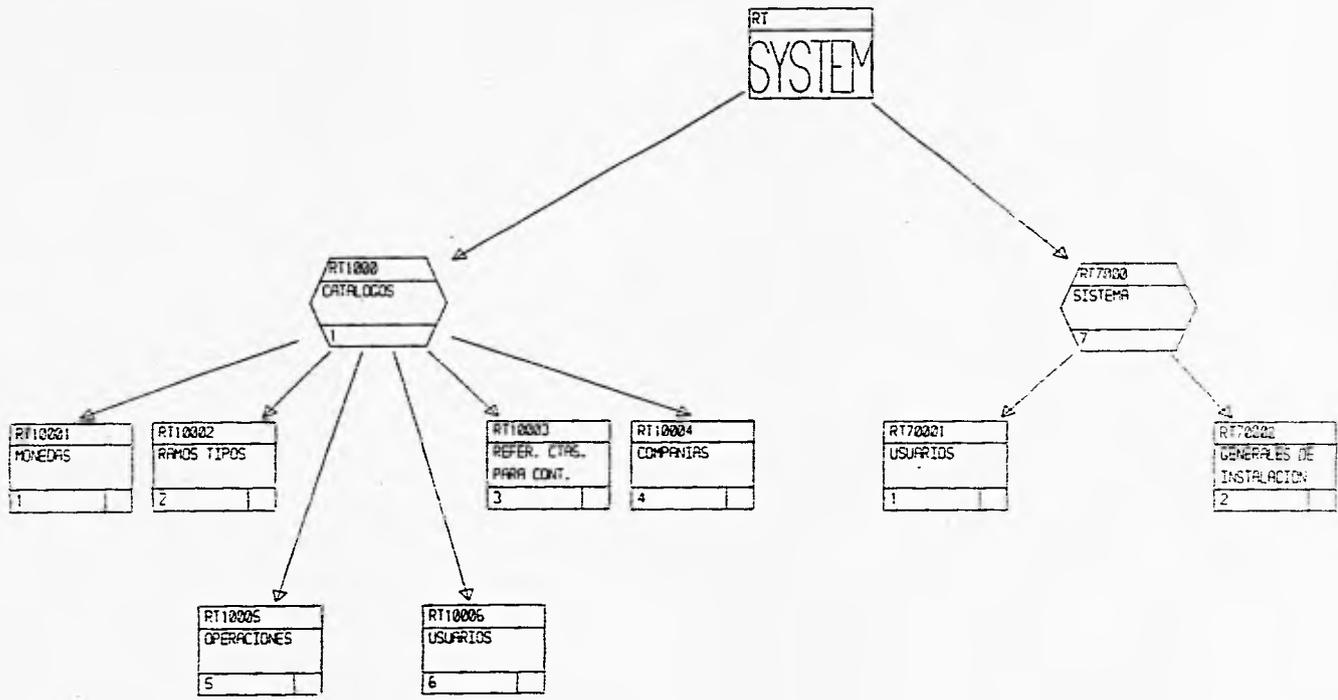


Figura 5.4 Menú de catálogos y sistema

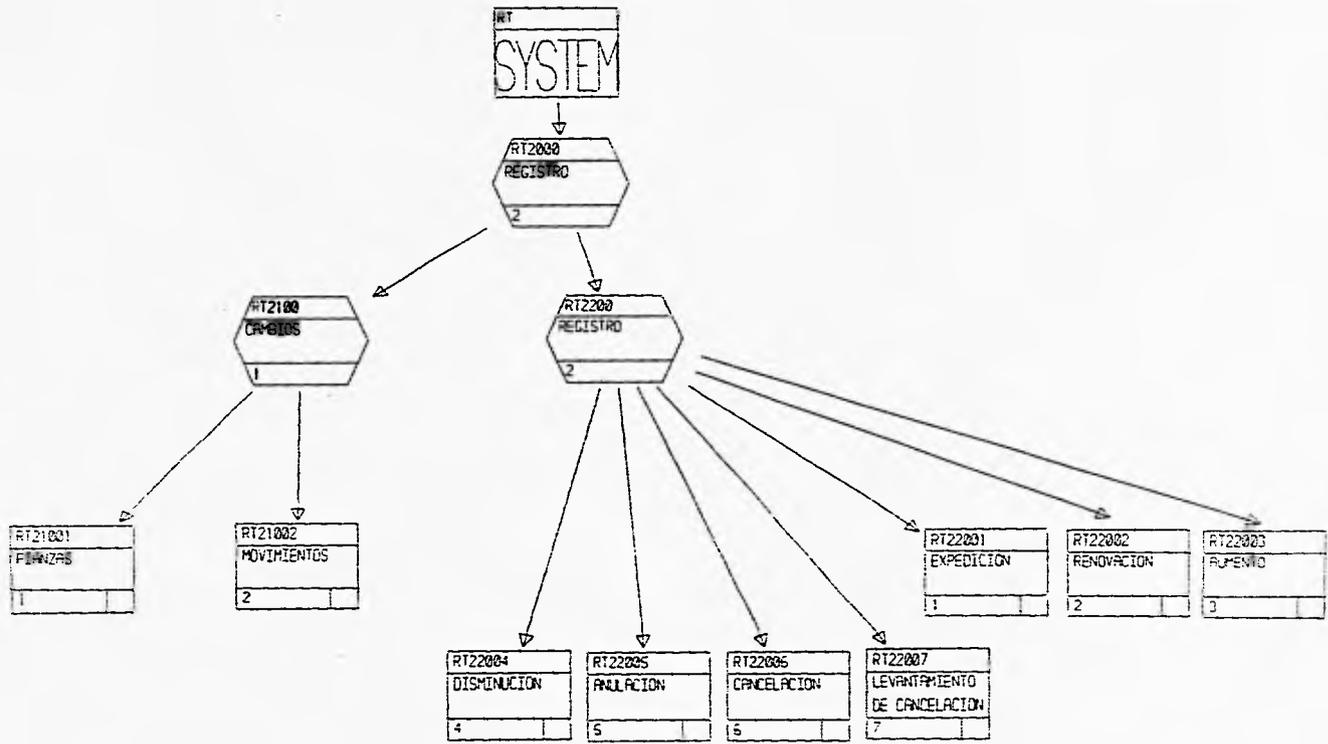


Figura 5.5 Menú de registro

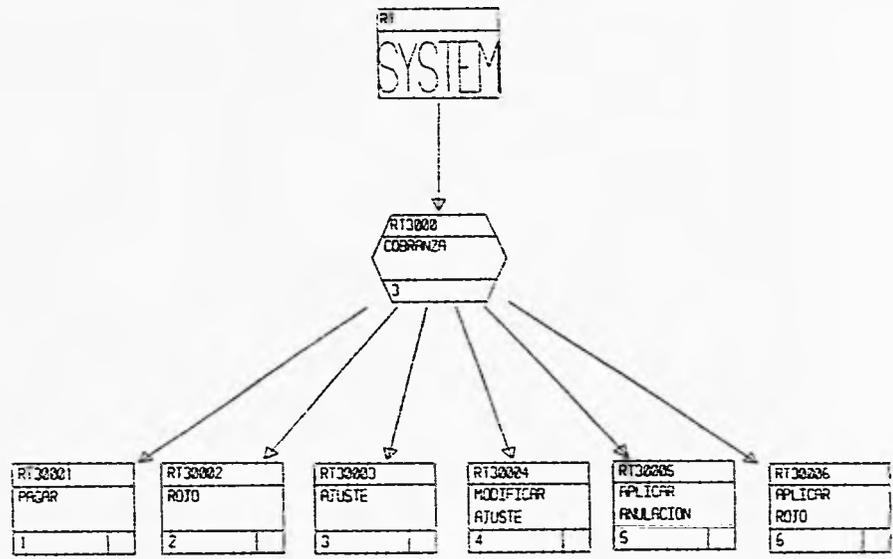


Figura 5.6 Menú de cobranza

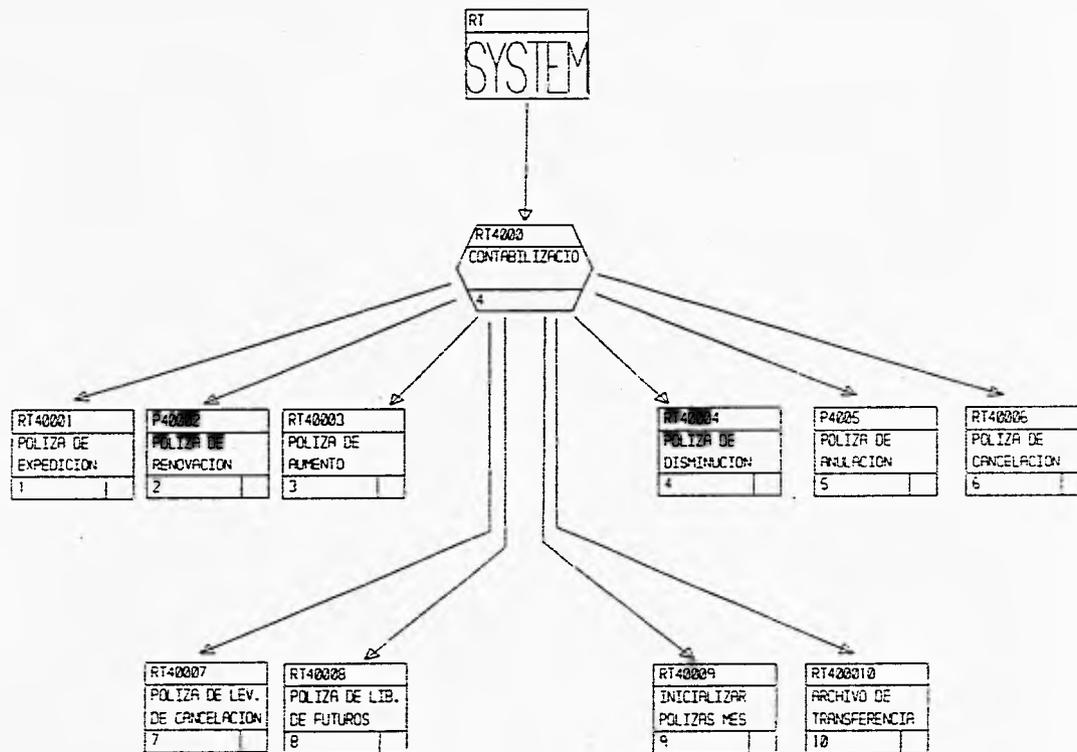


Figura 5.7 Menú de contabilización

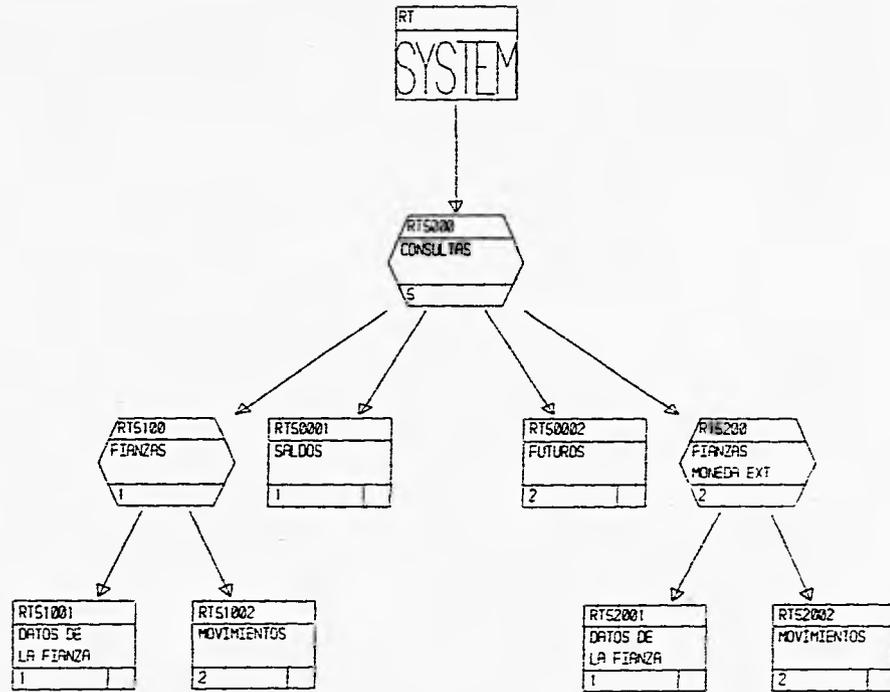


Figura 5.8 Menú de consultas

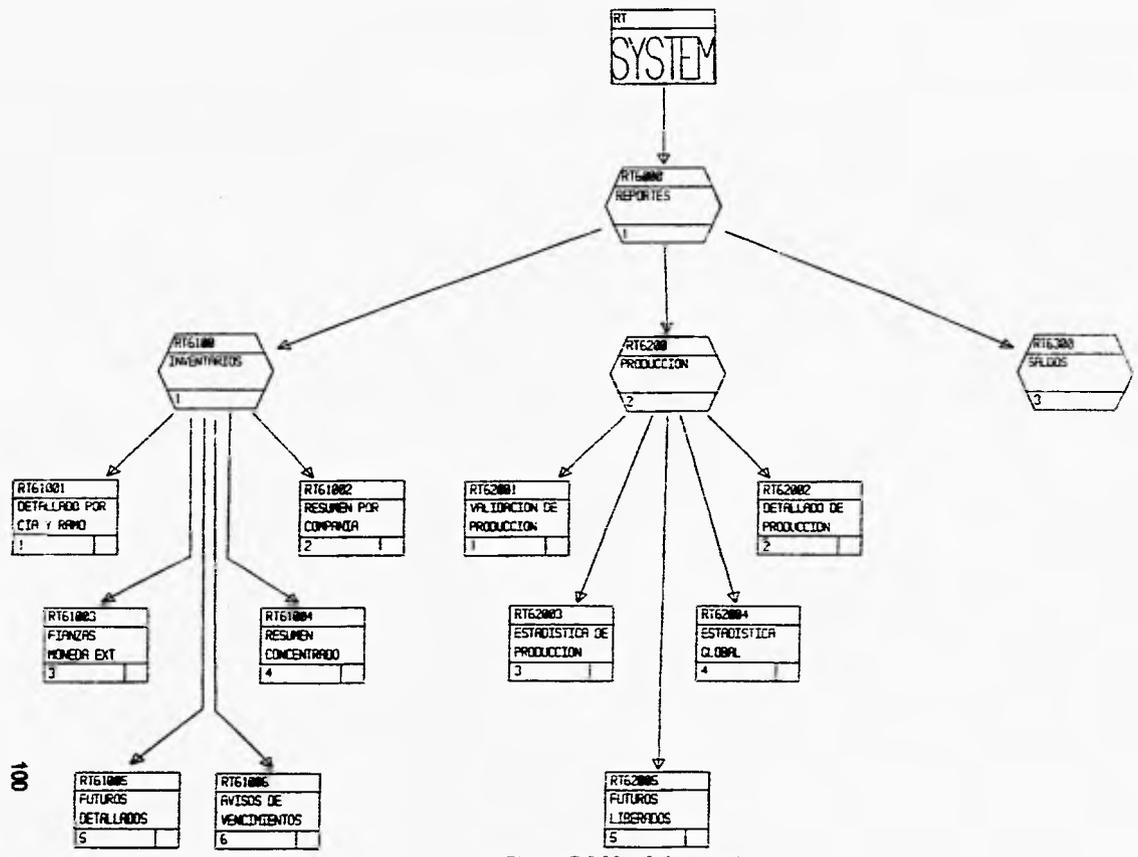


Figura 5.9 Menú de reportes

6 Desarrollo del sistema

Las etapas que se han presentado van dirigidas hacia un objetivo final: traducir las representaciones del software a una forma que pueda ser comprendida por la computadora, a esto se le llama codificación, y para ello es necesario de la construcción de una base de datos y una herramienta de desarrollo o lenguaje de programación. Las características del lenguaje de programación y el estilo de programación pueden afectar profundamente a la calidad y al mantenimiento del sistema, así como la construcción de la base de datos es importante para no tener redundancias en la información y que esté interrelacionada adecuadamente.

6.1 Construcción de la base de datos

El elemento más importante de cualquier sistema de información es precisamente la información o datos que maneja. Una forma de catalogar los sistemas es por la cantidad de datos (registros) que almacenan y procesan. Al ritmo que la tecnología de dispositivos electrónicos o sea de almacenamiento avanza, pronto será más barato guardar la información en archivos de computadora que en papel, cintas de audio o de video, no importando si esta información está en forma escrita, en imágenes o sonidos.

Una base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto con mínima redundancia; su finalidad es la de servir a un aplicación o más; los datos se almacenan de modo que resulten independientes de los programas que los usan, empleándose métodos bien determinados para incluir datos nuevos y para actualizar o recuperar los datos almacenados. Las bases de datos deben cumplir con ciertas características que son: redundancia mínima, independencia de datos, interconectividad, seguridad y accesibilidad.

Los métodos de almacenamiento han ido evolucionando al igual que las hardware y software, en un principio teníamos una simple organización secuencial de los datos en archivos completamente planos, donde la estructura física y lógica era igual, la programación era dependiente de estos archivos. Se evolucionó de manera que el acceso fue a nivel de campo o grupo, este acceso era por medio de llaves múltiples y la programación eran independientes de la organización física del archivo. La era en la que nos encontramos es donde se tiene un administrador de base de datos (DBMS), hay lenguajes específicos para manejo de la base de datos, se pueden tener bases de datos distribuidas y algunas actúan con la arquitectura cliente-servidor.

Los DBMS o sistemas administradores de bases de datos es un paquete de programas que corre en el sistema operativo como un largo y sofisticado programa de aplicación. La mayoría de los DBMS se pueden clasificar dentro de las tres siguientes categorías: relacional, red y jerárquica. La clasificación se basa en las estructuras lógicas que utiliza, y estas estructuras son las que el usuario percibe como base para su almacenamiento físico.

La construcción de la base de datos del sistema depende de las características propias del DBMS y del modelo de datos seleccionado para el sistema. En el proceso de diseño de datos contenido en la etapa del diseño, se definieron las tablas o estructuras de datos; es por esto y por el tipo de base de datos que se utilizará que veremos a más detalle el enfoque relacional de un DBMS.

6.1.1 Enfoque relacional

La ventaja principal del enfoque relacional sobre los red y jerárquico está en la simplicidad de su representación y en la flexibilidad para establecer relaciones de datos por medio de campos de conexión.

El enfoque relacional hace posible alcanzar mayor independencia de los datos usando campos de conexión en vez de apuntadores para entrelazar registros relacionados en diferentes archivos.

La arquitectura relacional consta de los siguientes componentes:

- El modelo relacional de datos está representado por una colección de relaciones almacenadas (tablas). Cada registro de tipo conceptual es un archivo almacenado distintamente.
- Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de uno o más escenarios o vistas para describir los datos requeridos por una aplicación.

Cada programa de aplicación está provisto de un buffer donde el DBMS puede depositar los datos momentáneamente para su procesamiento, antes de que el DBMS las actualice en la base de datos.

- El esquema de almacenamiento es peculiar ya que en el nivel interno, cada tabla se implanta como un archivo lógico almacenado. Para las recuperaciones se pueden establecer uno o más índices para acceder el archivo.
- El sublenguaje de datos es un lenguaje de manejo de datos para el sistema relacional. Existen principalmente dos sublenguajes de datos que son: el álgebra relacional y el cálculo relacional.

El álgebra relacional es un lenguaje de procedimientos de alto nivel que permite el uso de operadores para derivar la tabla deseada en diversos pasos desde las tablas base originales o de otras tablas intermedias.

El cálculo relacional es un potente lenguaje de consulta que permite al usuario la recuperación de datos mediante el establecimiento de condiciones de consulta sin necesidad de codificar en forma detallada los pasos de la recuperación.

El álgebra relacional aunque no es muy accesible proporciona los conceptos básicos para el desarrollo de lenguajes relacionales de manejo de datos. Los dos principales lenguajes relacionales, SQL (structured query language) y QBE (query by example), están implementados de manera que sea sencilla la recuperación de información.

Una de las características del SQL es que el lenguaje se puede usar no sólo como lenguaje de manipulación de datos (DML) para manejar los datos sino también como lenguaje de definición de estructuras (DDL) para definir tablas, vistas o los archivos almacenados en los niveles conceptual, externo e interno. El SQL es un lenguaje flexible cuyos comandos se pueden emitir ya sea interactivamente desde un editor o incluirse en el lenguaje del programa.

La mayoría de los DBMS relacionales constan de dos subsistemas principales:

- Sistema relacional de datos (Relational Data System)
- Sistema de búsqueda de almacenamiento (Research Storage System)

Como lo muestra la figura 6.1, el RDS es una interfase externa similar al sistema de control de la base de datos. Sus funciones principales son el análisis sintáctico y la optimización.

El sistema de búsqueda de almacenamiento (RSS) es una interfase interna que maneja los datos almacenados, índices, buffers, distribución del almacenamiento, etcétera.

Uno de los principales factores que afectan el funcionamiento de los programas, que interactúan con la base de datos, es la manera en la cual los datos se almacenan y se accesan. Un sistema de manejo de bases de datos generalizado utiliza algunos métodos de acceso del modelo interno junto con los métodos de acceso especializados disponibles a través del modelo externo. El diseñador de la base de datos tiene la flexibilidad de elegir uno o una combinación de varios métodos de acceso disponibles.

Al crecer el número de entradas/salidas físicas necesarias para recuperar los datos necesarios de un usuario, el desempeño del sistema decrece. Utilizando los métodos de acceso tanto del sistema operativo como de los modelos interno y externo, los datos de la base física se pueden recuperar y presentar al DBMS.

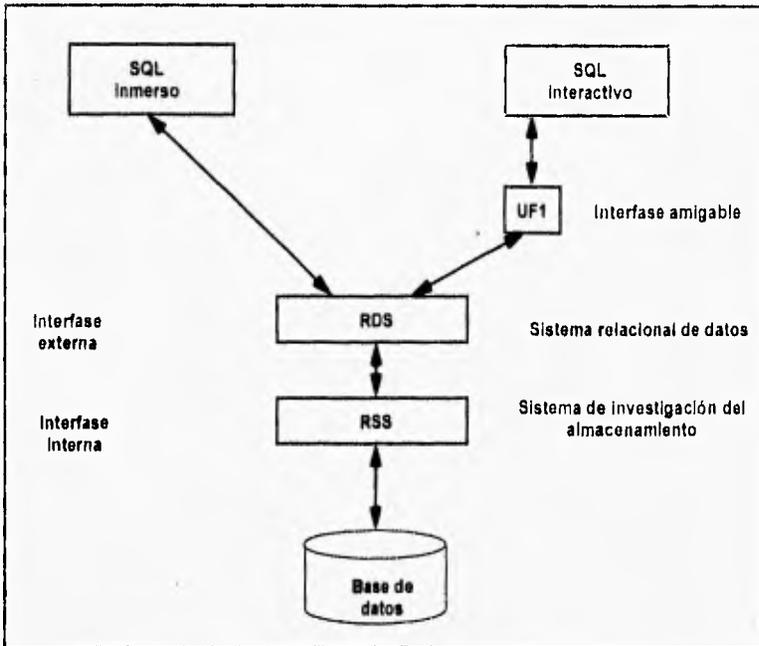


Figura 6.1 componentes de un sistema relacional

El diseñador de la base de datos debe considerar todos los métodos de acceso que el DBMS ofrece y determinar el óptimo para cada caso establecido en los programas de aplicación. En la mayoría de los DBMS el diseñador define índices como método principal de acceso a la información.

La base de datos que se utilizó para el desarrollo de este trabajo fue WATCOM SQL. WATCOM SQL es un sistema de base de datos completamente relacional que corre bajo windows en una PC. WATCOM SQL fue creada bajo el

estándar ANSI SQL89 pero tiene además características definidas en IBM DB2 y las especificaciones SAA y ANSI SQL92.

La implementación de la base de datos se hace a través de PowerBuilder. Se necesita dar un nombre a cada una de las tablas y columnas que contienen cada tabla.

Durante el proceso de diseño, se decidió que tablas se necesitan y que datos se deben almacenar en cada tabla. Ahora se necesita seleccionar un nombre para cada columna de datos a almacenar, especificando el tipo de dato y el tamaño.

Watcom acepta cualquier combinación de letras, números o símbolos para el nombre de una columna. Los tipos de datos y tamaños aceptados son los siguientes:

- enteros (int, integer, smallint)
- decimales (decimal, numeric)
- punto flotante (float, double, real)
- carácter (char, varchar, long varchar)
- binarios (binary, long binary)
- fecha/hora (date, time and timestamp)

6.2 Programación

El paso de codificación traduce una representación del software, dada por un diseño detallado, a una realización en un lenguaje de programación. Los lenguajes de programación son un vehículo de comunicación entre los humanos y las computadoras. El proceso de traducción continúa cuando un compilador acepta el código fuente como entrada y produce como salida un código objeto dependiente de la máquina. Más tarde, la salida del compilador es traducida a código máquina.

El paso inicial de traducción del diseño al lenguaje de programación es un punto fundamental. La interpretación equivocada de las especificaciones del diseño detallado puede conducir a un código fuente erróneo. La complejidad o las

restricciones de un lenguaje de programación pueden conducir a un código fuente muy complejo que resulte difícil de probar y de mantener.

Existen cientos de lenguajes de programación, que han sido utilizados en su momento, pero principalmente se jerarquizan en cuatro generaciones como lo muestra la figura 6.2 y analizamos brevemente a continuación.

La primera generación de lenguajes se remonta a los días en que se codificaba a nivel máquina. El lenguaje más usado dentro de esta generación era el lenguaje ensamblador, el problema con este lenguaje ensamblador, es que, existen igual número de lenguajes ensamblador como tipos de microprocesadores hay. Este tipo de lenguajes representan el menor nivel de abstracción con el que se puede representar un programa.

La segunda generación de lenguajes fue desarrollada a finales de los años 50 y principios de los 60 y ha servido como base para todos los lenguajes de programación. Esta generación se caracterizó por su amplio uso, la enorme cantidad de bibliotecas que se crearon y se siguen usando, así como la gran familiaridad y aceptación. FORTRAN, COBOL, ALGOL y BASIC son algunos ejemplos de lenguajes de segunda generación, y algunas empresas se siguen utilizando.

La tercera generación de lenguajes o también llamados lenguajes de programación estructurada están caracterizados por sus potentes posibilidades procedimentales y de estructuración de datos. Estos lenguajes se pueden dividir en tres amplias categorías: lenguajes de alto nivel de propósito general, lenguajes de alto nivel orientado a objetos, y lenguajes especializados.

Los lenguajes de alto nivel de propósito general han sido adoptados como lenguajes para un gran espectro de aplicaciones como para uso en el área de ciencias e ingeniería, comerciales y aplicaciones de sistemas de administración. Algunos ejemplos de estos lenguajes son PL/1, PASCAL, C, etc.

Los lenguajes orientados a objetos deben permitir la implementación tanto del análisis como del diseño orientado a objetos, como pueden ser la definición de clases, que permitan la herencia, la encapsulación y el paso de mensajes entre objetos. Los lenguajes orientados a objetos más importantes son C++, Smalltalk y Eiffel.

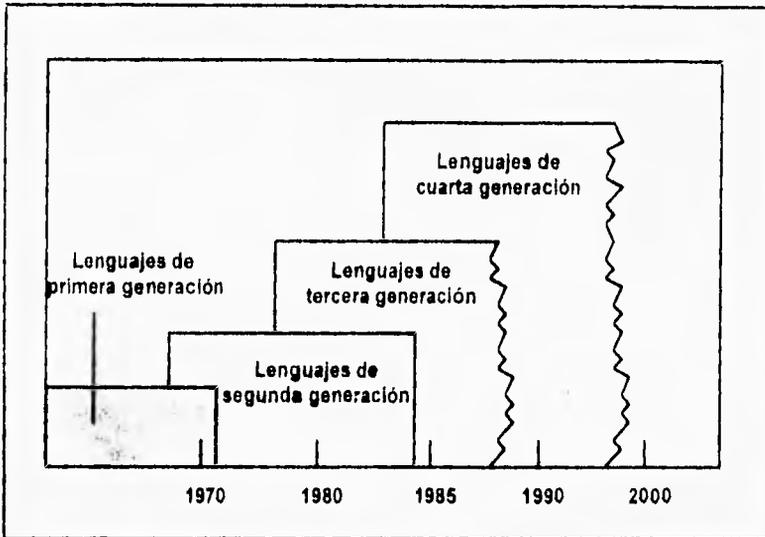


Figura 6.2 Generaciones de los lenguajes de programación

Los lenguajes especializados y de inteligencia artificial están caracterizados por su inusual formulación sintáctica que ha sido especialmente diseñada para una aplicación en especial. Entre los lenguajes que han encontrado aplicación se encuentran LISP, PROLOG, APL y FORTH.

Los lenguajes especializados tienen tantas ventajas como desventajas. Debido a que cada lenguaje especializado se ha diseñado para una aplicación específica, se puede facilitar la traducción de los requisitos del diseño a la implementación en código. Por otro lado, son apenas portables y son menos fáciles de mantener que los lenguajes de propósito general.

Los lenguajes de cuarta generación (4GL) han tratado de elevar el nivel abstracción en la programación y al igual que los lenguajes de inteligencia artificial, contienen una sintaxis distinta para la representación del control y la representación de las estructuras de datos.

Estos lenguajes combinan características procedimentales y no procedimentales. Es decir, el lenguaje permite al usuario especificar condiciones con sus correspondientes acciones, mientras que, al mismo tiempo, se pide al usuario que indique el resultado deseado, encontrando los detalles procedimentales mediante la aplicación de su conocimiento. Los lenguajes de cuarta generación se han dividido en tres grandes categorías:

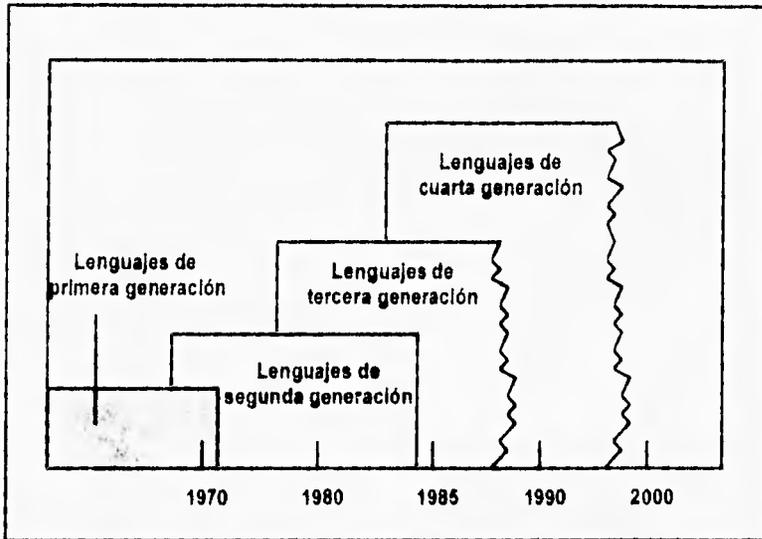


Figura 6.2 Generaciones de los lenguajes de programación

Los lenguajes especializados y de inteligencia artificial están caracterizados por su inusual formulación sintáctica que ha sido especialmente diseñada para una aplicación en especial. Entre los lenguajes que han encontrado aplicación se encuentran LISP, PROLOG, APL y FORTH.

Los lenguajes especializados tienen tantas ventajas como desventajas. Debido a que cada lenguaje especializado se ha diseñado para una aplicación específica, se puede facilitar la traducción de los requisitos del diseño a la implementación en código. Por otro lado, son apenas portables y son menos fáciles de mantener que los lenguajes de propósito general.

Los lenguajes de cuarta generación (4GL) han tratado de elevar el nivel de abstracción en la programación y al igual que los lenguajes de inteligencia artificial, contienen una sintaxis distinta para la representación del control y la representación de las estructuras de datos.

Estos lenguajes combinan características procedimentales y no procedimentales. Es decir, el lenguaje permite al usuario especificar condiciones con sus correspondientes acciones, mientras que, al mismo tiempo, se pide al usuario que indique el resultado deseado, encontrando los detalles procedimentales mediante la aplicación de su conocimiento. Los lenguajes de cuarta generación se han dividido en tres grandes categorías:

- **Lenguajes de petición.**- hasta ahora, la gran mayoría de los lenguajes de cuarta generación se han desarrollado para ser usados conjuntamente con aplicaciones de bases de datos. Tales lenguajes de petición permiten al usuario manipular de forma sofisticada y efectiva la información contenida en la base de datos previamente creada.
- **Generadores de programas.**- son otra clase de 4GL con más sofisticación. Más que utilizar una base de datos predefinida, un generador de programas permite al usuario crear programas en un lenguaje de tercera generación usando un menor número de sentencias. Estos lenguajes de programación de muy alto nivel hacen gran uso de la abstracción de datos y procedimientos. La gran mayoría de los generadores de programas se centran exclusivamente en aplicaciones de sistemas de información administrativos o de negocios.
- **Otros 4GL .-** Los lenguajes de soporte a la toma de decisiones permiten que los no programadores lleven a cabo una gran variedad de análisis, que van desde los simples modelos de hojas de cálculo bidimensionales hasta modelos estadísticos y de investigación de operaciones. Los lenguajes de prototipos se han desarrollado para apoyar el uso de prototipos facilitando la creación de interfaces de usuario y de diálogos, además de proporcionar medios para la modelización de datos.

La herramienta de cuarta generación con la que se contó para el desarrollo del sistema de reafianzamiento tomado es una combinación de los tipos de lenguajes de cuarta generación antes mencionados.

Powerbuilder es una herramienta gráfica para el desarrollo de aplicaciones en una computadora personal sobre ambiente cliente-servidor o standalone, dependiendo de la base de datos que se utilice. Powerbuilder permite crear aplicaciones con una base de datos sobre el ambiente de Microsoft-Windows, esto quiere decir que estas aplicaciones tienen una vista similar y su uso es igual al del sistema de Windows, por ejemplo la forma de abrir un menú, seleccionar un ícono, mover algún objeto, y todo por medio del ratón (mouse).

En Powerbuilder las entradas de una pantalla o los reportes son normalmente objetos Datawindow, que accesan y recuperan datos de la base

datos o alguna otra fuente de información y la despliega en pantalla. El usuario puede añadir, modificar o eliminar datos desde la Datawindow y entonces se actualiza la base de datos con los cambios.

Una aplicación de Powerbuilder es una aplicación de windows que normalmente contiene una o muchas ventanas con las que se comunica con el usuario. Para comunicarse con el usuario, la ventana puede contener controles ya definidos (como botones de comando, botones tipo radio, lista de opciones, textos, campos de captura, etc.) y además se pueden construir otros.

El ambiente del 4GL (PowerBuilder) queda definido por la integración de los siguientes módulos o herramientas:

Diseñador de Aplicaciones (Application Painter): sirve para construir objetos de aplicación. Los objetos de aplicación son el punto de entrada a Microsoft-Windows. Es el nivel más alto del sistema y cuando se corre este, el programa del nivel de aplicación, inicializa todas las actividades del sistema.

El primer paso para construir un sistema en PowerBuilder, es crear un objeto de aplicación, en este módulo:

- Se asigna el nombre del sistema y su icono.
- Se establecen los valores por omisión del color del texto, tamaño, estilos y fuentes para el sistema.
- Especifica las librerías que se pueden usar.
- Se crea el archivo ejecutable al finalizar el sistema.

Diseñador de la Base de Datos (Database Painter): este módulo sirve para definir y mantener tablas, índices, vistas y estructuras de transferencia de datos. Dentro de la construcción de tablas se pueden definir las características del campo como el tipo de dato, longitud, formato, presentación, máscara de validación, texto de ayuda, valores por omisión, valores válidos, etcétera.

El módulo permite definir tablas de la base de datos, las cuales quedan inmersas en la organización de la base de datos. Además se pueden recuperar y manipular datos salvándolos directamente a la base de datos.

Desde este módulo se puede controlar el acceso a bases de datos definidas en el DBMS y también acceder información de otras fuentes como pueden ser hojas de cálculo o archivos tipo texto, así como escribir o dibujar sentencias de SQL a la base que se está utilizando.

Diseñador de Ventanas de Datos (Datawindow Painter): en esta herramienta se definen las características tanto de las pantallas de captura o desplegado de información como de los reportes que están dentro de las ventanas. De forma gráfica se puede definir la fuente de los datos que se quieren, por medio de código de SQL, ya que están definidos los campos, éstos pueden ser modificados en su tamaño, fuente, color, etc.

Desde este módulo PowerBuilder genera el código de SQL de las selecciones que se hicieron desde el diseñador. Se pueden escribir programas para cada uno de los eventos de la ventana.

Módulo de Seguimiento (Debug): cuando se compila el programa, es normal que el compilador detecte errores y por lo tanto el programa no puede ser compilado hasta que sean arreglados esos errores. Existen errores que nos hacen caer en loops infinitos, que de ninguna manera podemos pararlos. Para resolver estos problemas el Debug nos ayuda a encontrar estos errores.

El Debug permite la suspensión del programa en un punto antes seleccionado y así poder revisar el contenido de las variables que están en uso.

Diseñador de Funciones (Function Painter): El lenguaje de programación de PowerBuilder tiene por omisión más de 300 funciones programadas, pero hay ocasiones que un sistema necesita que se codifique un proceso varias veces y en diferentes lugares. Cuando esto sucede se pueden crear o definir funciones por medio del diseñador de funciones.

Una función definida por el usuario puede estar compuesta por una colección de sentencias, comandos y funciones de uso común, tanto de PowerBuilder como de SQL; los argumentos pueden ser variables de todo tipo, objetos de una ventana de datos, estructuras y arreglos de datos.

Este diseñador permite salvar toda o parte de la función en un archivo tipo texto en MS-DOS con la extensión .FUN, para después poder se utilizada en otros sistemas.

Diseñador de Menús (Menu Painter): Los menús son listas de comandos u opciones que el usuario puede seleccionar en la ventana de uso. El contenido de los menús usualmente provee al usuario con otras acciones como son copiar, abrir, guardar como, etc. o también son usados como otra manera de realizar acciones que se pueden hacer con botones o iconos dentro de la ventana.

El diseñador de menús permite desplegar los menús como menús de barra, o una lista hacia abajo o en cascada; también permite definir teclas de aceleramiento y teclas de trabajo rápido.

Existen dos tipos de menús: tipo ventana y de arriba hacia abajo (Popup). Los menús de ventana son desplegados siempre con la ventana asociada, en cambio, los menús Popup necesitan un evento específico para ser desplegados.

Al salvar un menú este guarda el código compilado dentro de la librería del sistema.

Diseñador de Ventanas (Window Painter): las ventanas son la principal interfase entre el usuario y los sistemas creados por PowerBuilder. Las ventanas pueden ser de captura, desplegar información, o responder a una acción del mouse o el teclado.

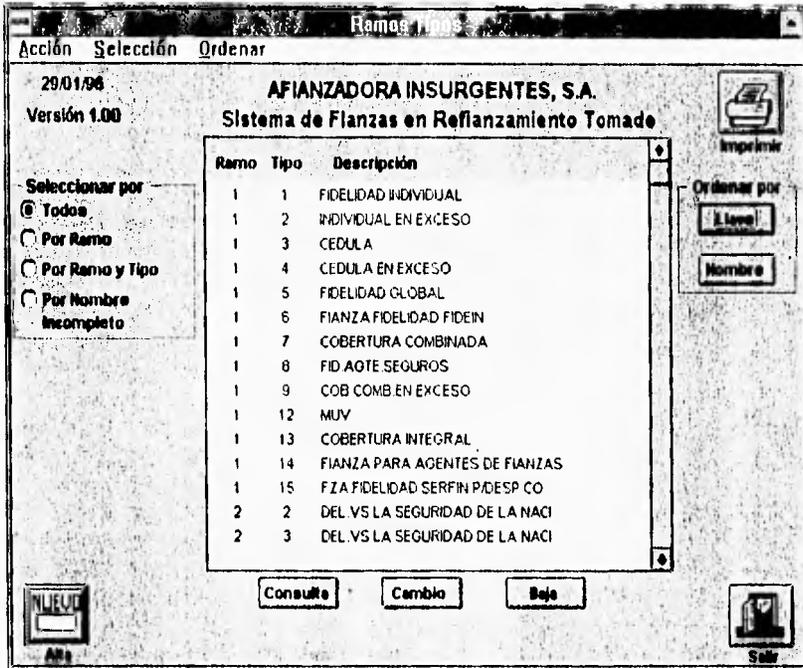
El diseñador de ventanas permite definir la apariencia, el estilo, el tipo, el tamaño y la posición de las mismas.

A las ventanas se les puede agregar controles y objetos gráficos. Los controles permiten a la ventana la comunicación con el usuario de manera que se pueda obtener algún dato del usuario o desplegar alguna información. Los objetos gráficos sirven para estilizar el diseño de la ventana. Después que se pone un control en la ventana este puede ser modificado en su estilo, tamaño y posición.

Una forma de crear una aplicación sencilla sería la siguiente:

1. - Crear una base de datos, siempre y cuando no exista una, si ya la hay sólo hay que conectarse a ella.
2. - Construir un objeto de aplicación. Construir una aplicación es sumamente sencillo con sólo apretar un botón y dar un nombre se crea; lo que nosotros podremos ver desde el sistema operativo es una librería con el mismo nombre de la aplicación y una extensión .PBL . La aplicación es el objeto más general que tenemos y se puede decir que están solo el nombre del programa.
3. - Crear una ventana. En esta ventana se pueden agregar controles para poder comunicarse con el usuario, con sólo apretar un control o seleccionar algún botón el usuario puede empezar una actividad, como una consulta, una alta o un ordenamiento.
4. - Crear una ventana de datos (Data window) para presentar, manejar y actualizar datos de la base de datos.
5. - Asociar la ventana de datos en la ventana anteriormente creada, y construir scripts (pequeños programas) para lograr conectarse a la base de datos y así manipular la información.
6. - Construir un menú para la ventana, de forma que las opciones de íconos y botones que se tengan, también se puedan usar por medio de menús y así no perder la estandarización de MS-Windows.
7. - Asociar el menú con la ventana.
8. - Crear un archivo ejecutable (.EXE), de manera que se pueda correr nuestra aplicación desde windows, sin necesidad de PowerBuilder, un centinela o un run-time.

A continuación se presenta el trabajo de programación que implica el catálogo de ramos tipos.



Esta pantalla esta compuesta por varios elementos, la parte que esta enmarcada y contiene a todo es una ventana (window), el objeto que contiene la información es una ventana de datos (data window), existen una serie de cuatro botones de radio (radio button) en la parte izquierda, que es para dar una selección específica. En la parte derecha de la pantalla hay unos botones para el ordenamiento deseado. Los botones de la parte de abajo son para generar las acciones señaladas. A continuación se muestran los programas (scripts) para cada uno de los elementos que conforman la ventana, así como lo que generan.

Empecemos por la parte de la ventana. Como ya se dijo la ventana es todo lo que engloba a los elementos y el siguiente programa que se encuentra en el evento de abrir (open) nos muestra la habilitación y deshabilitación tanto de botones, como de los menús y ventanas de datos, así como el inicio de transacción hacia la base de datos de las dos ventanas de datos (data window). También tiene el manejo de niveles según el usuario.

```

st_fecha.text=string(today(),"dd/mm/yy")
st_version.text=version
dw_tabular.settransobject(SQLCA)
dw_desplegar.settransobject(SQLCA)
dw_tabular.retrieve()

if nivel_u=1 then
  cb_cambio.enabled=false
  cb_baja.enabled=false
  pb_1.enabled=false
  m_ramos_tipos.m_accin.m_cambio.enabled=false
  m_ramos_tipos.m_accin.m_baja.enabled=false
  m_ramos_tipos.m_accin.m_alta.enabled=false
end if
if nivel_u=2 then
  cb_baja.enabled=false
  m_ramos_tipos.m_accin.m_baja.enabled=false
end if
pb_1.picturename=path + 'nuevo.bmp'
pb_1.disabledname=path + 'nuevo.bmp'

pb_2.picturename=path + 'salir.bmp'
pb_2.disabledname=path + 'salir.bmp'

pb_imprimir.picturename=path + 'impreso3.bmp'
pb_imprimir.disabledname=path + 'impreso3.bmp'

```

Existen una serie de menús que hacen exactamente la misma acción que los botones es por esto que sólo se explicarán las acciones de los botones.

El botón de impresión es un botón de imagen (picture button) y su función es el de mandar a Impresión lo que se encuentre en la ventana de datos (data window), su programa es el siguiente:

```

int job
job=printopen()
printdatawindow(job,dw_tabular)
printclose(job)

```

Lo que se realiza en este programa es un principio se declara una variable, después es asignada como un trabajo para impresión, se manda a impresión los datos que en ese momento se despliegan y por último se cierra el trabajo antes abierto.

Del lado izquierdo de la pantalla existen una serie de botones que sirven, para la selección y desplegado de datos en la ventana de datos, a continuación veremos los programas que se definieron para cada una de las opciones.

El siguiente programa fue realizado para el botón tipo radio que dice todos. Este es el programa para el evento de clicked.

```
m_ramos_tipos.m_seleccin.m_todos.check()
m_ramos_tipos.m_seleccin.m_llave.checked=false
m_ramos_tipos.m_seleccin.m_nombre.checked=false
m_ramos_tipos.m_seleccin.m_tipo.checked=false
rb_todos.checked=true
sle_texto_buscar.text=""
sle_texto_buscar.enabled=false
sle_texto_buscar.visible=false
dw_tabular.doItaobject="dw_ramos_tipos"
dw_tabular.settransobject(SQLCA)
dw_tabular.retrieve()
```

La primera parte del programa define las habilitaciones de los menús, la segunda parte deshabilita el campo de captura que se utiliza en las otras opciones y por último realiza la transacción con la base de datos y trae los datos a la ventana de datos.

El siguiente programa fue realizado para el botón tipo radio que dice por ramo. Este es el programa para el evento de clicked.

```
m_ramos_tipos.m_seleccin.m_llave.check()
m_ramos_tipos.m_seleccin.m_todos.checked=false
m_ramos_tipos.m_seleccin.m_nombre.checked=false
m_ramos_tipos.m_seleccin.m_tipo.checked=false
rb_seleccion_llave.checked=true
sle_texto_buscar.enabled=true
sle_texto_buscar.visible=true
sle_texto_buscar.text=""
seleccion=0
sle_texto_buscar.setfocus()
```

La primera parte del programa define las habilitaciones de los menús, la segunda parte habilita el campo de captura donde se leera el ramo a consultar y por último manda el cursor al campo de captura donde se realiza la transacción con la base de datos y trae los datos a la ventana de datos.

El siguiente programa fue realizado para el botón tipo radio que dice Ramo-Tipo. Este es el programa para el evento de clicked.

```
m_ramos_tipos.m_seleccin.m_tipo.check()
m_ramos_tipos.m_seleccin.m_todos.checked=false
m_ramos_tipos.m_seleccin.m_nombre.checked=false
m_ramos_tipos.m_seleccin.m_llave.checked=false
rb_seleccion_llave2.checked=true
sle_texto_buscar.enabled=true
sle_texto_buscar.visible=true
sle_texto_buscar.text=""
seleccion=1
sle_texto_buscar.setFocus()
```

La primera parte del programa define las habilitaciones de los menús, la segunda parte habilita el campo de captura donde se leera el ramo y el tipo a consultar y por último manda el cursor al campo de captura donde se realiza la transacción con la base de datos y trae los datos a la ventana de datos.

El siguiente programa fue realizado para el botón tipo radio que dice Nombre incompleto. Este es el programa para el evento de clicked.

```
m_ramos_tipos.m_seleccin.m_nombre.check()
m_ramos_tipos.m_seleccin.m_tipo.checked=false
m_ramos_tipos.m_seleccin.m_llave.checked=false
m_ramos_tipos.m_seleccin.m_todos.checked=false
rb_nom_incomp.checked=true
sle_texto_buscar.enabled=true
sle_texto_buscar.visible=true
sle_texto_buscar.text=""
seleccion=2
sle_texto_buscar.setFocus()
```

La primera parte del programa define las habilitaciones de los menús, la segunda parte habilita el campo de captura donde se leera la cadena de caracteres a consultar y por último manda el cursor al campo de captura donde se realiza la transacción con la base de datos y trae los datos a la ventana de datos.

El siguiente grupo de botones a examinar son los que se encuentran del lado derecho de la pantalla, y sirven para ordenar los datos de la ventana de datos. Como se vio en el grupo anterior de botones los programas son muy similares es por ello que sólo explicaremos uno de ellos.

El siguiente programa fue realizado para el botón de ordenamiento que dice por Llave. Este es el programa para el evento de clicked.

```
m_ramos_tipos.m_ordenar.m_llave1.check()
m_ramos_tipos.m_ordenar.m_nombre1.checked=false
dw_tabular.setsort("1 A.2A")
dw_tabular.sort()
```

Lo único que hace este pequeño programa es el definir un ordenamiento (sort) según los campos indicados y si es ascendente o descendente.

El programa que continúa es el de la ventana de datos (data window) y el evento programado es el clicked en su área de acción.

```
// OBTIENE EL RENGLON DESEADO
renglon=dw_tabular.getclickedrow()
//SI EL NUMERO ES VALIDO, OBTIENE LA CLAVE Y LO MARCA
if renglon>0 then
    clave1=dw_tabular.getitemnumber(renglon,1)
    clave2=dw_tabular.getitemnumber(renglon,2)
    actualizacion=true
    dw_tabular.setredraw(false)
    dw_tabular.selectrow(0,false)
    dw_tabular.selectrow(renglon,true)
    dw_tabular.setredraw(true)
end if
// INICIALIZA LAS BANDERAS
baja=false
consulta=false
cambio=false
alta=false
//DESHABILITA LA DW PARA DESPLEGAR Y LOS BOTONES: SALIR Y ACTUALIZAR
dw_desplegar.enabled=false
dw_desplegar.visible=false
cb_salir_ventana.enabled=false
cb_salir_ventana.visible=false
cb_actualizar.enabled=false
cb_actualizar.visible=false
```

El programa siguiente es para el evento de doble clicked (double clicked)

```
//REALIZA EL SCRIPT DEL EVENTO DE CLICKED
dw_tabular.triggerevent("clicked")
cb_consulta.triggerevent("clicked")
```

El siguiente grupo de programas se dedica al desarrollo de los botones que se encuentran en la parte baja de la pantalla y que son botones que realizan alguna acción en específico.

Comenzaremos con el botón de alta.

```
alta=true
  habilita_w_ramos_tipos(false)
  //DESMARCA EL RENGLON SELECCIONADO
  dw_tabular.setredraw(false)
  dw_tabular.selectrow(0,false)
  dw_tabular.setredraw(true)
  //HABILITA LA DW DE DESPLEGAR Y LOS BOTONES: SALIR Y ACTUALIZAR
  dw_desplegar.reset()
  dw_desplegar.enabled=true
  dw_desplegar.visible=true
  dw_desplegar.reset()
  cb_actualizar.visible=true
  cb_actualizar.enabled=true
  cb_salir_ventana.enabled=true
  cb_salir_ventana.visible=true
  //HABILITA EL CAMPO CLAVE
  dw_desplegar.settaborder("cve_rama",1)
  dw_desplegar.settaborder("cve_tipo",2)
  //INSERTA EL REGISTRO NUEVO
  dw_desplegar.insertrow(0)
  dw_desplegar.setFocus()
```

La siguiente pantalla es la llamada dw_deslegar que es la pantalla en donde se capturarán los datos necesarios para el alta

The image shows a graphical user interface form. It contains three input fields: 'Ramo:' (a small text box), 'Tipo:' (a small text box), and 'Descripción:' (a larger text area). Below these fields are two buttons: 'Actualizar' on the left and 'Salir' on the right.

El programa es para el botón de actualizar de la pantalla dw_desplegar.

```

int numero
//PREGUNTA EL TIPO DE ACCION A REALIZAR
if baja then
    dw_desplegar.deleteRow(0) //BORRA
end if
commit;
if dw_desplegar.update()!=1 then //ACTUALIZA
    commit;
    cb_salir_ventana.triggerevent(clicked!)
else
    rollback;
end if

```

El programa es del botón de salir

```

//VUELVE A DEJAR TODOS LOS PARAMETROS IGUAL QUE AL COMIENZO
habilita_w_amos_tipos(true)
dw_desplegar.enabled=false
dw_desplegar.visible=false
cb_salir_ventana.enabled=false
cb_salir_ventana.visible=false
cb_actualizar.enabled=false
cb_actualizar.visible=false
if alta or baja or cambio then
    rb_todos.triggerevent(clicked!)
end if
baja=false
consulta=false
cambio=false
alta=false
actualizacion=false

```

Botón de consulta

```

if not actualizacion then
    return
end if
consulta=true
habilita_w_amos_tipos(false)

```

```

//DESMARCA EL RENGLON SELECCIONADO
dw_tabular.setredraw(false)
dw_tabular.selectrow(0,false)
dw_tabular.setredraw(true)
// PERMITE VER LA DW PARA DESPLEGAR Y EL BOTON DE SALIR.
// PERO NO HABILITA LA DW NI EL BOTON DE ACTUALIZAR
if dw_desplegar.release(clave1,clave2)<= 0 then
    messagebox("ERROR","El registro ha sido actualizado por otro
    usuario",stopsign!,ok!)
end if

```

The image shows a screenshot of a graphical user interface window. It contains three input fields: 'Ramo:' with the value '1', 'Tipos:' with the value '1', and 'Descripción:' with the text 'FIDELIDAD INDIVIDUAL'. A 'Salir' button is located in the bottom right corner of the window.

Botón de salir.

```

//DEJA LOS PARAMETROS IGUAL
habilita_w_ Ramos_tipos(true)
dw_desplegar.enabled=false
dw_desplegar.visible=false
cb_salir_ventana.enabled=false
cb_salir_ventana.visible=false
cb_actualizar.enabled=false
cb_actualizar.visible=false
if alta or baja or cambio then
    rb_todos.triggerevent(clicked!)
end if
baja=false
consulta=false
cambio=false
alta=false
actualizacion=false

```

Botón de cambio

```

if not actualizacion then
    return
end if
cambio=true
habilita_w_ Ramos_tipos(false)

```

```

//DESMARCA EL RENGLON SELECCIONADO
dw_tabular.setredraw(false)
dw_tabular.selectrow(0,false)
dw_tabular.setredraw(true)

//HABILITA LA DW DE DESPLEGAR Y LOS BOTONES: SALIR Y ACTUALIZAR

if dw_desplegar.retrieve(clave1,clave2)<= 0 then
  messagebox("ERROR","El registro ha sido actualizado por otro
  usuario",stopsign!,ok!)
end if

```

The image shows a screenshot of a graphical user interface window. It contains three input fields: 'Ramo:' with the value '2', 'Tipo:' with the value '6', and 'Descripción:' with the text 'ADU. TERIO'. Below these fields are two buttons: 'Actualizar' on the left and 'Salir' on the right. The window has a simple border and a light background.

Botón de actualizar.

```

int numero
if baja then
  dw_desplegar.deleterow(0)
end if
commit;
if dw_desplegar.update()=1 then
  commit;
  cb_salir_ventana.triggerevent(clicked!)
else
  rollback;
end if

```

Botón de salir

```

habilita_w_ramos_tipos(true)
dw_desplegar.enabled=false
dw_desplegar.visible=false
cb_salir_ventana.enabled=false
cb_salir_ventana.visible=false
cb_actualizar.enabled=false
cb_actualizar.visible=false
if alta or baja or cambio then
  rb_todos.triggerevent(clicked!)
end if
baja=false

```

```
consulta=false
cambio=false
alta=false
actualizacion=false
```

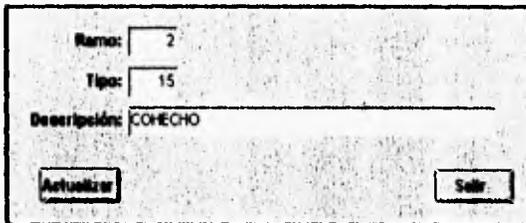
Botón de baja

```
if not actualizacion then
    return
end if
baja=true
habilita_w_ Ramos_tipos(false)

//DESMARCA EL RENGLON SELECCIONADO
dw_tabular.selectrow(false)
dw_tabular.selectrow(0,false)
dw_tabular.selectrow(true)

//HABILITA LA DW DE DESPLEGAR Y LOS BOTONES: SALIR Y ACTUALIZAR

if dw_desplegar.retrieve(clave1,clave2)<= 0 then
    messagebox("ERROR","El registro ha sido actualizado por otro
usuario",,stop,signl,ok!)
end if
```



The image shows a screenshot of a graphical user interface (GUI) window. It contains three input fields: 'Ramo:' with the value '2', 'Tipo:' with the value '15', and 'Descripción:' with the value 'COHECHO'. Below these fields are two buttons: 'Actualizar' on the left and 'Salir' on the right. The window has a standard title bar and a border.

Botón de actualizar

```
int numero
if baja then
    dw_desplegar.deleterow(0)
end if
commit;
if dw_desplegar.update{}=1 then
    commit;
    cb_salir_ventana.triggerevent(clicked!)
else
    rollback;
end if
```

Botón de salir

```
habilita_w_ramos_tipos(true)
dw_desplegar.enabled=false
dw_desplegar.visible=false
cb_salir_ventana.enabled=false
cb_salir_ventana.visible=false
cb_actualizar.enabled=false
cb_actualizar.visible=false
if alta or baja or cambio then
  rb_todos.triggerevent(clicked!)
end if
baja=false
consulta=false
cambio=false
alta=false
actualizacion=false
```

Y por último el botón salir general de la ventana.

```
close(parent)
```

6.3 Pruebas

La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los "costos" asociados a un fallo del mismo están motivando la creación de pruebas minuciosas y bien planificadas.

La prueba presenta una interesante anomalía para el ingeniero en computación. Durante las fases anteriores de definición y de desarrollo, el ingeniero intenta construir el software partiendo de un concepto abstracto y llegando a una implementación tangible. A continuación, llega la prueba. El ingeniero crea una serie de casos de prueba que intentan demoler el software ya construido. De hecho, la prueba es uno de los pasos de la ingeniería del software que se puede ver como destructivo en lugar de constructivo.

La gente que desarrolla software, es por naturaleza, constructiva. La prueba requiere que se descarten ideas preconcebidas sobre la "corrección" del

software que se acaba de desarrollar y se supere cualquier conflicto de intereses que aparezcan cuando se descubren errores.

La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

Una prueba tiene éxito si descubre un error no detectado hasta entonces.

El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Como ventaja secundaria, la prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la fiabilidad del software y, de alguna manera, indican la calidad del software como un todo.

La prueba no puede asegurar la ausencia de defectos; solo puede demostrar que existen defectos en el software.

Cualquier producto de ingeniería puede ser probado de dos formas:

1.- conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa.

2.- conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que "todas las piezas encajan"; o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

La primera forma es llamada prueba de la caja blanca y es un método de diseño de casos de prueba que usa la estructura de control del diseño procedural para derivar los casos de prueba. Mediante los métodos de prueba de la caja blanca, se pueden obtener cuatro casos de prueba: (1) que garanticen que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo, (2) ejercitan todas las decisiones lógicas en sus vertientes verdadera y falsa; (3) ejecuten todos los bucles en sus límites y con sus límites operacionales y (4) ejerciten las estructuras internas de datos para asegurar su validez.

Las siguientes ideas justifican el proceso de una prueba:

Los errores tienden a reproducirse en nuestro trabajo cuando diseñamos e implementamos funciones, condiciones o controles que se encuentran fuera de lo normal.

El flujo lógico de un programa a veces no es nada intuitivo, lo que significa que nuestras suposiciones intuitivas sobre el flujo de control y los datos nos pueden llevar a tener errores.

Los métodos de prueba de la caja negra se centran en los requisitos funcionales del software, es decir intenta encontrar errores de las siguientes categorías: (1) funciones incorrectas o ausentes; (2) errores de interfaz; (3) errores de estructuras de datos o en accesos a bases de datos externas; (4) errores de rendimiento y (5) errores de inicialización y terminación.

Los objetivos de las actividades de prueba son valorar y mejorar la calidad de los productos de software generados durante el desarrollo y modificación. Hay dos tipos de verificación la formal y la de ciclo de vida, esta última consiste en el proceso de determinar el grado en que los productos de trabajo de una fase dada del ciclo de desarrollo cumplen con las especificaciones establecidas durante las fases previas.

La verificación y validación implican la valoración de los productos de trabajo para determinar el apego a las especificaciones. Estas incluyen las especificaciones de requisitos, la documentación del diseño, diversos principios generales de estilo, estándares del lenguaje de instrumentación, del proyecto y organizacionales. La documentación del diseño debe verificarse con respecto a los requisitos y las convenciones notacionales; del código fuente debe revisarse su conformación con los requisitos, la documentación del diseño y las expectativas del usuario.

Las tres grandes clases de errores del software son las de requisitos, de diseño, y de implantación. Los errores de requisitos se provocan por una propuesta incorrecta de las necesidades del usuario, por falta de una especificación completa de los requisitos funcionales y de desempeño.

Los errores de diseño se introducen por fallas al traducir los requisitos en estructuras de solución correctas y completas, por inconsistencias dentro de las especificaciones del diseño y como entre las especificaciones de diseño y los requisitos.

Los errores de instrumentación son los cometidos al traducir las especificaciones del diseño en código fuente. Estos errores pueden producirse en

las declaraciones de datos, en las referencias a los datos, en la lógica de flujo, en expresiones, en interfaces y en operaciones de entrada y salida.

Las pruebas que se realizaron fueron tanto del tipo de la caja blanca, como las de tipo de la caja negra. Las pruebas fue un proceso muy exhaustivo debido a que psicológicamente el trabajo se sentía terminado y las pruebas realmente llevaron un tiempo que nunca se planeo durante el proyecto.

Las pruebas del tipo de la caja blanca se realizaron por el departamento de sistemas, o sea por la misma gente que influyó en su desarrollo, fueron pruebas que normalmente se le llaman pruebas de escritorio donde se realizaron a mano los recorridos de los programas.

Las pruebas de la llamada caja negra fueron más minuciosas, ya que se realizaron con el personal del departamento usuario, donde se operó con información real y se hicieron infinidad de pruebas antes de su liberación.

Aún durante la capacitación hubo un momento donde siguieron las pruebas, y se realizaron paralelamente las operaciones como normalmente se hacían, dando los mismos resultados el sistema quedo totalmente liberado.

No hay que perder de vista que la etapa de pruebas realmente nunca termina debido, a que el usuario constantemente esta usando o probando el sistema. A continuación analizaremos a más detalle el proceso de liberación del sistema, así como la capacitación a los usuarios.

7 Puesta en marcha

Después de las etapas de prueba que se le hicieron al sistema el paso siguiente es el de la puesta en marcha, que no es otra cosa que la fase de transición entre la creación del sistema a la operación normal de éste. Hasta este punto el producto de software se encuentra totalmente desarrollado y cumple con las especificaciones que le dieron origen, sólo falta insertarlo en el entorno operativo de la compañía e iniciar operaciones.

El proceso de puesta en marcha cuenta con dos importantes fases que son: la instalación o implantación del sistema y la capacitación del personal que lo utilizará; este proceso es tan importante como las etapas de desarrollo, ya que un inicio de operaciones deficiente se puede traducir como un rotundo fracaso inicial del software y pegar enormemente en la idea que hay sobre el sistema. Para la implementación es necesario tomar en cuenta los factores que intervienen en la operación de un sistema que son: el factor humano, el factor procedimental, y el factor hardware.

7.1 Implantación del sistema.

La implantación, al igual que el diseño, debe tener el objetivo principal de producir un software de comprensión, modificación y mantenibilidad sencillos. Es

por esto que hay que tomar en cuenta los diferentes actores que toman parte en la escena final. El factor humano es como se vio en los primeros capítulos la parte más importante de los sistemas ya que ellos serán los que lo manejaran y el objetivo principal de los sistemas es la ayuda directa en las tareas que se realizaban manualmente. Dentro del factor humano la capacitación de la operación del sistema es fundamental ya que si sistema no se sabe usar debidamente, pueden suceder dos cosas: (1) el sistema se esta subutilizando y/o (2) el sistema no arroja los resultados esperados.

El factor procedimental nos sirve para determinar o implementar procedimientos acordes con la operación del sistema, por ejemplo si se requiere algún cambio, o es necesario darle mantenimiento a alguna parte del sistema, este punto debe definir los estándares y procedimientos de manera que no afecte al uso normal del sistema, esto dará mayor seguridad al usuario ya que vera protegida sus información.

El factor de hardware es primordial, ya que se deben cumplir ciertas especificaciones para que el sistema pueda ser operado con facilidad, ya que en una máquina con capacidad mínima el sistema puede desde no correr a correr verdaderamente lento y esto le ocasionara un disgusto al usuario al grado de rehusarse a usarlo. Otros puntos importantes para el hardware es la red utilizada, el cableado, la comunicación y la máquina en donde será instalada la base de datos.

En particular con el sistema desarrollado la capacitación se explicará en el siguiente punto, el factor procedimental no altero los estándares ya usados en la afianzadora, donde se tienen hojas impresas en las cuales el usuario pide las modificaciones a través del nombre de la pantalla que se quiere modificar y un pequeño texto explicando por que se quiere la modificación. Después de este punto el gerente del área firma el requerimiento y es enviado directamente al área de sistemas. Para el hardware no existió problema alguno, ya que se instalo en clientes tipo PC 80486 - 33 Mhz y el servidor es una máquina VAX-3100, y se conectaron a través de una red tipo Ethernet de Digital llamada Pathworks, el cableado es entre pisos a través de cable coaxial y dentro de los pisos con par trenzado.

7.2 Capacitación del sistema.

Para el éxito de un sistema no debe dejarse a un lado la capacitación para la operación del sistema, ya que aún los sistemas bien diseñados y técnicamente completos pueden fracasar debido a un mala operación, por lo tanto, la calidad de la capacitación del personal involucrado ayuda o dificulta el éxito final.

La capacitación del usuario debe incluir en el caso que no tenga los conocimientos, desde la operación del equipo usado (microcomputadora e impresora) hasta un manejo bueno de Windows.

La mayor parte de la capacitación de los usuarios radica en la operación del sistema mismo. La capacitación en la codificación de datos destaca los métodos que han de seguirse para captación de datos, a partir de transacciones, o en la preparación de datos necesarios para las actividades de apoyo en toma de decisiones.

Las actividades a las que se deben prestar más atención dentro de la capacitación es la de recepción de datos, la edición de datos y la formulación de consultas y reportes, y la mayor parte del tiempo se dedica a estos puntos.

Como ya se dijo antes existen dos aspectos dentro de la capacitación del usuario: (1) la familiaridad con el entorno de la computación, es decir el equipo y el sistema operativo y (2) la capacitación en el empleo de la aplicación, es decir el software que acepta los datos, los procesa y produce información. Hay que tomar en cuenta que la carencia en cualquier aspecto de la capacitación terminarán por hacer fracasar el uso del sistema y no existe sustituto para la operación práctica del sistema mientras se aprende a usarlo.

La estrategia de capacitación seguida incluye por una parte cursos departamentales de uso de equipo y manejo de Windows y por otra cursos enfocados a los diferentes tipos de usuarios del sistema que son: capturistas, ejecutivos de cuenta y funcionarios.

Después de impartirse el curso básico a todos los integrantes del departamento de reafianzamiento, éstos se agrupan de acuerdo a la relación que tienen con el sistema, es decir, en base a las actividades que desarrollan y el apoyo que el sistema les brinda. De manera general los puntos vistos en la capacitación de los diferentes tipos de usuario son los siguientes:

1.- Capturistas

- Manejo de la PC e impresora
- Métodos y procedimientos del ciclo de vida de la fianza
- Módulo de registro.
- Módulo de cobranza.
- Mantenimiento de catálogos.

2.- Ejecutivos de cuenta

- Manejo de la PC e impresora
- Métodos y procedimientos del ciclo de vida de la fianza.
- Módulo de contabilización.
- Consultas.
- Reportes.
- Módulo de sistema.

3.- Funcionarios

- Manejo de la PC e impresora
- Métodos y procedimientos del ciclo de vida de la fianza.
- Consultas.
- Reportes.

Esta capacitación es impartida durante la instalación del equipo y sistema, e incluye aspectos teóricos y prácticos, aunque la información sólo es como ejemplo.

Para iniciar operaciones con el sistema, se vuelve a dar la capacitación pero ahora con la información real de la operación diaria y así poder cubrir aspectos que sólo se pueden cubrir hasta este momento. El grupo de capacitación permanece en el departamento durante el tiempo necesario hasta asegurarse que el sistema es utilizado correctamente.

Con este punto se da fin al sistema para el registro y control de la información de reafianzamiento tomado nacional; no hay que olvidar que el éxito o el fracaso de un sistema de información está dado por el grado de utilización y aceptación que tenga este. Afortunadamente y para el momento en el que estoy escribiendo estas líneas el programa es utilizado en su totalidad y se ha visto que la nueva manera de trabajo fue aceptada y se está pensando en hacer la migración de todos los sistemas de la afianzadora con esta arquitectura.

Nuestro siguiente punto y tal vez el más importante de toda este trabajo de tesis son las conclusiones a las que se llegaron.

CONCLUSIONES

Tomando en cuenta las experiencias que se obtuvieron al poner en marcha el sistema para el registro y control de la información de reafianzamiento tomado nacional, podemos concluir que los objetivos propuestos se cumplieron satisfactoriamente, reflejándose principalmente en los siguientes puntos:

- Se logró mayor eficacia en el desempeño del trabajo por el personal del área encargada.
- La afianzadora tuvo un mejor control sobre el manejo de la información así como un alto grado de confiabilidad en la misma, además de un acceso rápido, ordenado y clasificado a ésta.
- El sistema al realizar las tareas que se hacían manualmente, beneficia en mayor disposición de tiempo para atender las nuevas solicitudes de reafianzamiento y para analizar detenidamente los reportes.
- Se logró tener una explotación de la información registrada, para la toma de decisiones a través de reportes y estadísticas que anteriormente no se realizaban por no tener tiempo disponible ya que su realización resultaba laboriosa.
- Se obtuvieron de manera ágil y precisa las pólizas mensuales de producción por este tipo de fianzas y su fácil integración al sistema contable de la afianzadora, trabajo que antes del sistema consumía demasiados recursos obtenerlo.

El éxito rotundo del producto incrementó la cultura de sistemas en sus colaboradores, fomentando de esta manera el uso extensivo e intensivo de computadoras y creando importantes expectativas de desarrollo a los módulos faltantes por sistematizar en la afianzadora.

Actualmente en el mercado aparecen nuevas herramientas para apoyar cada una de las etapas de la ingeniería de software y la utilización de estas herramientas modificará el desarrollo de sistemas de información y lograr mejorar la calidad del software.

Digno de mencionarse es el grado de conocimiento del negocio, de las herramientas y la metodología utilizada que proporcionó el haber realizado esta tesis, de manera que es la punta de flecha necesaria para poder enfrentarse al mundo del trabajo profesional.

Aunque en algunos aspectos la formación adquirida en la Facultad de Ingeniería de la U.N.A.M. es un poco lejana al trabajo profesional, la realización del presente trabajo me ha permitido apreciar y valorar las bases adquiridas en la Universidad y particularmente en la Facultad de Ingeniería ya que comprobé su efectividad tanto técnica como humana en las diversas etapas del desarrollo de esta tesis.

Debido a esto el presente trabajo no es sólo la culminación profesional de un alumno más, sino el éxito de todo un grupo de personas y de la institución misma al ver cumplidos los objetivos de nuestra Alma Mater en una persona más.

Gracias a todos los que hicieron posible esta tesis.

BIBLIOGRAFIA

- Computer Associates.
Application development for the 90's, 1993.
- Computer Associates.
Computing architecture for the 90's, 1992.
- Date, C.J.
Introducción a los sistemas de bases de datos, 1990.
- Davo, Ramón
Personal Computing México 1994-1996
- Evans, Ronald W.
Byte, 1994-1995.
- Facultad de Ingeniería, U.N.A.M.
Apuntes "**Diseño de bases de datos**", 1992.
- Fairley, Richard.
Ingeniería de software, 1988.
- Hawkins, John L.
Data Based Advisor, 1994-1995.
- Powersoft
Introduction to Powerbuilder, 1993.
- Powersoft
Powerbuilder Function Reference, 1993.
- Powersoft
Powerbuilder Getting Started, 1993.
- Powersoft
Powerbuilder ODBC Connect, 1993.

- Powersoft
Powerbuilder User's Guide, 1993.
- Powersoft
Watcom SQL, 1993.
- Powersoft
Powerscript Language, 1993.
- Pressman, Roger S.
Ingeniería de software, 1993.
- Senn, James A.
Análisis y diseño de sistemas de Información, 1988.
- Tsai, Alice Y.H.
Sistemas de bases de datos, 1990.
- Yourdon, Edward
Análisis estructurado moderno, 1993.

ANEXO I

SISTEMA DE REGISTRO Y CONTROL DE REAFIANZAMIENTO TOMADO (RT) NACIONAL.

Manual de Usuario

El sistema de reafianzamiento tomado nacional pretende apoyar la venta de fianzas por parte de AISA, a otras compañías de seguros y fianzas que estén interesados en ceder parte de sus fianzas.

El objetivo principal del sistema de registro y control de reafianzamiento tomado es registrar los movimientos de fianzas que AISA toma de compañías afianzadoras o aseguradoras tanto nacionales como extranjeras, mantener el control de estas fianzas y asegurar su correcta aplicación en la contabilidad de la compañía.

El presente documento tiene por objetivo el apoyar la operación del sistema y servir de guía para los usuarios del mismo.

Para lograr claridad en este documento y facilitar su acceso para consulta, se encuentra organizado de la siguiente forma:

Documento	Sección
Menús	1
Detalle de operación	2
Catálogos	2.1
Registro	2.2
Cobranza	2.3
Contabilización	2.4
Consultas	2.5
Reportes	2.6
Sistema	2.7

1. MENUES

MENU PRINCIPAL

- 1.- CATALOGOS.**
- 2.- REGISTRO.**
- 3.- COBRANZA.**
- 4.- CONTABILIZACION.**
- 5.- CONSULTAS.**
- 6.- REPORTE.**
- 7.- SISTEMA.**

CATALOGOS

- 1.- MONEDAS.
- 2.- RAMOS TIPOS.
- 3.- REFERENCIAS DE CUENTAS.
- 4.- COMPAÑIAS.
- 5.- OPERACIONES.
- 6.- USUARIOS.

REGISTRO

- 1.- EXPEDICION.**
- 2.- RENOVACION.**
- 3.- AUMENTO.**
- 4.- DISMINUCION.**
- 5.- ANULACION.**
- 6.- CANCELACION.**
- 7.- LEVANTAMIENTO DE CANCELACION.**

COBRANZA

- 1.- PAGAR.
- 2.- ROJO.
- 3.- AJUSTE.
- 4.- MODIFICAR AJUSTE.
- 5.- BAJA DE SALDO POR DISMINUCION.
- 6.- BAJA DE ROJO.
- 7.- APLICAR DISMINUCION.
- 8.- APLICAR ANULACION.
- 9.- APLICAR ROJO.

CONTABILIZACION (POLIZAS)

- 1.- EXPEDICION.
- 2.- RENOVACION.
- 3.- AUMENTO.
- 4.- DISMINUCION.
- 5.- ANULACION.
- 6.- CANCELACION.
- 7.- LEVANTAMIENTO DE CANCELACION.
- 8.- LIBERACION DE FUTUROS.
- 9.- INICIALIZAR POLIZAS DEL MES.
- 10.- ARCHIVO DE TRANSFERENCIA

CONSULTAS

- 1.- FIANZAS.
- 2.- FIANZAS MONEDA EXTRANJERA.
- 3.- SALDOS.
- 4.- FUTUROS.

REPORTES

- 1.- INVENTARIOS.
- 2.- PRODUCCION.
- 3.- SALDOS.

INVENTARIOS

- 1.- DETALLADO POR COMPAÑIA-RAMO.
- 2.- RESUMEN POR COMPAÑIA.
- 3.- FIANZAS MONEDA EXTRANJERA.
- 4.- RESUMEN CONCENTRADO.
- 5.- FUTUROS DETALLADOS.
- 6.- AVISOS DE VENCIMIENTO.

PRODUCCION

- 1.- VALIDACION DE PRODUCCION.
- 2.- DETALLADO DE PRODUCCION.
- 3.- ESTADISTICA DE PRODUCCION.
- 4.- ESTADISTICA GLOBAL.
- 5.- FUTUROS LIBERADOS.

SISTEMA

1.- USUARIOS.

2.- GENERALES DE INSTALACION.

2. DETALLE DE OPERACION

2.1 ACCESO AL SISTEMA

El acceso al sistema está restringido a través de una clave de usuario y un password.

La clave de acceso y su password será proporcionado por el área de sistemas y sólo dicha área podrá cambiarlo.



Para entrar al sistema digite su clave presione la tecla de <enter>, digite su password y la tecla <enter>, si la clave existe y el password es correcto se presentará la siguiente pantalla.



Oprima el botón de la selección deseada o el botón de salir si quiere finalizar su sesión. En este caso oprima el botón de catálogos y al hacer esto aparece la siguiente pantalla.

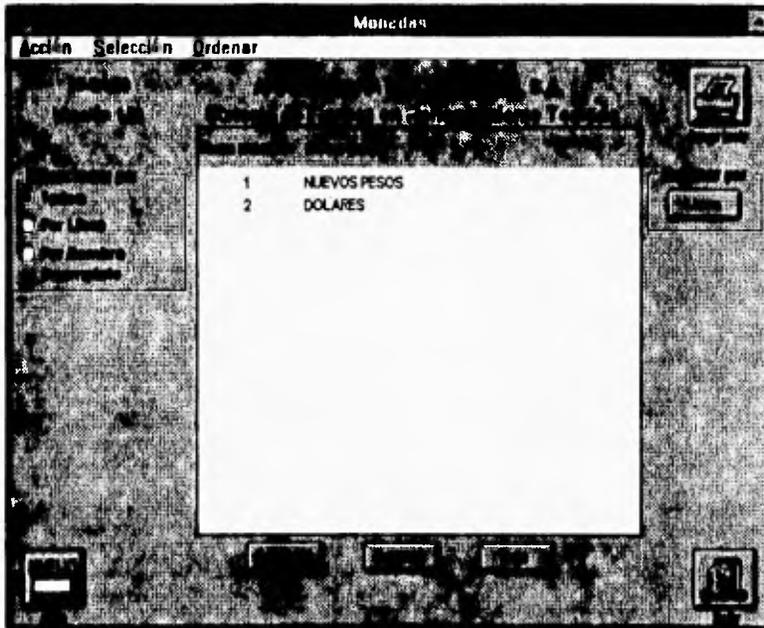
2.2 CATALOGOS

A continuación presentamos la pantalla de menú de catálogos y después se explica cada una las opciones de catálogos que se tienen mostrando las pantallas del sistema.



2.2.1 Catálogo de Monedas.

- Pantalla.



- Descripción General.

En esta opción se da mantenimiento al catálogo de monedas; esto es, se pueden dar de alta monedas nuevas, cambios a información ya existente o bien bajas de monedas ya registradas.



- Descripción de Campos.

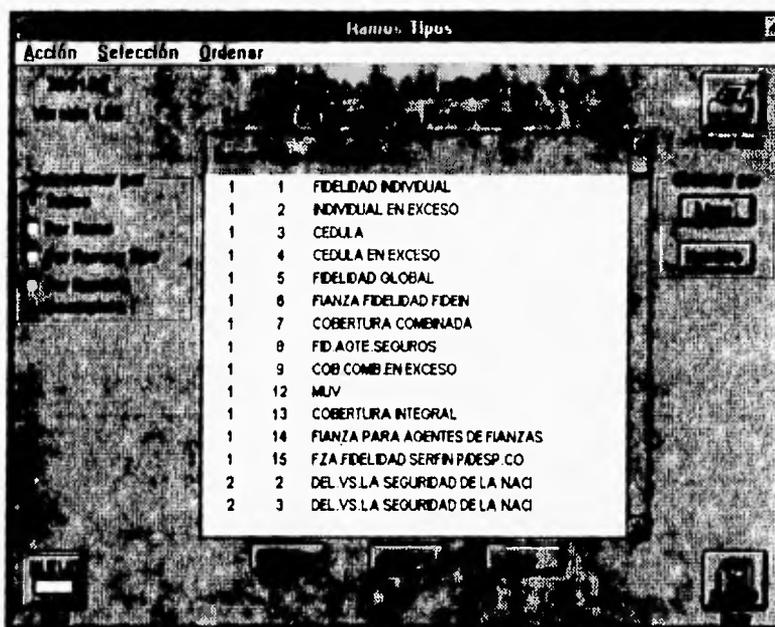
Clave Moneda.- Es la clave de la moneda.

Descripción de la moneda.- Nombre de la moneda.

Paridad para ajuste.- paridad al día con respecto al peso.

2.2.2 Catálogo de Ramos-Tipos.

- Pantalla.



The screenshot shows a software interface titled "Ramos Tipos". At the top, there are three menu options: "Acción", "Selección", and "Ordenar". The main area displays a list of insurance products with columns for "Ramo" and "Tipo".

Ramo	Tipo	Descripción
1	1	FIDELIDAD INDIVIDUAL
1	2	INDIVIDUAL EN EXCESO
1	3	CEDULA
1	4	CEDULA EN EXCESO
1	5	FIDELIDAD GLOBAL
1	6	FIANZA FIDELIDAD FIDEIN
1	7	COBERTURA COMBINADA
1	8	FID. AOTE. SEGUROS
1	9	COB. COMB. EN EXCESO
1	12	MUV
1	13	COBERTURA INTEGRAL
1	14	FIANZA PARA AGENTES DE FIANZAS
1	15	FZA FIDELIDAD SERFIN PADESP. CO
2	2	DEL. VS. LA SEGURIDAD DE LA NACI
2	3	DEL. VS. LA SEGURIDAD DE LA NACI

- Descripción General.

En esta opción se da mantenimiento al catálogo de ramos tipos. Un ramo tipo está definido por un ramo (ver anexo 1) y por un tipo (ver anexo 2). Esto es, se pueden dar de alta ramos tipos nuevos, cambios a información ya existente o bien bajas de ramos tipos ya registrados.



- **Descripción de Campos.**

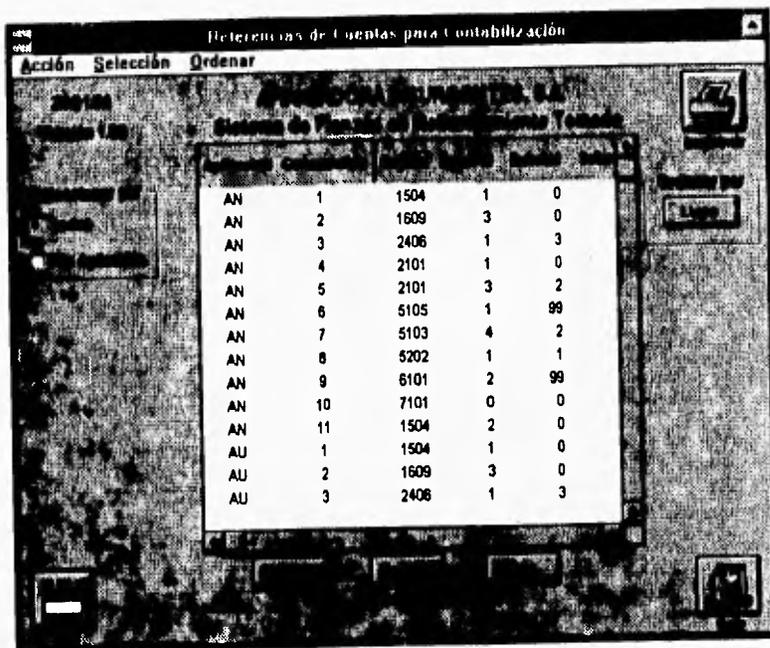
Ramo. - Es la clave del ramo de la fianza.

Tipo. - Es la clave del tipos de la fianza.

Descripción. - Es el nombre de la combinación de las claves del ramo tipo.

2.2.3 Referencia de cuentas para contabilización.

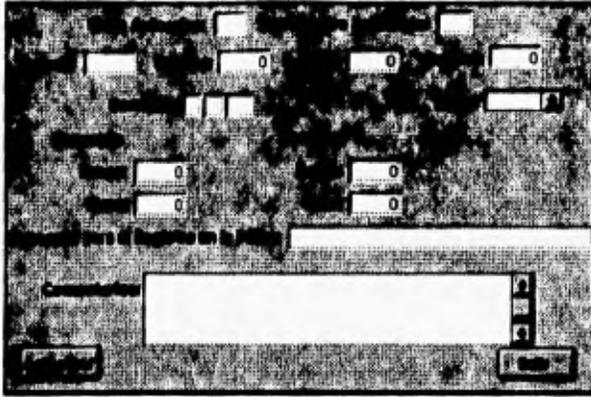
- Pantalla.



Cuenta	Subcuenta	Signo	Concepto	Colectiva
AN	1	1504	1	0
AN	2	1609	3	0
AN	3	2406	1	3
AN	4	2101	1	0
AN	5	2101	3	2
AN	6	5105	1	99
AN	7	5103	4	2
AN	8	5202	1	1
AN	9	6101	2	99
AN	10	7101	0	0
AN	11	1504	2	0
AU	1	1504	1	0
AU	2	1609	3	0
AU	3	2406	1	3

- Descripción General.

En esta opción se da mantenimiento al catálogo de referencias de cuentas que requiere la Comisión Nacional de Seguros y Fianzas. Esta referencia está definida por un conjunto de claves de cuentas y subcuentas, gerencia de expedición, signo, concepto y colectiva. Esto es, se pueden dar de alta referencias nuevas, cambios a información ya existente o bien bajas de referencias ya registradas.

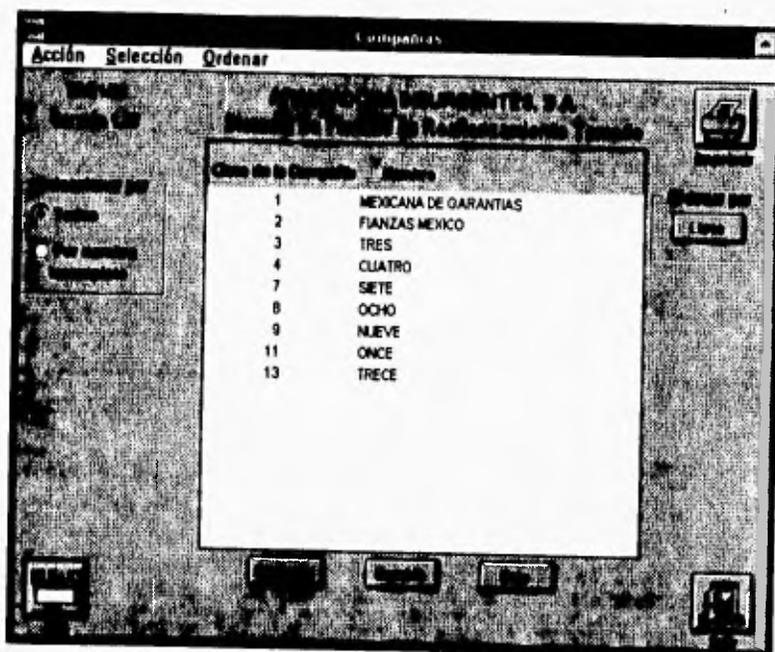


• **Descripción de Campos.**

- Clave aplicación.- Clave de aplicación.
- Consecutivo aplicación.- Consecutivo de la aplicación.
- Cuenta.- Clave de cuenta de la póliza.
- Subcta1.- Clave de la subcuenta uno de la póliza.
- Subcta2.- Clave de la subcuenta dos de la póliza.
- Subcta3.- Clave de la subcuenta tres de la póliza.
- Gerencia.- Clave de la gerencia de expedición.
- Signo.- Valor de debe o haber de la póliza.
- Concepto-Rango.- Rango del concepto.
- Concepto-Clave.- Clave del concepto.
- Colectiva-Rango.- Rango de colectiva.
- Colectiva-Clave.- Clave de colectiva
- Concepto para el registro de la póliza.- Leyenda o título de la póliza.
- Comentarios.- Descripción o comentarios especiales a la póliza.

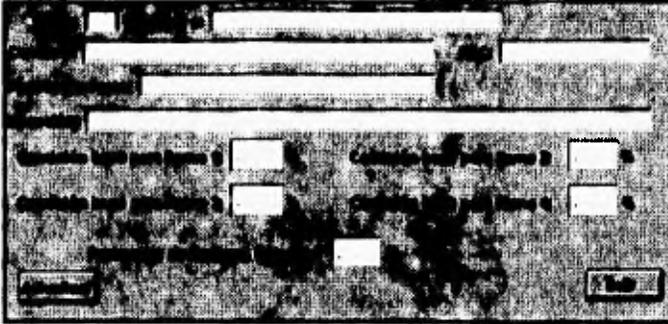
2.2.4 Catálogo de compañías.

- **Pantalla.**



- **Descripción General.**

En esta opción se da mantenimiento al catálogo de compañías; esto es, se pueden dar de alta compañías nuevas, cambios a información ya existente o bien bajas de compañías ya registradas.



• **Descripción de Campos.**

Clave .- Clave de la compañía.

Nombre.- Nombre de la compañía.

Contacto.- Nombre de la persona que atiende en la compañía.

RFC.- Registro Federal de Contribuyentes de la compañía.

Puesto Contacto.- Puesto del contacto dentro de la compañía.

Dirección.- Dirección de la compañía.

Comisión base para Ramo 1.- Comisión base de las ventas del Ramo 1.

Comisión base para Ramo 2.- Comisión base de las ventas del Ramo 2.

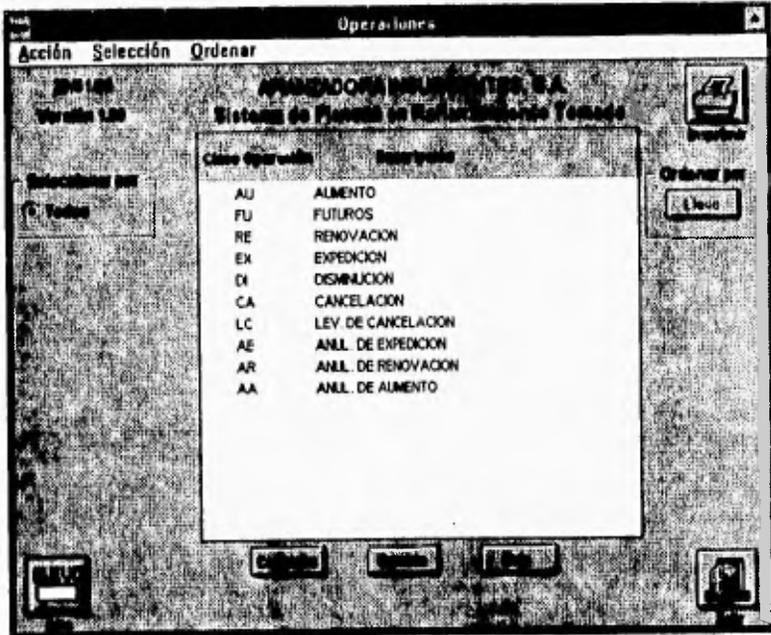
Comisión base para Ramo 3.- Comisión base de las ventas del Ramo 3.

Comisión base para Ramo 4.- Comisión base de las ventas del Ramo 4.

Porcentaje de Reserva Retenida.- Porcentaje de reserva según la siniestralidad de la compañía.

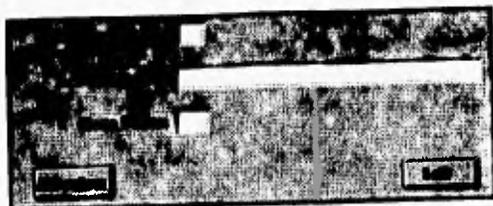
2.2.5 Catálogo de operaciones.

- Pantalla.



- Descripción General.

En esta opción se da mantenimiento al catálogo de operaciones de fianzas; esto es, se pueden dar de alta operaciones nuevas, cambios a información ya existente o bien bajas de operaciones ya registradas.



- **Descripción de Campos.**

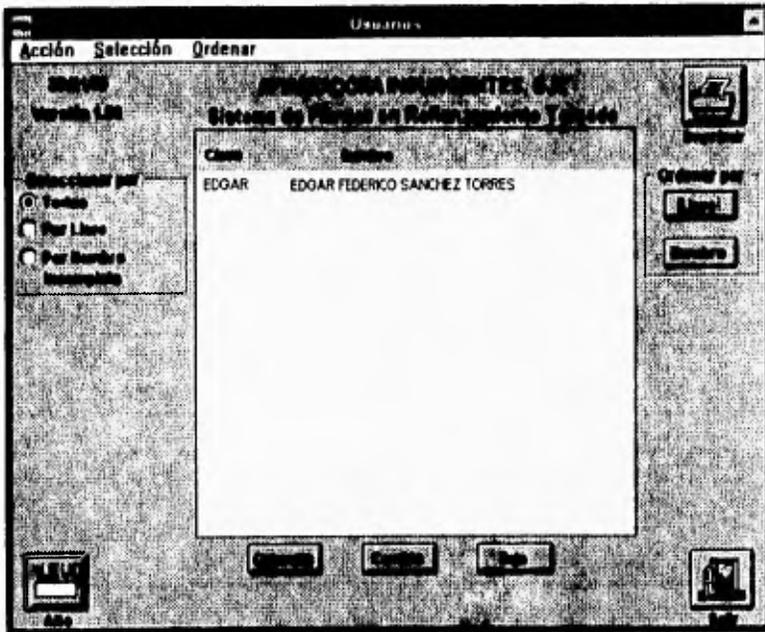
Clave operación.- Clave de la operación.

Descripción.- Descripción o nombre de la operación.

Clave Póliza.- Clave de póliza que le corresponde a esta operación.

2.2.6 Catálogo de usuarios.

- Pantalla.



- Descripción General.

En esta opción se da mantenimiento al catálogo de usuarios del sistema; hay que recordar que sólo se puede dar mantenimiento a usuarios con menor nivel al con el que se entro al sistema, esto es, se pueden dar de alta nuevos usuarios, cambios a información ya existente o bien bajas de usuarios ya registrados.



- **Descripción de Campos.**

Clave - Clave del usuario.

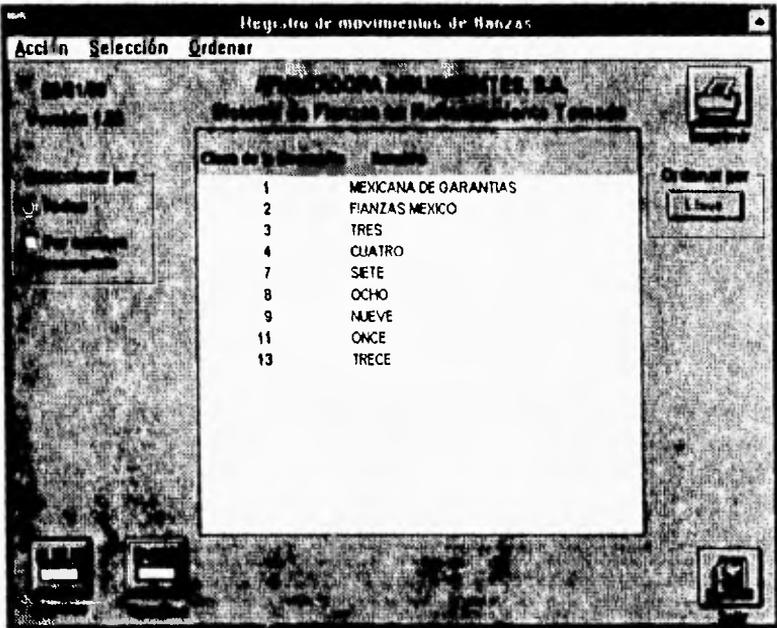
Password.- Llave de acceso al sistema del usuario.

Nivel Usuario.- Nivel de privilegio del sistema del usuario.

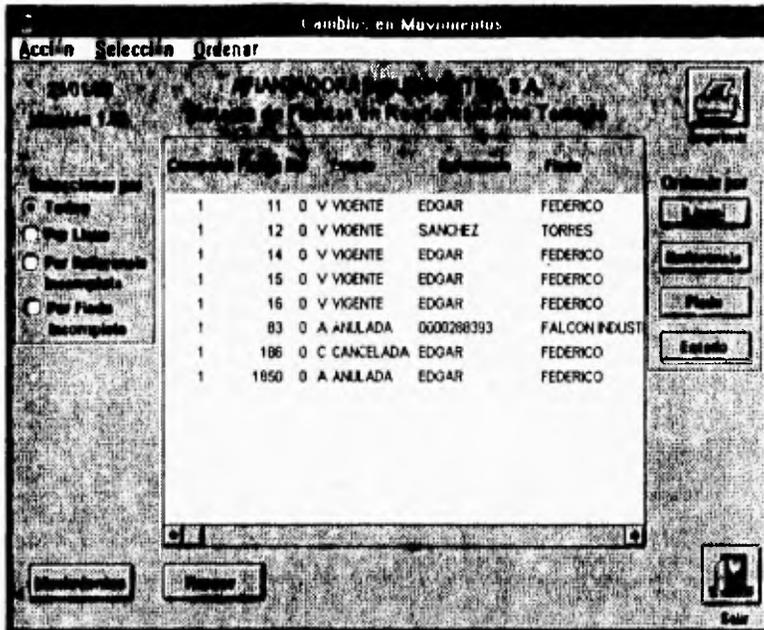
Nombre.- Nombre del usuario.

2.3 REGISTRO

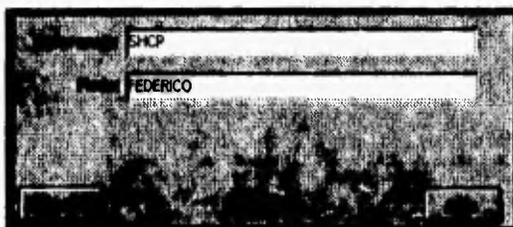
A continuación presentamos la pantalla de menú de registro y después se explican cada una las opciones de registro y cambio que se tienen y se mostrarán las pantallas del sistema.



Seleccione alguna compañía de la que se quiera registrar un movimiento, esto se hace haciendo clic con el mouse sobre la línea de esta compañía, a continuación seleccione el botón de cambios y este botón desplegará la siguiente pantalla.



De la misma manera que seleccionó la compañía ahora seleccione la fianza para hacer su cambio. Oprima el botón de Fianzas y este desplegará la siguiente pantalla.



Los únicos datos que se pueden cambiar en una fianza son la referencia de la fianza y su fiado, haga los cambios necesarios y después oprima el botón de actualizar o salir si no quiere actualizar los datos.

Si de la pantalla de cambios oprime el botón de movimientos la pantalla que aparece es la siguiente en donde, usted tiene que seleccionar el movimiento específico al que se quieran hacer cambios y después oprima el botón de cambio. Al realizar esto se despliega la siguiente pantalla

Cuentas	Plazo	No Plazo	Operación	Origen	Destino
1	12	0	EX	0111200	0111200
1	12	0	AU	0111200	0111200
1	12	0	AU	0111200	0111200

Botones:

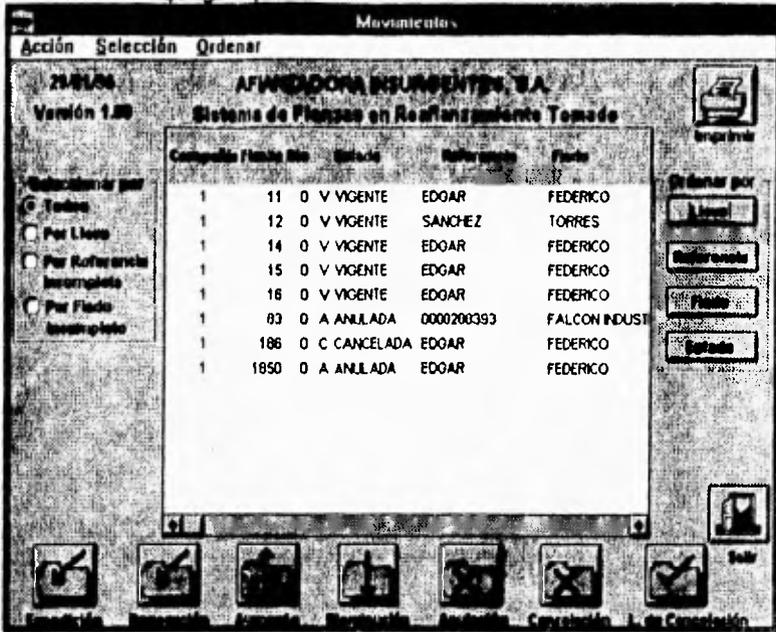
En esta pantalla usted puede hacer los cambios que le parezcan pertinentes en los campos que así lo permitan.

Cuenta: Operación: EX Fecha Base: 0000.00
 Plazo: 33
 No Plazo: 0
 Origen: Destino:
 Plazo Tipo: 0 10 CUMPLIMIENTO (OBRA)
 Moneda: NUEVOS PESOS

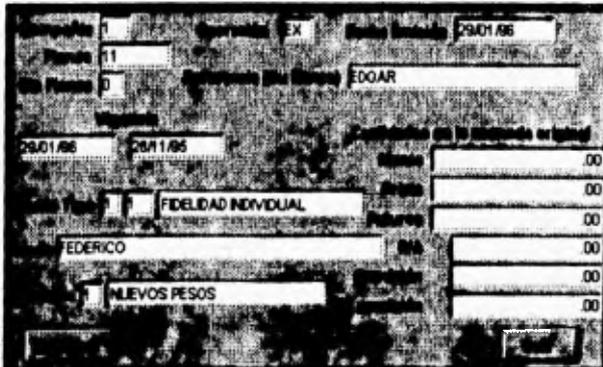
Cuentas en el momento original	
Monto	4,528,700.00
Primas	36,538.00
Polizas	00
Res	3,563.80
Seguros	00
Intereses	00

Botones:

Si en la pantalla principal de registro usted oprimió el botón de nuevo o fianzas la se despliega la pantalla de a continuación.

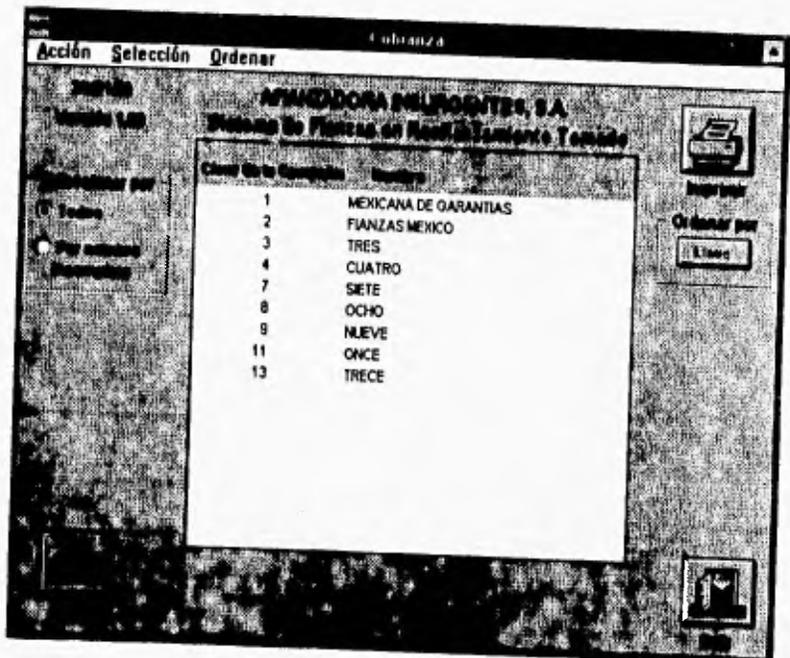


En esta pantalla se pueden capturar los movimientos de la fianza seleccionada. La pantalla de captura es la siguiente en todos los casos, lo único que cambia es la clave de operación, recuerde seguir con el ciclo de vida de la fianza (Una anulación no puede tener levantamiento de cancelación).

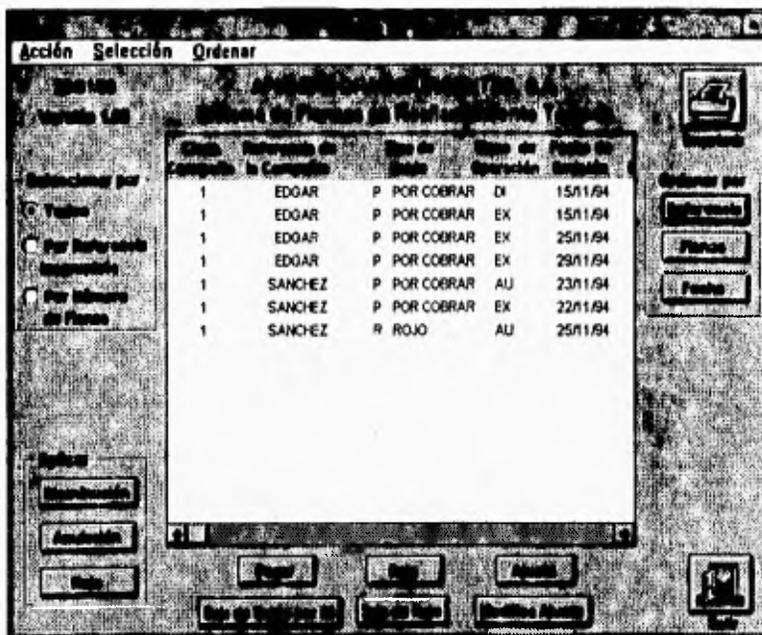


2.4 COBRANZA

A continuación presentamos la pantalla de menú de cobranza y después se explica cada una las opciones de cobranza que se tienen mostrando las pantallas del sistema.



En la pantalla anterior que es propiamente de presentación debe escoger la compañía con la que se quiere trabajar, y a continuación oprimir el botón de saldos. Esto desplegará la pantalla siguiente donde se encuentran las fianzas de la compañía escogida.



En esta pantalla se tienen varias opciones de pago, disminuciones, aplicaciones, modificaciones etc., todos los botones desplegarán la siguiente pantalla con distinta información en los campos y no todos los campos podrán captarse.

1 MERCADERIA DE GARANTIA																									
LANCHEZ																									
POR COBRAR																									
Fecha de Registro	22/11/04																								
Fecha de Pago	26/01/05																								
<table border="1"> <tr> <td>Primo</td> <td>0</td> </tr> <tr> <td>Primo Neto</td> <td>\$1,000.00</td> </tr> <tr> <td>Comisión</td> <td>\$100.00</td> </tr> <tr> <td>Por Intereses</td> <td>\$100.00</td> </tr> <tr> <td>Por Comisiones</td> <td>\$100.00</td> </tr> <tr> <td>Total</td> <td>\$1,300.00</td> </tr> <tr> <td>Primo</td> <td>0.00</td> </tr> <tr> <td>Primo</td> <td>0.00</td> </tr> <tr> <td>Primo</td> <td>\$100.00</td> </tr> <tr> <td>Total de Pago</td> <td>\$1,300.00</td> </tr> <tr> <td>Primo de Pago</td> <td>00-00-00</td> </tr> <tr> <td>Primo de Pago</td> <td>.00</td> </tr> </table>		Primo	0	Primo Neto	\$1,000.00	Comisión	\$100.00	Por Intereses	\$100.00	Por Comisiones	\$100.00	Total	\$1,300.00	Primo	0.00	Primo	0.00	Primo	\$100.00	Total de Pago	\$1,300.00	Primo de Pago	00-00-00	Primo de Pago	.00
Primo	0																								
Primo Neto	\$1,000.00																								
Comisión	\$100.00																								
Por Intereses	\$100.00																								
Por Comisiones	\$100.00																								
Total	\$1,300.00																								
Primo	0.00																								
Primo	0.00																								
Primo	\$100.00																								
Total de Pago	\$1,300.00																								
Primo de Pago	00-00-00																								
Primo de Pago	.00																								

En esta pantalla como ya se dijo, según el botón elegido anteriormente vendrán campos con información y campos donde no podrán hacerse cambios o captura.

2.5 CONTABILIZACION

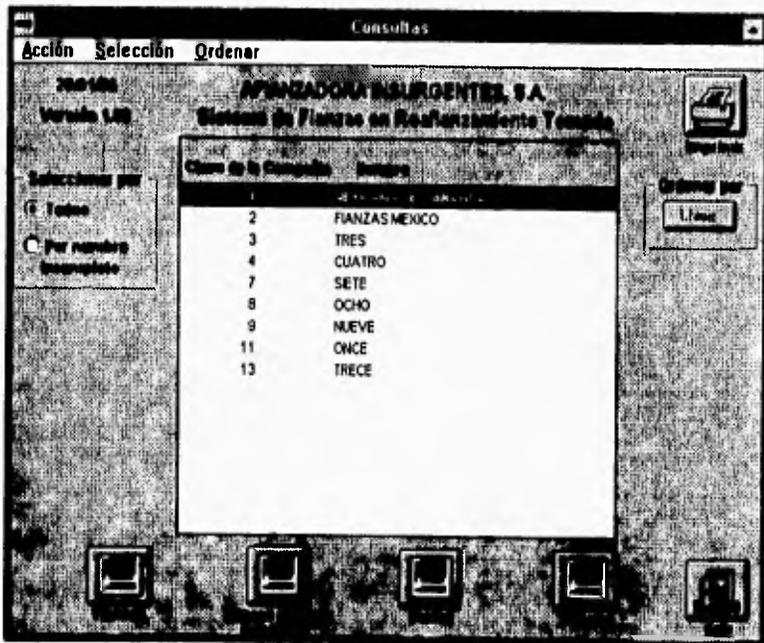
La pantalla es la pantalla principal de contabilización en esta pantalla se crean las pólizas contables que se migrarán al sistema central contable de la afianzadora.



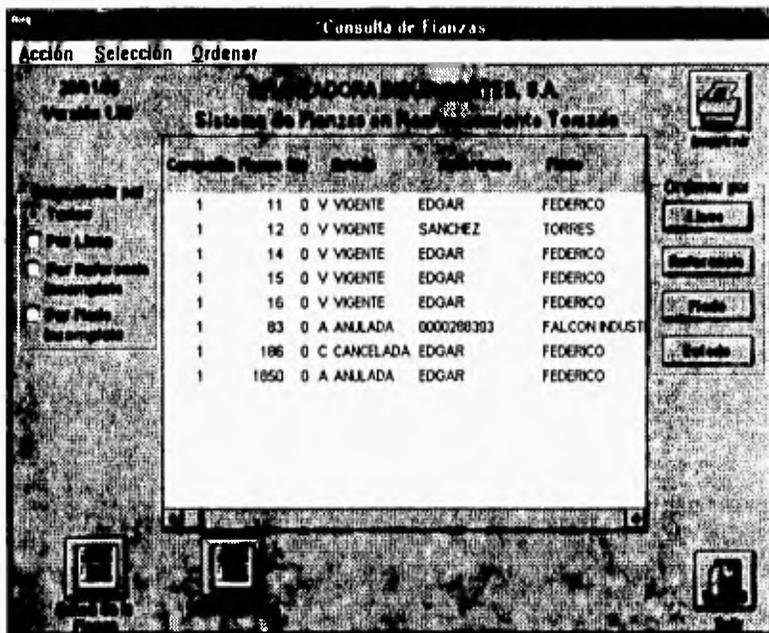
Las opciones de pólizas son las necesarias y exigidas por la comisión nacional de seguros y fianzas. Recuerde inicializar las pólizas del mes antes de sacarlas y tenga un disquete de alta densidad para su migración. También existe la opción de impresión de pólizas si se quiere revisar antes de migrar.

2.6 CONSULTAS

A continuación presentamos la pantalla de menú de consultas y después se explica cada una de las opciones de consulta que se tienen mostrando las pantallas del sistema.



En la pantalla debe escoger la compañía con la que se quiere trabajar, y a continuación oprimir el botón de consulta de fianzas. Esto desplegará la pantalla siguiente donde se encuentran las fianzas de la compañía escogida. Hay que hacer notar que la consulta de fianzas en moneda extranjera sigue exactamente la misma dinámica de consulta que la consulta de fianzas en moneda nacional y es por esto que sólo se explicará la consulta de fianzas en moneda nacional.

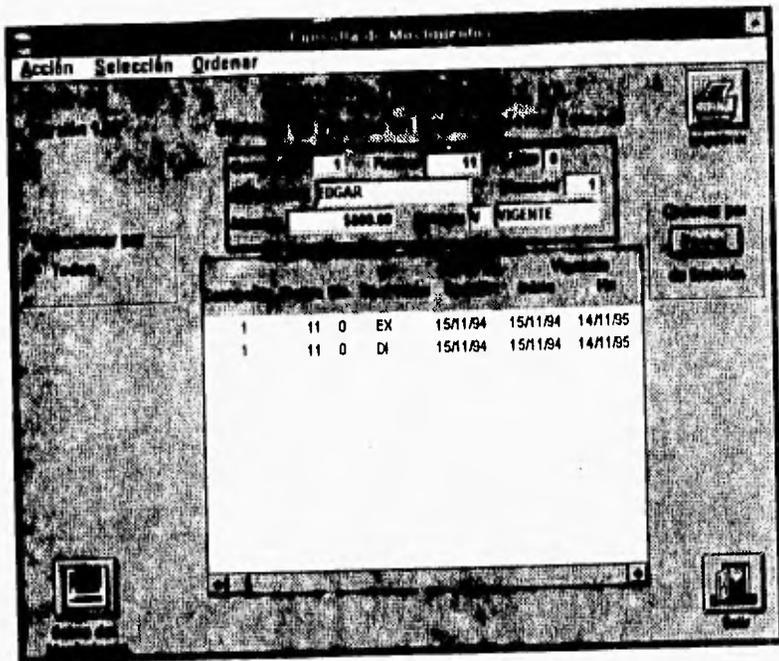


En esta pantalla se muestran los datos esenciales de la fianza, pero si su consulta debe ser más profunda entonces existen dos opciones, consultar los datos de la fianza y consultar sus movimientos. Si oprime el botón de datos de la fianza entonces se desplegará la siguiente pantalla.

Compañía	1 MEXICANA DE GARANTIAS	
Código	03	
Subcódigo	0	
Cuenta	0000200303	
Operación	3	10 CUMPLIMIENTO (OBRA)
Importe Bruto	11,000.00	Importe Neto 10,000.00
Monto	1 NUEVOS PESOS	
Monto Pagar	\$4,620,700.00	
Monto Pagar	\$34,630.00	
Cero Pagar	FALCON INDUSTRIES IN	
Cero Pagar	A ANULADA	
Importe Pagar	\$10,310.00	
Importe Net Pagar	0.00	
Reporte en moneda nacional		
		Salir

Esta pantalla muestra todos los datos de la fianza y recuerde que sólo es de consulta y aquí no se puede hacer ningún cambio.

Si su selección fue la de movimientos de la fianza entonces se despliega la siguiente pantalla donde se muestran todos los movimientos que se ha tenido hasta ese momento en forma resumida pero si su consulta quiere ser a detalle existe un botón de datos del movimiento donde podrá verse toda la información del movimiento seleccionado.



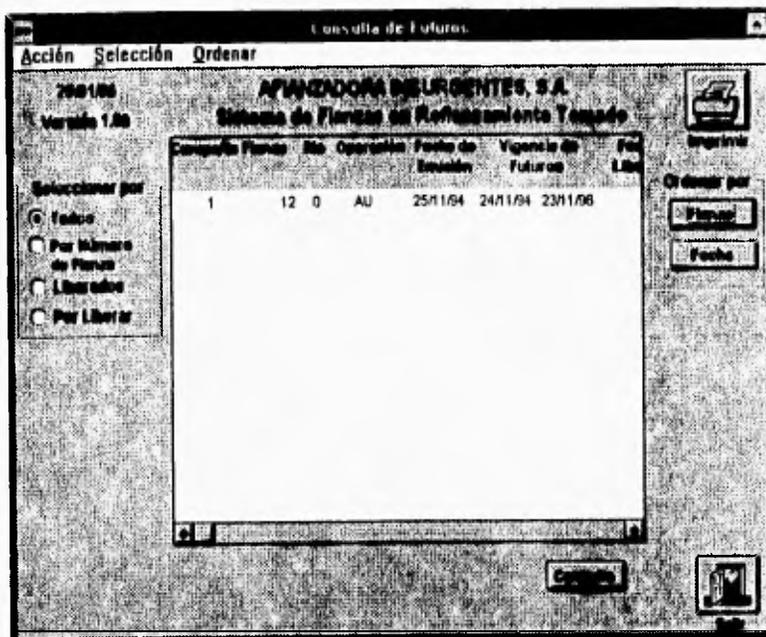
Los campos que se muestran en la pantalla son los que integran al movimiento y recuerde que en este momento usted no puede hacer cambios de información.

Compañía	1 MECANICA DE GARANTIAS	
Plano	11	0
Concepto	EX EXPEDICION	
Fecha de Emisión	15/11/84	
Deposito	MON	PTA
	15/11/84	14/11/84
Monto	\$1,000.00	
Primo	\$100.00	
Póliza	0.00	
Reserva	000.00	
Fecha de Anulación		
Reserva Retenido	0.00	
PVA Monto	\$10.00	
Cantidad Monto	\$10.00	
PVA Comis	\$1.00	
Fecha de Registro	15/11/84	
Plano	1	1 FIDELIDAD INDIVIDUAL

La pantalla que a continuación se presenta es la correspondiente a la consulta de saldos de fianzas de la compañía previamente escogida. En la parte baja de la ventana de datos se encuentra un scroll bar para ver los campos que no se ven a simple vista.

Clase	Seguro	Tipo de Seguro	Fecha de Emisión	Saldo de Fianza
1	EDOAR	P POR COBRAR	DI	15/11/94
1	EDOAR	P POR COBRAR	EX	15/11/94
1	EDOAR	P POR COBRAR	EX	25/11/94
1	EDOAR	P POR COBRAR	EX	29/11/94
1	SANCHEZ	P POR COBRAR	AU	23/11/94
1	SANCHEZ	P POR COBRAR	EX	22/11/94
1	SANCHEZ	R ROJO	AU	25/11/94

La pantalla que a continuación se presenta es la correspondiente a la consulta de futuros de fianzas de la compañía previamente escogida. En la parte baja de la ventana de datos se encuentra un scroll bar para ver los campos que no se ven a simple vista.



2.7 REPORTE

La pantalla es la pantalla principal de reportes en esta pantalla se crean los reportes requeridos por el área de reafianzamiento tomado.



Las opciones de reportes se subdividen en tres que son los inventarios, los reportes de producción y el reporte de saldo o estado de cuenta de la compañía.

Los reportes de inventarios son propiamente los reportes que utilizarán los funcionarios. Para sacar cualquiera de los reportes abajo mostrados en la pantalla sólo es necesario oprimir el botón del reporte deseado.

Después de oprimir el botón aparecerá una pequeña ventana donde se pedirá la fecha de corte del reporte o un rango de fechas según sea el caso.

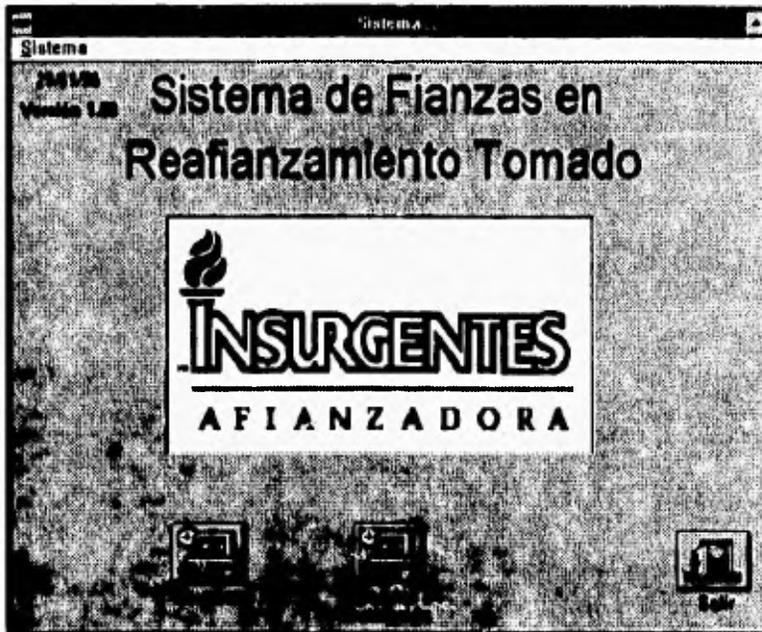




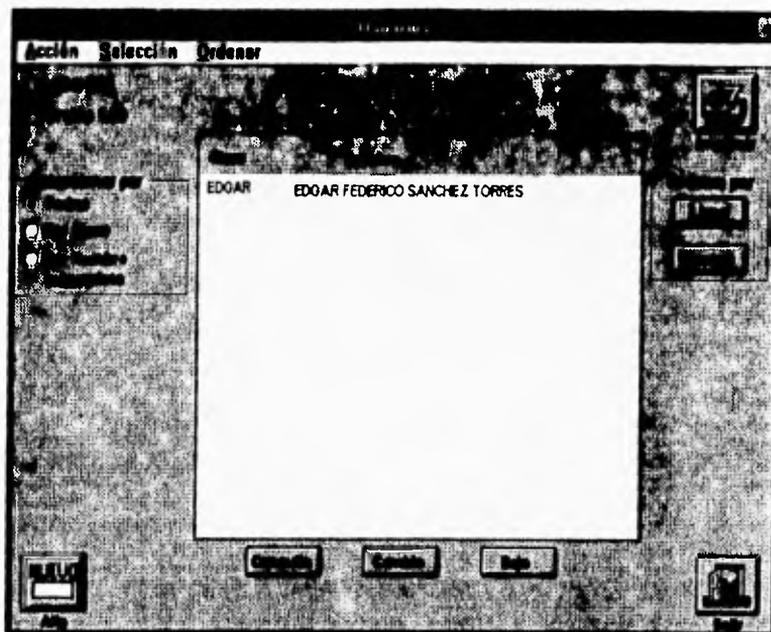
La pantalla muestra los reportes de saldos, en este caso solo existe el reporte de estado de cuenta de la compañía previamente seleccionada.

2.8 SISTEMA

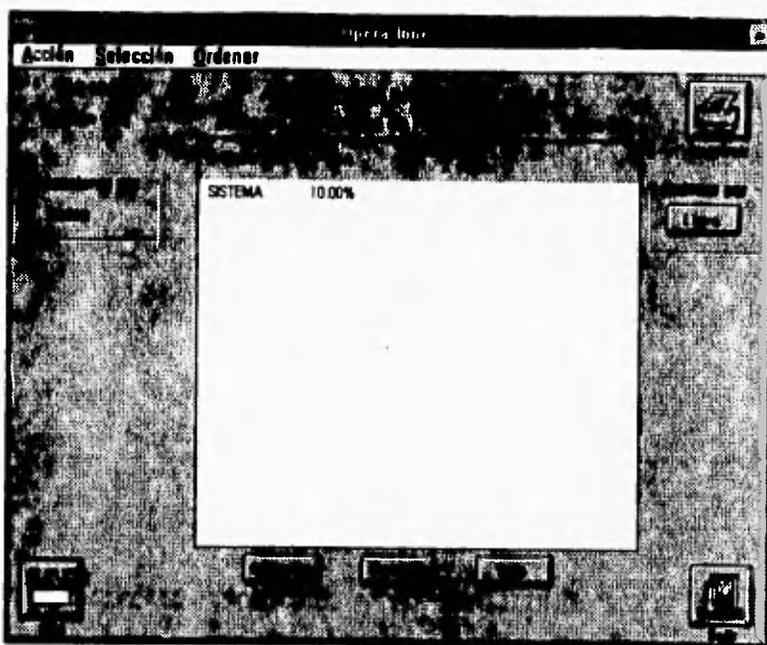
La pantalla de sistema es la siguiente y nos muestra las opciones que se tienen. En esta parte del sistema es posible dar de alta nuevos usuarios y también cambiar algunos datos generales del sistema como lo es el I.V.A..



Oprima el botón de usuarios y se desplegará la siguiente pantalla.



Esta pantalla muestra los usuarios de igual o menor categoría o jerarquía del sistema. En este punto usted puede dar de alta nuevos usuarios, se pueden hacer consultas de los mismos, así como dar de baja usuarios ya existentes.



En esta pantalla se pueden cambiar parámetros generales del sistema como lo son el porcentaje de I.V.A. o las comisiones. Esto se puede hacer para cada uno de las claves de usuario, para poder tener más opciones.