



18

UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO E IMPLANTACION DE UN SISTEMA DE  
CONTROL DE CARTERA PARA ASEGURADORAS

T E S I S

QUE PARA OBTENER EL TITULO DE  
INGENIERO EN COMPUTACION  
P R E S E N T A N :  
F R E D D Y C A N O R U I Z  
M A N U E L H U M B E R T O E S Q U I V E L R A M O S  
J O S E M I J A N G O S T O R R E S  
M A R I A T O R R E S H E R N A N D E Z



DIRECTOR DE TESIS: M. I. JUAN CARLOS ROA BEIZA

MEXICO, D. F.

JULIO 2001



Universidad Nacional  
Autónoma de México



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

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

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

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

FREDDY CANO

## **ESTA TESIS LA DEDICO**

### **A MIS PADRES (MARIA LUISA Y MANUEL):**

POR SU APOYO INCONDICIONAL EN LAS BUENAS Y EN LAS MALAS, SIN USTEDES NO ESTARIA DONDE ESTOY Y NO SERIA LO QUE SOY. USTEDES SIEMPRE FUERON Y SEGUIRAN SIENDO UN EJEMPLO DE VALENTIA ANTE LA VIDA. GRACIAS, SON LOS MEJORES. ESTA TESIS ES DE USTEDES.

### **A MIS HERMANOS (MANUEL, MAURICIO, DANIEL Y MONICA):**

POR SER COMO SON, POR SU RESPETO, POR SU APOYO Y POR SER LOS MEJORES HERMANOS QUE PUEDE HABER TENIDO.

### **A MIS SOBRINOS: (VANNESA, LUIS Y KATY):**

POR SU CARIÑO, POR SUS TRAVESURAS Y POR TODO LO QUE HACEN, USTEDES VINIERON A DAR MUCHA ALEGRIA A LA FAMILIA Y PARA MI SON UNA FUENTE DE INSPIRACION.

## **AGRADECIMIENTOS**

### **A GLORIA:**

POR TU IMPULSO, POR TU APOYO, POR TU AMOR, POR AGUANTAR MI CARACTER Y POR SER COMO ERES.

### **A MIS AMIGOS:**

POR SU AMISTAD, POR SU APOYO, POR ESTAR AHI CUANDO LOS NECESITE, NO QUISIERA LISTARLOS POR TEMOR A OMITIR A ALGUNO.

### **A MI ASESOR DE TESIS (ING. JUAN CARLOS ROA BEIZA):**

POR BRINDARNOS SU APOYO EN LA ELABORACION DE ESTA TESIS Y POR AGUANTAR NUESTROS RETRASOS.

### **A MIS COMPAÑEROS DE TESIS (MARY, MANOLO Y PEPE):**

POR SU RESPONSABILIDAD Y COMPROMISO Y POR SU AMISTAD.

## ***Agradecimientos Manuel***

### ***A mis padres:***

***María y Humberto por que siempre me han apoyado y en todo momento han estado junto a mí Impulsándome hacia delante, gracias por su esfuerzo y dedicación.***

### ***A mis hermanas:***

***Faby y Nadia que con sus sonrisas y su fuerza para vivir la vida me contagian de entusiasmo, gracias por su apoyo y comprensión.***

### ***A mis familiares:***

***Este esfuerzo es de todos y para todos.***

### ***A mis amigos:***

***Por que en el relajo y las cosas importantes de la vida siempre han sabido apoyar y estar conmigo.***

### ***A mis compañeros de tesis:***

***Por que hemos hecho un gran grupo nos espera un futuro muy brillante.***

A mis padres,

Les agradezco todo su apoyo, dedicación y desvelos incondicionales, ya que sin ellos no hubiera alcanzado esta meta, ni tendría el valor de plantearme ningún otro reto. Ya que son el pilar más importante en mi vida.

**Este trabajo esta dedicado a ellos.**

A mi hermano,

Le agradezco su ayuda incondicional en los momentos que lo necesite.

A mi esposa,

Le agradezco su apoyo y fuerza para soportar los desvelos que le provoqué, y la fuerza que me dio para concluir con esto.

A mi director de tesis,

Les agradezco su ayuda y orientación para concluir satisfactoriamente este trabajo, ya que es el soporte de mi tesis.

A Todos,

Mil Gracias, ya que si no están nombrados es porque no hay espacio, pero agradezco su apoyo.

**JOSÉ MIJANGOS TORRES**

## DEDICATORIA

**A papá:** Por tu apoyo incondicional, por tu amor, porque siempre me sacaste adelante, y por tu gran sabiduría. Te quiero mucho.

**A mamá:** Por ser siempre un ejemplo de fortaleza, por tus enseñanzas, tu amor, comprensión, tus consejos y apoyo. Te quiero mucho.

**A David:** Por tu dedicación, comprensión, apoyo, compañía, porque eres mi fuerza para seguir adelante. Pero sobre todo por ser el amor en mi vida. Te amo.

**A mis hermanos:** Porque todos son mi ejemplo como ser humano en la vida, y porque de cada uno de ustedes he aprendido algo valioso en mi vida.

**Este trabajo es para todos ustedes. Gracias infinitamente.**

## AGRADECIMIENTOS

**Ing. Juan Carlos Roa Beiza:** Por su apoyo en el desarrollo de este trabajo.

**Ing. Víctor D. Pinilla Morán:** Por el apoyo brindado desde mi ingreso a la UACT.

A todas las personas que de alguna manera contribuyeron al término de este trabajo.

**María.**



## ÍNDICE TEMÁTICO

### INTRODUCCIÓN

#### CAPÍTULO 1 MARCO TEÓRICO

1.1. Conceptos básicos sobre seguros	3
1.1.1. El riesgo clasificación y características	3
1.1.2. El seguro	6
1.1.3. La indemnización	6
1.1.4. Principales riesgos cubiertos por las compañías de seguros	7
1.1.5. Concepto de ramo	8
1.1.6. Elementos del contrato de seguro	8
1.1.7. Prima	9
1.1.8. Elementos fundamentales de la póliza	10
1.1.9. Conceptos importantes dentro del contrato de seguros	10
1.1.10. Metodología para la asignación de montos, tarifas y deducibles para los diferentes riesgos	11
1.1.10.1. Determinación de montos de tarifa (cuotas)	11
1.1.10.2. Determinación de deducibles	16
1.2. Conceptos básicos sobre contabilidad (cobranza)	18
1.2.1. Qué es cobranza	18
1.2.2. Qué es la cartera	20
1.2.3. Qué es un recibo	22
1.2.3.1. Tipos de recibo	22
1.2.4. Vencimiento de documentos	22
1.2.5. Recuperación de cartera	23
1.2.6. Aspectos legales	27
1.3. Teoría de base de datos relacional	28

#### CAPÍTULO 2 METODOLOGÍA DE DESARROLLO.

2.1. Metodología del diseño de la base de datos relacional	37
2.1.1. Elección de la metodología óptima	38
2.2. Sistemas operativos	51
2.2.1. Windows 9x / NT (Server/Workstation)	51
2.2.2. Windows 95	51
2.2.1. Windows NT	56
2.3. Tipos de redes y protocolos de comunicación	64
2.3.1. Antecedentes	64
2.3.2. Redes de computadoras, tipos y configuraciones	64
2.3.2.1. Tipos de Redes	64
2.3.2.2. Topología	66
2.3.2.2.1. Configuración estrella	66
2.3.2.2.2. Configuración anillo	67
2.3.2.2.3. Configuración bus	68
2.3.2.2.4. Configuración malla	69
2.3.2.3. Redes LAN	69
2.3.2.4. Redes WAN	71
2.3.2.5. Redes Ethernet	72
2.3.3. Comunicaciones	72
2.3.3.1. Modelos de interconexión	72
2.3.3.2. Protocolos de control de transferencia	72
2.3.3.3. Protocolos TCP/IP	74
2.3.3.4. Protocolos de transferencia de archivos simples	78
2.3.3.5. Protocolos de acceso a red	78
2.3.3.6. Arquitectura TCP/IP	80
2.3.3.7. Modelo OSI	81
2.3.3.8. Servicio en modo conexión	83
2.3.3.9. Suite de protocolo TCP/IP	84
2.4. Características, ventajas y desventajas de SQL SERVER	90
2.5. Características, ventajas y desventajas del Front End a elegir	103
2.5.1. Requerimientos de hardware y software	103
2.5.2. Características de Visual Basic 5.0	104



2.5.3. Entorno integrado de desarrollo de Visual Basic .....	109
2.5.3.1. Cuadro de herramientas .....	109
2.5.3.2. Barra de menús .....	112
2.5.3.3. Ventana explorador de proyectos .....	112
2.5.3.4. Ventana propiedades .....	112
2.5.3.5. Ventana de código .....	113
2.5.3.6. Asistentes .....	114
2.5.3.7. Cómo funciona el administrador de asistentes .....	114

### CAPÍTULO 3 PLANTEAMIENTO DEL PROBLEMA.

3.1. Definición del problema .....	117
3.1.1. Cobranza .....	117
3.1.2. Emisión de pólizas .....	118
3.1.3. Recuperación de recibos .....	119
3.1.4. Auditoría de agentes .....	119
3.1.5. Auditoría interna .....	120
3.1.6. Dirección .....	121
3.1.7. Clientes (Agentes y Promotores) .....	121
3.1.8. Planteamiento general .....	122
3.2. Situación actual .....	124
3.2.1. Cobranza .....	125
3.2.2. Emisión de pólizas .....	126
3.2.3. Recuperación de recibos .....	126
3.2.4. Auditoría de agentes .....	127
3.2.5. Auditoría interna .....	128
3.2.6. Dirección .....	129
3.2.7. Clientes (Agentes y Promotores) .....	129
3.2.8. Esquematización de la situación actual .....	130
3.2.9. Áreas de interés .....	132
3.3. Requerimientos del usuario .....	133
3.3.1. Planteamiento .....	134
3.3.2. Requerimientos gerenciales .....	134
3.3.3. Requerimientos de campo .....	135
3.3.4. Análisis de requerimientos .....	136
3.3.5. Justificación del análisis de requerimientos .....	136
3.3.6. Beneficios .....	137
3.4. Recopilación y análisis de la información .....	138
3.5. Descomposición funcional .....	145

### CAPÍTULO 4 PROPUESTA DE SOLUCIÓN.

4.1. Opciones de solución .....	151
4.2. Elección de la solución óptima .....	159
4.2.1. Comparación de los programas para desarrollar Back Ends .....	159
4.2.2. Comparación de los programas para desarrollar Front Ends .....	161

### CAPÍTULO 5 DESARROLLO E IMPLANTACIÓN DEL SISTEMA.

5.1. Base de datos .....	165
5.1.1. Diagrama de contexto .....	165
5.1.2. Diagrama de flujo de datos .....	165
5.1.3. Diccionario de datos .....	172
5.1.4. Diagrama Entidad-Relación .....	176
5.1.5. Normalización .....	179
5.2. Generación de código para el procesamiento de información .....	183
5.3. Diseño y construcción del Front End .....	196
5.3.1. Proyecto .....	196
5.3.2. Conexión .....	198
5.3.3. Consultas .....	199
5.3.4. Reportes .....	202
5.3.5. Catálogo .....	204
5.3.6. Ayuda .....	205
5.3.7. Módulos .....	205
5.3.8. Archivo ejecutable .....	206



## DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

5.4. Diseño e implantación de rutinas diagnóstico y evaluación de los datos	207
5.4.1. Introducción	207
5.4.1.1. Flujo de información para las pruebas	207
5.4.2. Tipos de pruebas	208
5.4.2.1. Prueba funcional	208
5.4.2.2. Prueba de recuperación	209
5.4.2.3. Prueba de desempeño	209
5.4.2.4. Prueba exhaustiva	209
5.5. Integración y pruebas del sistema	210
5.6. Factibilidad técnica y operativa	227
<b>CONCLUSIONES</b>	<b>237</b>
<b>APÉNDICES</b>	
Apéndice A. Manual de Usuario	243
Apéndice B. Manual Técnico	260
Apéndice C. Código del Sistema	268
<b>BIBLIOGRAFÍA</b>	<b>287</b>



## INTRODUCCIÓN

El objetivo general de este trabajo es el diseñar e implementar el sistema mencionado de control de cartera y así poder ofrecer un servicio mas competitivo para la atención a los agentes con la notificación de su cartera, y a la compañía con la recuperación de la misma, reduciendo en gran parte el problema de cartera vencida y la eliminación de la responsabilidad que no es pagada a la empresa. De igual forma se trata de mejorar los procesos actuales dando un apoyo mas preciso y oportuno de información.

El área de cobranza de una compañía de seguros ofrecerá a sus unidades regionales, promotorías y agentes un nuevo servicio, recibos en poder del intermediario, teniendo como base la recuperación de la cartera, este servicio dará oportunidad a los agentes y a las oficinas regionales de tener la capacidad de cobro que la empresa requiere, los principales servicios que ofrecerá el sistema son: consultas y reportes a cualquier nivel, además, el sistema contará con la información actualizada al día.

Esta institución cuenta con oficinas en gran parte de la república, por lo que tiene una cobertura nacional, la estructura de la compañía se compone de la dirección general, las oficinas divisionales, subdirecciones regionales y oficinas regionales, como se muestra en el anexo de la estructura de la organización, con este sistema la empresa planea recuperar un gran número de la cartera vencida, así como tener un mayor control sobre los promotores y agentes.

Cabe mencionar que actualmente se generan los recibos de cobro y estos son entregados a los agentes, y mientras estos no regresen los recibos que no se pueden cobrar o los ya cobrados, el cliente continúa asegurado, aún y que no se haya pagado la prima correspondiente, por lo que la responsabilidad de la empresa no concluye hasta la recuperación de estos recibos.

Para este sistema de control de cartera se propone desarrollar un sistema a la medida de las necesidades de la compañía aseguradora. Estas necesidades son la solicitud de la cartera individual de cada agente y resúmenes concentrados de promotorías, oficinas regionales y divisionales, así como la auditoría a cada uno de los recibos recuperados. Esto permitirá a la compañía aseguradora una mayor eficiencia en la recuperación de la cartera y el control sobre los adeudos de todos y cada uno de los agentes, que son la inyección de dinero a la empresa.

La principal función del sistema de control de cartera es administrar la cartera de los agentes, con el fin de evitar que la cartera vencida crezca y poder auditar a los agentes con el fin de evitar fraudes a la empresa. Con esto se le debe brindar confianza y seguridad a los clientes de que su dinero ingresa oportunamente a la compañía y su protección esta garantizada. El sistema tendrá cargas diarias de información, las cuales serán alimentadas del sistema principal de la empresa. Adicionalmente se tendrá registrada una historia sobre todos los recibos que hayan sido requeridos para una auditoria mas profunda ya que estos recibos pueden ser origen de un fraude.



El sistema contará con:

1. Consultas a cualquier nivel (agente, promotor, región o división).

El sistema generará tres tipos de reporte:

1. Reporte de la auditoría a un recibo.
2. Reporte detallado por agente.
3. Reporte concentrado por promotoría, división u oficina regional.

Al terminar el proceso de auditoría sobre algún recibo, este ya no será desplegado por el sistema, pero no perderá la información capturada.

El sistema de control de cartera deberá usar una plataforma Cliente/Servidor centralizada, conectando todas y cada una de las oficinas regionales con una red TCP/IP de estrella, el cliente (Front End) deberá ser una interfaz gráfica y amigable al usuario del sistema.



# CAPÍTULO I

## MARCO TEÓRICO



## 1.1. CONCEPTOS BÁSICOS SOBRE SEGUROS.

A continuación daremos una explicación de los seguros que se ofrecen en esta aseguradora, teniendo en cuenta que posiblemente en las demás aseguradoras manejen paquetes similares.

### 1.1.1. El riesgo: clasificación y características.

Desde que el hombre existe sobre la faz de la tierra ha vivido expuesto a infinidad de riesgos, los cuales han significado una amenaza constante para la preservación de su patrimonio familiar, el de su persona y el de sus seres queridos. De ahí que el concepto de riesgo sea precisamente el que señale el punto de partida para definir la actividad asegurada.

**Riesgo.** Es la exposición a cada una de las contingencias que pueden producir consecuencias negativas en lo material (figura 1.1.1.1).

El riesgo es una eventualidad que tiene cierta probabilidad de ocurrir y que pueda acarrear, como consecuencia una pérdida económica.

- Riesgos especulativos. Son aquellos cuyas consecuencias pueden ser favorables o adversas.
- Riesgos puros. A diferencia de los especulativos acarrear siempre consecuencias adversas.

**Siniestro.** Destrucción fortuita o pérdida importante que sufren las personas.



Figura 1.1.1.1 Riesgo.



Es la contingencia que, si sucede afecta al patrimonio familiar, produciendo un desequilibrio con consecuencias económicas negativas.

Los riesgos se pueden clasificar de la siguiente manera:

- Por la naturaleza de las pérdidas.
- Puros. Involucran solamente la probabilidad o posibilidad de pérdida.
  - Especulativos. Involucran la posibilidad tanto de pérdida como de ganancia.
- Cabe mencionar que los riesgos puros son asegurables y los especulativos no lo son.
- Por su origen o alcance.
- Catastróficos. Aquellos que afectarán a gran cantidad de personas.
  - Particulares. Cuando afectan únicamente a un individuo.
- Por su valoración.
- Riesgos objetivos. Son los que pueden ser medidos de acuerdo a estadísticas.
  - Riesgos especiales. Son los que debido a su magnitud o naturaleza exceden la posibilidad de cobertura de un seguro normal.
- Por su tiempo y lugar.
- Riesgos temporales.
  - Riesgos geográficos.
- Por el objeto sobre el que recae.
- Riesgo patrimonial. Aquel que afecta al patrimonio asegurado.
- Por su intensidad.
- Variable. Aquel que puede tener consecuencias diferentes.
  - Constante. Aquel que siempre tiene los mismos efectos.

El asegurador, por medio de la valoración y estadística del riesgo, puede obtener el factor conocido como cuota de riesgo puro. Es decir, agruparlos por su tipo y características y homogeneizarlos de acuerdo a su giro y género.

Las características de los riesgos se describen de la siguiente forma:

- Aleatorio o incierto. Sobre el riesgo debe haber incertidumbre pues el conocimiento de su existencia real haría desaparecer la aleatoriedad.
- Posible. A de existir la posibilidad de que el riesgo suceda. Ya que si no existe la posibilidad no hay riesgo que asegurar (figura 1.1.1.2)



Figura 1.1.1.2 El riesgo debe tener una posibilidad de ejecución.

Concreto.	Una designación ambigua del riesgo de asegurarse imposibilita el estudio y análisis previos a la aceptación del mismo, lo que trae como consecuencia que no pueda calcularse adecuadamente la cuota que deberá establecerse.
Lícito.	El riesgo no debe ir en contra de las leyes del país.
Fortuito.	El riesgo debe provenir de un acto o acontecimiento ajeno a la voluntad humana.
Produce una pérdida económica.	La realización del riesgo debe causar una pérdida económica.

Los siniestros pueden clasificarse por la intensidad del daño (figura 1.1.1.3):

Siniestro de pérdida total.	Aquel cuyas consecuencias han afectado completamente el objeto asegurado.
Siniestro de pérdida parcial.	Aquel que afecta parcialmente el objeto asegurado.



Perdida Parcial

Perdida Total



Figura 1.1.1.3 Siniestro de pérdida parcial y pérdida total.

### 1.1.2. El seguro.

Se define al seguro como una cobertura recíproca y colectiva por parte de muchas economías igualmente amenazadas por peligros comunes y tasables en dinero.

Legalmente, el seguro se define como un contrato que consiste en que una de las partes (ASEGURADOR) se obliga mediante un cierto precio (PRIMA) a indemnizar a la otra (ASEGURADO) por la pérdida o daño previsto en el contrato que pueda sobrevenir al ocurrir un siniestro.

Partiendo del punto de vista funcional, el seguro es un dispositivo social mediante el cual los riesgos inciertos de los individuos pueden combinarse en grupos para convertirse así en algo más certero, y en el que pequeñas contribuciones periódicas de los individuos proporcionan fondos de los cuales se echará mano para retribuir a aquellas que hayan sufrido pérdidas.

El objeto del seguro es la compensación del perjuicio económico experimentado por el patrimonio a consecuencia de un siniestro.

### 1.1.3. La indemnización.

La indemnización es el importe que el asegurador está obligado a pagar contractualmente en caso de producirse un siniestro (figura 1.1.3.1).

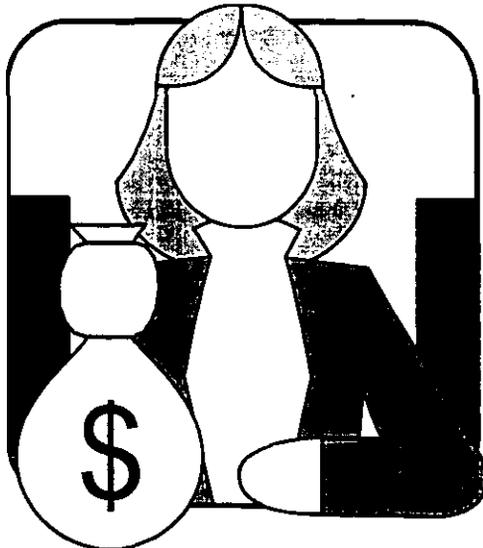


Figura 1.1.3.1 La indemnización.

Su finalidad es reponer económicamente el patrimonio del asegurado afectado por un siniestro. La indemnización está delimitada por las siguientes variables:



Deducible.	Por este medio el asegurado soportará parte de la pérdida con su patrimonio.
Sobreseguro.	Cuando el valor del objeto asegurado es superior al valor que el objeto realmente tiene.
Bajoseguro.	Cuando el valor del objeto asegurado es inferior a la suma asegurada.

#### 1.1.4. Principales riesgos cubiertos por las compañías de seguros.

##### Muerte prematura.

Su objetivo es prever la protección para nuestra familia ya que algunos gastos continúan independientemente de la desaparición de la cabeza de familia: manutención, educación, renta, deudas a largo plazo, últimos gastos, traslado de dominio, gasto de casa, etc.

##### Vejez sin recursos económicos.

La vejez sin recursos impide que el jubilado conserve su nivel de vida.

##### Invalidez.

La invalidez puede producir desajustes económicos durante un lapso que no es posible determinar.

##### Accidentes y Enfermedades.

Todos estamos expuestos a sufrir un accidente o una enfermedad, pero el seguro alivia las consecuencias económicas negativas.

Los riesgos que afectan al patrimonio suelen ser clasificados dentro de cuatro grupos generales.

Actos delictivos.	Los actos delictivos son un riesgo común causando pérdidas económicas graves.
Ordenamientos jurídicos.	El código de procedimientos civiles estipula que cuando una persona ocasiona un daño, esta obligada a reparar el perjuicio ocasionado a su conciudadano.
Riesgos consecuencias.	Cuando ocurre un siniestro podemos enfrentarnos a toda una serie de riesgos que son consecuencia del mismo.
Las propiedades materiales.	Las propiedades materiales están sujetas a riesgos porque en caso de suceder una contingencia que las afecta producirán pérdidas económicas.



### 1.1.5. Concepto de Ramo.

Ramo es el conjunto de seguros de naturaleza o características semejantes.

En la ley general de instituciones y sociedades mutualistas de seguros, en el artículo 7, determina las tres operaciones de seguros en que pueden ser autorizadas las compañías de seguros. Estas operaciones o ramos son:

Vida.	Los que tengan base del contrato, riesgos que afectan la persona del asegurado en su existencia, jubilación o retiro.
Accidentes y Enfermedades.	Los que tengan como base la lesión o incapacidad que afecte la integridad personal, salud o vigor vital del asegurado, ocasionado por accidente o enfermedad.
Daños.	Las operaciones de daños se dividen en nueve ramos: <ul style="list-style-type: none"><li>• Responsabilidad civil y riesgos profesionales.</li><li>• Marítimo y transportes.</li><li>• Incendio.</li><li>• Agrícola.</li><li>• Automóviles.</li><li>• Crédito.</li><li>• Diversos.</li><li>• Terremoto y otros riesgos catastróficos.</li><li>• Especiales (los que dicta la SHCP).</li></ul>

### 1.1.6. Elementos del contrato de seguro.

El contrato de seguro es el pacto o convenio entre dos o más partes por el que se obligan sobre una materia o cosa determinada o a cuyo cumplimiento se comprometen.

Por medio del contrato del seguro, la empresa se obliga mediante una prima a resarcir un daño o pagar una suma de dinero, al verificarse la eventualidad prevista en el contrato.

Los principales elementos del contrato del seguro son:

Asegurador.	Es la persona que asume las consecuencias dañosas, producidas por la realización del evento.
Contratante.	Es la persona que celebra con una entidad aseguradora un contrato y se obliga al pago de la prima.
Beneficiario.	Persona designada en la póliza por el asegurado o contratante como titular de los derechos indemnizatorios.
Asegurado.	Es la persona que en sí mismo o en sus bienes esta expuesto al riesgo.
Dañado.	Es la persona que a consecuencia de un siniestro sufre un daño o perjuicio.



### 1.1.7. Prima.

Es la aportación económica que deberá satisfacer el contratante o asegurado a la aseguradora, por concepto de contra presentación por la cobertura del riesgo.



Figura 1.1.3.1 La prima aportada por el asegurado.

Las tarifas se fijan tomando en cuenta la recopilación estadística con el fin de determinar un factor o tasa definitiva, la cual multiplicada por la suma asegurada nos dará como resultado la prima pura de riesgo.

A esta prima pura de riesgo hay que agregarle los costos de adquisición (comisiones de los agentes) y los costos de administración (gastos administrativos de la empresa) así como la utilidad que se espera, la desviación estadística y los gastos de redistribución de riesgos (coaseguro y reaseguro). Esta formara la prima de tarifa; a la cual deberán añadirse los impuestos correspondientes que dan origen a la prima total.

Por su situación contable las primas se clasifican en:

Prima emitida.	Corresponde al nacimiento o renovación del contrato.
Prima anulada.	Resulta de la cancelación de la póliza.
Prima cobrada.	Aquella que ha sido pagada por el seguro.
Prima por cobrar.	Aquella que esta pendiente de pago.
Prima neta emitida.	Prima emitida deduciendo impuestos, anulaciones o devoluciones.



Por su forma de pago las primas se clasifican en:

Fraccionaria.	Calculada por el tiempo de vigencia menor a un año.
Fraccionada.	Esta prima es liquidada mediante pagos periódicos más reducidos.
Anual.	Se paga cada 12 meses.
Única.	Su importe se satisface una vez por adelantado y el asegurado se libera de la obligación de pagar nuevas cantidades durante la vigencia del contrato.
Periódica.	Se satisface periódicamente. Son normalmente anualidades.

### 1.1.8. Elementos fundamentales de la póliza.

Existen diversos tipos de póliza por amplitud de los riesgos cubiertos.

- Póliza simple.
- Póliza combinada (o global).
- Póliza todo riesgo (o integral).
- Póliza individual.
- Póliza colectiva.

Las pólizas pueden dividirse en:

Condiciones generales.	Reflejan el conjunto de principios básicos que establece el asegurador para regular todos los contratos a seguir, que emita en el mismo tema o modalidad de garantía.
Condiciones particulares.	Comprende los aspectos relativos al riesgo que se asegura como: nombre, domicilio, beneficiario, giro, objetos asegurados, suma asegurada, primas, recargos, deducibles, impuestos, vigencia, etc.
Condiciones especiales.	En ella se incluirán algunas normas o convenios no usuales en el seguro como deducibles o coaseguros especiales, supresión de exclusiones, etc.
Endosos.	Documentos anexos a la póliza utilizados para modificar, ampliar o restringir las condiciones de la póliza original.

### 1.1.9. Conceptos importantes dentro del contrato de seguros.

En todo contrato de seguros debe existir:

- Interés asegurable. Uno de los fundamentos para la contratación de la póliza es la existencia de un interés.
- Bien asegurable. Es el objeto sobre el que recae la cobertura del seguro.
- Suma asegurada. Es el valor atribuido por el titular del contrato a los bienes cubiertos por la póliza.
- Valoración del riesgo. Cálculo o apreciación del valor económico de una cosa.



Existen varios tipos de valor:

Valor convenido o estimado.	Aquel que, de común acuerdo, establecen el asegurado y el asegurador.
Valor real.	El valor que tiene el bien tomando en cuenta la depreciación por uso.
Valor de reposición.	Valor del bien como nuevo.

- Alteración o agravación del riesgo. Es la situación que se produce cuando por el riesgo se vuelve más peligroso o con más probabilidad de que ocurra, causando una modificación en la prima.
- Disminución del riesgo. Cuando por el riesgo disminuye su peligrosidad o su probabilidad de ocurrencia produce una reducción en la prima.

#### **1.1.10. Metodología para la asignación de montos, tarifas y deducibles para los diferentes riesgos.**

Procedimiento general para la determinación de una tarifa lo podemos definir de acuerdo a los pasos que indicaremos en esta sección, aunque muchas veces este procedimiento puede diferir o cambiar dependiendo de la cobertura o riesgo que se está evaluando.

##### **1.1.10.1. Determinación de montos de tarifa (Cuotas).**

Dentro del sector de seguros y fianzas se manejan varios términos que identifican montos de dinero ante las autoridades correspondientes CNSF (Comisión Nacional de Seguros y Fianzas).

Dentro de este término se encuentra el llamado PRIMA, la cual representa un monto de dinero, que puede encontrarse en cualquier tipo de moneda (Dólares, Pesos Mexicanos, etc.) relacionando al riesgo que se desea asegurar o afianzar.

Existen varios tipos de primas:

Prima Pura de riesgo.	Refleja únicamente el reparto de costo neto por fallecimiento y se obtiene de la tabla de mortalidad
Prima natural ascendente.	Es aquella que su monto va aumentando conforme la edad se va incrementando.
Prima nivelada.	Esta prima fue diseñada para evitar el problema que representaba el sistema de pago de la prima natural ascendente debido a que, año con año se incrementa el costo de las primas, dicho costo puede llegar a ser prohibitivo con el transcurso del tiempo, por esto surge la prima nivelada, la cuál se comporta constante ya que es



resultado de promediar las diferentes primas puras de riesgo que el asegurado tendrá que pagar por los años en que participe del seguro.

Prima neta nivelada.

La reserva que se forma con el sistema de prima nivelada es invertida por la compañía, disminuyendo el costo de la prima por efecto de los intereses.

Prima de tarifa

Esta prima es el resultado de sumar la prima neta nivelada (P.N.N.), los gastos de adquisición (G.ADQ.) y los gastos de administración (G.ADMON.); entendiéndose como gastos de adquisición y administración los exámenes médicos, comisiones de agente, pago de luz, papelería, servicios de computo y otro más del mismo tipo que tiene la compañía.

Entonces:

Prima de tarifa = P.N.N. + G.ADQ. + G. ADMON.

Esta prima es la que se cobra a los asegurados por el riesgo contratado.

---

Cabe precisar que algunos conceptos relacionados con los cálculos de las primas son:

- Reserva. Es la diferencia que existe entre el total de primas recaudadas y el total de reclamaciones presentadas en un periodo de tiempo determinado. La reserva permite tener disponibles fondos suficientes para afrontar cualquier reclamación que se presente. Se considera la reserva como un elemento nivelador del costo del seguro o fianza.
- Dividendos. Es el derecho del asegurado a participar de las utilidades obtenidas por la compañía en la cartera de seguro de personas. El asegurado puede destinar estas utilidades en cuatro formas distintas. Aplicarlas al pago de primas, disminuyendo así su monto. Dejarlas en depósito y obtener un rendimiento adicional en efectivo o al finalizar el plazo. A un seguro saldado adicional y a un seguro temporal a un año.
- Prima anual. Es el costo que se genera de afianzar un riesgo por un periodo de un año. Esta prima se utiliza en el sector afianzador para determinar el cálculo de la reserva que se tiene que constituir bajo la supervisión de la C.N.S.F.
- Prima tiempo. Es el costo que se genera de afianzar un riesgo por un periodo mayor a un año, en el caso en que el riesgo que se desea cubrir su vigencia sea menor a un año, normalmente el sector afianzador cobra un año completo. Esta prima es utilizada en el sector afianzador para determinar lo que el Fiando o Cliente tiene que pagar por la fianza.



- Prima futura.** Es la parte de la prima tiempo que se devengará hasta el siguiente ejercicio fiscal de la compañía, es decir, es igual a prima tiempo – prima anual.
- Prima resultados.** Es la parte de la prima tiempo que se devengará en el actual ejercicio fiscal de la compañía, si el periodo fiscal comienza el mismo día que comienza la vigencia de la fianza, la prima de resultados sería igual a la prima anual, en caso contrario, será igual al monto del riesgo por la tarifa entre 365 por los días que se tenga entre la vigencia de inicio de la fianza y la vigencia de termino del ejercicio fiscal donde se ingresó la fianza.
- Prima neta.** Es el monto que se le cobrará al fiador por la expedición de la fianza. Este monto considera monto de derechos, monto de gastos de expedición, monto de bonificación e IVA.  
$$\text{Prima neta} = \text{Prima tiempo} + \text{Derechos} + \text{Gastos} - \text{Bonificación} + \text{IVA}$$
- Prima mínima.** Cantidad mínima que se cobra por concepto de prima se define una cantidad mínima a cobrar en el caso de que la prima tiempo resultante haya sido muy baja.

La parte fundamental para determinar los montos de una tarifa es contar con una base estadística confiable que contenga la emisión de pólizas por lo menos con un año de antigüedad.

Para el caso de determinar montos de tarifa de alguna cobertura de nueva creación, tendremos que suponer los datos que se requieren por el análisis, por tal motivo, es de suma importancia que la suposición de información se haga de una manera consiente y lo más cercano a la realidad pudiéndonos apoyar de datos de cualquier fuente confiable, que pudieran haber tenido una experiencia similar (otra compañía de seguros) o alguna fuente especializada en la explotación de información estadística (INEGI, Internet, etc.)

La información básica para determinar los montos (cuotas) de una tarifa es la siguiente:

- Utilidad (UT)** Técnica Este dato lo podemos determinar de acuerdo al margen de ganancia que deseamos obtener en nuestra tarifa.
- Prima cobrada (PC)** Este dato lo obtenemos de nuestra base estadística o de la suposición que hagamos considerando que en un año hayamos captado un cierto monto de dinero por concepto de la venta de la cobertura en análisis.
- Gastos de administración (G1)** Estos gastos representan toda la inversión que debe hacer la compañía de seguros para la elaboración de una póliza hasta que finalmente es presentada al cliente para su pago, como gastos de administración consideramos: luz, renta, papelería teléfono, etc.



Gastos de adquisición (G2) Estos gastos son el factor que representa la comisión del agente de seguros que se le otorga por la venta de pólizas.

Siniestros pagados (SP) Los siniestros pagados es el monto del dinero que la compañía de seguros entrega a sus clientes o asegurados por concepto de indemnización al haber sufrido un evento desafortunado (siniestro) y estuviera amparado por alguna de las coberturas que hubiera contratado en su póliza mediante el pago de la prima correspondiente.

Con la información antes definida podemos aplicarla a nuestro modelo matemático general para la integración de primas de tarifa, tal como lo mostraremos a continuación.

$$UT = \frac{PC(1-G1-G2) - SP}{PC} \dots\dots\dots(1)$$

Donde:

- UT = Utilidad técnica.
- PC = Prima cobrada.
- G1 = Gastos de administración.
- G2 = Gastos de adquisición.
- SP = Siniestros pagados.

En todas las coberturas del ramo de daños siempre debemos guardar un equilibrio el cual lo podemos mantener al utilizar nuestro modelo matemático definido en la ecuación 1.

Este equilibrio lo define el concepto denominado como siniestralidad, el cual esta determinado como el monto de siniestros pagados entre la prima cobrada.

$$\text{Siniestralidad} = \frac{\text{Siniestros Pagados}}{\text{Prima Cobrada}} = \frac{SP}{PC} \dots\dots\dots(2)$$

Podemos representar también la ecuación (1) de la siguiente manera:

$$\frac{SP}{PC} = \frac{1-(G1+G2+UT)}{PC} \dots\dots\dots(3)$$

Si combinamos las ecuaciones (2) y (3)

$$\text{Siniestralidad} = \frac{SP}{PC} = \frac{1-(G1+G2+UT)}{PC} \dots\dots\dots(4)$$

Con la ecuacione (4) nos podemos dar cuenta que la siniestralidad de alguna cobertura en particular esta determinada por los gastos de administración (G1), los gastos de adquisición (G2) y la utilidad técnica (UT).



Por ejemplo, si consideramos que la venta de la cobertura de robo de mercancías se le otorga al agente de seguros un porcentaje de comisión (G2) del 12% y tomando en cuenta lo que se indicó anteriormente, que los gastos de administración (G1) en el ramo de daños son del 15% y que la utilidad técnica (UT) en las coberturas del mismo ramo se determina con un 5%, tenemos que sustituyendo la ecuación (4) quedaría:

$$\text{Siniestralidad} = \frac{SP}{PC} = \frac{1-(0.15 + 0.12 + 0.05)}{1} = 0.68$$

Como podemos observar, el porcentaje de siniestralidad no puede ser mayor a un 68% para garantizar que la cobertura de robo de mercancía sea rentable. Los conceptos de gastos de administración (G1), gastos de adquisición (G2) y utilidad técnica (UT) no pueden variar en sus porcentajes; en determinado momento el único valor que podemos variar es el de la utilidad técnica en caso de que se quiera tener un margen mayor de utilidad.

El porcentaje de siniestralidad antes obtenido nos sirve para determinar cual es el máximo valor que debemos tener en nuestros resultados reales para que determinada cobertura sea rentable (estemos ganando).

En forma practica el resultado de la siniestralidad de cualquier cobertura esta determinada por la relación que existe de los siniestros pagados (SP) entre la prima cobrada (PC) tal y como se mostró en la ecuación (2).

Retomando nuestro ejemplo en donde calculamos que la siniestralidad máxima para ser rentable fue del 68% supongamos que el resultado de dividir los datos estadísticos de los siniestros pagados entre la prima cobrada nos arroja un porcentaje mayor al 68%.

$$\text{Siniestralidad} = \frac{SP}{PC} > 68\%$$

Analizando los factores que involucran esta relación tenemos:

- Siniestros pagados (SP). Es un dato que no podemos modificar directamente ya que representa el monto que tuvimos que desembolsar para pagar los siniestros que reclamaron nuestros asegurados. De una manera indirecta podemos llegar a disminuir este monto implementando medidas preventivas (en la medida de lo posible) que tiendan a reducir el monto en este concepto.
- Prima cobrada (PC). Este dato es reflejo de las primas que cobramos por concepto de una cobertura amparada a nuestros asegurados. El cálculo de la prima cobrada viene de la siguiente ecuación:

$$\text{Prima Cobrada} = \text{Cuota X Suma Asegurada} \dots\dots\dots(5)$$



Donde:

Cuota	Factor de riesgo determinado para cada cobertura.
Suma Aseg.	Monto de dinero determinado por el asegurado en el cual estima el valor de los bienes que esta asegurando.

Como se puede observar de nuestra ecuación (2) el valor de siniestralidad se puede disminuir en una forma directa si aumentamos el monto de la prima cobrada. Por tal motivo, si observamos la ecuación (5), se puede aumentar el monto de la prima cobrada si incrementamos la cuota, ya que la suma asegurada no puede ser aumentada porque estaríamos mintiendo en el valor de los bienes que se están asegurando. De la ecuación (2) y (5), tenemos:

$$\text{Siniestralidad} = \frac{\text{SP}}{\text{Cuota X Suma Asegurada}} \dots\dots\dots(6)$$

Por lo tanto la nueva cuota que debemos obtener para tener rentabilidad en determinada cobertura la podemos calcular de la siguiente manera:

$$\text{Cuota Nueva} = \frac{\text{SP}}{\text{Siniestralidad X Suma Asegurada}} \dots\dots\dots(7)$$

### 1.1.10.2. Determinación de deducibles.

El concepto de deducible lo podemos dividir en dos maneras diferentes:

Participación en la pérdida.	Es el monto en dinero que tendrá que desembolsar el asegurado por la reclamación de algún siniestro que le haya ocurrido y se encuentre amparado por alguna cobertura contratada en su póliza. Este monto será calculado sobre la base de un porcentaje predeterminado y el cual será aplicado al monto de los daños sufridos.
Deducible.	Es el monto en dinero que tendrá que desembolsar el asegurado por la reclamación de algún siniestro que le haya ocurrido y se encuentra amparado por alguna cobertura contratada en su póliza. Este monto será calculado sobre la base de un porcentaje predeterminado y el cual será aplicado al monto de la suma asegurada que haya definido nuestro cliente.

Como podemos observar la determinación de estos dos conceptos de deducibles se aplican en forma muy diferente, y se pueden utilizar dependiendo del caso que se presenten.

Otra forma de determinar deducibles es cambiándolos con montos mínimos y máximos, los cuales pueden aparecer de la siguiente manera:



- Deducible del 10% de la suma asegurada con mínimo de 10 días de SMGVDF (salario mínimo general vigente en el distrito federal).
- Deducible del 1% de la suma asegurada con máximo de 750 días de SMGVDF.

Estos ejemplos de deducible son muy usuales dependiendo de la situación o la cobertura de que se trate.



## 1.2. CONCEPTOS BÁSICOS SOBRE CONTABILIDAD (COBRANZA)

### 1.2.1. Que es cobranza.

Las cuentas por cobrar representan derechos exigibles originados por ventas, servicios prestados, otorgamiento de préstamos o cualquier otro concepto análogo.

Dentro del concepto general anterior se incluyen los documentos por cobrar a clientes que representan derechos exigibles, que han sido documentados con letras de cambio o pagarés.

Las cuentas por cobrar se clasifican atendiendo a su origen. Por ejemplo: a cargo de clientes, a cargo de compañías afiliadas, empleados etc. Separando a los que provengan de ventas y servicios de los que tengan otro origen.

Las cuentas por cobrar representan aplicaciones de recursos de la empresa que se transformarán en efectivo para terminar el ciclo financiero a corto plazo.

Las cuentas por cobrar a clientes incluyen una estimación para cuentas malas o incobrables para determinar el valor que se espera cobrar:

~~Cuentas por cobrar \$ 20,000.00 — Estimación de incobrables 1,800.00 = Cuentas por cobrar netas \$ 18,200.00~~

La condición para que sea considerada cuenta por cobrar, consiste en que sea cobrable dentro de un periodo de un año o dentro del ciclo normal de operaciones de la empresa.

Los documentos por cobrar, importes por cobrar por la venta de activo no circulante y otras partidas cuyo vencimiento sea mayor de un año o del ciclo de operaciones normales, deberán presentarse inmediatamente después de la sección de activo circulante, dentro de la sección de activo no circulante. Algunas empresas lo subdividen en "Clientes", "Deudores" y "Documentos por cobrar".

La cobranza constituye uno de los conceptos más importantes del activo circulante. La importancia está relacionada con el giro del negocio.

Generalmente en la empresa la inversión en cuentas por cobrar representa una inversión muy importante.

Unas medidas para conocer la inversión que se tiene en cuentas por cobrar es la relación de la inversión al capital de trabajo y al activo circulante, y se obtienen a través de pruebas de liquidez como en el ejemplo siguiente:

$$\frac{\text{Cuentas por cobrar netas}}{\text{Capital de trabajo}} = \frac{6,150}{9,014} = 68.2\%$$



El resultado de 68.2% representa la dependencia que el capital de trabajo tiene en la inversión de cuentas por cobrar y es una buena medida en la influencia que tienen las cuentas por cobrar en la estructura financiera del capital de trabajo.

$$\frac{\text{Cuentas por cobrar netas}}{\text{Activo Circulante}} = \frac{6,150}{15,620} = 39.4\%$$

El resultado de 39.4% representa la importancia que la inversión en cuentas por cobrar tiene en el total de la inversión del activo circulante.

La inversión de las cuentas por cobrar se determina por el volumen de las ventas a crédito y por el promedio de días que transcurren entre las fechas de venta y la cobranza.

Para medir la inversión en cuentas por cobrar se efectúan pruebas de liquidez. Se calcula generalmente la relación que existe entre las cuentas por cobrar y las ventas y la rotación de las cuentas por cobrar como sigue:

$$\frac{\text{Cuentas por cobrar (neto)}}{\text{Ventas netas}} = \frac{6,150}{18,420} = 33.4\%$$

El resultado representa el porcentaje de las ventas anuales que están pendientes de ser liquidadas por los clientes.

Para ser una medida correcta deben tomarse las ventas netas a crédito, las cuales están relacionadas con las cuentas por cobrar a clientes; sin embargo, como este dato resulta difícil de obtener la mayor parte de las veces, es aceptable tomar el total de ventas netas siempre y cuando no hubiera una influencia importante de ventas de contado. En algunas ocasiones en el numerador se emplea el promedio de cuentas por cobrar, aunque es más correcto usar el saldo de fin de año o periodo. Cuando las cuentas por cobrar incluyen impuesto (IVA), éste se debe deducir para que sean comparables con las ventas.

### Rotación de cuentas por cobrar

El promedio de cobranza se obtiene tomando como base el resultado de la razón "cuentas por cobrar ventas" multiplicado por los días del año (365) para obtener los días de crédito, o por 12 para obtener los meses.

$$\text{Cuentas por cobrar a ventas} \times \frac{\text{días en el año}}{\text{días cartera}} = \frac{\text{días en el año}}{\text{días cartera}}$$

$$0.357 \times \frac{365}{130} = 10.0$$

O bien:

$$\text{Cuentas por cobrar a ventas} \times \frac{\text{meses en el año}}{\text{mes en cartera}} = \frac{\text{meses en el año}}{\text{mes en cartera}}$$

$$0.357 \times \frac{12}{4.3} = 10.0$$



Esta medida es válida en empresas cuyo ciclo financiero no implica variaciones importantes en sus ventas mensuales promedio. En caso contrario el procedimiento que debe seguirse es restarle a la inversión de cuentas por cobrar el total de la venta neta del último mes, que representa 30 días, y así sucesivamente hasta dejar un residuo que debe dividirse entre las ventas del mes y multiplicarse por 30 para obtener el número de días que representa:

		Días
Cuentas por cobrar (neto)	\$ 6150	
menos:		
Venta neta del último mes	<u>2 100</u>	30
	4 050	
Venta neta del penúltimo mes	<u>3 000</u>	30
	1 050	
Venta neta del antepenúltimo mes	3 150	
	<u>1, 050</u> =	<u>10</u>
	3, 150	20

El promedio de los días que transcurren entre las fechas de la venta y la cobranza ~~dependen en parte de las condiciones económicas y de los factores controlables~~ conocidos como variables de las políticas de crédito.

La buena o mala administración de las cuentas por cobrar afecta directamente la liquidez de la empresa, ya que un cobro es el final del ciclo comercial donde se recibe la utilidad de una venta realizada y representa el flujo del efectivo generado por la operación general, además de ser la principal fuente de ingresos de la empresa base para establecer compromisos a futuro.

### 1.2.2. Que es la cartera.

Cartera es una combinación de activos, y la teoría de cartera se ocupa de la selección de carteras óptimas que proporcionen los mayores rendimientos posibles para cualquier grado específico de riesgos o el menor riesgo para cualquier tasa específica de rendimiento. Dicha teoría se elaboró en forma más amplia para los activos físicos, y, desde luego, estos conceptos son también pertinentes en la preparación de presupuestos de capital.

Supóngase que se toman en consideración N activos, en donde N es cualquier número mayor que 1. Esos activos se pueden combinar en un número casi ilimitado de carteras y cada cartera posible tendrá una tasa esperada de rendimiento de,  $E(K_p)$ , y un riesgo de  $\sigma$ . El conjunto hipotético de todas las carteras posibles se define como "conjunto factible" que podemos ver en la zona sombreada de la figura 1.2.2.

Dado el conjunto total de las carteras potenciales que se pueden construir mediante los



activos disponibles, ¿qué cartera debemos escoger? , la elección implica dos decisiones separadas:

- Determinar el conjunto eficiente de carteras.
- Elegir del conjunto eficiente la cartera que sea mejor para el inversionista individual.

Es eficiente una cartera que proporciona los rendimientos esperados más elevados que sean posibles para cualquier grado de riesgo o el grado más bajo posible de riesgo para cualquier rendimiento esperado. En la figura 1.2.2 el límite BCDE define el conjunto eficiente de la cartera. Las carteras situadas a la izquierda del conjunto eficiente no son posibles, puesto que se encuentran fuera del conjunto factible; o sea no hay un conjunto de  $K_i$  valores que proporcione una cartera con una tasa esperada de rendimiento  $E(K_p)$  y riesgos  $\sigma_p$  representado mediante un punto situado a la izquierda de BCDE.

Las carteras situadas a la derecha del conjunto eficiente son ineficientes porque alguna otra cartera podría producir rendimientos más altos con el mismo grado de riesgo o un riesgo más bajo con la misma tasa de rendimiento.

### Rendimientos esperados

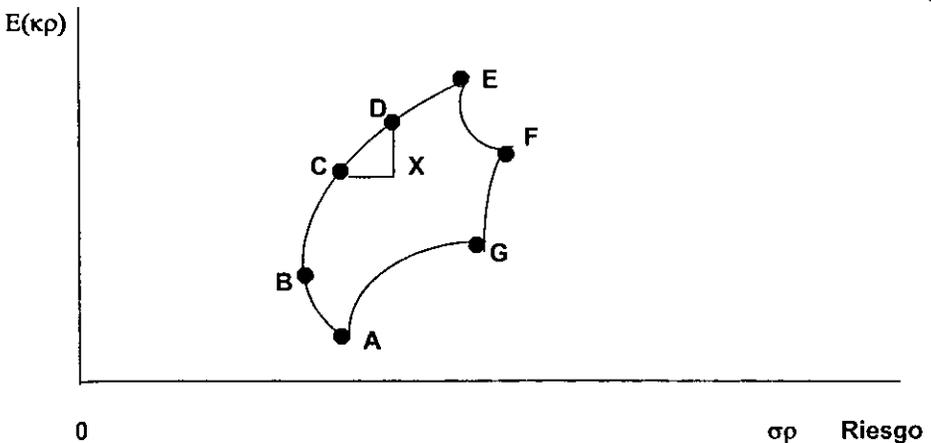


Figura 1.2.2 El conjunto eficiente de inversiones.

En la figura 1.2.2 la cartera C proporciona la misma tasa de rendimiento que la X; pero la C entraña menor riesgo. Por otra parte la cartera D es tan arriesgada como la X pero ofrece una tasa esperada de rendimiento más alta. Los puntos C y D (y otros puntos situados en el límite del conjunto eficiente entre C y D) dominan al punto X.



### 1.2.3. Que es un recibo.

Los documentos recibidos en pagos de ventas de mercancías son más deseables que las cuentas por cobrar, puesto que por lo general causan interés y representan pruebas escritas de los adeudos, una característica que les da mayor fuerza ante los tribunales, es que están firmados por el deudor. Además en la mayoría de los casos los documentos pueden descontarse más fácil y cómodamente que las cuentas por cobrar.

Un recibo de cambio y las aceptaciones comerciales son órdenes a plazo giradas a cargo de clientes y aceptadas por ellos. Un recibo por cobrar puede no representar una transición normal, sino la aceptación de un documento por liquidar o una cuenta por cobrar vencida.

Un recibo aceptado es preferible generalmente a un simple recibo por cobrar, puesto que constituye un documento aceptado por el comprador en el momento de la venta, y, por lo tanto, es más fácilmente aprobado para fines de descuento en un banco. Cuando se descuenta, este recibo es un documento con dos obligados; el aceptante tiene una obligación primaria y el girador una obligación secundaria. Un recibo aceptado se conoce como un "papel de liquidación propio", puesto que es otorgado por el cliente a cambio de mercancías que han de volver a vender en el curso ordinario de los negocios. El producto de estas ventas usualmente estará disponible para pagar la aceptación comercial al vencimiento.

#### 1.2.3.1. Tipos de recibos.

Existen tres clasificaciones de recibos :

- Anticipados.
- Vencidos.
- Amparados.

**Anticipados.** Son los recibos que son pagados antes de su fecha de vencimiento y por lo general la empresa otorga un descuento por el pronto pago.

**Vencidos.** Son los recibos que no fueron pagados en el plazo de tiempo establecido (vencimiento) y generan intereses sobre el monto establecido.

**Amparados.** Son aquellos recibos que no fueron pagados en el tiempo establecido, y que se encuentran en proceso legal para su recuperación.

#### 1.2.4. Vencimiento de documentos.

Los vencimientos es el cumplimiento del plazo de una obligación o deuda.

Los vencimientos a corto plazo son aquellos que su cobro es en el término de un año o



dentro del ciclo financiero a corto plazo de la entidad si éste es mayor a un año.

### **1.2.5. Recuperación de cartera.**

La recuperación de cartera tiene que fijar políticas, siendo las principales:

- Reducir al máximo la inversión en cuentas por cobrar en días de cartera, sin afectar las ventas y la comercialización de las mercancías o servicios.
- Administrar el crédito que se otorga a los clientes con procedimientos ágiles y términos competitivos que permitan a la empresa obtener un valor agregado.
- Para lo anterior se requiere de un sistema de información con todos los datos necesarios y actualizados, que permita tomar decisiones firmes y ágiles. El cliente debe sentirse confiado y seguro en las relaciones comerciales.
- Evaluar el crédito en forma objetiva que de como resultado una mínima pérdida por cuentas incobrables.
- Mantener la inversión en cuentas por cobrar al corriente. La inversión en cuentas por cobrar debe ser vigilada en cada cliente en particular, de conformidad a lo pactado o acordado tomando acción sobre la cartera vencida.
- Vigilar la exposición de las cuentas por cobrar ante la inflación y la devaluación de la moneda. Las cuentas por cobrar son activos monetarios que están expuestos a la inflación y las cuentas por cobrar en moneda extranjera están además expuestas a la devaluación de la moneda. Este concepto del activo circulante tiene generalmente una influencia muy importante en las utilidades o pérdidas monetarias que reportan las empresas.

### **Administración de Cartera**

En la Administración de Cartera se captan fondos de uno o más inversionistas, para ser colocados en una cartera común de inversión, administrada por una Casa de Corredores de Bolsa autorizada que se encarga de colocarlos en diferentes emisores con el objetivo de obtener una mayor rentabilidad y minimizar el riesgo del capital. Los inversionistas son propietarios de una parte proporcional de la cartera de inversión en que han sido colocados los fondos aportados. Por sus servicios, la sociedad cobra ciertas comisiones que pueden variar dependiendo de la cantidad invertida y del acuerdo pactado de manejo del fondo.

### **Ventajas de la administración de cartera**

**Diversificación de las Inversiones.** La diversificación consiste en el proceso de reducir el riesgo al distribuir los activos de los fondos entre diferentes tipos de inversión. Ello se debe a que se pueden obtener mejores resultados en cierto tipo de colocaciones bajo



ciertas condiciones económicas, y se reduce el riesgo financiero de acuerdo al dicho "no ponga todos sus huevos en una misma canasta". La canasta de títulos valores denominada portafolio de inversión puede consistir en opciones con diferentes características en cuanto a sus riesgos y sus rendimientos.

Administración Profesional de los Fondos. Otra ventaja que ofrece la Administración de Cartera es que es realizada por profesionales especializados, los cuales tienen un mayor conocimiento del Mercado de Valores, y por lo tanto, pueden realizar mejores decisiones en cuanto a en que tipo de valores invertir.

Costos Menores. Debido a que las inversiones en administración de cartera se realizan a gran escala, conjuntamente con todos los fondos aportados por varios inversionistas particulares, los administradores de cartera pueden obtener un precio relativamente más bajo para la adquisición de los títulos.

### **Las políticas de crédito**

Las políticas de crédito deben tener como objetivo elevar al máximo el rendimiento sobre la inversión. Las políticas que otorgan plazos de crédito muy reducidos, normas crediticias estrictas y de una administración que otorga o rechaza el crédito con lentitud restringen las ventas y la utilidad de manera que a pesar de la reducción de la inversión en cuentas por cobrar, la tasa de rendimiento sobre la inversión de los accionistas será más baja de la que se puede obtener con niveles más altos de venta y cuentas por cobrar.

El otorgamiento de crédito conlleva las pérdidas por cuentas incobrables, los costos de investigación del crédito del cliente, los gastos de cobranza y la financiación de las cuentas por cobrar. La investigación y operación de estos factores disminuyen la tasa de rendimiento sobre la inversión de los accionistas.

Para evaluar el crédito los administradores deben considerar: La solvencia moral del acreditado, la capacidad financiera de pago, las garantías específicas y las condiciones generales de la economía o del mercado en que opera.

La información se obtiene de varias fuentes incluyendo la experiencia anterior. Si la cuenta es nueva es una práctica normal el solicitar estados financieros auditados, declaraciones de impuestos si no se tienen estados financieros, cartas de solvencia de las instituciones de crédito con las que opera.

El volumen principal de las ventas de crédito de bienes de consumo se hacen a través de la práctica comercial de cuenta abierta. Al cliente se le abre una cuenta en donde se registran sus transacciones sin pedir un reconocimiento formal de sus deudas como pagarés o letras de cambio.

En ventas a crédito en bienes de capital o de uso duradero como pueden ser maquinaria, equipos, automóviles, vehículos, refrigeradores, televisiones, etc. La venta



generalmente se documenta con títulos de crédito que pueden ser negociados con posterioridad con instituciones de crédito. El cargar intereses por el crédito otorgado o por falta de pago oportuno en la fecha de vencimiento, es una práctica común.

### **Descuentos por pago anticipado.**

Las empresas generalmente tienen como meta capturar mercado, tener una participación mayor, lo que conlleva el otorgar un crédito similar al de la competencia. Como estrategia agresiva se puede reducir el período de crédito o, en caso extremo, a volver el negocio de "venta de contado" si se reduce el precio de venta en forma significativa y se mantiene la calidad del producto y servicio. En el país existen empresas que lo han logrado.

Es práctica común el otorgar por ejemplo el 2% de descuento cuando los clientes pagan anticipadamente sus facturas. El descuento debe tener como referencia el costo del dinero en el mercado.

Si la empresa tiene créditos con intereses, una buena estrategia financiera es otorgar a los clientes mayores a los del mercado, pero menores a lo que está pagando la empresa a terceros para poder generar un flujo de efectivo adicional a la operación normal y liquidar anticipadamente los pasivos contratados que están generando tasas de interés superiores.

Por ejemplo: La empresa otorga normalmente un plazo de crédito de 30 días fecha de factura. El descuento por pago anticipado del mercado es del 2%. La tasa mensual que paga la empresa al banco por créditos contratados es del 4%.

En este caso existe un diferencial del 2% entre el descuento por pronto pago del mercado(2%) y el interés mensual que la empresa está pagando al banco (4%). En estas condiciones el otorgar un descuento del 3% trae un beneficio a la empresa del 1% al liquidar anticipadamente los créditos contratados.

En épocas de inflación alta, la que debe considerarse da más de un dígito, otorgar descuentos por pago anticipado o por pago de contado es una práctica financiera muy aceptable. El convertir la venta en efectivo para hacer uso de éste en la compra de inventarios, representa una estrategia financiera que protege a la empresa de los efectos de la inflación y la devaluación de la moneda al estar convirtiendo activo monetario( cuentas por cobrar y efectivo) en activos monetarios (inventarios), porque los inventarios son bienes que son susceptibles de modificar su precio.

Los descuentos por pago al contado en efectivo ( no tarjeta de crédito) tienen un beneficio directo en la financiación de las cuentas por cobrar, que se compensa con la reducción del ingreso. Esto no es aplicable a los pagos con tarjeta de crédito porque conlleva un costo que cobra el banco. Los descuentos por pago de contado tienen un efecto en la promoción de ventas, incrementándose el volumen y mejorándose generalmente la rentabilidad de la empresa.



Debe estudiarse el efecto que pueda tener la demanda sobre los aumentos o disminuciones en precios y la repercusión que tendrían los aumentos o disminuciones por volumen sobre los rendimientos. Para esto deben aplicarse las fórmulas y análisis financieros de sensibilidad precio-volumen.

### **Vigilancia sobre los saldos pendientes de cobro**

El departamento de crédito tiene a su cargo vigilar y supervisar las cuentas de los clientes para que las facturas sean presentadas en su oportunidad para su cobro, y gestionar con base en las políticas de crédito el cobro de las cuentas de los clientes que están vencidas.

Cuando el cliente tenga vencida su cuenta, el departamento de crédito debe tener la autoridad suficiente para suspender los embarques futuros hasta que esté al corriente su saldo o tenga un arreglo para liquidar su adeudo.

Cuando las cuentas son importantes y los clientes reconocen su adeudo pero están en dificultades financieras temporales para hacer el pago total, con frecuencia se establecen financiamientos especiales con el fin de separar el monto que no puede ser liquidado temporalmente y mantener la cuenta al corriente. Generalmente el monto que no puede ser liquidado temporalmente se documenta y en muchas ocasiones se cargan intereses.

---

La administración de la cobranza se realiza a través de un informe conocido como "análisis de antigüedad de saldos" en el que aparecen todas las cuentas a cobrar, analizadas por su antigüedad. Actualmente con la ayuda de la computadora este informe se obtiene con cierta facilidad y oportunidad.

Una práctica necesaria y muy importante es comunicarles a los clientes el estado de su cuenta con fines de información y para que concilien su importe con su contabilidad, y así se elimine cualquier operación mal registrada u omitida y se mantenga una cartera al corriente con información oportuna y veraz.

Los estados de cuenta tienen generalmente la siguiente información:

Saldo del mes anterior, cargos por factura u otros conceptos del mes, créditos por pagos de facturas y notas de crédito, y el saldo actual. Existen estados que también muestran el monto de la antigüedad de las facturas pendientes de pago.

Una práctica de control interno es que los estados de cuentas mensuales de los clientes sean manejados y controlados por el departamento de auditoría interna, quien enviará a los clientes con una leyenda que se conoce como de confirmación negativa: "Si en un término de 15 días no se recibe comunicación alguna sobre el saldo de esta cuenta, se considera que es correcta".

En el caso de observaciones, el cliente debe enviar su comunicación directamente al departamento de auditoría interna, generalmente con porte pagado a una casilla de



correos destinada para tal objeto.

Con el análisis de antigüedad de saldos se tendrá información importante para poder juzgar y establecer la provisión para las cuentas de cobro dudoso y así poder presentar una valuación adecuada de las cuentas por cobrar de conformidad a principios de contabilidad.

Es necesario conocer sobre la situación que guardan las cuentas vencidas a más de 120 días. Periódicamente el departamento legal debe dar un dictamen sobre la recuperabilidad de las cuentas que tienen para su gestión de cobro.

La provisión para las cuentas de cobro dudoso cuando el número de clientes es numeroso, se estima generalmente aplicando porcentos de pérdida a las cuentas vencidas de diferente antigüedad y sumando las que se consideren de muy difícil recuperación. La experiencia obtenida en los años anteriores es un factor importante para considerarse en el cálculo.

#### **1.2.6. Aspectos Legales.**

##### **Las políticas de cobro.**

Generalmente está política es muy variable y está condicionada al mercado y giro del negocio en que opera la empresa.

En condiciones normales en la primera semana de vencido el recibo se le hace un llamado al cliente, a los 15 días se le envía un recordatorio amable, a los 30 días un nuevo recordatorio más enérgico y durante el mes siguiente se puede gestionar la cobranza a través de un representante de la empresa. Si el cliente no liquida su adeudo, éste debe turnarse al departamento legal para su cobro.

El proceso de cobro puede ser costoso pero se requiere de firmeza para no prolongar la gestión de cobro y reducir al máximo las pérdidas por cuentas incobrables

Debe tomarse en cuenta el costo de la cobranza tramitada a través de procedimientos legales o con arreglos a que se puedan llegar para que el cliente liquide su adeudo.

En caso de que el cliente se atrase en su pago, deben conocerse las razones reales de la morosidad. Si el cliente tiene capacidad de pago hay que emplear procedimientos de cobro enérgicos.

No debe concederse más crédito a un cliente que deje de cumplir sólo por que se pensó que era un riesgo aceptable.



## 1.3. TEORÍA DE BASE DE DATOS RELACIONAL

### Conceptos básicos

**Relación** es un término matemático para representar una tabla bidimensional, como una colección de registros, se define mediante filas y columnas. Cada columna de una relación contiene los valores de un atributo con nombre y cada fila se llama tupla. Si se trata de  $n$ -tuplas, esto es, si la tabla tiene  $n$  columnas, entonces la relación es de grado  $n$ . El número de filas o tuplas de una relación es su cardinalidad, y el número de columnas es su grado. Los elementos individuales de una relación son valores de atributo. Si consideramos una relación  $m \times n$  ( $m$  filas y  $n$  columnas), tendremos una relación de grado  $n$  y cardinalidad  $m$ , esto es una relación con  $n$  columnas y  $m$  tuplas.

Estas relaciones son matrices rectangulares que pueden ser descritas matemáticamente, estas poseen las siguientes propiedades generales:

- Cada entrada de la tabla representa un ítem de datos, no hay grupos repetitivos.
- Son homogéneas por columna, es decir, todos los ítems de una columna son de la misma clase.
- Cada columna tiene nombre propio.
- Todas las filas son diferentes, no se admiten filas duplicadas.
- Tanto las filas como las columnas pueden ser consideradas en cualquier secuencia y en cualquier momento, sin afectar por ello el contenido de información ni la semántica de cualquier función que utilice la tabla.

**Extensión** es el conjunto de tuplas de la relación en un determinado momento, por tanto, varía con el tiempo, pues se van insertando o borrando con el tiempo las tuplas. La extensión se encuentra en los datos de la base de datos.

**Intención** se refiere a las características permanentes de una relación como son el nombre de la relación o los nombres de los atributos, pero con las restricciones de integridad pertinentes, la intención se encuentra en el esquema conceptual de la base de datos, y esta define todas las extensiones permisibles.

**Base de datos relacional** es un conjunto de relaciones normalizadas que varían con el tiempo (las extensiones de las relaciones cambian con el tiempo), de diversos grados. En otras palabras, una base de datos relacional es una base de datos para la cual los operadores a disposición del usuario operan sobre estructuras relacionales. No significa que por fuerza los datos estén almacenados en forma de tablas físicas.

**Tupla** debe estar asociada con una clave que permita su identificación, a veces la tupla se identifica mediante un único atributo, en otras ocasiones hay que recurrir a más de un atributo para lograr la identificación inequívoca de una tupla, por tanto la clave debe cumplir con dos requisitos:

- Identificación unívoca: En cada tupla de una relación el valor de la clave debe identificar unívocamente esa tupla.



- **No redundancia:** Ningún atributo de la clave podrá ser descartado sin destruir la propiedad de identificación unívoca.

En cada tupla puede existir más de un conjunto de atributos capaz de satisfacer estos dos requisitos, estos conjuntos son llamados claves candidatas o llaves candidatas.

**Clave primaria** o llave primaria es la clave candidato elegida para identificar una entidad como única, si hay solamente una clave candidata para una relación, entonces esta clave debe ser la clave primaria. Una restricción del modelo relacional es que una clave primaria no puede tener valores nulos, porque un valor nulo no puede identificar una tupla, Date llama a esto integridad de entidad.

**Clave primaria** si un atributo o una colección de atributos de una relación o entidad contiene el valor de clave primaria de otra relación, a ese atributo se llama clave externa o llave externa, estas claves externas sólo pueden tener dos valores posibles, el valor de la clave primaria pertinente o un valor nulo. Esto se conoce como integridad referencial porque los valores se obtienen por referencia a los valores de clave primaria del propietario.

### **Regla de integridad 1 (integridad de la entidad)**

Ningún componente de un valor de una clave primaria puede ser nulo. Todas las entidades deben ser distinguibles por definición, es decir, deben tener alguna identificación única de alguna clase. Las claves primarias realizan la función de identificación única en una base de datos relacional, un identificador (clave primaria) nulo sería una contradicción de términos, porque equivaldría a decir que hubo alguna entidad que no tuvo ninguna identificación única, por tanto no era de otras entidades.

### **Regla de integridad 2 (integridad de referencia)**

Introduciremos primero la noción de dominio primario, un dominio específico puede designarse como primario si y sólo si existe alguna llave primaria de un solo atributo definida sobre ese dominio. Cualquier relación que incluya un atributo que se defina sobre un dominio primario debe obedecer a la restricción de la regla de integridad referencial, donde  $D$  es un dominio primario, y  $R_1$  una relación con un atributo  $A$  (el cual puede ser una clave externa) que se define sobre  $D$ . Entonces, en cualquier instante dado, cada valor de  $A$  en  $R_1$  debe ser o bien nulo, o bien igual a  $V$ , por ejemplo, donde  $V$  es el valor de la llave primaria de alguna tupla de alguna relación  $R_2$  ( $R_1$  y  $R_2$  no son por fuerza distintas) con llave primaria definida sobre  $D$ .

$R_2$  debe existir por definición de dominio primario, también la restricción se satisface de modo trivial si  $A$  es la llave primaria de  $R_1$ .

### **Normalización**

Es necesario normalizar las relaciones para minimizar la redundancia de los datos, un dato sin normalizar es redundante, la redundancia de datos causa problemas de



integridad, las transacciones pueden no ser consistentes en todas las copias de los datos causando inconsistencia en cualquier base de datos, la normalización ayuda a identificar entidades y relaciones faltantes.

La normalización es un proceso reversible paso a paso para transformar progresivamente una relación no normalizada, en relaciones con una estructura más simple. Como el proceso es reversible, no se pierde ninguna información durante la transformación. Inicialmente se definieron tres tipos de propiedades indeseables – agrupaciones de datos, dependencia parcial de clave y dependencia indirecta de clave – estas pueden eliminarse en tres etapas llamadas primera forma normal (1NF), segunda forma normal (2NF) y tercera forma normal (3NF), actualmente existe también una cuarta forma normal (4NF) e incluso una quinta forma normal (5NF).

Se llama relación no normalizada cuando contiene agrupaciones de datos. En cada etapa de normalización una relación se descompone en otras varias, sin embargo, es posible que una relación esté en tercera o incluso en forma superior al comenzar. Como este proceso es sucesivo, una relación en la  $n$ -ésima forma normal está también en la  $(n-1)$ -ésima forma normal (Figura 1.3.1).

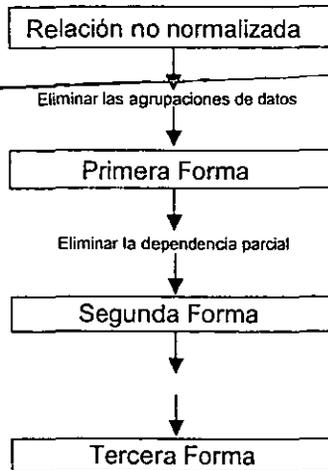


Figura 1.3.1 Tres niveles de normalización

### Dependencia funcional

La expresión funcionalmente dependiente es del siguiente modo: El atributo B de una relación R es funcionalmente dependiente del atributo A de R si, en cada instante, cada valor de A está asociado con no más de un valor de B dentro de la relación R.

Decir que B es funcionalmente dependiente de A es equivalente a decir que A identifica a B, esto quiere decir que si en cualquier instante es conocido el valor de A, el valor de B queda determinado.



## **Dependencia funcional completa**

Un atributo o colección de atributos B, de una relación R es dependiente funcional completo de otra colección de atributos A, de la relación R, si B es funcionalmente dependiente del total de A pero no de ningún subconjunto de A.

## **Primera forma normal**

Una relación está en primera forma normal si cada uno de sus componentes es atómico, la finalidad de la primera forma normal es simplificar la estructura de una relación, asegurando que contenga solamente datos elementales y no agrupaciones de datos.

Cualquier relación normalizada está en 1NF, es decir, una relación en 1NF que, además no esta en 2NF y por tanto, tampoco lo está en 3NF.

## **Segunda forma normal**

Una relación R se halla en la segunda forma normal si está en la primera forma normal y cada uno de sus atributos no primos es dependiente funcional completo de cada clave candidata.

## **Tercera forma normal**

Una relación R se halla en la tercera forma normal si se encuentra en la segunda forma normal y cada uno de los atributos no primos son dependientes no transitivos de cada clave candidata R.

Suelen presentarse algunas anomalías aun cuando la relación se halle en la segunda forma normal. Para eliminarlas, se recurre a un último paso de normalización, con el que se va de la segunda a la tercera forma normal. Con este paso se elimina lo que se llama la dependencia transitiva.

Una forma normal n-ésima óptima se define como el número mínimo de relaciones en esa forma normal para la relación original no normalizada.

El concepto de tercera forma normal tuvo un gran impacto en áreas de procesado de datos más allá del modelo relacional, especialmente en análisis de datos. Se considera que las principales ventajas de la tercera forma normal son la fácil identificación de entidades y claves, y la conversión de estructuras complejas de datos en formas más simples. Un tipo de registro, desprovisto de grupos de repetición, pero que tiene una clave única, está en tercera forma normal si cada dato elemental fuera de la clave es identificable:

- Por la clave
- Por la clave completa
- Y por nada mas que la clave



## Álgebra Relacional

Una operación del álgebra relacional tiene como operandos una o más relaciones y las manipula de modo que forman una nueva relación. Con el fin de obtener un resultado determinado, el usuario especificará cierta secuencia de operaciones de esta clase.

La notación original sobre álgebra relacional que utilizó Codd fue principalmente para transmitir los conceptos más que para proporcionar una sintaxis para la implementación. Sin embargo la notación que se utilizará será la del álgebra basada en una implementación llamada PRECI **Algebraic Language** (Lenguaje Algebraico) (PAL), esto es porque la notación original de Codd a veces da impresiones equivocadas respecto a su implementabilidad.

Las operaciones se expresan en un sistema de notación y pueden utilizarse para recuperar información de una o más relaciones, o para actualizar una tupla de una relación. Describiremos siete operaciones; las tres primeras, unión, intersección y diferencia, son operaciones clásicas de conjuntos, y las otras cuatro, proyección, selección, join (salto) y división son menos comunes. Las siguientes operaciones se evalúan de izquierda a derecha, con el siguiente orden de precedencia:

La parte derecha de los operadores de comparación = < ≥ > ≠

- |   |   |                                    |
|---|---|------------------------------------|
| • Operadores de comparación.                      | } | NOT                                |
| • Operadores lógicos<br>(en orden de precedencia) |   | AND                                |
|   |   | OR                                 |
| • Selección                                       |   | :                                  |
| • División  |   | /                                  |
| • Reunión (join)                                  |   | *                                  |
| • Proyección                                      |   | %                                  |
| • Intersección                                    |   | !                                  |
| • Unión y diferencia                              |   | ++--                               |
| • Asignación                                      |   | :=(no es una operación relacional) |

También pueden utilizarse paréntesis para indicar un orden diferente de prioridad en las operaciones.

La **unión** de la relación A con la relación B, indicada por A++B, es el conjunto de todas las tuplas (en A y B) sin repetición. Puede utilizarse para añadir una nueva tupla a una relación.

La **intersección** de la relación A con la relación B, indicada por A!B, es el conjunto de todas las tuplas que pertenecen a ambas. Ésta puede utilizarse para encontrar un conjunto de tuplas común a ambas relaciones.

La **diferencia** entre las relaciones A y B, indicada por A--B, es el conjunto de todas las



tuplas que pertenecen a A pero no a B. Ésta puede utilizarse para eliminar una tupla. Como la unión, una operación de diferencia puede aplicarse sólo si las relaciones son compatibles para la unión.

**Proyección** es la extracción de una o varias columnas con nombre de una relación, en un orden especificado. Hablando con rigor, en todas las operaciones relacionales las tuplas duplicadas han de ser eliminadas de la relación resultado, pues se supone que una relación no tiene tuplas duplicadas. Sin embargo, la eliminación de duplicados implica un procesado extra, excepto a veces que la presencia de duplicados en la respuesta no tiene importancia. Por eso se utilizan dos tipos de proyección; uno se llama proyección indicada por %, y el otro proyección única, indicada por %%. Los duplicados se eliminan de la proyección única, pero no de la proyección.

**Reunión** La reunión de dos relaciones R y S según una cualificación Q multi-atributo es el conjunto de las tuplas del producto cartesiano  $R * S$  que satisfacen la cualificación Q. Existen varios casos particulares de la reunión:

La equi-reunión de R y S sobre los atributos  $A_i$  y  $B_m$  es la reunión según la cualificación  $A_i = B_j$

La  $\Theta$  reunión de R y S sobre los atributos  $A_i$  y  $B_j$  es la reunión según la cualificación  $A_i \Theta B_j$ .

La auto-reunión de R sobre el atributo  $A_i$  es la reunión de R consigo misma según la cualificación  $A_i = A_i$ .

La reunión natural de R y S es  $R \bowtie S$ , es la equireunión de R y S sobre todos los atributos que tienen el mismo nombre en R y en S, seguida de la proyección que permite conservar uno solo de estos atributos iguales del mismo nombre.

**Selección** se utiliza para extraer tuplas de una relación sujeta a condiciones especificadas en un predicado y por lo tanto es algo complementario a la proyección que extrae columnas. La forma de la operación es Relación : Predicado que se interpreta como "seleccionar aquellas tuplas cuyo predicado es verdadero". Los dos puntos significan "donde" o "tales que"

**División** el operador de división divide una relación dividiendo A de grado  $m+n$  entre una relación divisor B de grado n, y produce una relación resultado de grado m. El  $(m+1)$ -ésimo atributo de A y el l-ésimo atributo de B (l en el rango de 1 a n) deben estar definidos sobre el mismo dominio. Considérense los primeros m atributos de A como un solo atributo compuesto X, u los últimos n como otro, Y; entonces A puede considerarse como un conjunto de pares de valores  $\langle x,y \rangle$ . Asimismo, B puede considerarse como un conjunto de valores simples  $\langle y \rangle$ . Entonces el resultado de dividir A entre B es el conjunto de los valores x tales que el par  $\langle x,y \rangle$  aparece en A para todos los valores y que aparecen en B. Los atributos del resultado tienen los mismos nombres con calificación que los primeros m atributos de A.

### Cuestiones de implementación

El modelo relacional proporciona una técnica para describir datos de forma lógica, junto



con un conjunto de potentes lenguajes de alto nivel para manipular los datos, pero sin especificar ninguna arquitectura o estrategia de almacenamiento. El modelo puede ser visto a dos niveles: en primer lugar como una facilidad de lenguaje relacional con un esquema externo relacional por encima de otros modelos de datos y, en segundo lugar, como un modelo de datos independiente con una arquitectura definida por el implementador y facilidades que la acompañan.

### **Cuestiones de arquitectura**

El modelo relacional puede construirse para soportar registros de entidad y asociaciones para un esquema conceptual. Sin embargo, el modelo original no incluía ninguna técnica para las declaraciones de privacidad e integridad, y aunque las investigaciones subsiguientes han producido algunas construcciones efectivas, no hay ninguna especificación única coherente que pueda describirse como el enfoque relacional.

Como el modelo relacional no especifica ningún esquema de almacenamiento, tal esquema está definido por el implementador. Una implementación eficiente que armonice la potencia de un lenguaje relacional requiere la solución de una serie de problemas relacionados con la optimización.

~~Mientras que el modelo relacional original no hizo ninguna distinción entre un esquema conceptual y un esquema externo, los trabajos más recientes han producido algunas facilidades de aplicación, generalmente conocidas como "vistas".~~

Una atracción especial del modelo relacional es la independencia de datos que proporciona, separando las relaciones y los lenguajes relacionales de las estructuras de almacenamiento.



# **CAPÍTULO II**

## **METODOLOGÍA DE DESARROLLO**



## 2.1. METODOLOGÍA DEL DISEÑO DE LA BASE DE DATOS RELACIONAL.

Es sencillo diseñar una base de datos, pero a menudo hay que reconsiderar posteriormente la estructura de los datos, lo cual ocasiona retrasos y modificaciones. Es más lento la obtención de un diseño que sea óptimo en lo posible, pero el tiempo invertido se recupera al no tener que volver atrás para replantearse el diseño de los datos. Un buen diseño es la clave para iniciar con buen pie el desarrollo de una aplicación basada en una base de datos o la implementación de un sistema.

Es de destacar la importancia de un buen diseño. Un diseño apresurado o simplemente bosquejado puede mostrarse inservible o muy mejorable cuando la aplicación ya está parcialmente codificado, o el administrador de la base de datos ya tiene organizados el mantenimiento y el control de acceso a los datos.

**Esquema:** diseño general de la base de datos a nivel lógico. Incluye el tipo de datos y las relaciones entre ellos. Es de naturaleza fija y solo se altera excepcionalmente. El esquema se define y se mantiene utilizando el lenguaje de definición de datos (DDL).

**Instancia:** contenido concreto de la base de datos en un momento dado. Varía con el tiempo, al añadir, eliminar o modificar datos, utilizando el lenguaje de modificación de datos (DML).

El diseño de una base de datos se realiza a dos niveles. El primero es el **nivel conceptual**, en la cual se contempla una estructura abstracta y no implementable directamente. El segundo es el **nivel físico**, en el cual la base de datos es ya implementable. Detalladamente, las fases del diseño de una base de datos son las siguientes:

- Descripción en lenguaje natural.
- **Diagrama Entidad-Relación (E-R).** También conocido como "diagrama de Chein". Estos diagramas modelizan el problema mediante entidades asociadas por relaciones. Adoptan la forma de grafos donde los datos se relacionan mediante flechas. El diagrama E-R no depende del modelo de datos.
- Elección del modelo de datos (usualmente el relacional)
- Conversión del diagrama E-R al modelo relacional (tablas)
- **Normalización** (eliminar diversos defectos de diseño).
- **Optimización** (según criterios de almacenamiento interno, como el espacio en disco y el tiempo medio de acceso).

Las tres primeras fases pertenecen al nivel conceptual del diseño de bases de datos mientras que las tres últimas se relacionan con el nivel físico.



### 2.1.1. Elección de la metodología óptima.

**Modelo de datos:** estructura general de los datos y técnicas de acceso. Hay tres modelos de datos posibles:

- Chein
- Mericce
- Yourdon

#### Metodología de Chein

En la metodología de Chein se utilizan diagramas de contexto además su modelo ambiental dice que cualquier sistema tiene relación con el medio ambiente en que se sitúa. Esta interrelación dificulta la versión del sistema que se está desarrollando, por lo tanto es importante definir con claridad, ¿Qué forma parte del sistema?, y, ¿Qué forma parte del medio ambiente?.

Podemos decir que el modelo ambiental define la frontera entre el sistema y el resto del mundo. La definición de los límites del sistema comienza con la elaboración del objetivo y del análisis del sistema. Dentro del análisis del sistema es necesario determinar: ¿Qué es parte del sistema y qué no?

Además de determinar que está en el exterior y que está en el interior ( que se logra definiendo la frontera y el medio ambiente), también se necesita saber qué información entra al sistema del exterior y cuál es la que produce como salida.

Otro aspecto importante del modelo ambiental es identificar los eventos que ocurren en el ambiente al cuál debe responder el sistema. Sólo se deben considerar los eventos que ocurren en el medio ambiente exterior y que requieren respuesta del sistema.

La frontera entre un sistema y su medio ambiente es arbitraria, sin embargo, puede considerarse alguna política administrativa, negociaciones políticas u otras estrategias para definir una frontera.

Comúnmente, se tiene una idea de la frontera general entre el sistema y el ambiente, pero a menudo se dice que existe un área gris que es la que está sujeta a consideraciones, se trata de un área sobre la cuál pueden presentarse los siguientes casos:

- No hay incertidumbre.
- No se ha pensado.
- Hay ideas preconcebidas que hay que reflexionar.

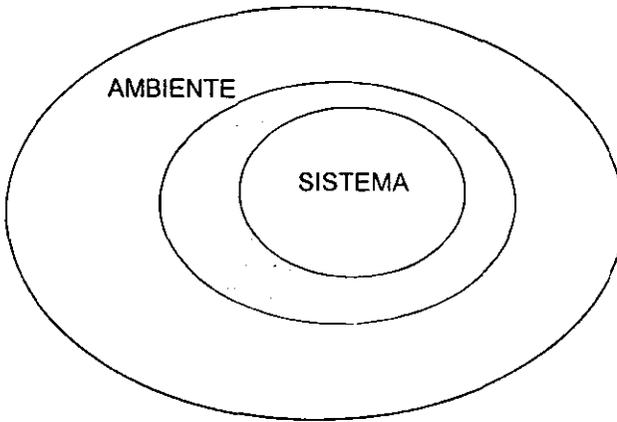
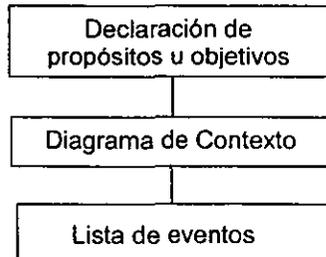


Figura 2.1.1 Área gris entre el sistema y el ambiente

Los componentes del sistema ambiental son:



- **Objetivo.** Es el primer componente del modelo ambiental y es una declaración textual breve y concisa del propósito del sistema ( puede constar de una o varias frases, jamás debe llegar a más de un párrafo ya que no es una descripción detallada), dirigida al nivel administrativo superior, la administración de los usuarios, y otros que no están directamente involucrados con el desarrollo del sistema.
- **Lista de eventos.** Es una descripción de los estímulos que recibe el sistema y que provoca una relación en él.

Existen dos tipos de eventos: Tipo de Flujo de Control y Tipo flujo temporal.

Los eventos orientados a flujo se identifican como aquellos que proporcionan datos al sistema y que al ser captados hace que realice algún proceso de estos eventos identificados con (F).

Los eventos de tipo temporal, son aquellos que periódicamente proporcionan



información a las entidades con las cuales se relaciona el sistema, estos eventos se marcan con (T). Debe notarse que los eventos temporales no se inician con flujos de datos de entrada; puede imaginarse que el sistema tiene un reloj interno con el cuál puede determinar el paso del tiempo, sin embargo debe considerarse que también un evento temporal podrá requerir que el sistema solicite entradas de una o más terminales y podrán asociarse uno o más flujos de datos con un evento temporal, aunque los flujos de datos en sí, no representan el evento mismo.

Los eventos de control deben considerarse un caso especial del evento temporal. A diferencia de un evento temporal normal, el evento de control no se asocia con el paso temporal del tiempo, por lo que el sistema no puede anticiparlo utilizando un reloj interno. Y a diferencia de un evento de flujo normal, el de control indica su presencia con el arribo de los datos.

- **Diagrama de Contexto.** Este es el siguiente paso del modelo ambiental y es un caso especial del diagrama de flujo.

Los terminales se representan con rectángulos en el diagrama de contexto. Se comunican con el sistema a través de los flujos de datos o de control.

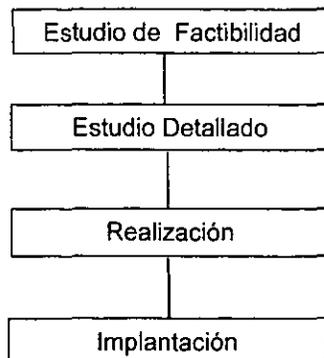
Los flujos que aparecen en el diagrama de contexto se incluyen para:

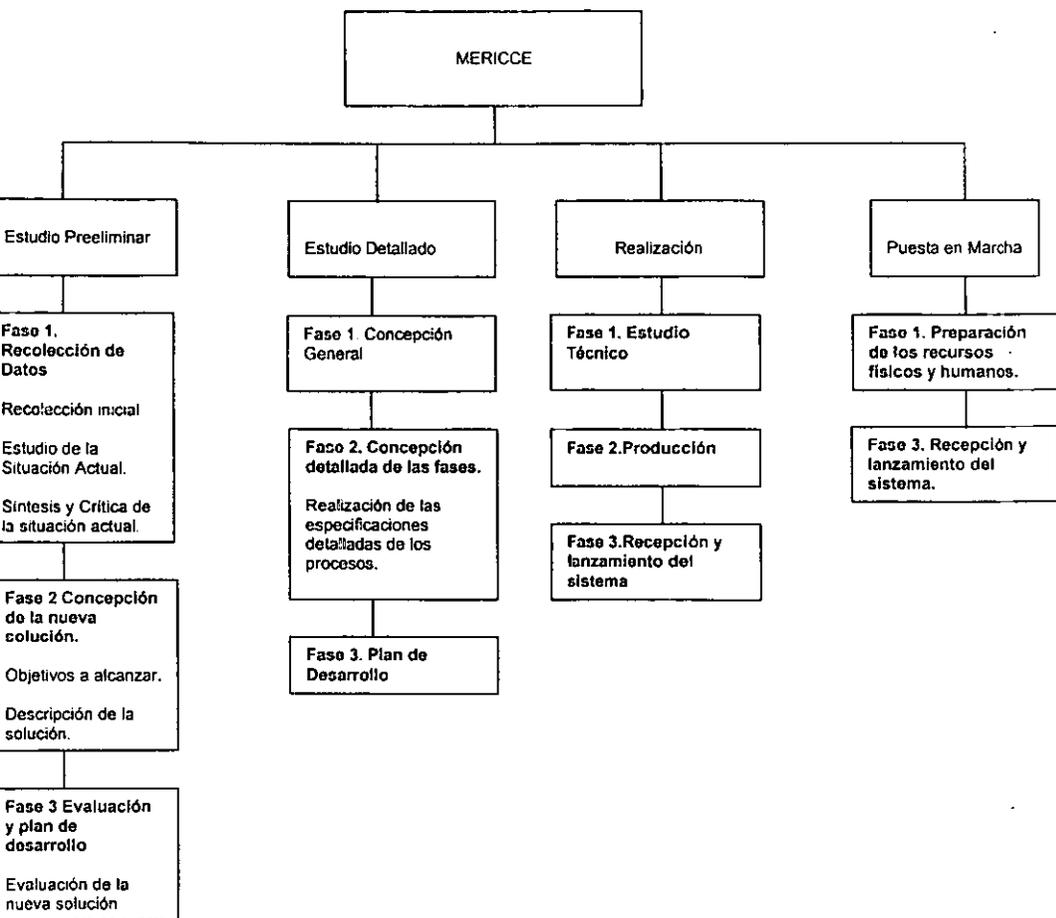
- Detectar un acontecimiento en el ambiente en que deba responder el sistema, o si se ocupan para producir una respuesta.
- Ilustrar los datos que son transportados entre los terminales y el sistema.

### Metodología de Mericce

La metodología de Mericce surge de una petición de la Administración Pública para el diseño y desarrollo de sistemas informáticos en Francia en 1997. Esta metodología utiliza el concepto de diagrama entidad – relación.

Las etapas para el desarrollo de un proyecto según Mericce son:





Podemos hacer una descripción del sistema en tres niveles:

- **Conceptual.** En este nivel se tratan básicamente los datos y la manera de manejarlos. La descripción de los datos reflejará la información existente en su entorno y las relaciones entre ellas. La manera de manejarlos como su nombre lo indica, se refiere a los procesos que se realizarán con los datos así como su secuencia en el tiempo.
- **Lógico u organizativo.** En este nivel se desglosan las operaciones descritas en el nivel conceptual en procedimientos funcionales, es decir, en tareas realizadas sucesivamente en un puesto de trabajo; en este nivel se definen las tareas que hay que realizar. Se realizan estructuras lógicas de datos para su tratamiento por el sistema.



- **Físico u operativo.** En este nivel se indica como hay que realizar las actividades indicadas. Se detallan las indicaciones para todo tipo de procedimientos así como las normas para realizarlos correctamente.

Mericce utiliza algunas técnicas para modelar los elementos ( datos y manejo de ellos) a todos niveles:

- Representación gráfica de diagramas de flujo.
- Modelado de datos(conceptual, lógico, físico).
- Modelado de tratamientos(conceptual, lógico y operacional).

Utilización de grupos de trabajo para el desarrollo del proyecto. Mericce realiza la descripción de los grupos de trabajo, es decir, el personal implicado en el desarrollo del sistema a todos niveles, habrá entonces:

- **Comité director.** Iniciará los objetivos a cubrir por el sistema en las primeras fases de concepción.
- **Comité de usuarios.** Aportará el punto de vista de los futuros utilizadores del mismo.
- ~~Grupo de desarrollo.~~ Es el personal técnico encargado de satisfacer las necesidades a través del sistema informático.

Para lograr el objetivo deseado, Mericce se apoya en los siguientes principios generales:

- Desglose del desarrollo en etapas.
- Definición de los documentos estándar de cada etapa.
- Uso del modelo entidad relación y sus principios para la representación de datos.
- Uso de las redes de Petri para la representación de procesos y tratamientos.

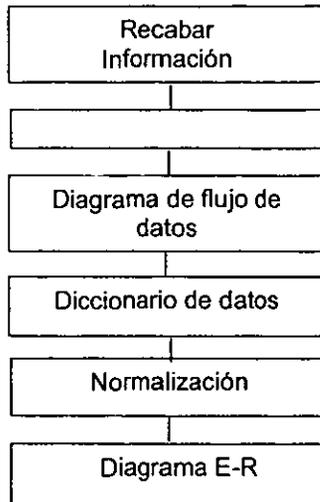
Las redes de Petri, es un método de representación muy completo ya que por cada acción – operación a realizar determina:

- Los sucesos que desencadenan la ejecución de procesos.
- Bajo que condiciones de esos sucesos se ejecuta el proceso.
- En qué consiste el proceso, es decir, que operaciones hay que realizar.
- Quien ejecuta el proceso.
- Resultados obtenidos o salidas del proceso.



- Definición de grupos de trabajo y reparto de las responsabilidades y funciones a lo largo del desarrollo.
- Especificación del reparto de tareas y tratamientos entre los usuarios y ordenador.
- Definición de los flujos de información entre las unidades del sistema.

### Metodología de Yourdon



### Diagrama de Contexto

El diagrama de contexto es un caso especial del diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema.

Para llegar al objetivo planteado y realizar las funciones necesarias se requiere que el sistema interactúe con otros sistemas u organizaciones, los cuales son llamados terminadores o entidades y están representados en el diagrama de contexto por rectángulos, dentro de los cuales se coloca el nombre de la entidad a la que

representan. Los datos que proporciona cada una de las entidades se muestran mediante flechas que representan a su vez los principales flujos de entrada y salida del sistema.

La parte más difícil de un diagrama de contexto es el proceso, que consiste de un círculo en un diagrama de flujo de datos. El nombre de tal proceso suele ser el nombre completo del sistema o un acrónimo convenido. Los nodos terminales se representan con rectángulos, los cuales se comunican con él a través de flujos de datos o de control, y no deben comunicarse directamente entre sí, esto es incorrecto dentro de un diagrama de contexto.



Los nodos terminales deben tener las siguientes características:

- Amplio número de entradas y salidas. Para evitar un diagrama saturado, conviene dibujar el nodo terminal más de una vez y ser marcado por un asterisco o una diagonal.
- Si el nodo terminal es una persona individual es preferible indicar el rol que tiene más que su entidad.
- Es importante distinguir entre almacén de datos (bases de datos y archivos) y manejadores de bases (mecanismos, dispositivo o medio físico para transportar datos hacia dentro o fuera del sistema) de datos.
- Los flujos de datos en el diagrama de contexto se incluyen para detectar un acontecimiento en el medio ambiente en que deba responder el sistema, o si se ocupan (como datos) para producir una respuesta y para ilustrar datos que son transportados entre los terminales y el sistema.
- El diagrama de contexto de un modelo debe evitar (hasta donde sea posible) mostrar los manejadores cercanos a la implementación que introducen y sacan datos del sistema. Además tampoco deberá mostrar los mensajes y medios específicos de coordinación que el sistema y los terminadores pasan entre sí ~~para indicar que están entre sí las entradas o salidas~~.

### Tipos de flujo de datos

- Flujo de datos de entrada. Son aquellos que el sistema requiere para un determinado proceso y que fluyen de una entidad externa.
- Flujo de datos de salida. Son los datos que el sistema produce como resultado del procedimiento de datos de entrada y control, y que fluyen hacia una entidad externa al sistema.
- Flujo de control. Puede considerarse como un flujo de datos binario. Está encendido o apagado y puede cambiar de un estado a otro en cualquier momento, señalando así al sistema que se necesita tomar alguna acción inmediata.

### Diagrama de flujo de datos

El diagrama de flujo de datos (DFD) es una de las herramientas gráficas de modelado más importante del análisis estructurado que permite visualizar un sistema como una red de procesos funcionales conectados entre sí por "conductos y tanques de almacenamiento" de datos y constan de cuatro componentes.

- El proceso indica la manera que una o más entradas se transforman en salidas. Se representan por medio de un círculo.
- El flujo, que se usa para describir el movimiento de bloques o paquetes de



información de una parte del sistema a otra. Se representa como una flecha que entra y otra que sale de un proceso.

- El almacén que se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por un rectángulo con los vértices curvados.
- El terminador que representa entidades externas con las cuales el sistema se comunica como rectángulo.

El DFD es usado principalmente en sistemas operacionales, en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que este maneja.

### Diccionario de Datos

Para realizar el objetivo de tener los datos correctos, redundancia mínima y control del uso de los datos, es indispensable un mecanismo central de control. Este mecanismo nos da la posibilidad de controlar y manejar la información sobre los datos en las fases de diseño, implantación, operación y expansión de una base de datos, dicho mecanismo recibe el nombre de diccionario de datos.

El diccionario de datos almacena información sobre los datos relativos al origen de estos, descripción, relación con otros datos, uso, responsabilidad y formato. Es la misma base de datos la que almacena datos sobre datos. El diccionario es una guía y contiene el mapa de la ruta hacia la base de datos.

Es aconsejable comenzar a recabar la información sobre los datos en un diccionario de datos el mismo día en que el proyecto se hecha a andar. El diccionario de datos se puede usar para almacenar la información sobre los campos de los datos en un lugar central, para establecer una comunicación efectiva entre todas las partes involucradas.

Los dos objetivos básicos de un diccionario de datos son la administración y el control de los datos como un recurso, en un lugar central, a través de las fases de diseño, realización y operación, así como el establecimiento de una comunicación efectiva entre todos los que estén interesados en la base de datos.

El diccionario de datos (DD) también se puede usar en un medio ajeno a la base de datos, la información almacenada en este es sobre los datos almacenados en la base, mientras que en un medio ajeno a una base de datos, la información almacenada en el DD es la relativa a los archivos de datos.

Es necesario instalar software para crear y manejar el diccionario de datos de una base de datos. El paquete de DD se puede integrar dentro de un sistema de manejo de bases de datos o tratarse aisladamente. Los dos planteamientos, el paquete de diccionario de datos independiente y el integrado, ofrecen ventajas y desventajas.

### Ventajas de un sistema de diccionario de datos integrado

- Las descripciones de los datos no están almacenadas redundantemente en un



paquete de diccionario de datos y en el sistema de manejo de la base de datos. Esto reduce la ocurrencia de errores debido a las fallas en la actualización en los dos lugares.

- Tiene acceso a los datos de la base de datos. Un uso potencial del diccionario de datos puede ser en el área de seguimiento del acceso a los datos al proporcionar estadísticas valiosas para mejorar el funcionamiento.
- Un diccionario de datos puede servir como una herramienta de control mucho más poderosa cuando está integrado con el DBMS, ya que el diseñador de la base de datos y los usuarios tendrán que reforzar el diccionario de datos como una herramienta para la documentación y el control de los mismos.

### **Ventajas de un sistema de diccionario de datos independiente**

Hay menos riesgo al implantar en el sistema de manejo de la base de datos, un diccionario independiente que uno integrado. También la implantación de un diccionario independiente es más sencilla, ya que el diccionario no tiene que ajustarse a las características de implantación de un DBMS.

Un diccionario de datos integrado necesita al mismo tiempo todas las descripciones que se le pueden proporcionar por etapas al diccionario de datos independiente, en el medio actual de procesamiento de datos no todos los archivos de datos se convierten al mismo tiempo a la base de datos. Una base de datos existe junto con

algunos archivos de base de datos convencionales. Es deseable tener un paquete de diccionario de datos con el que pueda vigilar a los datos en la base de datos, tal como lo haría un diccionario de datos integrado y que además vigile los datos contenidos en archivos como lo haría un diccionario de datos independiente.

Ya sea que el diccionario de datos sea independiente o integrado, es necesario conocer lo relativo a sus interfaces en un medio de la base de datos ideal.

### **El diccionario de datos y sus interfaces**

Consideraremos las interfaces en un diccionario de datos en un medio que dispone de un único sistema de manejo de la base de datos, como se muestra en la Figura 2.1.2

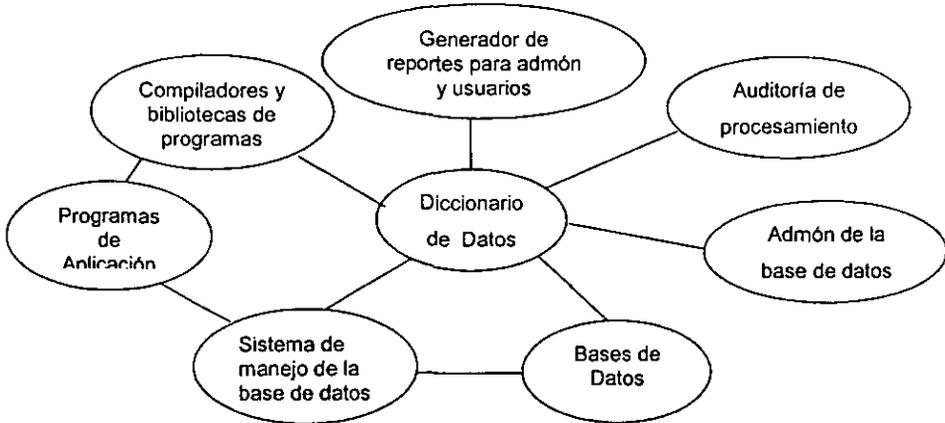


Figura 2.1.2. Interfaces del diccionario de datos

Existen dos tipos de interfaces:

- La interfaz con el usuario, por ejemplo, el administrador de la base de datos, el programador de sistemas y el usuario final.
- La interfaz con el software, por ejemplo el sistema de manejo de la base de datos, los compiladores, el sistema operativo y los generadores de informes.

Los informes deben contener información sobre lo siguiente:

- Los campos de datos y las entidades.
- Las relaciones entre los campos de datos y entre las entidades.
- Responsabilidad de los usuarios de proporcionar datos correctos.
- Frecuencia del uso y descripciones textuales de los elementos.
- Información de control de acceso.
- Informes de contabilidad del sistema.
- Informes sumarios predefinidos.
- Informes adecuados diseñados por los usuarios.
- Informes de referencia cruzada.
- Informes sobre cambios.
- Informes de error.



## Recursos y recuperación de informe

Cualquier herramienta de comunicación y documentación puede ser efectiva si proporciona recursos sencillos de recuperación y de informe. Los informes pueden incluir las siguientes partes:

- Lista de los campos en secuencia alfabética o en cualquier otra deseable.
- Listado de referencia cruzada entre los campos de datos, grupos de campos de datos, programas que los están usando y departamentos o gente responsable de mantener válidos los contenidos.
- Descripciones de datos para los programas en lenguaje huésped y los enfoques lógicos para los modelos externos que contiene a los programas de aplicación.

## Captura de datos como entrada al diccionario de datos

Idealmente el diccionario de datos debe ser usado como herramienta por el administrador de la base de datos, los programadores de aplicaciones, los usuarios. Los datos almacenados en el diccionario de datos deben ser introducidos a través de un lenguaje de entrada orientado al usuario.

## Información del control de acceso

---

El diccionario de datos puede contener información tomando en cuenta el control de acceso, especificando quien puede tener acceso a qué parte de la base y en qué forma. La información puede usarse de la siguiente manera:

- Para localizar con precisión cualquier mal uso de los datos por parte de la base de datos compartida por múltiples usuarios, implántese un conjunto riguroso de restricciones de seguridad.
- Si el diccionario de datos va a usarse para el control de acceso, el mismo debe estar sujeto a este último.

## Partes básicas de un diccionario de datos

- **Campo de datos.** Atributo que describe a una entidad. Cada campo tiene un nombre o etiqueta única. Estos deben ser descriptivos y consisten de palabras claves o abreviaciones tomadas de listas autorizadas.
- **Campo de grupo.** Agrupación de campo de datos lógicamente relacionados. Para un campo de grupo, todos los subelementos (elementos y campos de grupo), deben estar especificados explícitamente.
- **Campo derivado.** Es un campo de datos que se origina como resultado de una fórmula o de un cálculo. Para todos los campos derivados, la fórmula o el cálculo deben estar plenamente identificados, junto con todos los campos de datos usados en la derivación. Estos campos también deben estar definidos en el diccionario.



- **Sinónimo o alias.** Es un campo de datos con designación distinta pero con el mismo significado que otro campo. La descripción de un sinónimo debe identificar las designaciones de otros campos de datos de los cuales este es sinónimo.
- **Homónimos.** Son dos campos con la misma designación pero con distinto significado. Al introducir las designaciones del campo en el diccionario, los problemas de sinónimos y homónimos deben ser identificados por el diccionario y resueltos por el equipo de diseño de base de datos.

Es necesario además de los componentes básicos del diccionario de datos, conocer la descripción de ellos.

- Descripción del **modelo conceptual.** Este modelo describe las entidades de la empresa, las relaciones entre ellas y el flujo de información.
- Descripción del **modelo lógico.** Es un modelo interno, es la conversión física de la base de datos. Describe las relaciones entre las entidades, los métodos de acceso que se han de usar, los elementos de grupo y otros detalles de conversión física.

### Diagrama Entidad – Relación

Los diagramas entidad-relación (DER) son una técnica para representar gráficamente la estructura lógica de una base de datos. La descripción de los objetos que se usaron para su elaboración se muestra en la Tabla 2.1.1

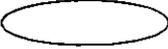
Concepto	Definición	Simbología
Entidad	Se define como algo que se puede identificar fácilmente. Estas se clasifican como entidades regulares y débiles. La entidad débil es aquella cuya existencia depende de otra entidad.	
Propiedad	Las entidades tienen propiedades conocidas como atributos que son elementos de información que describe a una entidad. Cada tipo de propiedad toma sus valores de un conjunto de valores correspondientes. Estas pueden ser simples o compuestas; clave, univaluadas o multivaluadas, faltantes, base o derivadas.	
Interrelación	Es una vinculación de entidades del número de participantes en una interrelación se define el grado de esa interrelación. El grado de su interrelación puede ser De uno a uno De uno a muchos De muchos a muchos	

Tabla 2.1.1 Descripción de los elementos de un DER.



## Normalización

Cada elemento de información es listado y organizado en estructuras de archivos lógicas. Existen técnicas que pueden simplificar la organización de estos archivos, tal es el caso de la llamada normalización, la cuál simplifica la estructura lógica de los datos. Esta técnica identifica los resultados redundantes existentes y determina claves únicas necesarias para el acceso a los elementos de datos. Se tienen tres niveles de organización.

Primera Forma Normal. Para normalizar una lista se separan todos los grupos de datos de forma que ningún archivo tenga grupos repetidos.

Segunda Forma Normal. Para conseguirse deben reorganizarse las relaciones de forma que ningún dato que no sea clave, sea completo y funcionalmente dependiente.

Tercera Forma Normal. Se realiza si todas las condiciones para la segunda forma normal se cumplen y ningún elemento que no sea clave, puede derivarse de una combinación de otros elementos que no son claves en ninguna de las relaciones.



## 2.2. SISTEMAS OPERATIVOS.

### 2.2.1. Windows 9x / NT (Server/Workstation)

Un sistema operativo (SO) puede ser contemplado como una colección organizada de extensiones del software y hardware, consistente en rutinas de control que hacen funcionar a una computadora y proporcionan un entorno para la ejecución de los programas. Otros programas se apoyan en las facilidades proporcionadas por el sistema operativo para obtener acceso a los recursos del sistema informático, tales como archivos y dispositivos de entrada/salida (E/S).

El rango y la extensión de los servicios proporcionados por SO dependen de varios factores. Entre otras cosas, las funciones visibles al usuario de un SO están en gran medida determinadas por las necesidades y características del entorno objetivo que el SO esta destinado a soportar. Por ejemplo, un SO destinado al desarrollo de programas en un entorno interactivo puede tener un conjunto bastante diferente de llamadas y órdenes que el SO diseñado para soporte en tiempo de ejecución de una aplicación de tiempo real dedicada, tal como el control del motor de un automóvil.

Internamente, un SO actúa como gestor de los recursos del sistema informático, tales como el procesador, la memoria, los archivos y los dispositivos de E/S. En esta función el SO lleva la cuenta del estado de cada recurso y decide quién obtiene un recurso, durante cuánto tiempo y cuándo. En sistemas que soportan ejecución concurrente de programas, el SO resuelve las peticiones conflictivas de recursos de manera que preserve la integridad del sistema, y al hacerlo intenta optimizar el rendimiento final.

En general, el objetivo primario de los SO es incrementar la productividad de un recurso de proceso tal como el hardware de la computadora, o los usuarios del sistema informático. En sistemas de un solo usuario, el énfasis está en hacer el sistema informático más fácil de utilizar proporcionando una interfaz de usuario gráfico y obviamente más intuitivo.

Podemos resumir las funciones del SO así: administra los recursos de que disponga la computadora, asigna un lugar en la memoria a los programas y a los datos del usuario y pone en marcha los programas de aplicación tales como procesadores de texto, hojas de calculo, etc.

Para el desarrollo de este proyecto se ha elegido trabajar en un ambiente Cliente/Servidor por lo cuál se adopto el SO Windows NT como servidor y como cliente el SO Windows 95. A continuación se describen ambos a grandes rasgos.

### 2.2.2. WINDOWS 95.

Los SO, como ya hemos visto, funcionan como intermediarios entre el usuario y la computadora y, en consecuencia, cumplen dos funciones: la primera es comunicarse con el usuario interpretando sus instrucciones, y la segunda consiste en comunicarse con la máquina desarrollando cada instrucción y convirtiéndola en un amplio conjunto



de instrucciones que la computadora puede llevar a cabo.

Podríamos resumir este proceso diciendo que el usuario solo se ocupa de dar la orden precisa y el SO de dar los pasos necesarios para que la computadora pueda cumplir adecuadamente la orden, liberando al usuario de tener que crear un programa que especifique en detalle como se tienen que dar todos esos pasos.

Para facilitar la comunicación entre el usuario y la máquina, se han diseñado últimamente los llamados entornos operativos, basados en gráficos, que permiten presentar los distintos elementos con los que trabaja la computadora con dibujos más cercanos al mundo real. Estos entornos operativos se denominan GUI (Graphic User Interface, Interface Gráfica de Usuario).

Windows 95 es un entorno gráfico de usuario para computadoras PC que intenta facilitar la comunicación entre el usuario y la computadora. Para ello, la pantalla de la computadora se convierte en un escritorio, un conjunto de carpetas llenas de papeles y una papelerera.

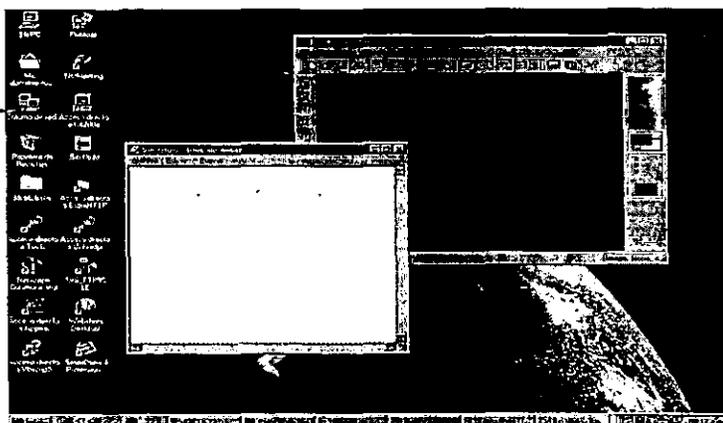


Figura 2.2.1.1 Entorno gráfico de Windows 95.

Cuando queramos trabajar con un papel determinado basta que, mediante el ratón, lo saquemos de su carpeta correspondiente y lo pongamos sobre el escritorio. Cuando alguno de los papeles que utilizemos ya no nos sirva, lo hecharemos simplemente a la papelerera.

Mientras que utilizando el SO MS-DOS o anteriores versiones de Windows, nos veríamos obligados a etiquetar nuestros archivos con palabras de solo 8 letras, ahora es posible etiquetarlos como deseemos.

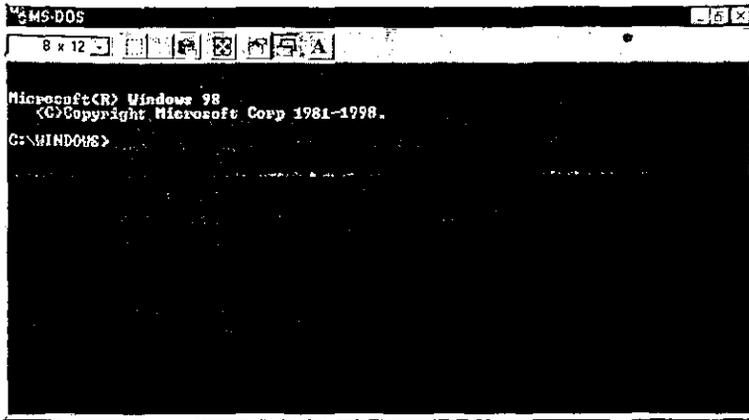


Figura 2.2.1.2 Entorno DOS dentro de Windows 95.

Windows 95 pone también a nuestro alcance la tecnología Plug and Play que nos libera de la agotadora tarea de configurar los diversos dispositivos periféricos conectados a nuestra computadora: impresoras, CD-ROM, tarjetas de sonido, etc. Basta con que conectemos el dispositivo y Windows 95 se ocupará de identificar el modelo y la marca y adaptarse a él para que todo funcione correctamente.

Windows 95 ofrece otras posibilidades interesantes: esta especialmente preparado para el trabajo en red, para la conexión a través de un módem con las redes informáticas internacionales y para que usuarios con discapacidades físicas lo puedan utilizar. Otra ventaja de Windows 95 sobre los sistemas operativos tradicionales es su capacidad de integrar los distintos programas con los que trabaja el usuario (procesadores de texto, programas de diseño gráfico, hojas de cálculo, bases de datos, etc.), facilitando el intercambio de datos y evitando así las laboriosas tareas de exportación e importación de archivos de datos. Esto quiere decir que un dibujo realizado con una hoja de cálculo, por ejemplo, puede colocarse fácilmente en una cierta posición de un texto escrito con un programa procesador de textos mediante opciones denominadas cortar y pegar en la jerga informática.

Finalmente, otra de las ventajas de Windows 95 consiste en la posibilidad de trabajar con algunas de las tecnologías más interesantes y con mayor futuro de la informática actual: las tecnologías multimedia, que permiten la manipulación por la computadora de algunos tipos de información diferentes de los utilizados tradicionalmente tales como gráficas de alta resolución, sonido musical e imágenes de video.



Figura 2.2.1.3 Panel de control de Windows 95.

Windows 95 dispone de tres elementos básicos: los iconos, las ventanas y los menús. Los diferentes iconos representan a los diversos documentos y programas, siempre que estos no estén abiertos en ese momento sobre la mesa del escritorio. Un icono es un pequeño dibujo con un rotulo situado debajo de él. Utilizando adecuadamente el ratón, podemos convertir el icono en una ventana.

Podemos tener varias ventanas abiertas a la vez en la pantalla, de la misma forma que podemos tener abiertas simultáneamente varias carpetas sobre un escritorio.

Las computadoras actuales son lo suficientemente rápidas como para poder ejecutar frente al usuario dos o más tareas en paralelo. Al subdividir la pantalla en ventanas, cada programa se puede ejecutar en una ventana, y los datos que se van generando se pueden cambiar de una a otra.

El tercer elemento gráfico que los entornos operativos ponen a su disposición del usuario son los menús. Un menú es un conjunto de opciones que un programa ofrece al usuario para que éste elija la opción que quiere poner en marcha. Los menús se despliegan mostrando en cada línea una opción.

Podemos dividir las tareas del entorno Windows 95 en tres grupos. En primer lugar, aquellas que realiza un sistema operativo tradicional, administración de programas, administración de archivos y control de impresión de los diferentes dispositivos conectados a la computadora. En segundo lugar las tareas específicas que los entornos operativos realizan para facilitar al usuario una serie de herramientas de las que no disponían los SO tradicionales. Por último, las tareas propias de la tecnología multimedia.

Quando decimos que un entorno operativo permite administrar programas, nos referimos a que permiten organizarlos en grupos de programas afines. También permite crear un nuevo grupo de programas, eliminar uno ya existente o modificar las



características de un cierto grupo de programas. Igualmente es posible cambiar un programa de un grupo a otro grupo y poner marcha la ejecución de un cierto programa.

Windows 95 permite realizar tareas de administración de archivos de forma fácil y cómoda. Muchas de las funciones se realizan mediante el explorador a través del cual se puede:

- Examinar los archivos que tenemos guardados en los discos.
- Buscar con rapidez el archivo que nos interesa.
- Copiar archivos.
- Moverlos, es decir, que los archivos aparezcan en el lugar de destino y que desaparezcan del lugar de origen.
- Eliminarlos provisionalmente arrastrándolos con el ratón hasta la papelera de reciclaje.
- Recuperarlos de la papelera si nos arrepentimos.
- Eliminarlos definitivamente, vaciando la papelera.
- Cambiarles el nombre.
- Ver su contenido de forma rápida.
- Ejecutar programas, es decir, ponerlos en funcionamiento.
- Hacer copias en disquetes.
- Formatear disquetes.

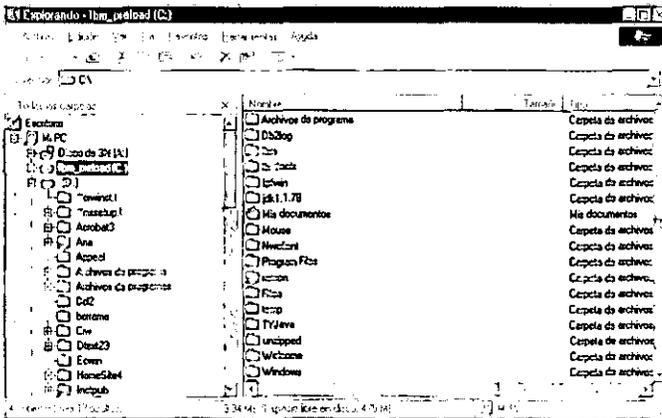


Figura 2.2.1.4 Explorador de Windows 95

El control de impresión es una de las tareas que los entornos operativo intentan facilitar al usuario. De forma sencilla, éste podrá conseguir interrumpir la impresión de texto o gráfico temporalmente, reanudarla, eliminar un archivo de la cola de impresión, conectar la impresora a una red local o desconectarla de ella, ver la relación de archivos enviados a la impresora de la red local por todas las computadoras de la red y, consecuentemente ver el lugar que ocupa en la cola de impresión el archivo que ha enviado el usuario, etc.

Los entornos operativos permiten también controlar, de forma más sencilla que con un



SO tradicional, los distintos dispositivos de la computadora: teclado, pantalla, puertos en serie, y en paralelo, ratón, tipos de letra que se imprimen, etc.

Además Windows 95 ofrece una serie de herramientas que facilitan el trabajo del usuario que van desde una calculadora hasta un reproductor multimedia con múltiples aplicaciones.

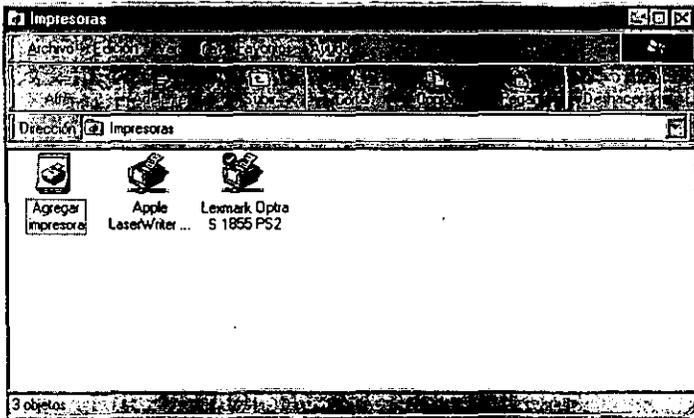


Figura 2.2.1.5 Control de Impresoras de Windows 95.

### 2.2.3. Windows NT.

Windows NT no es, como Windows 95, una ampliación de las anteriores versiones de DOS y Windows, sino un SO desarrollado totalmente nuevo. Las primeras versiones requerían un hardware tan caro e insólito que pocos usuarios estaban dispuestos a instalarlo. Con el descenso de los precios del hardware y sobre todo la memoria fue creciendo el atractivo de Windows NT.

Gracias al desarrollo del nuevo procesador Pentium Pro de Intel, Windows NT ha conseguido una plataforma eficaz y compite con las estaciones de trabajo Unix. En los últimos años, la versión de Windows NT Server se ha establecido básicamente como servidor de archivos y de aplicaciones en redes. En este ámbito, Windows NT podía ganar puntos frente a NetWare, que dominaba el mercado, y se ha ido convirtiendo cada vez más en un serio competidor, especialmente en las nuevas instalaciones de redes. El entorno de usuario totalmente gráfico facilita notablemente las tareas de administración diarias y reduce el periodo de formación.

Por lo que respecta a ofrecer servicios de Internet, las soluciones de Windows NT cada vez despiertan mayor interés. Si bien hace algún tiempo se encontraba casi exclusivamente servidores de Internet basados en Unix, actualmente aparecen cada vez más sitios trabajando con Windows NT en la red global de datos.

A diferencia de la mayoría de los SO, Windows NT trabaja de forma independiente de



la arquitectura del procesador. Las partes de programa específicas para cada procesador solo se encuentran en una pequeña porción del código de programa, denominada **Hardware Abstraction Layer** (capa de abstracción de hardware). Esta independencia del SO en sí hace posible la compatibilidad con varios procesadores.

La arquitectura del micrónúcleo de Windows NT puede ser considerada como sistema cliente-servidor. El SO en sí está compuesto por una serie de subsistemas que se ejecutan en modo privilegiado (núcleo). El modo núcleo también se denomina modo "supervisor" o modo "anillo 0". Las aplicaciones y áreas del servidor se ejecutan en modo usuario o aplicación.

En este punto, es importante destacar la diferencia entre el modo núcleo y el micrónúcleo en sí. El modo núcleo controla la ejecución de los sistemas elementales, concediéndoles siempre una mayor prioridad que el resto de aplicaciones o subsistemas adicionales que se ejecutan en modo usuario. De esta forma, una sola aplicación no puede paralizar el subsistema como tal.

El micrónúcleo propiamente dicho es una colección de subsistemas que ejecutan las tareas elementales del sistema operativo, por ejemplo la cooperación con el hardware y la funcionalidad de red. Los componentes del micrónúcleo están rigurosamente separados entre sí y no se reparten datos. En los SO puramente basados en micrónúcleo solamente se pueden ejecutar funciones esenciales en modo núcleo.

Sin embargo en Windows NT, se han incorporado numerosos subsistemas en el modo núcleo especialmente protegido, a fin de protegerlos ante accesos a través de aplicaciones. Es por ello que Windows NT también se denomina SO de "micrónúcleo modificado" o "macrónúcleo".

El resultado de esta ampliación del micrónúcleo es una estabilidad y una seguridad de acceso mayores, puesto que entre el modo usuario, que puede asignarse mediante las aplicaciones y los usuarios, y el modo núcleo, que es utilizado por el propio SO, existe un intermediario: una interfaz de software bien definida.

Con la versión 4.0 de Windows NT, se ha dado un paso hacia delante en lo que se refiere a un SO puramente de micrónúcleo. Dos partes esenciales del subsistema WIN32, que funciona en modo usuario, se han trasladado al micrónúcleo y funcionan en modo núcleo:

- El archivo USER.EXE (administrador de ventanas), y
- El archivo GDI.EXE (GDI, Interfaz de dispositivos gráficos), el sistema de salida de gráficos de Windows NT.

Los SO de alto rendimiento como Windows NT viven de su fiabilidad. A menudo se emplean en aplicaciones delicadas o en servidores y, por lo tanto, deben ser particularmente estables. Además, deben proteger los datos importantes de accesos no autorizados a estos ordenadores. Para cumplir con estos requisitos se precisan



determinados elementos en el SO.

Todos los procesos de un sistema operativo requieren memoria RAM auténtica o direcciones de memoria en el procesador. La mayoría de fallos de los SO como Windows 95 o Windows para Trabajo en grupo se basan en que dos aplicaciones o procesos intentan utilizar la misma memoria, interfiriéndose mutuamente. El resultado de ello son los fallos de protección general (suficientemente conocidos).

En Windows NT, todos los procesos se desarrollan en áreas de memoria rigurosamente separadas. Además se produce una separación entre los procesos en modo usuario y en modo núcleo:

- Los procesos en modo usuario sólo puede leer y escribir en el área de memoria que tienen asignada. No pueden utilizar la memoria de otra aplicación en modo usuario ni acceder a áreas de memoria utilizadas por los subsistemas del modo núcleo.
- Los procesos del modo núcleo, que deben ser utilizados por todas las aplicaciones, emplean básicamente el área de memoria asignada al modo núcleo. Esta área de memoria está protegida contra accesos mediante aplicaciones y subsistemas en modo usuario. Las aplicaciones del modo núcleo también pueden acceder en determinadas situaciones al área de memoria de las aplicaciones del modo usuario y abrir subprocesos en ellas.

---

La organización de los importantes archivos del sistema GDI.EXE y USER.EXE en modo núcleo garantiza una seguridad y una estabilidad del SO máximas, puesto que estos archivos están protegidos ante acceso de la mejor forma posible.

El precio que deberá pagar por ello como usuario son unos requerimientos del sistema más elevados. A pesar de que los accesos del hardware puedan acelerarse a través de la organización de modo núcleo, un modo núcleo mayor representa unos requerimientos mayores en cuanto a procesador y memoria disponible.

El peligro resultante de esta nueva forma de organización es que los procesos de los archivos USER.EXE y GDI.EXE ahora también pueden escribir en áreas de memoria de otras aplicaciones del núcleo. Por lo tanto, en un principio un GDI.EXE bloqueado puede dejar inutilizado todo el sistema.

No se han desplazado los archivos USER.EXE y GDI.EXE al núcleo en vano. Con el nuevo entorno de Windows 95 se ha dotado a Windows NT de un entorno gráfico mucho más potente que en las versiones anteriores. Esto es debido directamente al acceso al hardware. En un procesador Pentium 90, un acceso al hardware procedente del modo usuario requiere unos 70 microsegundos, mientras que un acceso del modo núcleo solo necesita unos pocos microsegundos. El nuevo entorno implica un mayor número de accesos y las necesidades de recursos del mismo aumentan claramente. Sin el truco de desplazar los archivos USER.EXE y GDI.EXE al modo núcleo, Windows NT seguramente habría sido más difícil de comercializar: nadie querría un sistema que en la versión para estación de trabajo sólo funcione correctamente con más de 32 MB



de memoria RAM, un Pentium y una tarjeta de aceleración de gráficos con más de 2 MB.

Un efecto secundario del desplazamiento de los archivos USER.EXE y GDI.EXE al modo núcleo es una necesidad de memoria más reducida de esas dos aplicaciones. No obstante, el espacio ahorrado de este modo es utilizado por el Explorador del entorno Windows 95, de forma que Windows NT 4.0 en total no emplea más memoria que versiones anteriores.

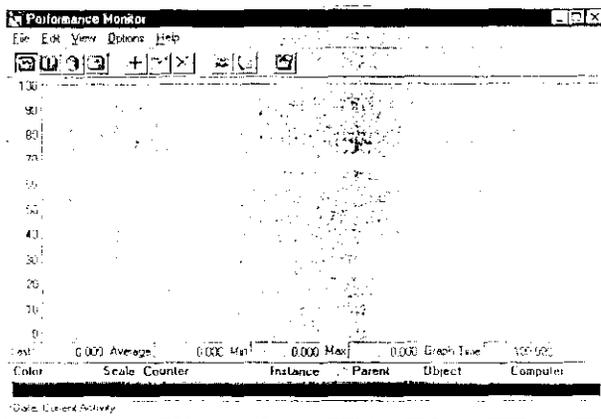


Figura 2.2.1.6 Monitor de Rendimiento de Windows NT.

Windows NT utiliza el método multitarea prioritario (auténtico). Con el mismo se ejecuta simultáneamente varios procesos con distintas prioridades. Así, es posible, por ejemplo, dar formato a los disquetes de inicio en el programa de instalación mientras se copian archivos temporales en el disco duro. Las tareas se distribuyen, en las unidades de control, en lapsos de tiempo.

Los conceptos multitarea sencillos utilizan el método multitarea cooperativo. Con el mismo también se asumen y gestionan varios procesos al mismo tiempo. Sin embargo, en cada momento sólo se edita un proceso, mientras que el resto se trasladan a una cola de espera.

Windows 95 utiliza un procedimiento de doble vía. Si se utilizan únicamente aplicaciones de 32 bits, es posible utilizar también multitarea auténtica. Sin embargo, sólo con que inicie una aplicación de 16 bits Windows 95 pasará todas las aplicaciones a multitarea cooperativa. Aquí Windows NT es más cómodo para el usuario. Aunque se abran aplicaciones de 16 bits, las aplicaciones de 32 bits emplean la multitarea auténtica. El procesamiento paralelo de varias tareas en un procesador se puede mejorar todavía. Si se realizan al mismo tiempo distintas tareas en un programa de aplicación, de poco sirve el método multitarea por sí solo. Para el procesador, estas tareas son una sola, por lo que solamente se ejecutarán una tras otra.



El remedio para este problema es el Multisubproceso. Con el mismo, el programa de aplicación diferencia distintas tareas mediante los denominados subprocesos. El número de subprocesos que puede administrar al mismo tiempo una aplicación depende de los recursos del SO y del software de la aplicación. Algunos programas gestionan hasta 17 subprocesos.

Junto con el rendimiento, las características de calidad esenciales de un SO son la estabilidad y la seguridad. La causa de pérdida conjunta de información se debe a que las aplicaciones no se ejecutan en áreas de memoria separadas, suele haber dos aplicaciones distintas que intentan acceder simultáneamente a la misma área de memoria.

Windows NT resuelve este problema ejecutando cada aplicación en un área de memoria distinta. Durante el funcionamiento normal no puede interferir en territorio ajeno. Una aplicación que haya dejado de responder puede ser terminada (o finalizada) con el administrador de tareas, sin que el resto de aplicaciones o el propio SO finalicen.

Aunque se ejecuten aplicaciones en distintas áreas de memoria pueden producirse fallos en el sistema. Por ejemplo, si dos programas acceden al mismo tiempo al archivo de sistema USER.EXE, pueden interferirse mutuamente provocando una pérdida de datos y dejar inutilizable el archivo de sistema para cualquier otro uso posterior. En estas situaciones Windows NT también ofrece soluciones. El servidor de sistema, responsable de las tareas fundamentales del SO, se ejecuta en un área de memoria que no puede ser utilizada por ninguna otra aplicación. Incluso si hubiera dañado el propio entorno de Windows NT mediante el bloqueo de otras aplicaciones, el servidor del sistema puede activar el Explorador y hacer funcionar el sistema.

Lamentablemente, Windows NT no se puede iniciar orientado por caracteres como OS/2, por ejemplo. Esto permite a los usuarios iniciar el sistema e intentar llevar a cabo alguna medidas de reconstrucción y rescate después de dañar el entorno gráfico en el disco duro.

Naturalmente, la gestión separada de programas en áreas de memoria propias requiere unos recursos del sistema mayores que para dividir la memoria. Cada aplicación en Windows NT toma como mínimo 3MB de memoria RAM. Si se ejecutan algunos servicios estándar o incluso de servidor en segundo plano, la memoria RAM se consume rápidamente.

La gestión central de datos de configuración también contribuye a la seguridad y a la estabilidad de un SO. Windows NT utiliza una base de datos de configuración central, la base de datos de registro. Los antiguos usuarios de Windows 3.x se beneficiaron al no tener que volver a editar algún archivo como WIN.INI, SYSTEM.INI, AUTOEXEC.BAT o un CONFIG.SYS, etc.

Para ello es necesario trabajar con cuidado con la complicada base de datos del registro. Si se realizan configuraciones incorrectas se puede bloquear el sistema.



Aunque en este caso no esta todo perdido: al volver a iniciar Windows NT basta con recurrir a la última configuración buena conocida, guardada en el sistema como copia de seguridad.

La administración de discos duros en Windows NT viene encuadrada por los conceptos de tolerancia a errores y RAID (Redundat Array of Inexpensive Disks).

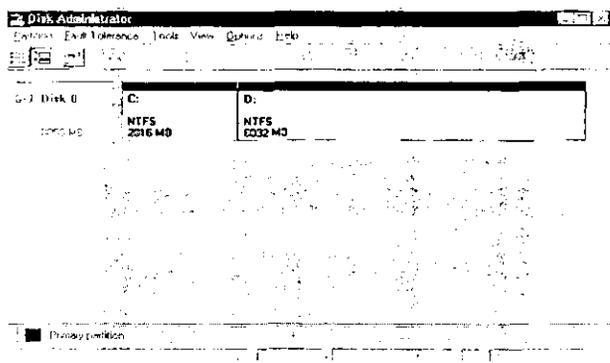


Figura 2.2.1.7 Administrador de discos de Windows NT.

El primer concepto comprende posibilidades de protección de datos como discos duros espejo y los conjuntos de bandas. RAID es el concepto general para distintos niveles de organización del disco duro como reflejar el disco duro o dividir una sola partición en varios discos duros.

Windows NT es un SO de Red. Windows NT administra grupos de trabajo y dominios. Si un servidor NT debe trabajar en una auténtica red cliente-servidor, uno de los servidores debe asumir el papel de controlador principal de dominio. Por lo tanto, para administrar usuarios y recursos es preciso trabajar a nivel de dominios.

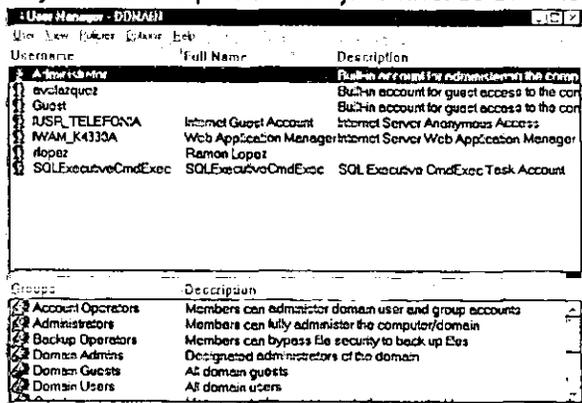


Figura 2.2.1.7 Administrador de Usuarios de Windows NT.



Las redes de área local modernas utilizadas en empresas con frecuencia no sólo conectan computadoras con el mismo SO, sino muchos distintos. Así, es posible encontrar en una red estaciones de trabajo de Windows para trabajo en grupo o Windows NT junto con computadoras con Windows 95, Macintosh, OS/2 y quizás también Unix. Además, puede también haber un servidor de archivos NetWare. Este entorno mixto debe comunicarse entre sí y permitir que se compartan los recursos mediante protocolos unitarios. Windows NT ofrece para ello todos los requerimientos: además del protocolo NetBEUI utilizado en redes de Microsoft, también son compatibles IPX/SPX (Novell) y TCP/IP (Unix, OS/2). TCP/IP también se encarga de conectar una computadora con Windows NT a Internet, y con ello, la red más grande y más heterogénea.

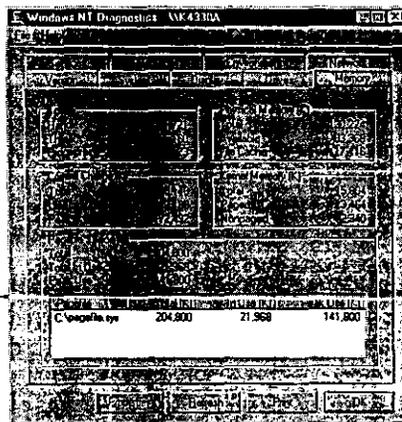


Figura 2.2.1.8 Ventana de Diagnostico de Windows NT.

Como las redes de NetWare tradicionalmente están muy extendidas en la empresa, Windows NT ofrece servicios de cliente y puerta de enlace para la cooperación con este tipo de redes.

Con el protocolo de control de vinculación de datos (DLC), Windows NT permite la vinculación con productos mainframe en LAN y WAN y, por ejemplo, el acceso a impresoras de red.

En una red trabajan varias personas que comparten recursos como el espacio de disco duro, el rendimiento de la computadora, la impresora y los programas de aplicación. Es evidente que, en una red de este tipo, no todo el mundo puede hacerlo todo. Para controlarlo Windows NT soporta:

- Cuentas de usuario, que establecen exactamente quién puede acceder a qué recursos y de qué forma.
- Registros de usuarios, con los cuales se comprueba quién ha utilizado qué recursos y cuándo.
- Registro de sucesos generales, que ayudan a encontrar el error si se produjera un fallo.

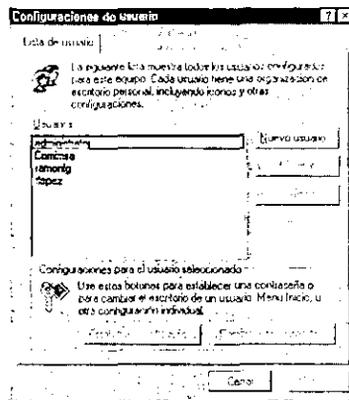


Figura 2.2.1.9 Configuraciones de usuario en Windows NT.

Como SO verdaderamente multiusuario, Windows NT ofrece esta administración de usuarios para todos los usuarios locales. Por lo tanto, para cada usuario que se sienta directamente delante de una computadora con Windows NT puede establecer los derechos de acceso y perfiles de usuario.



## 2.3. TIPOS DE REDES Y PROTOCOLOS DE COMUNICACIÓN.

### 2.3.1. Antecedentes.

Se puede definir a una red como un conjunto de computadoras ( y terminales, en general) interconectadas a través de uno o varios caminos o medios de transmisión. Si se analiza el concepto, se concluye que los elementos básicos de una red de cómputo: son los sistemas de cómputo, los medios de transmisión y los dispositivos que permitan interconectarlos.

En la década de los 80's se comercializaron las computadoras personales (PC's) y se generó una gran cantidad de software de aplicación específica y los sistemas operativos, que permitieron conectarlas en red. Se desarrollaron los sistemas multiusuarios y surgieron las redes de área local o LAN's (Local Area Networks) que posteriormente serían utilizadas en todo el mundo.

En la década de los 90's las redes de cómputo se convirtieron en una necesidad para pequeñas y medianas empresas en el desarrollo de una cultura de sistemas de información.

### 2.3.2. Redes de computadoras, tipos y configuraciones.

#### 2.3.2.1. Tipos de Redes

Anteriormente las computadoras eran muy costosas, las organizaciones no podían entregar a todo el personal una computadora para su uso individual. En lugar de ello la Unidad Central de Procesamiento (CPU) tenía que estar compartida, lo anterior dio origen a las redes.

En general las redes se pueden clasificar en:

- Centralizadas.
- Distribuidas.

Las primeras redes fueron centralizadas como se puede ver en la figura 2.3.2.1. En este tipo de redes el centro de la actividad de cálculo está localizado en la computadora principal o anfitrión. Esta puede ser tanto una computadora grande como una minicomputadora. Los usuarios acceden al anfitrión a través de las terminales "tontas", ya que no pueden realizar ningún proceso por sí mismos, su principal objetivo es proporcionar una interfaz entre el anfitrión y los usuarios.

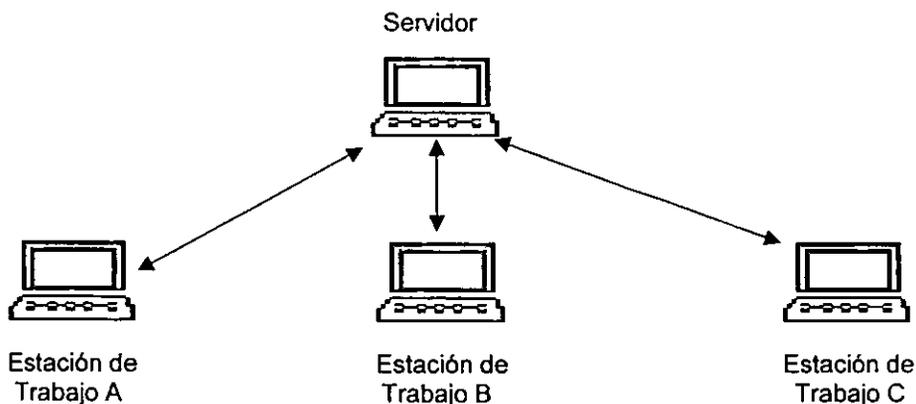


Figura 2.3.2.1 Redes Centralizadas

Las redes Centralizadas como su Nombre lo indica, proporcionan un proceso centralizado pero a la vez están limitadas en algunos puntos muy importantes. Los usuarios del procesador central están limitados por las aplicaciones del anfitrión. La habilidad de los usuarios para realizar procesos de análisis será condicionada, por que los cambios en los programas del anfitrión son a menudo muy costosos, son grandes consumidores de tiempo y además el programa del anfitrión debe satisfacer las necesidades de todos los usuarios de la red.

Por otra parte, las redes distribuidas permiten que los tiempos de análisis y de proceso sean mucho más rápidos y efectivos, además de existir la ventaja de que las aplicaciones y la información no se concentren en una sola computadora, como es el caso del anfitrión en las redes centralizadas, en este tipo de redes sigue existiendo una computadora principal pero, a diferencia de las redes centralizadas, el resto de las computadoras no son terminales "tontas", ya que cada una puede procesar información lo que podemos ver en la figura 2.3.2.2

Las redes distribuidas también ofrecen la ventaja de que los recursos de las mismas, por ejemplo las impresoras, puedan ser utilizadas con mayor efectividad y sin tanto consumo de tiempo y recursos.

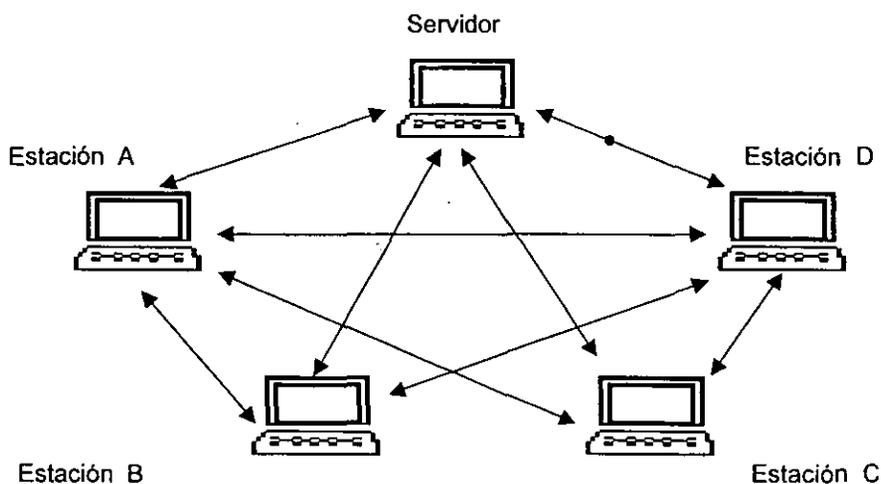


Figura 2.3.2.2. Redes Distribuidas

Por las características anteriormente mencionadas y el hecho de que son más sencillas de crecer, las redes distribuidas son ideales para trabajar como Redes de Área Local (LAN's).

### 2.3.2.2. Topología.

La topología de una red hace referencia a la ruta por la que discurren los datos a través de la red. Hay tres tipos básicos de topologías:

- Estrella
- Bus
- Anillo

#### 2.3.2.2.1. Configuración estrella.

En este tipo de conexión, el elemento central es el Servidor con sus Periféricos, como lo podemos ver en la figura 2.3.3.1.1 se mantiene preguntando constantemente a cada estación de trabajo mediante comunicación exclusiva y por un turno si desea transmitir información, y en caso afirmativo, la atiende, al terminar prosigue con otra su interrogatoria permanente.

Para este caso de pregunta a una –respuesta- se realiza de forma individual, la regla de comunicación se le denomina protocolo POOLING (poleo), y es comúnmente empleado en las "minis".

En el despertar de las redes, esta topología es la que se utilizó primero, pero resultaba



una de las más caras.

La gran desventaja de esta topología es depender en su totalidad del nodo central por lo que al fallar este, falla toda la red. Esta topología es usada en gran escala en los sistemas telefónicos.

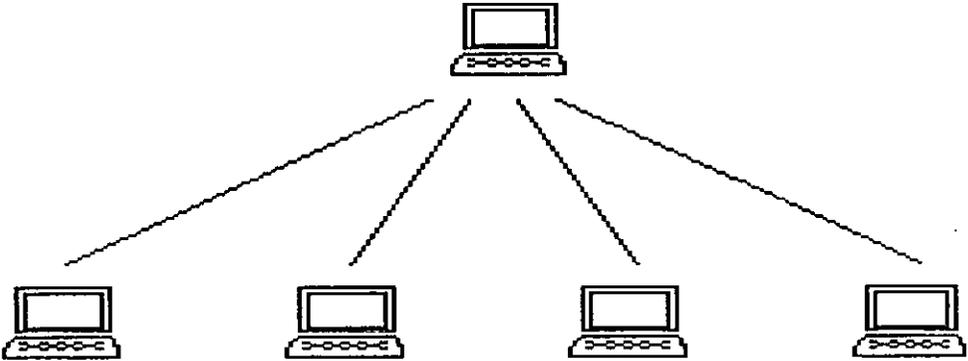


Figura 2.3.2.2.1.1. Configuración Estrella

### 2.3.2.2.2. Configuración anillo.

En esta conexión, la información viaja ordenadamente en un solo sentido a través de un solo cable, describiendo una trayectoria de 360° en cuyo anillo imaginario, están conectadas en serie las estaciones de trabajo y el Servidor.

Una señal llamada TOKEN (Receptáculo, a modo de estafeta) va circulando por la RED y pasando por cada estación, si la primera resultó ser la solicitante, previa identificación entrega la información, de lo contrario la deposita en un "sobre cerrado" para que ésta a su vez la envíe a la siguiente, llevando la consigna de entregarla hasta identificar a la solicitante.

Cada estación de paso, cuando más, colecta información adicional enviándola a la siguiente y así se pasa la señal cerrando los ciclos "circulares" por ello el protocolo apropiado para este caso se conoce como TOKEN PASSING.

Como podemos observar en la figura 2.3.3.1.2.1, en una configuración de anillo las computadoras se conectan entre sí, como su nombre lo indica. Los mensajes que se mandan alrededor del anillo deben estar específicamente dirigidos al nodo destino, debido a que las computadoras conectadas tienen acceso al mensaje, de similar forma que la configuración de bus.

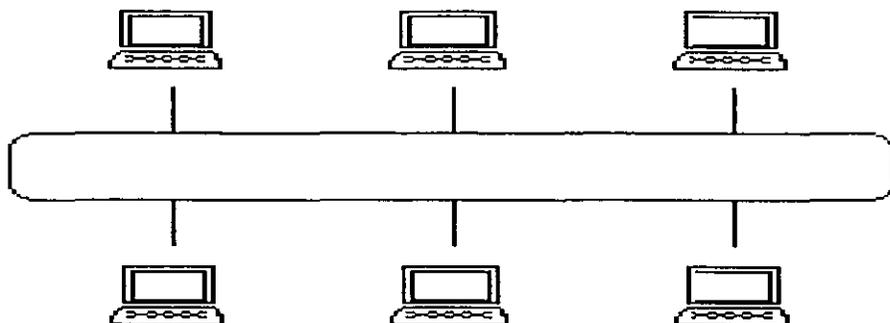


Figura 2.3.2.2.1. Configuración Anillo

### 2.3.2.2.3. Configuración bus.

Esta configuración se considera que es la más sencilla de todas, donde las Estaciones de Trabajo y el Servidor, están enlazadas por un solo cable (Configuración par trenzado), y la configuración viaja en ambos sentidos, por lo que es necesario prevenir las colisiones.

~~Por ello el Protocolo asociado es el CSMA/CD (Carrier Sense Múltiple Access/CollisionDetection).~~

Con este Protocolo, cualquier NODO transmite y espera a que se le confirme que la configuración fue recibida configuración, de otra forma, se detecta la posible colisión, espera un tiempo a que el canal esté desocupado y la configuración se transmite.

Es decir en la topología de bus, cada dispositivo está directamente conectado a un canal de configuración común, como podemos ver en la figura 2.3.3.1.3.1. Las señales que son transmitidas sobre el canal son llamados mensajes. Todos los mensajes pasan a lo largo del canal, por lo cual todas las computadoras se dan cuenta de lo que se transmite sobre el canal. El mensaje contiene una dirección la cual indica a que dispositivo se traslada en caso de que el nodo al que pasa el mensaje es el indicado por la dirección se procesa en caso contrario se ignora y el mensaje sigue su camino.

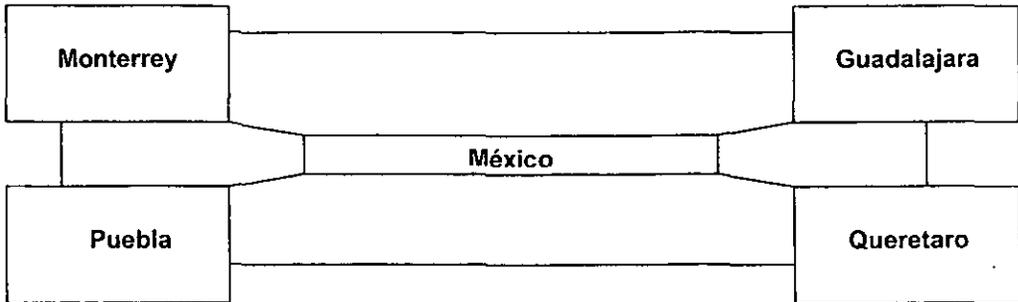


Figura 2.3.2.2.3.1. Configuración Bus



#### 2.3.2.2.4. Configuración malla.

La topología se puede llevar a cabo en forma local sin embargo con frecuencia este tipo se utiliza en las redes amplias o metropolitanas (MAN, WAN) que necesitan conectar diferentes puntos en forma remota. Se utilizan ruteadores para seleccionar el mejor trayecto de un punto a otro a través de la malla, figura 2.3.3.1.4.1.



#### 2.3.2.3. Redes LAN.

Es un sistema de transmisión de información con el objetivo de compartir recursos con los que trabaja una computadora normalmente, es decir, archivos, directorios, impresoras, plotters, escáners, etc, entre computadoras conectadas entre si o bien mediante redes conectadas entre sí.

La palabra local se refiere a que el conjunto de computadoras se encuentra próximo geográficamente hablando es decir, que se encuentra en el espacio físico de un mismo centro.

En general una red local está caracterizada por una distancia corta entre computadoras, un medio de comunicación entre estas, una velocidad de conexión elevada, la utilización de cables de conexión simples (como los coaxiales o los telefónicos).

Cuentan con la facilidad de su instalación, de su administración y de su bajo precio.

En la mayoría de los casos una red se usa para compartir entre varias computadoras una unidad de almacenamiento enorme o en general cualquier dispositivo periférico del que hagan uso varias personas de un mismo grupo de trabajo, de esta forma no es necesario comprar ese periférico para cada computadora, por ejemplo una impresora láser.

Además constituye un valor añadido a la hora de compartir la información y distribuir tareas.



### **Ventajas de utilizar una red de área local.**

A menudo la instalación de una red de área local se justifica inicialmente como un medio de compartir los dispositivos periféricos. También pueden compartirse otros dispositivos de salida caros o especiales, como impresoras láser, fax, módems de alta velocidad o las impresoras de color.

Pero conforme la red va creciendo y se integra en la organización, el compartir los dispositivos es insignificante en comparación con las demás ventajas de la conexión de las redes. Las redes de área local se enlazan tanto a las personas como a los equipos físicos de la computadora. El resultado es tanto una red electrónica como una red humana. Las redes de área local proporcionan una herramienta efectiva para la comunicación entre los componentes del grupo de trabajo a través del empleo del correo electrónico y software para desarrollo de aplicaciones o para el trabajo en oficinas. Los mensajes se envían instantáneamente a través de la red.

Un beneficio colateral muy importante de las redes de área local es que el sistema operativo, las aplicaciones y los datos son mucho más fáciles de mantener y proteger que los de un entorno visualizado. Los datos críticos pueden salvarse diariamente o incluso cada hora si fuese necesario con el objetivo de mantener la integridad de los mismos.

---

Construir una red de área local no es difícil, pero requiere una cuidadosa planificación, que en general consta de cinco puntos básicos:

- Seleccionar la topología y el equipo físico.
- Instalar el equipo físico y el sistema operativo de la red.
- Configurar el sistema y cargar las aplicaciones.
- Crear el entorno del usuario.
- Establecer una administración de la red.

El primer paso consiste en diseñar la arquitectura física de la red. Trabajando con un instalador de redes, se debe decidir en cuales oficinas o locales deben tenerse los cables y donde colocarse los dispositivos claves (Ej. El servidor). También se deben seleccionar los tipos de computadoras que deben usarse como terminales y el propio esquema de la red.

El siguiente paso es instalar el equipo físico y unir las computadoras como los cables y las tarjetas de interfaces. Llegados a este punto, debe cargarse el sistema operativo en el disco duro de la computadora que se haya elegido como servidor, configurándola para reconocer los demás dispositivos (Ej. Las impresoras). Una vez hecho lo anterior, se puede crear la estructura del subdirectorio y organizar el disco duro para preparar la carga de la aplicación y de otros datos.

Posteriormente se debe crear el entorno del usuario a través de las pantallas que van apareciendo, a partir de cuando el usuario inicia la sesión y de los menús que lo



ayudan y lo guían entre las muchas opciones disponibles; también se necesita determinar los procedimientos de seguridad para proteger la integridad de los datos almacenados en la red.

Finalmente ya que la red local requiere una administración constante, se necesitan establecer los procedimientos de soporte de la red.

Por lo general las LAN tienen características particulares que son:

- Un campo de acción cuyo tamaño no es mayor de unos cuantos kilómetros.
- Una velocidad total de datos, de cuando menos varios Mbps.
- Una pertenencia a una sola organización.
- Un campo de acción reducido, tamaño del orden de los kilómetros.
- Velocidad total de datos de varios Mbps (10 - 100).
- Propiedad de una sola organización.

### 2.3.2.4. Redes WAN.

Las redes WAN (Red de área extendida) abarcan países enteros, tiene una velocidad de datos inferior a 1 Mbps y pertenecen a múltiples organizaciones (Los proveedores de servicios portadores son propietarios de las subredes de comunicación y numerosos anfitriones pertenecen a los clientes).

Una red de área extendida o WAN (Wide Area Network) es una colección de LAN's interconectadas. Las WAN pueden extenderse a ciudades, estados, países o continentes. Las redes que comprenden una WAN utilizan encaminadores (routers) para dirigir sus paquetes al destino apropiado. Los encaminadores son dispositivos de hardware que enlazan diferentes redes para proporcionar el camino más eficiente para la transmisión de datos. Estos encaminadores están conectados por líneas de datos de alta velocidad, generalmente, líneas telefónicas de larga distancia, de manera que los datos se envían junto a las transmisiones telefónicas regulares.

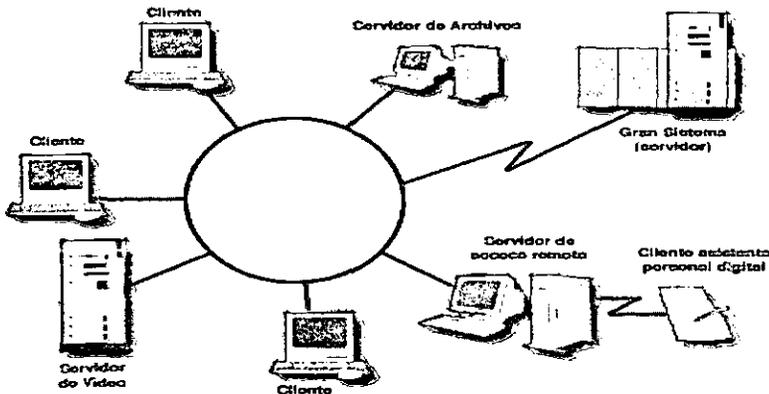


Figura 2.3.2.4. Redes WAN.



### 2.3.2.5. Redes Ethernet.

Las redes Ethernet emplean el esquema de contención en la gestión de la transmisión en la red. Pueden estructurarse en las topologías de bus o de estrella empleando cables de tipo coaxial, par trenzado o fibra óptica.

Una ventaja clave de Ethernet es la rapidez. Con transmisiones del orden de 10-100-1000 megabytes por segundo (Mbps) Ethernet es una de las redes de área local más rápidas que pueden encontrarse.

### 2.3.3. Comunicaciones.

#### 2.3.3.1. Modelos de interconexión.

Existen diversas formas en las que podrían organizarse las redes, en cuanto a la conexión física y esto es la topología de la red.

#### 2.3.3.2. Protocolos de control de transferencia.

~~ICMP~~ Internet es un sistema autónomo que no dispone de ningún control central, el protocolo ICMP ("Internet Control Message Protocol = Protocolo de Control de Mensajes de Internet"), proporciona el medio para que el software de host y gateway intermedios se comuniquen.

El protocolo de ICMP tiene su propio número de protocolo (número 1) que lo habilita para utilizar el IP directamente.

La implementación de ICMP es obligatoria como un subconjunto lógico del protocolo IP. Los mensajes de error de este protocolo los genera y procesa TCP/IP, y no el usuario.

#### Formato del mensaje ICMP

Cada mensaje ICMP esta compuesto por los siguientes campos:

- Tipo
- Código
- Checksum
- Otras variables

#### Solicitud de Eco

Un host puede comprobar si otro host es operativo mandando una solicitud de eco. El receptor de la solicitud la devuelve a su origen. Esta aplicación recibe el nombre de



ping. Esta utilidad encapsula la solicitud de eco del IMCP (tipo 8) en un datagrama IP y lo manda a la dirección IP.

El receptor de la solicitud de eco intercambia las direcciones del datagrama IP, cambia el código a cero y lo devuelve al origen.

### Informes de destinos inalcanzables

Si un gateway no puede enviar un datagrama a la dirección de destino, este manda un mensaje de error ICMP al origen. El valor del campo tipo es 3, y el tipo de error viene dado por el campo código.

Código	Descripción
0	Red no alcanzable
1	Host no alcanzable
2	Protocolo no alcanzable
3	Puerto no alcanzable
4	Necesaria fragmentación con la opción DF
5	Fallo de la ruta origen
6	Red de destino desconocida
7	Host de destino desconocido
8	Fallo del host de origen
9	Red prohibida administrativamente
10	Host prohibido administrativamente
11	Tipo de servicio de Red no alcanzable
12	Tipo de servicio de host no alcanzable

Tabla 2.3.3.2.1. Códigos de Inalcanzable

Formato del mensaje: El formato del mensaje de control de flujo es idéntico al mensaje de inalcanzable, excepto que el tipo es 4 y el código es 0.

### Cambio de ruta (redireccionamiento)

Los gateways en cualquier internet contienen las tablas de redireccionamiento más comunes. Cuando la ruta por defecto no es la más adecuada, el gateway puede enviar al host un mensaje de redireccionamiento ICMP que contiene la ruta correcta.

Formato del mensaje: El formato del mensaje ICMP de control de flujo es igual al del mensaje de inalcanzable, excepto que el tipo es 5 y el valor del código es variable entre 1 y 3. Los motivos para la redirección y sus códigos son los de la tabla siguiente:

Código	Razón para la dirección
1	Por el host
2	Por el tipo y servicio de red
3	Por el tipo y servicio de host

Tabla 2.3.3.2.1. Códigos de Redirección.



**Tiempo de vida excedido:** Para prevenir bucles en la dirección, el datagrama IP contiene un tiempo de vida definido por el origen. A medida que cada gateway procesa el datagrama, el valor del campo disminuye en una unidad. Posteriormente el gateway verifica si el valor del campo es 0. cuando se detecta un 0, el gateway manda un mensaje de error ICMP y descarta el datagrama.

**Formato del mensaje:** El formato del mensaje de error es igual al del mensaje de inalcanzable, pero el tipo es 11 y el código es igual a 0 (contador sobrepasado), o 1 (tiempo de reensamblaje de fragmento excedido).

### 2.3.3.3. Protocolos TCP/IP.

Aunque poca gente sabe lo que es TCP/IP todos lo emplean indirectamente y lo confunden con un solo protocolo cuando en realidad son varios, de entre los cuales destaca y es el más importante el protocolo IP.

Bajo este nombre (TCP/IP) se esconde uno de los protocolos más usados del mundo, debido a que es el estándar de internet y está muy extendido en el sistema operativo UNIX

Para que este protocolo pueda funcionar debe existir comunicación entre las distintas computadoras conectadas a internet, deben usar el mismo protocolo de comunicaciones y deben fraccionar la información que se transmite en los que se insertarán las direcciones de las computadoras origen y destino, asegurándose de que la información transmitida llegue intacta a su destino. Para ello elige las rutas más convenientes hasta el receptor y tras hacer comprobaciones en la información original de forma que esta quede como inicialmente se envió.

Este protocolo, en el momento de enfrentarse con un problema, el sistema que utiliza es el de dividir el problema en pequeñas porciones hasta que finalmente aborda cada uno de estos problemas hasta solucionarlo.

Algunos de los motivos de su popularidad son:

- Independencia del fabricante.
- Soporta múltiples tecnologías.
- Puede funcionar en máquinas de cualquier tamaño.
- Estándar de EEUU desde 1983.

**Capa de transporte:** Provee comunicación extremo a extremo desde un programa de aplicación a otro. Puede proveer un transporte confiable asegurándose de que los datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente de manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota. En esta capa se encuentran los protocolos UDP y TCP.



**UDP** ("User Datagram Protocol = Protocolo de datagrama de usuario"). El protocolo UDP (User Datagram Protocol) proporciona aplicaciones con un tipo de servicio de datagramas orientado a transacciones. El servicio es muy parecido al protocolo IP, en el sentido de que no es fiable y no está orientado a la conexión. El UDP es simple, eficiente e ideal para aplicaciones como el TFTP y el DNS. Una dirección IP sirve para dirigir el datagrama hacia una máquina en particular, y el número de puerto de destino en la cabecera UDP se utiliza para dirigir el datagrama UDP a un proceso específico localizado en la cabecera IP. La cabecera UDP también contiene número de puerto origen, que permite al proceso recibido conocer como responder al datagrama.

**Formato del mensaje:** El datagrama UDP contiene cuatro campos, que son el número de puerto de origen, número de puerto de destino, longitud del mensaje y Checksum.

**Formato del mensaje UDP:** Números de puerto de origen y destino. Estos números, junto con las direcciones IP definen el punto final de la comunicación. El número de puerto origen, puede tener valor cero si no se usa. El número de puerto destino solo tiene sentido en el contexto de un datagrama UDP y una dirección IP en particular.

El número de puerto origen es de un campo de 16 bits. El puerto destino tiene la misma longitud.

**Longitud del mensaje:** este campo tiene una longitud de 16 bits y contiene el número total de octetos que forman el datagrama, incluida la cabecera.

**Checksum:** el uso del checksum es opcional, y este campo debe ponerse a cero si no es utilizado. Mientras que el checksum del datagrama IP solo tiene en cuenta la cabecera del mensaje, el UDP tiene su propio checksum para garantizar la integridad de los datos. La longitud de este campo es de 16 bits, y esta formado por la suma de los campos del UDP, y algunos de los campos del IP. Para incluir los campos del IP, se construye una pseudo cabecera UDP. Esta pseudo cabecera de 12 octetos se utiliza únicamente a efectos de calcular la suma.

**TCP** El protocolo TCP proporciona un servicio de comunicación que forma un circuito. TCP proporciona un circuito virtual el cual es llamado una conexión.

Al contrario de los programas que utilizan UDP, los que utilizan TCP tienen un servicio de conexión entre los programas llamados y los que llaman, un chequeo de errores, control de flujo y capacidad de interrupción.

**Interfaces TCP:** existen dos tipos de interfaces entre la conexión TCP y los otros programas.

El primero es utilizar la pila de los programas de la capa de red. Como en esta capa solo esta el protocolo IP, la interface lo determina este protocolo. El segundo tipo es la interfaz del programa de usuario. Esta interfaz puede variar según el sistema operativo, pero en general tiene las siguientes características:



La interfaz envuelve el programa de usuario llamando a una rutina que introduce entradas en una estructura de datos llamada bloque de control de transmisión (TCB). Las entradas se realizan inicialmente en la pila de hardware y transferidas al TCB por medio de una rutina de sistema. Estas entradas permiten al TCP asociar un usuario con una conexión en particular, de modo que pueda aceptar comandos de un usuario y mandarlos a otro usuario en la otra parte de la conexión. TCP utiliza unos identificadores únicos para cada parte de la conexión. Al usuario se le asigna un nombre de conexión para utilizarlo en futuras entradas del TCB. Los identificadores para cada extremo de la conexión se llaman sockets. El socket local se construye conectando la dirección IP de origen y el número de puerto de origen. El socket remoto se obtiene conectando la dirección IP de destino y el número de puerto de destino.

El par de sockets de una conexión forman un único número en internet. El UDP tiene los mismos sockets, pero no los recuerda. Esta es la diferencia entre un protocolo orientado a conexión y otro a no-conexión. A continuación se explican los comandos más usuales:

- **Open:** Inicia una conexión o comienza a escuchar un socket. El usuario tiene un nombre de conexión local que actúa como un puntero dentro del TCB.
- **Send:** El comando Send manda datos del buffer especificado.
- **Recibir:** El comando Recibir es un mensaje de error si el nombre local proporcionado no es utilizado antes con el comando Open.
- **Close:** El comando Close hace que se cierre una conexión. Se produce un error si la conexión especificada no ha sido abierta, o si no se tiene autorización para cerrar la conexión.
- **Status:** El comando Status solo tiene una variable asociada, que es el nombre de la conexión.
- **Abort:** El comando Abort hace que todos los comandos Send y Receive asociados al nombre de la conexión local se interrumpan. La entrada del usuario del TCB se elimina y se envía un mensaje especial de reinicio a la entidad del otro lado de la conexión.

El TCP recuerda el estado de cada conexión por medio del TCB. Cuando se abre una conexión, se efectúa una entrada única en el TCB. Un nombre de conexión se le asigna al usuario para activar los comandos de la conexión. Cuando se cierra una conexión se elimina su entrada del TCB.

**Control de flujo:** El protocolo TCP puede controlar la cantidad de datos que debe enviar mediante el campo "Window = Ventana". Este campo indica el número máximo de octetos que pueden ser recibidos. El receptor de un segmento con el campo window a cero, no puede enviar mensajes al emisor, excepto mensajes de prueba. Un mensaje de prueba es un mensaje de un solo octeto que se utiliza para detectar redes o host inalcanzables.

**Formato del segmento TCP:** El segmento TCP consiste en una cabecera y datos. A continuación se describen los campos del segmento TCP.



- Número de puerto del Origen/Destino (Source/Destination Port Numbers): Este campo tiene una longitud de 16 bits.
- Números de Secuencia (Sequence Numbers): Existen dos números de secuencia en la cabecera TCP. El primer número de secuencia es el número de secuencia inicial (SSN) El SSN es el número de 32 bits. El otro número de secuencia es el número de secuencia esperado por la recepción, (también llamado número de reconocimiento "acknowledgement number")
- Longitud de la cabecera: Este campo tiene una longitud de 4 bits y contiene un entero igual al número de octetos que forman la cabecera TCP dividido por cuatro.
- Código de Bits (Code Bits): El motivo y contenido del segmento TCP lo indica este campo. Este campo tiene una longitud de 6 bits.
- Window: Este campo contiene un entero de 32 bits. Se utiliza para indicar el tamaño del buffer disponible que tiene el emisor para recibir datos.
- Opciones (Options): Este campo permite que una aplicación negocie durante la configuración de la conexión característica como el tamaño máximo del segmento TCP. Si este campo tiene el primer octeto a cero, esto indica que no hay opciones.
- Relleno (Padding): Este campo consiste en un número de octetos (De uno a tres), que tienen valor cero y sirven para que la longitud de la cabecera sea divisible por cuatro.
- Checksum: Mientras que el protocolo IP no tiene ningún mecanismo para garantizar la integridad de los datos, ya que solo comprueba la cabecera del mensaje. El TCP dispone de su propio método para garantizar dicha integridad. Como el checksum del protocolo TCP también se incluyen campos del protocolo IP, es necesario construir una pseudo-cabecera TCP que se considera únicamente a efectos de cálculo.

**Estados del TCP:** El inicio, mantenimiento y cierre de una conexión requiere que el TCP recuerde toda la información relativa a cada conexión. Esta información se almacena en una entrada para cada conexión dentro del TCB. Cuando se abre una conexión, la entrada en el TCB se realiza con todas las variables inicializadas con sus respectivos valores. Durante la conexión, la entrada del TCB es actualizada a medida que cambia la información.

- 0. CLOSED: No existe, solo para referencia.
- 1. LISTEN: esperando solicitud de conexión de un TCP remoto.
- 2. SYN-SEN: Esperando un mensaje de solicitud de conexión, después de haber enviado una solicitud de conexión.
- 3. SYN-RECEIVED: Esperando confirmación de un reconocimiento de solicitud de conexión, después de haber enviado y recibido una solicitud de conexión.
- 4. ESTABLISHED: Representa una conexión abierta. Los datos recibidos pueden ser enviados a un protocolo de una capa superior. Este es el estado normal de la fase de transferencia de la conexión.



- 5. FIN-WAIT-1: esperando la solicitud de fin de conexión de un TCP remoto, o un reconocimiento de una solicitud de fin de transmisión enviada anteriormente.
- 6. FIN-WAIT-2: Esperando una solicitud de fin de conexión de un TCP remoto.
- 7. CLOSE-WAIT: Esperando una solicitud de fin de conexión de un protocolo de capa superior.
- 8. CLOSING: Esperando el conocimiento de una solicitud de final de conexión de un TCP remoto.
- 9. LAST-ACK: Esperando el conocimiento de una solicitud de final de conexión enviada anteriormente al TCP remoto.
- 10. TIME-WAIT: Esperando el tiempo necesario para que el TCP remoto haya recibido el conocimiento de la solicitud del fin de conexión.

#### 2.3.3.4. Protocolos de transferencia de archivos simples.

FTP ("File Transfer Protocol = Protocolo de Transferencia de Archivos") es un programa de transferencia de archivo en entornos TCP/IP. FTP es un componente fundamental en los sistemas TCP/IP y se encuentra ubicado en el nivel de aplicación del modelo OSI. Su forma de trabajo se basa en el programa telnet y en TCP.

FTP se encuentra disponible para una gran variedad de sistemas y servidores como protocolo genérico de transferencia de archivos entre sistemas.

Las utilidades de FTP incorporan las siguientes:

- Soporte para archivos ASCII, EBCDIC, binarios y con formato.
- Capacidad de lectura y escritura para que los usuarios visualicen, creen y eliminen archivos y directorios.
- Protección de contraseñas.
- Compresión de datos.

Los archivos se transfieren como archivos de texto ASCII a menos que se elija un modo de transferencia diferente. Hay que escribir "binary" antes de escribir la orden para transferir un archivo binario, y escribir ascii para realizar la transferencia de un archivo de texto. Existen dos modos de transferencia utilizados: el modo de transferencia por flujo, envía las operaciones sin más operaciones; mientras que el modo de bloques realiza una verificación de errores en los bloques de datos.

#### 2.3.3.5. Protocolos de acceso a red.

**ARP** El protocolo ARP ("Address Resolution Protocol = Protocolo de resolución de dirección"), es el encargado de convertir las direcciones IP en direcciones de la red física.

El funcionamiento del protocolo ARP es bastante simple. Cuando una máquina desea enviar un mensaje a otra máquina que esta conectada a través de una red ethernet se



encuentra con un problema: la dirección IP de la máquina en cuestión es diferente a la dirección física de la misma. La máquina que quiere enviar el mensaje sólo conoce la dirección IP del destino, por lo que tendrá que encontrar un modo de traducir la dirección IP a la dirección física. Esto se hace con el protocolo ARP.

Este protocolo utiliza una tabla denominada Tabla de Direcciones ARP, que contiene la correspondencia entre direcciones IP y direcciones físicas utilizadas recientemente. Si la dirección buscada no está en la tabla el protocolo ARP envía un mensaje a toda la red. Cuando una computadora reconoce su dirección IP envía un mensaje de respuesta que contiene la dirección física. Cuando la máquina origen recibe este mensaje ya puede establecer la comunicación con la máquina destino, y esta dirección física se guarda en la tabla de direcciones ARP.

**Formato del mensaje ARP:** El mensaje ARP está formado por 28 octetos. En los campos que se describen a continuación se supone una interfaz ethernet.

**Tipo de Hardware:** El campo hardware indica el tipo de interfaz de hardware. Por ejemplo, el valor de una red ethernet es 1.

Tipo	Descripción
1	Ethernet (10mb)
2	Experimental Ethernet (3mb)
3	Amateur Radio X.25
4	Porteon ProNET Token Ring
5	Chaos
6	IEEE 802 Network
7	ARCNET
8	Hyperchannel
9	Lanstar
10	Autonet Short Address
11	LocalTalk
12	LocalNet

ESTA RESOLUCIÓN CALIFICA  
DE LA ENFERMERÍA

**Números de Protocolo:** El campo protocolo identifica el protocolo Ethernet usado. Por ejemplo el valor de la interfaz Ethernet es 0800 hex.

**Longitud de la dirección de Hardware:** El valor para Ethernet es de 6, lo que proporciona 48 bits para una dirección Ethernet (12 semi octetos)

**Longitud del protocolo:** este campo se usa para definir la longitud de la dirección de red. Para una red IP es 4.

**Operación:** Especifica el código de la operación. La solicitud ARP tiene valor 1, y la respuesta ARP tiene valor 2.

**Dirección Hardware del Origen:** Los campos de dirección del Hardware de origen,



Dirección IP del Origen, y Dirección IP del Destino los completa el emisor (si los conoce). El receptor añade la dirección del hardware del destino y devuelve el mensaje al emisor con el código de operación 2 (El código de la respuesta ARP)

### 2.3.3.6. Arquitectura TCP/IP.

TCP/IP ("Transmisión Control Protocol/Internet Protocol= Protocolo de Control de Transmisión/Protocolo Internet"), es junto con OSI una arquitectura de protocolos que ha sido determinante y básica en el desarrollo de los estándares de comunicación. Es la arquitectura más adoptada para la interconexión de sistemas. Al contrario de lo que ocurre con OSI, el modelo TCP/IP es software, es decir, es un modelo para ser implementado en cualquier tipo de red. Facilita el intercambio de información independientemente de la tecnología y el tipo de subredes a atravesar, proporcionando una comunicación transparente a través de sistemas heterogéneos. Por todo esto, TCP/IP no define una capa física ni de enlace; este protocolo define solamente tres capas que funcionarán en los niveles superiores a las capas físicas y de enlace para hacerlo así un modelo independiente del hardware en el que se implemente.

La arquitectura de un sistema en TCP/IP tiene una serie de metas:

- La independencia de la tecnología usada en la conexión a bajo nivel y la ~~arquitectura de la computadora.~~
- Conectividad universal a través de la red.
- Reconocimientos de extremo a extremo.
- Protocolos estandarizados.

### Estructura interna

El modelo básico en internet es el modelo Cliente/Servidor. El cliente es un programa que le solicita a otro que le preste un servicio. El Servidor es el programa que proporciona este servicio.

La arquitectura de Internet está basada en capas. Esto hace más fácil implementar nuevos protocolos. El conjunto de los protocolos TCP/IP, al estar integrado plenamente en internet, también dispone de este tipo de arquitectura. El modelo de capas de TCP/IP es algo diferente al propuesto por ISO para la interconexión de los sistemas abiertos.

**Capa Internet(IP):** En situaciones en las que dos dispositivos estén conectados a redes diferentes, se necesitarán una serie de procedimientos para permitir que los datos atraviesen las diferentes redes interconectadas. Esta será la función de esta capa. El protocolo internet(IP) se utiliza en esta capa para ofrecer el servicio de encaminamiento a través de varias redes.

**Capa de origen-destino o de transporte (TCP):** Sería deseable asegurar que todos los datos llegaran a la aplicación destino y en el mismo orden en el que fueron enviados.



Los mecanismos necesarios para ofrecer la seguridad son esenciales, independientemente de la naturaleza de la aplicación. El protocolo TCP es el más utilizado para proporcionar estas funciones.

**Capa de aplicación:** Contiene toda la lógica necesaria para llevar a cabo las aplicaciones de usuario. Para cada tipo específico de aplicación, como es por ejemplo la transferencia de un archivo, se necesitará un módulo en particular dentro de esta capa.

### ARQUITECTURA TCP/IP                      EQUIVALENTE / OSI

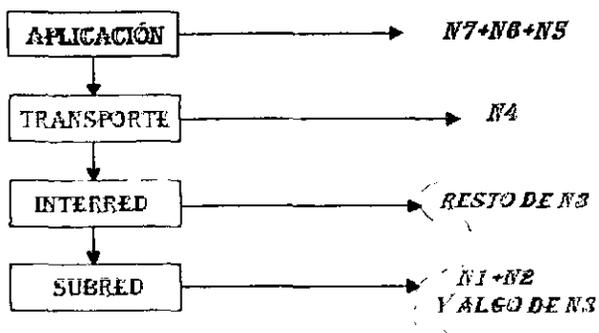


Figura 2.3.3.6.1. Arquitectura TCP / IP y su equivalente OSI

### 2.3.3.7. Modelo OSI.

La necesidad de intercambiar información entre sistemas heterogéneos, ejemplo, entre sistemas cuyas tecnologías son muy diferentes entre sí, llevó a la ISO ("International Standard Organization = Organización de Estándares Internacionales"), a buscar la manera de regular dicho intercambio de información. El modelo de referencia OSI ("Open Systems Interconnection = Interconexión de Sistemas Abiertos"), surge en el años de 1983 y es el resultado del trabajo de la ISO para la estandarización de los protocolos de comunicación.

El modelo OSI consta de 7 capas o niveles. Las características generales de las capas son las siguientes:

- Cada una de las capas desempeña funciones bien definidas.
- Los servicios proporcionados por cada nivel son utilizados por el nivel superior.
- Existe una comunicación virtual entre 2 mismas capas, de manera horizontal.
- Existe una comunicación vertical entre una capa de nivel N y la capa de nivel N+1.
- La comunicación física se lleva a cabo entre las capas de nivel 1.

Las 7 capas del modelo OSI y sus funciones son:



1. Capa Física
  - Transmisión del flujo de bits a través del medio. No existe estructura alguna.
  - Maneja voltajes y pulsos eléctricos.
  - Especifica cables, conectores y componentes de interfaz con el medio de transmisión.
2. Capa de Enlace de Datos
  - Estructura el flujo de datos de bits bajo un formato predefinido llamado trama.
  - Para formar una trama, el nivel de enlace agrega una secuencia especial de bits al principio y al final del flujo inicial de bits.
  - Transfiere tramas de una forma confiable libre de errores (utiliza reconocimientos y retransmisión de tramas).
  - Provee control de flujo.
3. Capa de Red (Nivel de paquetes)
  - Divide los mensajes de la capa de transporte en paquetes y los ensambla al final.
  - Utiliza el nivel de enlace para el envío de paquetes: un paquete es ~~encapsulado en una trama~~.
  - Enrutamiento de paquetes.
  - Envía los paquetes de nodo a nodo usando ya sea un circuito virtual o como datagramas.
  - Control de congestión.
4. Capa de Transporte
  - Establece conexiones punto a punto sin errores para el envío de mensajes.
  - Permite multiplexar una conexión punto a punto entre diferentes procesos del usuario (puntos extremos de una conexión).
  - Provee la función de difusión de mensajes(broadcast) a múltiples destinos.
  - Control de flujo.
5. Capa de Sesión
  - Permite a usuarios en diferentes máquinas establecer una sesión.
  - Una sesión puede ser usada para efectuar un login a un sistema de tiempo compartido remoto, para transferir un archivo entre dos máquinas.
  - Controla el diálogo (quién habla, cuándo, cuánto tiempo, comunicación unidireccional y bidireccional).
  - Función de sincronización.
6. Capa de Presentación



- Establece una sintaxis y semántica de la información transmitida.
- Se define la estructura de los datos a transmitir( define los campos de un registro: nombre, dirección, teléfono, etc).
- Define el código a usar para representar una cadena de caracteres (ASCII, EBCDIC, etc).
- Compresión de datos.
- Criptografía.

#### 7. Capa de Aplicación

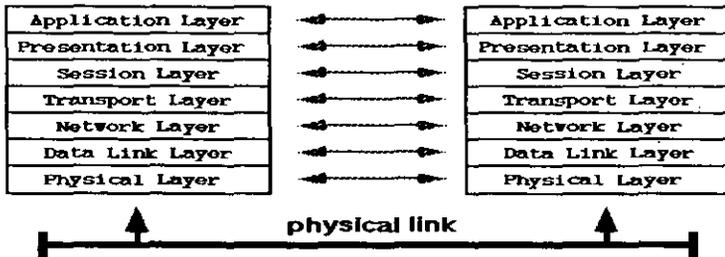
- Transferencia de archivos.
- Login remoto.
- Correo electrónico.
- Acceso a bases de datos, etc.

MODELO OSI	MODELO TCP / IP
Application	DNS. FTP. HTTP, SNMP, SMTP, TELNET etc.
Presentation	
Session	
Transport	TCP. UDP
Network	IP
Data Link	(Uses various other protocols)
Physical	(Uses various other protocols)

Figura 2.3.6.1. Capas del Modelo OSI en comparación con TCP / IP

#### 2.3.3.8. Servicio en modo conexión.

Una red orientada a conexión es aquella en la que inicialmente no existe una conexión lógica entre los elementos de la red. Es decir la conexión entre los elementos se encuentra en estado libre.



Las redes no orientadas a la conexión (también llamadas datagramas) pasan directamente del estado libre al modo de transferencia de datos, finalizando con el retorno al estado libre. La principal diferencia con la clase anterior es la ausencia de las fases de establecimiento y liberación. Además, las redes de este tipo no ofrecen confirmaciones, control de flujo ni recuperación de errores aplicables a toda la red, aunque estas funciones si existen para cada enlace en particular. Evidentemente el costo de una red no orientada a conexión es mucho mayor.

El compromiso de una red orientada a la conexión y una red no orientada a la conexión radica en la sobrecarga frente a las funciones obtenidas. Una red orientada a conexión ofrece muchas funciones, aunque ello se traduce en un mayor costo del sistema. Por el contrario, una red no orientada a conexión resulta más económica, ya que las funciones de apoyo que ofrece el proceso de aplicación del usuario son limitadas.

### 2.3.3.9. Suite de protocolo TCP/IP.

#### Aplicación y utilidades

Capa de aplicación. Esta capa corresponde a las aplicaciones que están disponibles para los usuarios.

**BOOTP** ("Bootstrap Protocol = Protocolo detección de arranque")

#### Información general

En lugar de utilizar el protocolo ARP, una máquina que acaba de ponerse en funcionamiento por primera vez, puede utilizar el protocolo bootstrap para obtener la dirección IP e información sobre su sector de arranque. Este método tiene algunas ventajas respecto al del protocolo ARP, por ejemplo, el BOOTP no requiere un acceso directo a la red.



## Formato del mensaje

### Descripción de los campos:

- Tipo (Type): Este campo identifica si el mensaje es una solicitud o una respuesta.
- Cabecera (Header): Este campo identifica el tipo de dirección de hardware.
- Longitud – H (H-Length): Este campo identifica la longitud de la dirección de hardware en octetos.
- Contador de saltos (Hop count): Se utiliza cuando el protocolo BOOTP se utiliza a través de varios Gateways. Cada paso por un Gateway aumenta en uno el contador.
- ID de Transacción (Transacción ID): Lo utiliza la estación de trabajo para asignar las respuestas a las solicitudes.
- Segundos (Seconds): Se utiliza para calcular el tiempo transcurrido desde el envío de la solicitud hasta la recepción de la respuesta.
- Dirección IP del cliente (Client IP address): Este campo lo completa el cliente, si la conoce. En otro caso se pone a cero.
- Dirección IP del servidor (Server IP Address): Puede ser introducido por el cliente, si la conoce. Cuando el valor es diferente de cero, solo el servidor especificado puede contestar a la solicitud. Esta es una forma de forzar al servidor para que proporcione la información de arranque.
- Dirección IP del Gateway (Gateway IP address): Este campo lo pone a cero el cliente, y si la solicitud la obtiene un gateway, este escribe su dirección en este campo.
- Dirección de Hardware del Cliente (Cliente Hardware Address): Este campo lo completa el cliente.
- Nombre del servidor Host (Server Host Name): Este campo es opcional, y puede ponerlo a cero tanto el servidor como el cliente.
- Nombre del archivo de arranque (Boot File Name): Puede ponerlo a cero el cliente, o poner un nombre genérico. El servidor reemplazara este campo por la ruta completa del archivo completo.
- Area del fabricante (Vendor-specific area): puede tener un código escrito por el cliente.



### Formato del mensaje BOOTP

Octet + 0								Octet +1								Octet +2								Octet +3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Type								Header Type								H-Length								Hop Count							
Transaction ID																															
Seconds																Zero															
Client IP Address																															
Response IP Address																															
Server IP Address																															
Gateway IP Address																															
Client Hardware Address (16 Octets)																															
Server Host Name (64 Octets)																															
Boot File Name (128 Octets)																															
Vendor-Specific Area (64 Octets)																															

Tabla 2.3.3.9.1. Formato del mensaje BOOTP.

NTP ("Network Time Protocol = Protocolo tiempo de red")

El NTP se utiliza para sincronizar los servidores en nanosegundos.

### Formato del mensaje

El mensaje NTP esta formado por los siguientes campos:

Indicador de Ajuste (Leap Indicator) (LI): Es un campo de 2 bits que indica el ajuste debido al periodo de rotación de la tierra.

Valor	Significado
00	Sin advertencias
01	-1 segundo
10	+1 segundo
11	Condición de alarma (Reloj no sincronizado)

Tabla 2.3.3.9.2. Indicador de ajuste.

- Numero de versión (Versión Number) (VN): Es un campo de 3 bits que indica el número de versión.
- Reservado (Reservado): Es un campo de 3 bits, que tiene valr cero.

Estrato (Stratum): Este campo tiene una longitud de 8 bits, y se utiliza para indicar el estrato local del reloj.



Valor	Significado
0	Sin especificar
1	Referencia primaria
2-n	Referencia secundaria (vía NTP)

Tabla 2.3.3.9.3. Estrato.

- Poll: Este campo tiene una longitud de 8 bits. Indica el intervalo máximo de tiempo entre mensajes.
- Precisión: Este campo tiene una longitud de 8 bits e indica la precisión del reloj local.
- Distancia de Sincronía (Sincroniza distance): Este es un campo de 32 bits, que indica el retraso aproximado de la primera ruta de sincronización.
- Nivel de velocidad aproximado (Estimated Drift Rate) : Es un campo de 32 bits que indica el nivel de velocidad del reloj local.
- Identificador del reloj de referencia(Reference Clock Identifier): Campo de 32 bits que indica un reloj de referencia particular.

Valor	Código	Significado
0	DCN	Determinado por el algoritmo DCN
1	WWVB	Radio Reloj WWVB (60 KHz)
1	GOES	Reloj de satélite GOES (450 Mhz)
1	Radio Reloj WWV	WWV(5/10/15 MHz)

Tabla 2.3.3.9.4. Identificador de Reloj.

Fecha y Hora (Timestamps): Existen 3 Timestamps (Fecha y Hora) de 64 bits cada uno.

**SNMP** ("Simple Network Management Protocol = Protocolo simple de administración de red")

El protocolo SNMP se utiliza para administrar múltiples redes físicas de diferentes fabricantes, es decir internet, donde no existe un protocolo común en la capa de enlace.

La estructura de este protocolo se basa en utilizar la capa de aplicación para evitar el contacto con la capa de enlace.

**IGMP** El IGMP ("Internet Group Management Protocol = Protocolo de administración de grupos") es un protocolo que funciona como una extensión del protocolo IP.

Se utiliza exclusivamente por los miembros de una red multicast para mantener su status de miembros, o para propagar información de direccionamiento.

Un Gateway multicast manda mensajes una vez por minuto como máximo. Un host



receptor responde con un mensaje IGMP, que marca al host como miembro activo. Un host de direccionamiento de la red multicast.

**Repetidores.** En las redes de computadoras las señales de comunicación se atenúan cuando pasan por algún medio de transmisión, esto debido a las características mismas del medio. Sin embargo, se puede amplificar la señal utilizando un dispositivo de amplificación llamado repetidor.

**Concentradores.** Se utiliza como un lugar central donde conectar las estaciones de trabajo y de este modo gestionar más fácilmente la red; para lograr esta gestión existen tres tipos de concentradores los cuales se describen a continuación:

**Concentrador pasivo.** Conecta entre sí varios elementos de la red. Pero no se realiza amplificación de la señal, y cada computadora de la red recibe la información enviada por todas las demás computadoras.

**Concentrador activo.** Además de generar la señal, también realizan las funciones como selección de las rutas de transmisión óptimas, y de esta manera contar con diferentes rutas para la entrega de información.

**Puentes.** Es un elemento de la red que permite el paso de señales entre segmentos de red. Las funciones que realizan:

- a) Recibir todas las señales enviadas de un segmento de red.
- b) Filtrar las señales que están direccionadas a otros nodos en ese mismo segmento de red.
- c) Retransmitir todas las demás señales a través del puerto apropiado.
- d) Realizar las mismas funciones con los datos de otros segmentos conectados.

Los puentes cumplen con estas tareas mediante la determinación de la localidad física de las computadoras origen y destino en la red.

**Routers.** Los routers, o ruteadores son elementos de la red que sirven para conexión de dos o más redes separadas lógicamente (subredes).

Los ruteadores se utilizan cuando se debe mantener separada la información de distintas subredes, pero al mismo tiempo tener enlaces ocasionales entre ellas; cada subred tiene asignada una dirección de red diferente.

Los ruteadores realizan una función muy similar a la de un puente. Típicamente, los ruteadores tienen un procesamiento más intenso que los puentes. Como resultado, sus velocidades de procesamiento no son tan altas. Por otro lado, los ruteadores tienen una capacidad mucho más sofisticada de selección de rutas óptimas de transmisión.



**Brouters.** Los brouters son esencialmente ruteadores que también pueden hacer la conexión entre redes. Un brouter checa primero si puede soportar el protocolo que está siendo utilizado por la información de entrada, si el protocolo no es soportado el brouter transmite la información usando información del direccionamiento físico.

**Switch Digital.** En la actualidad para construir redes de área amplia se busca utilizar una infraestructura de conmutación de paquetes (switches), con tecnologías como X.25, FrameRelay y actualmente con ATM ("Asynchronous Transfer Mode = Modo de Transferencia Sincronía") de conmutación de celdas.

En este esquema todos los ruteadores funcionan como equipos de acceso a una red conmutada, y cada uno de los sitios se conectan al proveedor de servicio y de esta manera hacen uso de la infraestructura de telecomunicaciones para transmitir otro tipo de información.



## 2.4. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE MICROSOFT SQL SERVER.

### Generalidades

El Microsoft SQL Server es un sistema Manejador de Bases de Datos Relacionales (RDBMS – **Relational Database Management System**) inteligente, basado en la arquitectura cliente/servidor, que soporta aplicaciones de procesamiento de transacciones en línea y brinda apoyo en la toma de decisiones a través de múltiples ambientes de red.

Este software combina el poder, la escalabilidad y manejabilidad del sistema operativo Windows NT. La seguridad y garantía del procesamiento de transacciones en SQL Server lo hacen ideal para las más exigentes aplicaciones de misión crítica.

Las capacidades avanzadas de SQL Server tales como disparadores, procedimientos almacenados, reglas, **defaults** y su tecnología de llamado a procedimiento remoto (RPC), hacen de SQL Server la herramienta ideal para el desarrollo de aplicaciones dorsales en estaciones de trabajo, minicomputadoras y **mainframes**. Su arquitectura diseñada para el alto rendimiento, multitarea, asegura un rendimiento consistente y estable que se mantiene aún cuando el número de usuarios aumenta.

Los puentes SQL Server (**gateways**), también proveen la tecnología clave para la integración de microcomputadoras con minicomputadoras y mainframes de larga escala.

Microsoft SQL Server 7.0 constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, continuando con la base sólida establecida por SQL Server 6.5. Como la mejor base de datos para Windows NT, SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software que construyen aplicaciones de negocios.

Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

### Requerimientos de SQL Server

- Sistema utilizando un procesador Intel Pentium o Alpha AXP.
- Microsoft Windows NT Server, 3.51 o posterior (requiere NT Server Enterprise 4.0 para utilizar Microsoft Cluster Server, el soporte para sistemas SMP y memoria adicional)
- 32 MB de memoria (64 MB con replicación)
- 80 MB de espacio disponible en disco (95 MB si se instalan los manuales en línea)
- Unidad de CD-ROM.



### **Soporte en red.**

Las siguientes redes están soportadas utilizando protocolos nativos:

- Microsoft Windows NT Server
- Microsoft LAN Manager
- Novell NetWare
- Redes basadas en TCP/IP
- IBM LAN Server
- Banyan VINES
- Digital PATHWORKS
- AppleTalk

### **Clientes soportados.**

- Sistema Operativo Microsoft Windows versión 3.1
- Microsoft Windows 95
- Microsoft Windows para Trabajo en Grupo
- Microsoft Windows NT Workstation
- Sistema Operativo Microsoft MS-DOS
- UNIX
- Apple Macintosh
- IBM OS/2

### **Lanzamientos SQL Server Recientes**

Esta sección provee una historia concisa de los lanzamientos SQL Server recientes. Una historia completa del desarrollo de SQL Server, desde sus comienzos hasta el lanzamiento del 6.5 se encuentra disponible en Dentro del Microsoft SQL Server 6.5, de Ron Soukup, publicado por Microsoft Press, ISBN 1-57231-331-5.

- El Standard Edition de SQL Server fue lanzado en abril de 1996.
- El Enterprise Edition fue lanzado en diciembre de 1997. Se incluyeron características adicionales en esta edición tales como soporte para Microsoft Cluster Server, sintonización de 4 GB RAM, English Query y soporte para sistemas de hasta 8 procesadores.
- El Service Pack actual para SQL Server 6.5 es SP4, lanzado en diciembre de 1997. SP3 fue lanzado en junio de 1997, SP2 en diciembre de 1996 y SP1 en agosto de 1996.
- Beta 1 fue lanzado en junio de 1997 a 200 clientes. Este grupo incluía un número limitado de proveedores Independientes de software (ISV), autores de libros, diseñadores de materiales para cursos y algunas cuentas corporativas. No se pusieron copias a disposición de la prensa o analistas. Este lanzamiento enfocó las pruebas de funcionalidad de bajo nivel y programación de interfaces.
- Beta 2 fue lanzado a finales de diciembre de 1997 a 3000 clientes. El cubrimiento de las cuentas corporativas y de la comunidad ISV fue incrementado



ampliamente, y se agregaron cuentas internacionales. Se entregaron copias de Beta 2 a la prensa y a los analistas en el Taller de Examinadores.

- La versión Beta de Mercadeo fue lanzada el segundo trimestre de 1998 con alta disponibilidad.
- El lanzamiento a fabricantes fue planeada para la segunda mitad del año 1998.
- Las ediciones Standard y Enterprise de SQL Server 7.0 serán lanzadas simultáneamente.

## Características de SQL Server

### El motor de Base de Datos para la Empresa.

Microsoft SQL Server revoluciona el concepto de Base de Datos para la Empresa. Reúne en un sólo producto la potencia necesaria para cualquier aplicación empresarial crítica junto con unas herramientas de gestión que reducen al mínimo el coste de propiedad.

- Con Microsoft SQL Server, la empresa tiene todo de serie.
- Miles de Soluciones Disponibles: Tendrá libertad de elección, ya que todas las aplicaciones de gestión del mercado corren sobre Microsoft SQL Server.
- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles.
- Empresas centralizadas u oficinas distribuidas, replicando cientos de sitios.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server. Posee los mejores registros de los **benchmarks** independientes (TCP) tanto en transacciones totales como en coste por transacción.
- Gestión: Con una completo interface gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

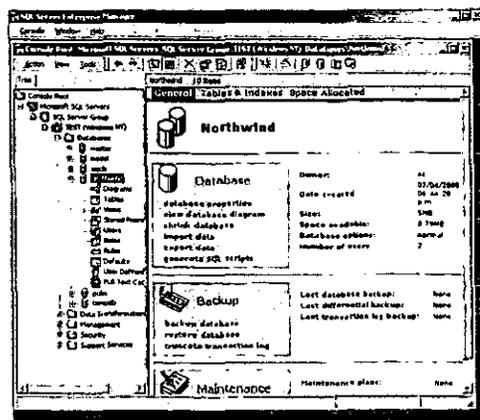


Figura 2.4.1 Ambiente gráfico de Microsoft SQL Server.



- Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.

### **La mejor base de datos para Internet, Intranet y Extranet.**

- Diseñada desde su inicio para trabajar en entornos Internet e Intranet, Microsoft SQL Server es capaz de integrar los nuevos desarrollos para estos entornos específicos con los desarrollos heredados de aplicaciones "tradicionales". Es más, cada aplicación que desarrollemos para ser empleada en entornos de red local puede ser utilizada de forma transparente -en parte o en su totalidad- desde entornos Internet, Intranet o Extranet.
- Plataforma de desarrollo fácil y abierta: integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server y con las mejores herramientas de gestión y desarrollo para Internet como FrontPage97, Microsoft Office97 y Visual Interdev.
- Diseñada para INTERNET: Es el único gestor de base de datos que contiene de forma integrada la posibilidad de generar contenido HTML de forma automática.
- La Base de Soluciones Integradas: La Integración total con BackOffice permite resolver toda las necesidades de infraestructura de la empresa con un sólo paquete.
- Potente y Escalable: Microsoft SQL Server es la única base de datos cuyo rendimiento sobre Internet está publicado, ofreciendo registros espectaculares.
- Mínimo coste de Propiedad: La sencillez de la instalación, y la potencia de sus herramientas de gestión y el menor coste de toda la industria para entornos Internet, hacen de Microsoft SQL Server la mejor opción con el menor coste.

### **Arquitectura RDBMS.**

- Arquitectura de servidor simétrico y paralelo con balanceo automático de carga en múltiples procesadores.
- Kernel multiproceso real para mejor rendimiento transaccional y escalabilidad.
- Soporte grandes bases de datos (VLDB) (+1 TB).
- Completo proceso transaccional interactivo con rollback automático y recuperación de roll-forward.
- Optimizador de consultas mejorado basado en coste.
- Puesta a punto mejorada para una mejor capacidad de datos y tiempo de respuesta.
- Soporte E/S asíncrono para acceso en paralelo a múltiples dispositivos de disco para una mejor capacidad.
- Bloqueo a nivel fila y página con escalación de bloqueos; resolución automática de bloqueos.



### Datos distribuidos y replicación.

- Llamadas a procedimientos remotos servidor-a-servidor (procedimientos almacenados remotos).
- Replicación asíncrona o continua basada en registros, o sincronización planificada de tablas en el momento.
- Configuración de replicación gráfica y características de gestión.
- Replicación de subscritores ODBC, incluyendo IBM DB2, ORACLE, SYBASE y Microsoft Access.
- El Distributed Transaction Coordinator gestiona transacciones que involucran a dos o más servidores SQL (proceso Two Phase Commit 2PC) transparente.
- Replicación de tipos de datos Texto e Imagen.

### Integración Internet y correo electrónico.

- MAPI, permitiendo aplicaciones de flujo de trabajo y notificación de cambio de datos automática.
- Compatibilidad con Microsoft Internet Information Server y otros servidores Web populares.
- SQL Web Assistant, para el retorno automático de datos en formato HTML.
- Procedimientos almacenados para generar páginas HTML o actualizar datos en plantillas Web.
- Posibilidad de poblar automáticamente carpetas públicas de Microsoft Exchange con datos.

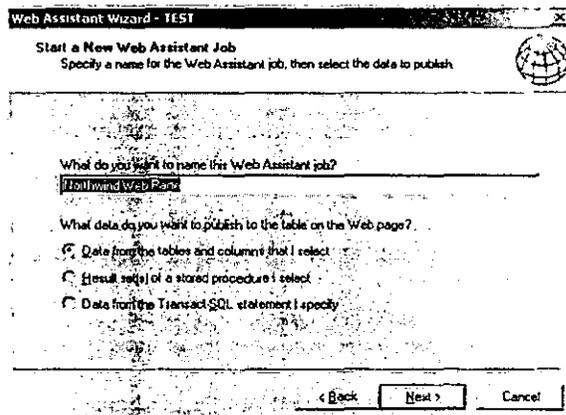


Figura 2.4.2. Asistente para Web de Microsoft SQL Server.

### Data Warehousing y amplio soporte de datos.

- Ejecución transaccional en paralelo, carga e indexación de datos, y verificación de integridad.



- Respaldo y restauración en paralelo de alta velocidad que soporta hasta 32 discos o dispositivos de cinta.
- Las extensiones de consulta OLAP, CUBE, y ROLLUP permiten conjuntos de resultados multidimensionales.
- EXEC e INTO ahora permiten almacenar en tablas resultados desde procedimientos almacenados.
- Copias de seguridad en cinta y recuperación de tablas individuales.
- Reconstrucción de índice sin realizar destrucción y recreación del índice.

### Gestión y administración centralizada de bases de datos.

- SQL Enterprise Manager, es una consola de gestión y motorización de 32-bits visual basada en Windows.
- Un único punto de configuración y gestión de control de datos remotas.
- SQL Executive, planificador de trabajos y monitor para gestión proactiva de servidores distribuidos.
- Operaciones remotas desatendidas mediante un "agente inteligente" que incluye evento-alerta- respuesta.
- Scripts Visual Basic a través de SQL-Distributed Management Objects (SQL-DMO) basados en OLE.
- DBA Assistant, para el mantenimiento automático rutinario en una única tarea planificada.

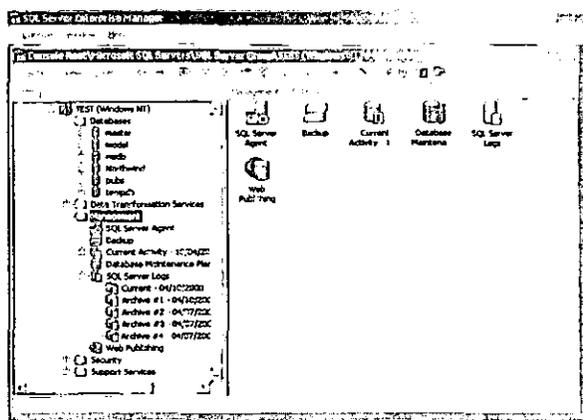


Figura 4.2.3. Administración grafica y fácil.

- SQL Trace, para monitorear consultas cliente-servidor mediante SQL almacenadas en archivos de registros.
- Soporte de MIB y traps SNMP para monitorear SQL Server desde herramientas de gestión basadas en SNMP.



### Disponibilidad, fiabilidad y tolerancia a fallos.

- Mirroring de dispositivos de base de datos con recuperación automática para tolerancia a fallos de dispositivos.
- Copias de seguridad en línea desatendidas garantizando la consistencia de datos para la más alta disponibilidad.
- Contextos de usuario protegidos, que pueden aislar los fallos a un proceso de un único usuario.
- Recuperación en el momento para restaurar bases de datos o transacción logs en un intervalo de tiempo.
- Tolerancia a fallos de servidor, permitiendo la recuperación automática a un servidor de respaldo o en espera.

### Mejoras en la programación y lenguaje.

- Disparadores, procedimientos almacenados, disparador de eventos antes y después de conexiones.
- Procedimientos almacenados extendidos (funciones definidas por el usuario) utilizando C/C++.
- Cursores basados en el motor con avance hacia adelante y atrás; posicionamiento absoluto y relativo.
- Sentencias DLL permitidas dentro de transacciones.
- Transacciones distribuidas dentro de interfaces DB-Library, ODBC, Transact-SQL, XA y OLE Transaction.
- Procedimientos almacenados OLE Automation.

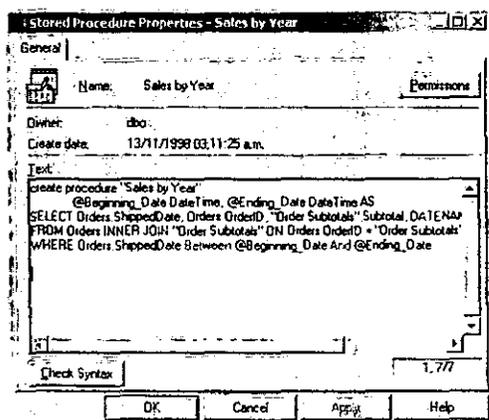


Figura 2.4.4. Propiedades de un Procedimiento Almacenado.



## Seguridad.

- Un único ID de usuario tanto para red como para la DB para mejorar la seguridad y facilitar la administración.
- Contraseña y encriptación de datos en red para mejorar la seguridad.
- Encriptación de procedimientos almacenados para la integridad y seguridad de código de aplicación.
- Interoperabilidad e integración con computadoras de escritorio.
- API estándar DB-Library totalmente soportada: estándar ODBC Nivel 2 totalmente soportado como API nativa.
- Gateway Open Data Services (ODS) programable para acceso transparente a fuentes de datos externas.
- Gateways de Microsoft y de terceros para fuentes de datos relacionales y no-relacionales, incluyendo IBM DB2.
- Soporte de importantes estándares de mercado como ANSI SQL-92, FIPS 127-2, XA, SNMP.

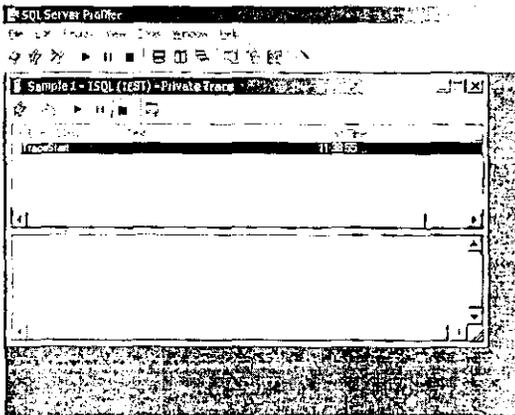


Figura 2.4.5. SQL Server Profile

## Ventajas de SQL Server

### Objetivos del Diseño de SQL Server

Los clientes están buscando soluciones para sus problemas de negocios. La mayoría de las "soluciones" de bases de datos solamente traen múltiples niveles de costos y complejidad. La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.

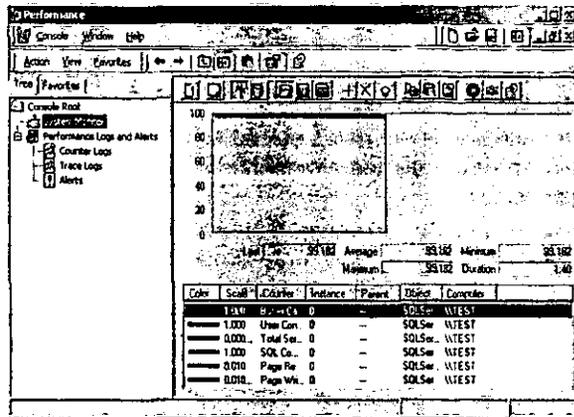


Figura 2.4.6. Monitor de rendimiento.

SQL Server 7.0 disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos; seguridad integrada; y scripting administrativo. Esta versión también libera al administrador de base de datos para aspectos más sofisticados del trabajo al automatizar las tareas de rutina. Al combinar estos poderosos servicios de administración con las nuevas características de configuración automática, Microsoft SQL Server 7.0 es la elección ideal de automatización de sucursales y aplicaciones de base de datos insertadas.

Los clientes invierten en sistemas de administración de bases de datos, en forma de aplicaciones escritas para esa base de datos y la educación que implica para la implementación y administración. Esa inversión debe protegerse: a medida que el negocio crece, la base de datos deberá crecer y manejar más datos, transacciones y usuarios. Los clientes también desean proteger las inversiones a medida que escalan aplicaciones de base de datos hacia equipos portátiles y sucursales.

Para cumplir con estas necesidades, Microsoft ofrece un motor de base datos único que escala desde una computadora portátil que ejecuta Windows 95 o Windows 98, hasta clusters de procesadores múltiples simétricos de terabyte que ejecutan Windows NT Server Enterprise Edition. Todos estos sistemas mantienen la seguridad y confiabilidad que exigen los sistemas de negocios de misión crítica.

El lanzamiento de Microsoft SQL Server 7.0 es una versión de baja memoria con capacidades de replicación de multi-sitio. Se ajusta muy bien a las necesidades cada vez mayores del mercado de la computación móvil. Las otras características tales como bloqueo a nivel de línea dinámico, el paralelismo intra-query, query distribuido, y mejoras para las bases de datos muy grandes (VLDB) hacen que el SQL Server 7.0 sea la elección ideal para sistemas OLTP de alta tecnología y sistemas de data warehousing.



Mientras los sistemas de procesamiento siguen siendo un componente clave para las infraestructuras de bases de datos corporativas, las compañías también están invirtiendo bastante en mejorar la comprensión que tienen de sus datos. La estrategia de Microsoft consiste en reducir el costo y la complejidad del Data Warehousing mientras hace que la tecnología sea más accesible a una mayor cantidad de público.

Microsoft ha establecido un enfoque total a todo el proceso de Data Warehousing (almacenamiento de datos). El objetivo es facilitar la construcción y diseño de soluciones de Data Warehousing costo efectivas a través de una combinación de tecnologías, servicios y alianzas con los proveedores.

La Alianza Microsoft para Data Warehousing es una coalición que une a los líderes en la industria de almacenamiento de datos y aplicaciones. El Microsoft Data Warehousing Framework constituye un conjunto de interfaces de programación diseñadas para simplificar la integración y administración de soluciones de Data Warehousing.

Las innovaciones del producto en SQL Server 7.0 mejoran el proceso de Data Warehousing: Servicios de Transformación de Datos; manejo mejorado de las consultas complejas y bases de datos muy grandes; procesamiento analítico en línea e integrado; y el Microsoft Repository. Otro componente esencial es el soporte extenso para integración de terceros.

Las innovaciones permiten que SQL Server 7.0 sea el líder en varias de las categorías de aplicación de rápido crecimiento en la industria de base de datos. Estas incluyen comercio electrónico, computación móvil, automatización de sucursales, aplicaciones de línea de negocios insertadas y mercados de datos.

#### **Las áreas de liderazgo e innovación en el Microsoft SQL Server 7.0 incluyen:**

- La primera base de datos en escalar desde la computadora portátil hasta la empresa utilizando la misma base de código y ofrecer el 100% de compatibilidad de código.
- La primera base de datos en soportar la auto-configuración y auto-sintonización Primera base de datos con OLAP integrado.
- La primera base de datos con Servicios de Transformación de Datos integrado
- El Data Warehousing Framework constituye el primer enfoque comprensivo al problema de metadatos.
- La primera base de datos en proveer administración de multi-servidor para cientos de servidores.
- La más amplia gama de opciones de replicación de cualquier base de datos.
- La mejor integración con Windows NT Server.
- La mejor integración con Microsoft Transaction Server.

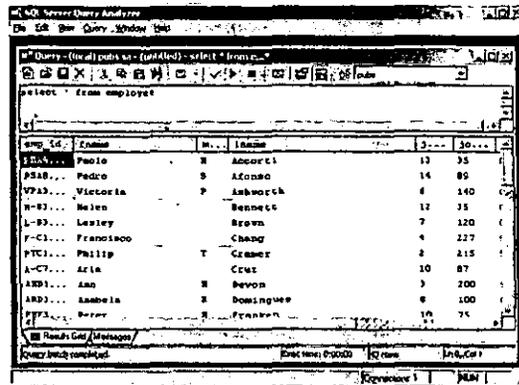


Figura 2.4.7. Analizador de queries.

### Desventajas de SQL Server

La instalación y operación requiere del Internet Explorer (IE) 4.0. Le guste o no, la interfaz del navegador de Web sigue siendo cada vez más habitual, y su uso es lo último en desarrollo de interfaces. Podemos entender por qué Microsoft quiere usarlo con el Servidor SQL, ya que también es un producto de la compañía. Sin embargo, ~~no tenemos ninguna utilidad para un navegador de Web en nuestro servidor de la base de datos~~, y su instalación es un problema que posiblemente, a más de uno le gustaría evitar. En nuestra opinión, la configuración de los canales de IE 4.0 para las características que nunca habíamos pensado usar nos parece algo frívolo.

La migración requiere un reinicio de la base de datos. El reinicio de todos los datos en una base de datos es un trabajo serio que invita a la potencial pérdida de datos. Cuanto más grande sea la base de datos, más onerosa será esta obligación. Sin embargo, después de mirar las herramientas de migración del SQL Server 7.0, es obvio que Microsoft se ha planteado esta operación como algo muy serio.

Ausencia de integridad referencial declarativa en cascada. La ausencia de una integridad referencial en cascada podría ser la desventaja más grande del Servidor SQL en comparación con las otras bases de datos dentro del mercado NT. Incluso Access ofrece soporte de este estilo. Se pueden utilizar disparadores para compensar esta desventaja, aunque en otras bases de datos esta técnica no es necesaria, así que no es lógico que deba utilizar para trabajar con SQL Server 7.0. Al considerar las otras nuevas características de SQL Server 7.0, es una pena que ésta no este incluida.

A continuación se describen algunas funciones interesantes del Microsoft SQL Server:

- Dentro del ambiente podemos existen objetos los cuales pueden ser una base de datos, una tabla, un procedimiento almacenado, etc., dichos objetos tienen propiedades las cuales podemos modificar de una manera muy sencilla como se ve en la figura 2.4.8. para el caso de una base de datos.

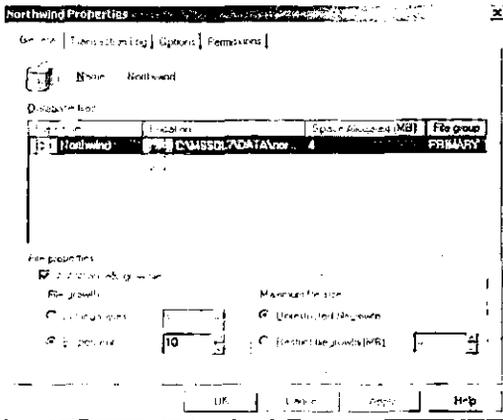


Figura 2.4.8. Propiedades de una Base de Datos.

- Otra parte interesante en la visualización grafica del la Base de Datos a través de un diagrama Entidad/Relación el cual se puede manipular e imprimir, este diagrama se genera automáticamente después de que se llevo a cabo la creación de una nueva Base de Datos.

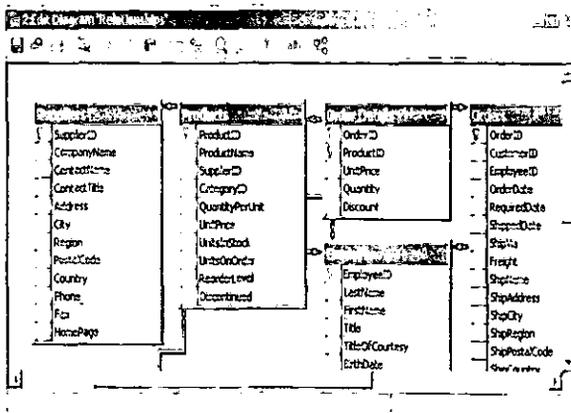


Figura 2.4.9. Diagrama Entidad/Relación de una Base de Datos.

- Dentro del ambiente de Microsoft SQL Server existe la posibilidad de utilizar una gran cantidad de asistentes o Wizards ya sea para la creación de Bases de Datos o para alguna otra tarea en específico. Todos estos asistentes nos llevan de la mano y en muy pocos pasos podemos hacer alguna tarea que de otra manera nos llevaría mucho tiempo. Existe un catálogo de Wizards dentro del SQL Enterprise Manager como lo podemos ver en la figura 2.4.10:

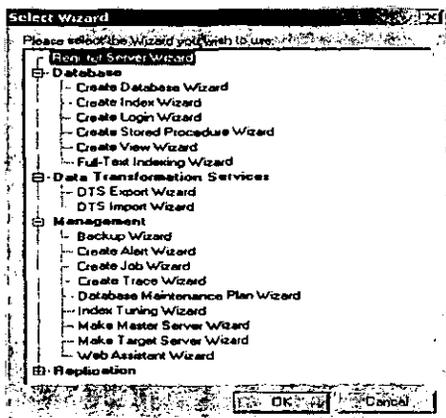


Figura 2.4.10 Catálogo de Wizards.

Para cada Wizard existe una pantalla principal que nos indica que es lo que podemos hacer con éste, dicha pantalla es como la que se muestra en la figura 2.4.11.

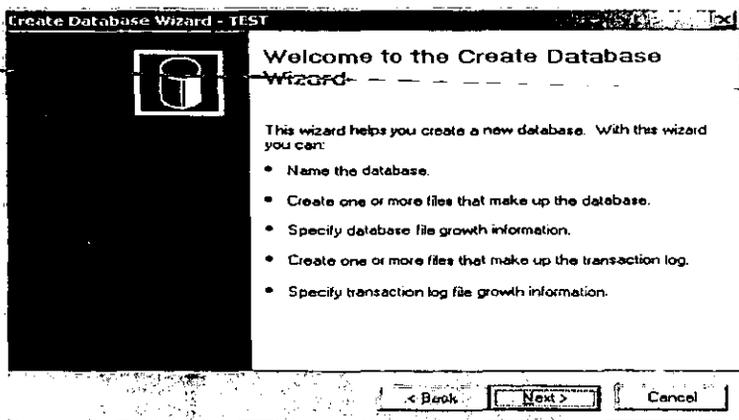


Figura 2.4.11. Pantalla principal de un Wizard.



## 2.5. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL FRONT END A UTILIZAR

Cualquier sistema informático cuenta generalmente con una metodología para el desarrollo y la implementación del mismo, estas metodologías, tienen un ciclo de vida; el cual garantiza su buen funcionamiento.

Un **Front End** (fachada) es una herramienta destinada a facilitar el desarrollo de sistemas, este front end automatiza esencialmente las etapas de planificación, análisis y diseño, es de las primeras etapas del desarrollo de un sistema.

Existen algunas características de los front end que facilitan el desarrollo de las aplicaciones de los sistemas de bases de datos, como lo son la gestión de memoria, la independencia de hardware, intercambio de datos entre aplicaciones, el entorno homogéneo, y que son multitarea. Estas características se dan porque el front end no tiene que controlar los periféricos ni manejar memoria, funciones que le corresponden al sistema operativo.

Microsoft Visual Basic 5.0 es un lenguaje de programación orientado a eventos, el cual tomaremos como el front end a utilizar para nuestro sistema, ya que cumple con las características necesarias para el desarrollo del mismo.

Visual Basic es el entorno más popular de la (GUI) **Graphical User Interface** (interfaz de usuario gráfica), mecanismo de comunicación entre el usuario y la aplicación. Es un sistema de desarrollo diseñado especialmente para crear aplicaciones con interfaz gráfica, de una forma rápida y sencilla. Para soportar este tipo de desarrollos, utiliza fundamentalmente dos herramientas, una que permite realizar los diseños gráficos y un lenguaje de alto nivel (en este caso QBasic o QuickBasic).

### 2.5.1. Requerimientos de hardware y software

Para las aplicaciones de Visual Basic se requiere el siguiente hardware y software:

- Microsoft Windows NT 3.51 o posterior, o Microsoft Windows 95 o posterior.
- Microprocesador PENTIUM o superior.
- Pantalla VGA o de resolución superior compatible con Microsoft Windows.
- 32 MB de RAM para aplicaciones (esto variará dependiendo de las bibliotecas de tipos o los archivos DLL específicos que se incluyan en las aplicaciones).
- 64 MB de RAM para el entorno de desarrollo de Visual Basic.
- Un **mouse** (ratón) u otro dispositivo de puntero
- Una unidad de CD-ROM.
- Un disco duro con un mínimo de 50 megabytes de espacio disponible para una instalación completa.



## 2.5.2. Características de Visual Basic 5.0

Para la realización de una aplicación con Visual Basic, se crean ventanas y sobre ellas se dibujan controles (etiquetas, botones, cajas de texto, etc.) y a continuación se escribe el código fuente relacionado con cada objeto. Esto es, cada objeto está ligado a un código que permanece inactivo hasta que se da el suceso que lo activa.

Visual Basic provee de herramientas que permiten crear ventanas y controles sin escribir código. También incluye un entorno de desarrollo que permite ejecutar todas las tareas de edición, ejecución y mantenimiento de programas de una forma fácil.

Las características de Visual Basic más sobresalientes son:

- Controles en tercera dimensión
- Soporte para intercambio de datos con aplicaciones diversas **DDE Dynamic Data Exchange** (Intercambio Dinámico de Datos).
- Soporte para establecer enlaces con rutinas escritas en varios lenguajes y con Windows DLL **Dynamic Link Libraries** (Librerías de Enlace Dinámicos)
- Una interfaz para múltiples documentos que permite crear una aplicación con una ventana de aplicación y múltiples ventanas de documento, por ejemplo Microsoft Word. **MDI Multiple Document Interface** (Interfaz de Documentos Múltiples)
- Soporte para objetos incrustados y vinculados **OLE Object Linking and Embedding** (Ligas a Objetos Embebidos). Lo que permite construcción de aplicaciones utilizando un software autocontenido en pequeñas y potentes unidades.
- Visualización y manipulación de datos con otras aplicaciones sobre Windows utilizando controladores OLE.
- Acceso y manipulación de objetos incrustados y vinculados suministrados por otras aplicaciones utilizando la automatización de OLE.
- Un compilador que nos permite añadir ayuda en línea creando herramientas de ayudas estilo Windows **HC Help Compiler** (Compilador de ayudas).
- Biblioteca para SQL, permitiendo manipular bases de datos relacionales tales como Microsoft Access **SQL Structured Query Language** (Lenguaje de Búsqueda Estructurada).
- Sistema de ayuda en línea.
- Acceso a bases de datos por medio de *Data Control* (Control de Datos) y a través de objetos de acceso a datos. Ambos pueden utilizar el acceso a través del motor de access o de controladores **ODBC Other Data Base Controller** (Controladores de Bases de Datos).
- Una biblioteca de iconos para trabajos profesionales.
- Manipulación de bases de datos en varios formatos, tales como: Microsoft Access, dBASE III y IV, etc. con el **Data Manager** (Administrador de Datos).
- Menús desplegables flotantes.
- Un asistente **Setup Wizard** (Asistente de Configuración), para crear los discos de distribución para sus aplicaciones.



- Un control que permite utilizar las cajas de diálogo más comúnmente utilizadas.
- Creación de aplicaciones de 16 bits y de 32 bits.
- Un generador de informes **Crystal Reports** (Reportes) para la creación de listados, informes y documentos a medida utilizando datos de una base.

El desarrollo de aplicaciones con Visual Basic se realiza básicamente en dos etapas:

- Interfaz con el usuario. Dentro de las aplicaciones nuevas ésta es la etapa inicial de diseño, pues se define la interfaz gráfica de la aplicación antes de escribir el código que determinará su comportamiento. En esta etapa es donde se trabaja con los diferentes elementos de Visual Basic que permiten crear la parte Visual del desarrollo de la aplicación, es decir, sólo se define la parte de los controles contenidos en las ventanas (denominadas comúnmente formas) de la aplicación.
- Código de las formas y controles. En esta parte es en la cual se introducirá el código necesario para el funcionamiento del programa.

En la figura 2.5.2.1 observamos la interfaz que se ve al arrancar Visual Basic 5.0, en esta interfaz se distinguen los siguientes elementos:

- Barra de menús. Visualiza las ordenes que se utilizan para desarrollar una aplicación.
- Barra de herramientas. Esta facilita el acceso a las órdenes más comunes utilizadas, al final de la barra de herramientas se visualizan la posición y el tamaño del objeto seleccionado.
- Ventana del proyecto. Es el conjunto de todos los ficheros (formularios, módulos recursos y clases) que forman una aplicación, el cual se denomina aplicación o proyecto.
- Ventana de propiedades. Cada objeto lleva asociado un conjunto de propiedades (nombre, posición, tamaño, color, etc.)
- Caja de herramientas. Provee de un conjunto de herramientas que permiten colocar los controles en un formulario durante el diseño de la interfaz gráfica de usuario.
- Formulario. Es la ventana sobre la que se colocan los controles de la interfaz gráfica que el usuario utiliza para comunicarse con la aplicación.

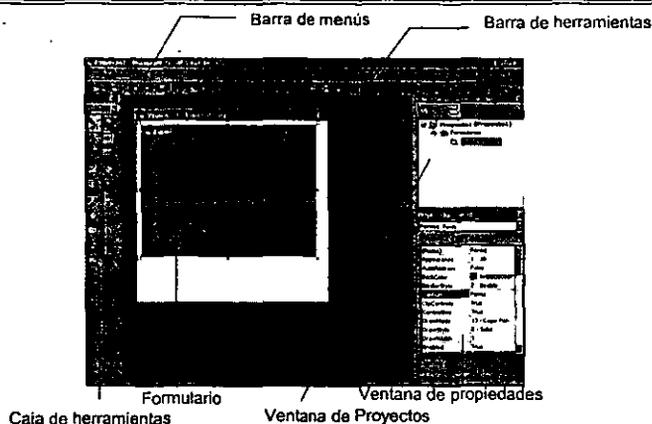


Figura 2.5.2.1. Interfaz al arrancar Visual Basic 5.0

Un único proyecto puede contener hasta 32.000 identificadores, que incluyen entre otros formularios, controles, módulos, variables, constantes, procedimientos, funciones y objetos. Los nombres de variables en Visual Basic no puede tener más de 255 caracteres y los nombres de formularios, controles, módulos y clases pueden tener un máximo de 40 caracteres. Visual Basic no impone ningún límite en cuanto al número de objetos distintos de un proyecto.

El número máximo de controles permitidos en un único formulario depende del tipo de controles que se utilicen y de los recursos disponibles del sistema, sin embargo hay un límite fijo de 254 nombres de control por formulario. En este límite una matriz de controles sólo cuenta como uno, ya que todos los controles en la matriz comparten un único nombre de control.

El límite de índices de matrices de controles es de 0 a 32.767 para todas las versiones de Visual Basic. Además no aceptará mas de seis niveles de controles anidados.

La tabla 2.5.2.2. muestra las limitaciones de propiedades aplicables a determinados controles de Visual Basic:



PROPIEDAD	APLICABLE A	LIMITACIÓN
List y ListCount	Controles de cuadro de lista y cuadro combinado	El número máximo de elementos es 32KB; el límite del tamaño de cada elemento es 1KB (1024 bytes).
Text	Control de cuadro de texto	El límite es de 64 KB.
Caption	Control de etiqueta	Limitado a 1024 bytes.
	Controles de botón de comando, casilla de verificación, marco y botón de opción	Limitados a 255 caracteres. Se truncan todos los títulos que superen ese límite. Los títulos en las propiedades de los controles personalizados tiene un límite de 32 KB.
	Control de menú	Limitado a 2356 caracteres.
Tag	Todos los controles	Limitado sólo por la memoria disponible.
Name	Todos los controles	Limitado a 40 caracteres.

Tabla 2.5.2.2. Limitaciones de propiedades para algunos controles de Visual Basic

La cantidad de código que se puede cargar en un formulario, clase o módulo estándar está limitada a 65.534 líneas. Una única línea de código puede contar de 1.023 bytes como máximo. Puede haber hasta 256 espacios en blanco delante del texto en una única línea y no se pueden incluir más de 25 caracteres de continuación de línea ( ) en una única línea lógica.

No hay límite en cuanto al número de procedimientos por módulo. Cada procedimiento puede contener hasta 64 KB de código. Si un procedimiento o módulo excede este límite, Visual Basic genera un error en tiempo de compilación. Si se encuentra con este error, se puede evitar dividiendo los procedimientos extremadamente grandes en varios procedimientos más pequeños o trasladando las declaraciones de nivel de módulo a otro módulo.

Visual Basic utiliza tablas para almacenar los nombres de los identificadores (variables, procedimientos, constantes, etc.) en el código. Cada tabla está limitada a 64KB.

Cada módulo de código y formulario utiliza una tabla que contiene una estructura que describe un punto de entrada de DLL. Cada estructura usa aproximadamente 40 bytes, con un límite total de tamaño de 64KB, lo que permite unas 15,000 declaraciones por módulo.

La aplicación entera utiliza una única tabla que contiene todos los nombres, estos incluyen:

- Nombres de constantes
- Nombres de variables
- Nombres definidos por el usuario de definición de tipo
- Nombres de módulos
- Nombres de declaración de procedimientos de DLL



La tabla de nombres del proyecto no tiene límite en cuanto al tamaño total, pero está limitada a un total de 32KB de entradas únicas con distinción entre mayúsculas y minúsculas. Si se llega a este límite, vuelva a utilizar identificadores privados en módulos diferentes para limitar el número de entradas únicas a 32KB.

Cada referencia a un identificador en un módulo diferente crea una entrada en la tabla de importación. Cada una de esas entradas tiene como mínimo 24 bytes y tiene un límite máximo de 64KB, lo que da como resultado unas 2.000 referencias por módulo. Una tabla de entradas de módulo acepta hasta 125 bytes por módulo, con un límite total de 64KB, que da como resultado unos 400 módulos por proyecto.

Las siguientes limitaciones son aplicables a las variables en el lenguaje Visual Basic.

- Cada formulario, módulo estándar y módulo de clase tiene su propio segmento de datos que puede ser como máximo de 64KB. Este segmento de datos contiene los siguientes datos:
  - Variables locales declaradas con **Static** (Estáticas).
  - Variables a nivel de módulo que no sean matrices y cadenas de longitud variable.
  - 4 bytes para cada matriz a nivel de módulo y cadena de longitud variable.
- Si un procedimiento o un módulo excede el límite de 64KB, Visual Basic generará un error de tiempo de compilación. Si se encuentra con este error, puede evitarlo dividiendo los procedimientos extremadamente grandes en varios procedimientos más pequeños o trasladando las declaraciones a nivel de módulo a otro módulo.
- Ninguna variable de un tipo definido por el usuario puede exceder los 64KB, aunque la suma de las cadenas de longitud variable en un tipo definido por el usuario puede exceder de 64KB (las cadenas de longitud variable sólo ocupan 4 bytes cada una en el tipo definido por el usuario; el contenido real de una cadena se almacena por separado). Los tipos definidos por el usuario se pueden definir en términos de otros tipos definidos por el usuario, pero el tamaño total de los tipos no puede exceder los 64KB.
- Los argumentos y las variables locales en los procedimientos ocupan espacio de pila en tiempo de ejecución. Las variables estáticas y a nivel de módulo no ocupan espacio de pila porque se encuentran en el segmento de datos para los formularios o los módulos. Todos los procedimientos de DLL a los que se llame utilizan esta pila mientras se están ejecutando. Visual Basic utiliza parte de la pila para sus propios usos, como el almacenamiento de valores intermedios al evaluar expresiones.

Algunas limitaciones de Visual Basic y las aplicaciones creadas con él están impuestas por Microsoft Windows. Estas limitaciones pueden cambiar cuando se instala una versión diferente de Microsoft Windows.

Cada ventana abierta usa recursos del sistema (áreas de datos utilizadas por Microsoft Windows). Si se agotan los recursos del sistema, se producirá un error en tiempo de



ejecución. Las aplicaciones también pueden llamar a la función **Get Free System Resources** (Recursos Gratuitos de Sistema) de la API de Windows para reclamar recursos del sistema, cerrar ventanas (como formularios abiertos y ventanas de código, así como ventanas de otras aplicaciones) y finalizar la ejecución de aplicaciones.

### 2.5.3. Entorno integrado de desarrollo de Visual Basic

El entorno integrado de desarrollo de Visual Basic (IDE) consta de los siguientes elementos:

#### 2.5.3.1. Cuadro de Herramientas:

El cuadro de herramientas de Visual Basic proporciona un conjunto de herramientas que puede usar durante el diseño para colocar controles en un formulario. Además del diseño del cuadro de herramientas predeterminado Visual Basic permite crear un diseño propio personalizado.

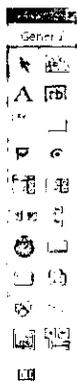


Figura 2.5.3.1.1. El cuadro de herramientas de Visual Basic

Hay tres categorías generales de controles en Visual Basic:

- Controles intrínsecos
- Controles ActiveX
- Objetos insertables

En la tabla 2.5.3.1.2. se enumeran los controles intrínsecos del cuadro de herramientas de Visual Basic.



Icono	Nombre del control	Nombre de clase	Descripción
	Casilla de verificación	<b>CheckBox</b>	Presenta una opción de tipo Verdadero o Falso, o Si o No. Puede activar varias casillas de verificación en un mismo formulario al mismo tiempo.
	Cuadro combinado	<b>ComboBox</b>	Combina un cuadro de texto y un cuadro de lista. Permite que el usuario escriba una selección o seleccione un elemento de una lista desplegable.
	Botón de comando	<b>CommandButton</b>	Ejecuta un comando o una acción cuando un usuario hace clic en él.
	Datos	<b>Data</b>	Permite conectar con una base de datos existente y presenta información de ella en los formularios.
	Cuadro de lista de directorios	<b>DirListBox</b>	Presenta directorios y rutas de acceso, y permite que el usuario los seleccione.
	Cuadro de lista de unidades	<b>DriveListBox</b>	Presenta unidades de disco válidas y permite que el usuario las seleccione.
	Cuadro de lista de archivos	<b>FileListBox</b>	Presenta una lista de archivos y permite que el usuario los seleccione.
	Marco	<b>Frame</b>	Proporciona un contenedor visual y funcional para otros controles.
	Barras de desplazamiento horizontal y vertical	<b>HScrollBar y VScrollBar</b>	Permite que un usuario agregue barras de desplazamiento a controles que no las tienen de forma automática. (No son las barras de desplazamiento incorporadas que se incluyen en muchos controles.)
	Imagen	<b>Image</b>	Presenta mapas de bits, iconos o metarchivos de Windows, archivos JPEG o GIF, y actúa como un botón de comando cuando se hace clic en él.



Icono	Nombre del control	Nombre de clase	Descripción
	Etiqueta	<b>Label</b>	Presenta texto con el que el usuario no puede interactuar ni modificar.
	Línea	<b>Line</b>	Agrega un segmento de línea recta a un formulario.
	Cuadro de lista	<b>ListBox</b>	Presenta una lista de elementos entre los que el usuario puede elegir.
	Contenedor OLE	<b>OLE</b>	Incrusta datos en una aplicación de Visual Basic.
	Botón de opción	<b>OptionButton</b>	El control <b>OptionButton</b> , como parte de un grupo de opciones con otros botones de opción, presenta varias opciones entre las que el usuario sólo puede elegir una.
	Cuadro de imagen	<b>PictureBox</b>	Presenta mapas de bits, iconos o metarchivos de Windows, archivos JPEG o GIF. También presenta texto o actúa como contenedor visual para otros controles.
	Forma	<b>Shape</b>	Agrega un rectángulo, un cuadrado, una elipse o un círculo a un formulario, marco o cuadro de imagen.
	Cuadro de texto	<b>TextBox</b>	Proporciona un área para escribir o presentar texto.
	Cronómetro	<b>Timer</b>	Ejecuta eventos periódicos a intervalos de tiempo especificados.

Tabla 2.5.3.1.2. Controles intrínsecos del cuadro de herramientas.



### 2.5.3.2. Barra de Menús:

La barra de menús visualiza ordenes que son necesarias para desarrollar la aplicación. Esta barra de menús se conforma de:

- Menú Archivo
- Menú Edición
- Menú Ver
- Menú Proyecto
- Menú Formato
- Menú Depuración
- Menú Ejecutar
- Menú Herramientas
- Menú Complementos.
- Menú Ventana.
- Menú Ayuda

### 2.5.3.3. Ventana Explorador de Proyectos:

Esta ventana enumera los formularios y módulos del proyecto actual. Un proyecto es la colección de archivos que usa para generar una aplicación.

En la figura 2.5.3.3.1. observamos la ventana del explorador de proyectos que presenta Visual Basic, en el entorno de desarrollo.

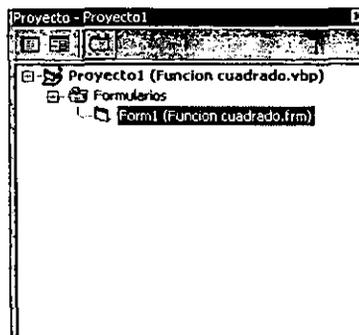


Figura 2.5.3.3.1. Ventana del Explorador de Proyectos.

### 2.5.3.4. Ventana Propiedades:

Esta ventana enumera los valores de las propiedades del control o formulario seleccionado. Una propiedad es una característica de un objeto, como su tamaño, título o color, la ventana propiedades se muestra en la figura 2.5.3.4

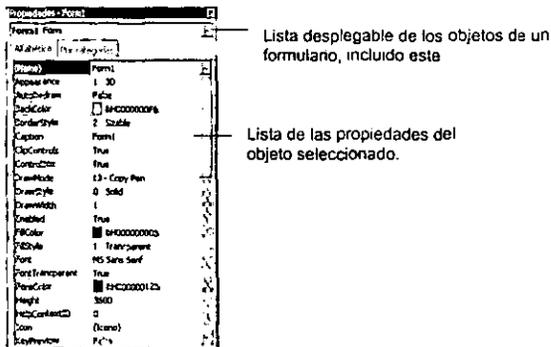


Figura 2.5.3.4 Ventana de Propiedades.

Esta ventana consta de los siguientes elementos:

- Object Box (Caja de objetos).
- Properties List (Lista de propiedades).
- Forms (Forma).

### 2.5.3.5. Ventana de Código:

Esta Ventana de código, sirve para poder ver los posibles sucesos mientras se diseña una aplicación, para esto se requiere conocer cuales van a ser los eventos que afectarán a cada objeto y que realizarán cuando estos sucedan.

Este código consta de declaraciones, constantes y sentencias, en la figura 2.5.3.5.1 se observa un ejemplo de la ventana de código de una forma.

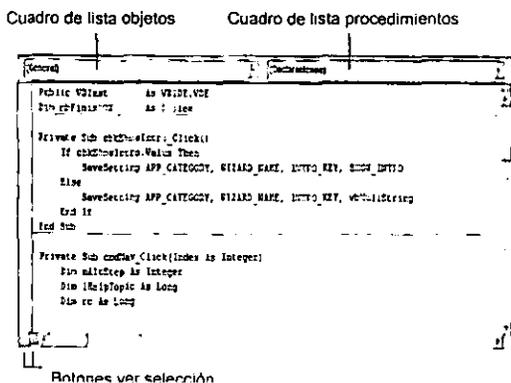


Figura 2.5.3.5.1. Ventana de Código



### 2.5.3.6. Asistentes:

Los **wizards** (asistentes) son herramientas que guían al usuario paso a paso a través de una tarea de programación. Si se desea crear asistentes que se asemejen y actúen de manera similar a los empleados en productos de Microsoft, es posible utilizar el Administrador de asistentes.

### 2.5.3.7. Cómo funciona el Administrador de asistentes

El Administrador de asistentes sólo emplea un formulario para todo el asistente. Este formulario (que debe llamarse "frmWizard") tiene un área visible con el mismo tamaño que el marco de cada paso. Cuando se presenta un paso, su propiedad **Left** (izquierda) se establece al valor de esa zona visible del formulario, de forma que el usuario la pueda ver. Las propiedades **Left** de pasos que no se utilizan se apartan del área visible del formulario. Se puede considerar como ver una pila de imágenes desde una mesa cercana, donde cada vez se coloca una foto frente al usuario para verla, para luego volverla a poner en la pila de la mesa cuando haya terminado. De manera similar, el paso actual se mueve al área visible y luego al área de "almacenamiento" cuando se tiene acceso al siguiente paso.

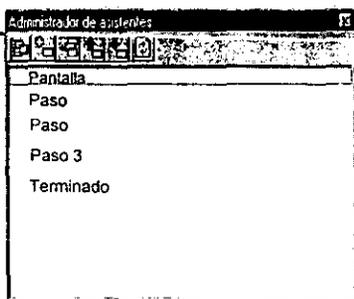


Figura 2.5.3.8.1. Administrador de asistentes

Cada paso proporcionado por el Administrador de asistentes consta de tres áreas:

- Un control **Image** (imagen).
- Un control **Label** (etiqueta).
- Un control **TextBox** (caja de texto).



# **CAPÍTULO III**

## **PLANTEAMIENTO DEL PROBLEMA**



### 3.1. DEFINICIÓN DEL PROBLEMA.

El objetivo planteado para el presente trabajo, consiste en desarrollar un sistema que integre la información que se necesita para la generación de reportes, con el fin de auditar a los agentes y monitorear el estado de la cartera, a continuación definiremos cual es la información necesaria y específicamente de donde lo podemos obtener, ya que se ven involucradas varias áreas, las cuales juegan un papel importante para la integración de dicha información.

En la figura 3.1.1 se muestra que áreas se relacionan para la recopilación de información, mismas que definiremos detalladamente a continuación.

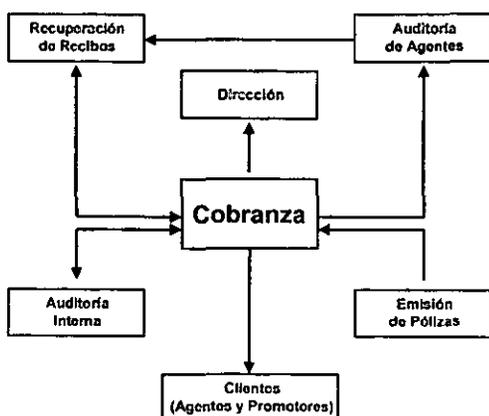


Figura 3.1.1 Áreas involucradas.

#### 3.1.1. Cobranza.

Actualmente esta área es la encargada de controlar toda la información de los adeudos generados por la emisión de pólizas de seguro, la forma de pago de la póliza, que es definida en la emisión de la misma y es la que indica la cantidad de adeudos generados, ya que cuando una póliza es anual únicamente se genera un adeudo y si la póliza es mensual se generan doce adeudos. En esta área se generan los recibos que se entregarán a los clientes a través de los agentes, estos recibos son facturas que se deben de cobrar individualmente, al momento de emitir el recibo se emiten todos los recibos, aún y que su cobro o reclamación sea varios meses después.

Las principales operaciones que realiza esta área son (figura 3.1.1.1):

- Generación de recibos de pago.
- Reportes de cartera vencida.
- Reportes de cartera por vencer.

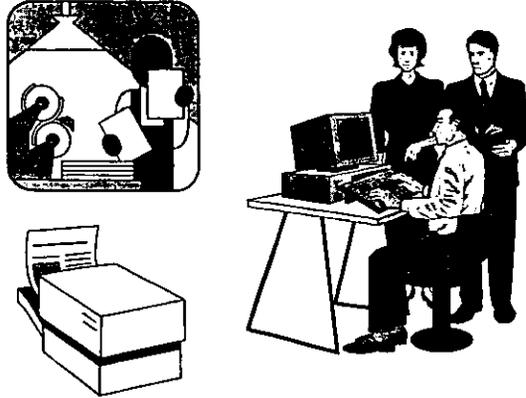


Figura 3.1.1.1 Actividades del área.

### 3.1.2. Emisión de pólizas.

Esta es el área que alimenta toda la información para el área de cobranza, aquí se desarrolla la emisión de pólizas de asegurados principalmente, la información que se genera en la emisión de cada póliza es muy importante para el cobro de la misma, el tipo de pago y el periodo de pago, estos datos se mandan al área de cobranza para la generación de los recibos.

Las principales operaciones que realiza esta área son (figura 3.1.2.1):

- Generación de pólizas.
- Control de pólizas.
- Cierre del negocio con el cliente.



Figura 3.1.2.1 Generación de pólizas.



### 3.1.3. Recuperación de recibos.

Esta es el área que se encarga de recuperar los recibos generados y que no son cobrables, ya que al momento de la emisión se generan todos los recibos, y en caso de cancelación de la póliza se deben recuperar esos recibos, ya que son facturas previamente emitidas, y si no son cobrables, son documentos que se deben recuperar.

La recuperación de estos recibos impide también que los agentes cometan algún tipo de fraude en contra de la compañía, ya que pueden seguir cobrándole al asegurado y ante la compañía cancelan la póliza, si el asegurado pago sus recibos aunque no hayan sido reportados a la compañía; la compañía se obliga a pagar al asegurado en caso de siniestro, dejando el problema legal con el agente para después.

Las principales operaciones que realiza esta área son (figura 3.1.3.1):

- Recuperación de recibos.
- Prevención de fraudes.



Figura 3.1.3.1 Recuperación de recibos incobrables.

### 3.1.4. Auditoría de agentes.

Esta área se encarga de monitorear a los agentes que tienen una cartera vencida elevada y recuperar los recibos no cobrables, así como los adeudos a los que están sujetos estos agentes morosos.

La auditoría a los agentes permite evitar fraudes, y depende mucho de la oportunidad con la que los reportes de estatus de cartera de agentes y promotorias sean generados, ya que si no se tiene un estatus de la cartera del intermediario al momento de la auditoría, estos agentes evaden el pago de la misma.

Los recibos recuperados son enviados al área de recuperación de recibos para darles el seguimiento pertinente.



Las principales operaciones que realiza esta área son (figura 3.1.4.1):

- Recuperación de recibos.
- Monitoreo de agentes.
- Recuperación de la cartera vencida.
- Prevención de fraudes.

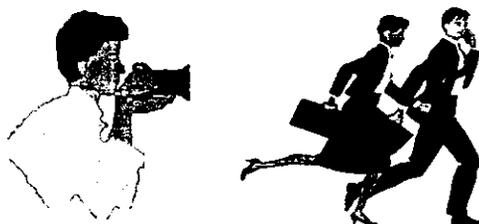


Figura 3.1.4.1 Monitoreo de agentes.

### 3.1.5. Auditoría Interna.

Esta área se encarga de monitorear el estado de los recibos recuperados por el área de recuperación de recibos, valida que se encuentren completos, que no hayan sido pagados por el cliente directamente, analizar el problema de la cartera vencida y plantear soluciones para evitarla, ~~fünge como el intermediario entre las oficinas regionales y su correspondiente división.~~

Las principales operaciones que realiza esta área son (figura 3.1.5.1):

- Auditoría de oficinas.
- Auditoría de recibos recuperados.
- Conocimiento total del negocio.



Figura 3.1.5.1 Auditoría interna.



### 3.1.6. Dirección.

La dirección es la encargada de la toma de decisiones, define el rumbo que tomará la compañía y corrige el rumbo en caso de existir desviaciones en el funcionamiento de la misma.

La importancia de esta área es sobrada, por lo que los reportes que genera el área de cobranza son indispensables en la toma de decisiones, ya que definen las acciones correctivas o preventivas para el correcto funcionamiento del negocio.

Las principales operaciones que realiza esta área son (figura 3.1.6.1):

- Toma de decisiones.
- Dirección de la compañía.



Figura 3.1.6.1 Dirección general.

### 3.1.7. Clientes (Agentes y Promotores).

Unos de los principales clientes de las compañías aseguradoras son los agentes y los promotores (intermediarios), estos son los que se encargan de llevar el negocio a la empresa, cobran, y son los personajes que hacen el negocio crecer, por lo que el área de cobranza debe mantenerlos informados acerca de los adeudos que tienen con la compañía, esta información debe ser oportuna, confiable y completa.

Las principales operaciones que realiza esta área son (figura 3.1.7.1):

- Venta.
- Cobro.
- Trato con el cliente final.



Figura 3.1.7.1 Fuerza de ventas.

### 3.1.8. Planteamiento general.

El área de cobranza es la encargada de concentrar la información de las pólizas generadas por la emisión y generar los recibos de pago, por lo que es el área que concentra la información de la cartera. Esta distribuye los reportes a las correspondientes áreas y tiene el conocimiento del estado financiero de la compañía, por lo que es un área estratégica del negocio, de la que debe fluir la información completa y oportuna.

La dirección solicita los reportes de cartera, agrupados a nivel compañía, división y oficina regional, con el fin de poder establecer la directriz que la compañía seguirá en el próximo mes, pero estos reportes son entregados una semana después y con información de una semana anterior, por lo que el desfase de la información es de medio mes, y esto tiene un costo bastante alto en la toma de decisiones.

El área de auditoría de agentes es un control importante en la recuperación de la cartera y el control sobre los agentes, por lo que al igual que la dirección, no se conoce con exactitud la cartera de cada agente (figura 3.1.8.1), por el desfase de medio mes en los reportes, y los agentes comienzan a dejar de reportar estos movimientos, ocultando información, y la compañía de seguros comienza a perder dinero, por lo que la planeación hecha por la dirección en la estrategia de recuperación de cartera no es la adecuada, y los auditores no conocen en su totalidad la cartera de los agentes, ni tampoco la cartera ya pagada, a su vez, los recibos recuperados al momento de la auditoría son enviados al área de recuperación de recibos y llegan a extraviarse.

El área de recuperación de recibos funciona de una forma muy parecida al área de auditoría de agentes, solo que la marcación es sobre los recibos y no sobre los agentes, aquí la importancia de contar con información completa y oportuna, y esta información esta incompleta y con 15 días de desfase, por lo que el desempeño de esta área se ve disminuido por el tiempo perdido en la investigación de la cartera de los agentes.



Figura 3.1.8.1 Desconocimiento de la cartera de agentes.

El área de auditoría interna trabaja sobre los recibos recuperados, y su trabajo es lento, costoso y riesgoso, ya que debe monitorear los recibos incompletos sobre los agentes y promotores a través de la oficina regional, estos recibos viajan en valija de la región a la división y viceversa, por lo que los recibos llegan a extraviarse, además de que el viaje del recibo es costoso y su auditoría es lenta.

Cualquier agente o promotor puede solicitar los reportes de su cartera, por desconocer la misma, con el objetivo de evadir su responsabilidad de pago, ya que todos los agentes conocen su cartera sin necesidad de informársela, por lo que es importante notificarles la cartera (figura 3.1.8.2) en su poder con el fin de evitar que no paguen sus recibos.

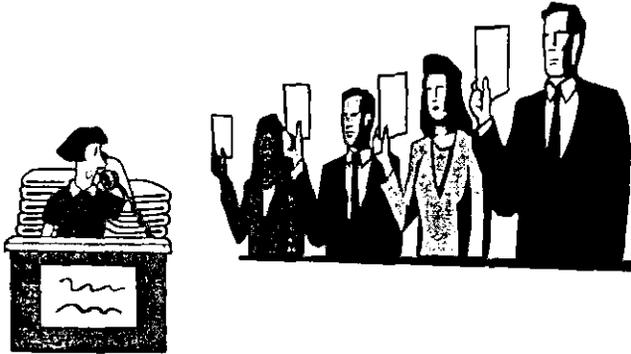


Figura 3.1.8.2 Notificación de su cartera al agentes.

La compañía esta formada por:

- Dirección general.
- Oficinas divisionales.
- Gerencias regionales.
- Oficinas regionales.

Estas oficinas tienen su jerarquía, y requieren conocer la información de todos los agentes que están por debajo de ellos, reportes concentrados, con el fin de conocer su posición dentro de la compañía.



### 3.2. SITUACIÓN ACTUAL.

En este punto, se realizará un estudio en el que se analizará la situación actual de la compañía, esto con el fin de poder tomar esta información y junto con los requerimientos planteados por los usuarios en el capítulo 3.3, desarrollar un modelo de solución el cual se detallará en el capítulo 4.

Es importante entender cuales son las áreas de influencia de la compañía de seguros con el fin de ofrecer una solución de buena calidad, buscando con esto, optimizar el proceso que actualmente se realiza en las diferentes áreas en forma manual. La importancia de entender el problema radica en que no siempre la solución de automatización significa la solución total de la problemática planteada. Muchas veces, la propuesta de solución debe ir acompañada de nuevas políticas de operación, el cambio de mentalidad de las personas involucradas y los costos en los que se incurrirá para la implantación de la nueva solución.

A través de la esquematización del problema es más sencillo determinar las deficiencias en el control adecuado de la información. A continuación, se enunciarán los principales problemas a los que se enfrentan, cada una de las áreas involucradas, en su operación diaria (figura 3.2.1).

Actualmente no se cuenta con una forma confiable de emisión de reportes de cartera para los intermediarios (promotores, agentes y auditores), ya sea por la oportunidad o versatilidad de los movimientos, ya que no todos los clientes son iguales y las necesidades de las áreas involucradas son diferentes, lo anterior se ve reflejado en la cartera personal de cada agente y los agentes que componen una promotoría, así como, los tipos y formas de ventas, además del nivel de concentración de información que requiere la dirección y el área de auditoría.

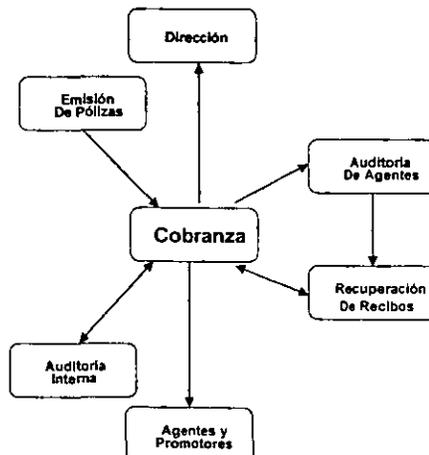


Figura 3.2.1 Áreas involucradas.



Por lo tanto, la atención a los diversos clientes del área de cobranza, no es adecuada, ya que el tiempo y la calidad de la información es muy deficiente.

A continuación, se detallará la operación de cada una de las áreas involucradas, y como se ven afectadas en sus respectivas actividades.

### 3.2.1. Cobranza.

Esta área es la de mayor importancia dentro del área operativa que se ve afectada por el problema de la cartera, ya que es la encargada de la generación de los reportes de cartera.

Actualmente se generan archivos con la información de los recibos generados y no hayan sido ya cubiertos, los cuales son alimentados en una hoja de cálculo, en esta hoja de cálculo se clasifican de acuerdo a su fecha de vencimiento en recibos anticipados, pendientes o amparados. Los recibos anticipados son aquellos que aún no rebasan la fecha de vencimiento, los recibos pendientes son aquellos que ya vencieron y que no tienen más de tres meses vencidos, y los recibos amparados son aquellos que su fecha de vencimiento es superior a tres meses.

Posteriormente se generan los reportes de la cartera desde la hoja de cálculo por agente, y manualmente se agrupan por promotora, obteniendo además un total por promotor, es decir, un concentrado de recibos y estos reportes se imprimen en tres tantos. En esta labor se consume un tiempo estimado de tres días entre 20 personas aproximadamente.

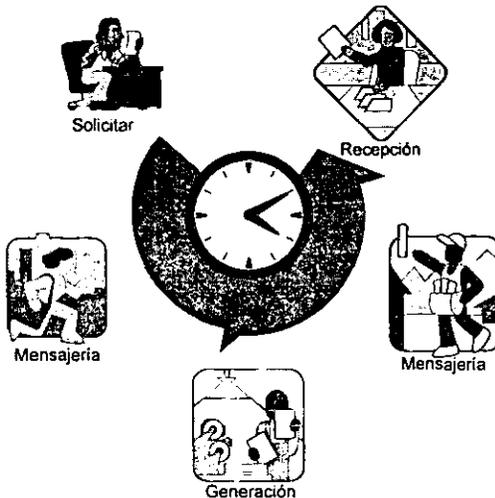


Figura 3.2.1.1 Entrega de reportes al solicitante.



Posteriormente, uno de estos reportes se envía por valija a la oficina regional, otro a la oficina de auditoría de cada división y el último a la dirección, lo cual demora aproximadamente dos días en llegar a su destino.

Esto implica un tiempo de cinco días, desde que la información llega al área de cobranza y los reportes sean entregados a su solicitante (figura 3.2.1.1).

### 3.2.2. Emisión de pólizas.

Esta área se encarga de la emisión de pólizas, y es la base de la existencia del área de cobranza, como se muestra en la figura 3.2.2.1, ya que es la encargada de alimentar al sistema central toda la información de las pólizas, y de esta forma poder emitir en el área de cobranza los recibos de pago que serán entregados a los intermediarios y posteriormente al usuario final.

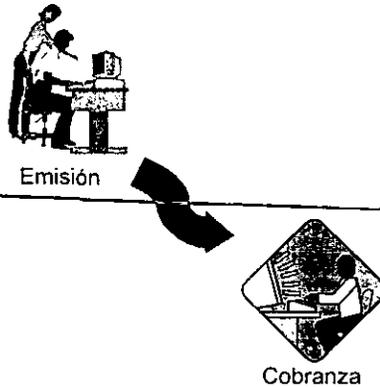


Figura 3.2.2.1 Principal alimentación de información.

Ya que los recibos hayan sido impresos, esta área es la encargada de verificar que el número total de recibos se haya emitido.

### 3.2.3. Recuperación de recibos.

Esta área solicita al área de cobranza los reportes de cartera vencida (recibos amparados) de los agentes, con el fin de dar seguimiento a la recuperación de la misma ó retirar estos recibos de la circulación, ya que cada uno de estos, es una factura, y esta factura no es cobrable.

La recuperación de estos recibos es muy importante para la compañía, ya que evita que los agentes cometan algún tipo de fraude en contra de la compañía, puesto que pueden seguir cobrándole al asegurado y reportar este dinero ante la compañía, y si el asegurado tiene estos recibos, en caso de un siniestro, la compañía se obliga a pagar al asegurado como se ve en la figura 3.2.3.1.



Figura 3.2.3.1 Pago de siniestro.

### 3.2.4. Auditoría de agentes.

Esta área es la encargada de monitorear a los agentes con cartera vigente elevada (vencida o no) y trata de recuperar los recibos no cobrables (vencidos o no), así como de recuperar los adeudos vencidos y que los agentes no entregan a la compañía.

Por lo que esta área también depende de los reportes de cartera de los agentes, teniendo con esto, un mayor control sobre la fuerza de ventas, ya que si esta área al momento de auditar no cuenta con toda la información, los agentes evaden su responsabilidad hasta que les son solicitados estos recibos completos.

Los recibos recuperados son enviados al área de recuperación de recibos para darles el seguimiento pertinente, y con el mal servicio de respuesta que da el área de cobranza, los agentes únicamente son auditados sobre recibos con un importe superior a cincuenta pesos, por lo que una parte de la cartera queda fuera como se observa en la figura 3.2.4.1.



Figura 3.2.4.1 Reportes de dinero incompletos.



### 3.2.5. Auditoría Interna.

En esta área se monitorean todos los recibos recuperados, ya que los agentes regresan recibos incompletos (figura 3.2.5.1), les falta una de sus tres partes, y esto puede ser el principio de algún fraude.

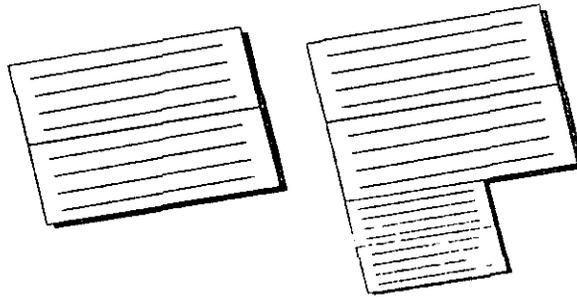


Figura 3.2.5.1 Recibos incompletos.

Los recibos, que son recuperados por las oficinas regionales directamente de los agentes, son enviados a las oficinas divisionales para darle seguimiento, los recibos son enviados por valija (mensajería) a las oficinas divisionales y una vez en la oficina divisional es validado, en caso de existir una anomalía en el recibo, este es regresado a la oficina regional, también por valija, con el fin de poder solicitarle al agente una explicación acerca de este recibo, este ciclo (figura 3.2.5.2) de ida y vuelta del recibo llega a ser ejecutado hasta unas diez veces, por lo que este ciclo tarda en cerrarse un tiempo considerable, ya que cada envío del recibo tarda 2 días y en el mejor de los casos la situación del recibo llega a aclararse en unos tres ciclos de ida y vuelta del recibo, por lo que tarda doce días.

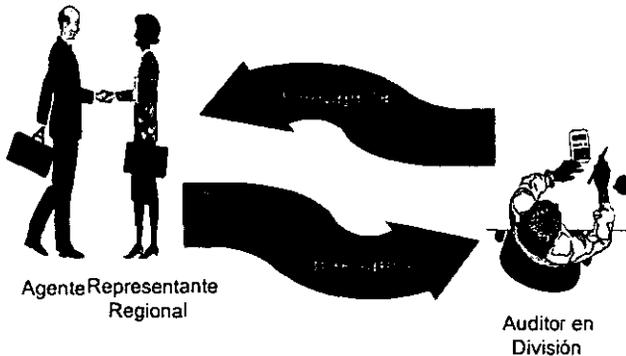


Figura 3.2.5.2 Ciclo de ida y vuelta de un recibo incompleto.



### 3.2.6. Dirección.

La toma de decisiones que realiza la dirección general de la compañía, en la mayoría de las veces se hace con información desactualizada, por lo que las decisiones tomadas pueden ser erróneas y las desviaciones en el funcionamiento de la compañía no pueden ser corregidas oportunamente, por lo que la corrección en la desviación en el rumbo de la compañía es más lenta.

El rumbo que toma la compañía (figura 3.2.6.1) siempre debe de ser exitoso, por lo que las decisiones tomadas basándose en los reportes generados por el área de cobranza son definitivas en la recuperación del dinero o de riesgos amparados y no pagados, por lo que la información desactualizada es un riesgo muy crítico a la hora de tomar una decisión que le puede costar mucho dinero a la empresa.



Figura 3.2.6.1 La dirección general debe orientarse al éxito.

### 3.2.7. Clientes (Agentes y Promotores).

Los agentes dependen de los reportes oportunos y confiables proporcionados por el área de cobranza con el fin de conocer su cartera, aún y que la conocen, pero con el fin de evadir la responsabilidad de pago dicen desconocerla (figura 3.2.7.1), por lo que la compañía debe proporcionar esta información oportuna, confiable y completa. El retraso en la entrega de estos reportes por el área de cobranza es grave, ya que esto ocasiona que nuestros agentes no cobren oportunamente a los asegurados y se incrementa el riesgo de no recuperarlo a través de las primas.

Así como la fuerza de ventas es la que genera el negocio, también es la que más lo perjudica, ya que al evadir la responsabilidad de pago, el ingreso no entra a la compañía y el riesgo es cubierto por la empresa y el dinero que debería estar ya en la empresa, no es ingresado hasta que se le exige al agente su pago.



Figura 3.2.7.1 Fuerza de ventas morosa.

### 3.2.8. Esquematización de la situación actual.

En la figura 3.2.8.1 se muestra como fluye el proceso actual de información el cual consiste en:

Primero el departamento de emisión genera las pólizas con la información necesaria para la recuperación de sus primas. Posteriormente el área de cobranza obtiene del sistema administrador de la compañía la información necesaria para la generación de reportes e integra la información con el fin de explotarla. Enseguida genera los reportes por agente y promotor, así como por región, división y compañía. La dirección toma los reportes generados por el área de cobranza a nivel región, división y compañía para tomar decisiones. Los agentes solicitan los reportes generados para cada uno de ellos con el fin de recuperar su cartera. La auditoría de agentes también toma los reportes por agente para auditar a los mismos y recuperar la cartera cobrada por el agente y que no a entregado aún, así como los recibos que tiene en su poder y no son cobrables, estos recibos recuperados en las auditorías son enviados al área de recuperación de recibos. El área de recuperación de recibos solicita a los agentes los recibos que no son cobrables y una vez que los tiene en su poder los agrega a los entregados por el área de auditoría y esta los entrega al área de cobranza. El área de cobranza integra todos los recibos de todas las regiones y los entrega al área de auditoría interna, la cual revisa la integridad de los recibos y su estado físico, devolviendo estos al área de cobranza con sus respectivas observaciones, los recibos que no tienen problemas son cancelados y los que tienen alguna anomalía son regresados al área de recuperación de recibos para solicitar al agente una respuesta al problema que presentó el recibo, para comenzar otra vez el ciclo de solicitud del recibo con el fin de auditarlo hasta su cancelación.



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

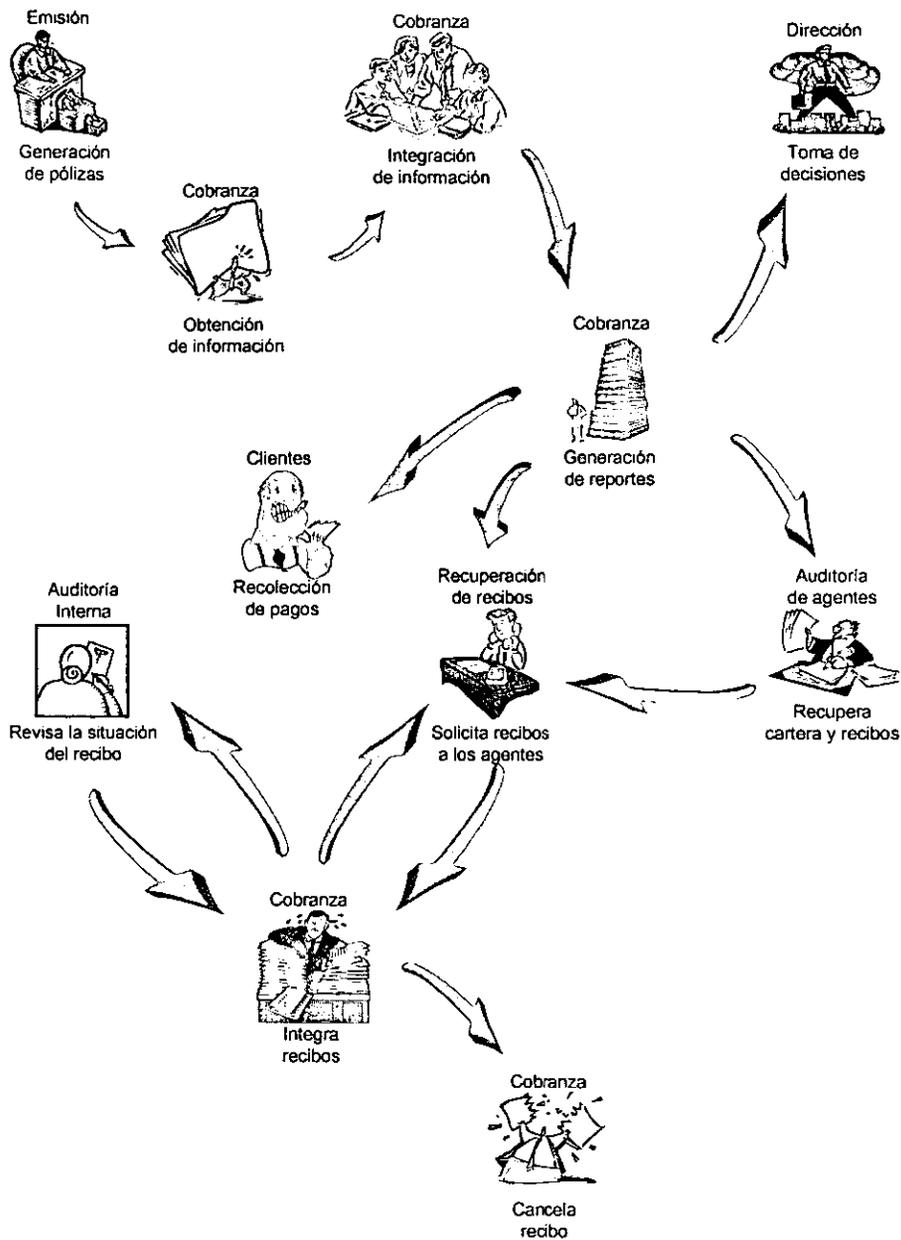


Figura 3.2.8.1 Flujo de información actual.



### 3.2.9. Áreas de interés.

De acuerdo a lo descrito en todos los puntos del capítulo 3.2 (situación actual), el proceso actual de impresión de reportes, es un proceso lento y con falta de oportunidad, ya que los reportes que solicita el agente al área de cobranza tardan demasiado tiempo en generarse y tenemos un retraso en el cobro de la cartera por el tiempo que se retrasa el área de cobranza en la generación de estos reportes (figura 3.2.8.1). La auditoría que se les realiza a los agentes no cuenta con la última información, ni con la información más completa, por lo que se deben generar reportes con más oportunidad y con más información. El viaje que los recibos hacen, es muy lento y costoso, por lo que se pierde mucho tiempo en la auditoría a cada uno de los recibos que necesitan aclaración.

La aseguradora tiene muchas áreas involucradas en el monitoreo de la cartera de los agentes, y con todo esto no cuenta con un flujo de información muy óptimo, también la dependencia de todas estas áreas sobre el área de cobranza, genera una gran oportunidad para la optimización del proceso de generación de reportes y la utilización de la infraestructura de la compañía para el transporte de recibos.

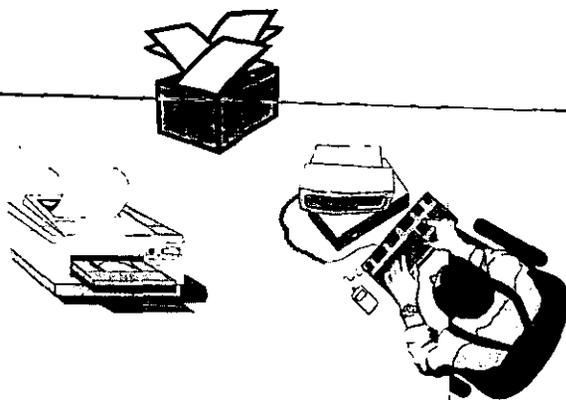


Figura 3.2.9.1 Generación de reportes manual.

En la sección 3.1 se vio a detalle como se ven involucradas las áreas de la problemática actual y la relación que guardan entre ellas.



### 3.3. REQUERIMIENTOS DEL USUARIO

Todo sistema de información debe tener un ciclo de vida, el cual entenderemos como un conjunto de fases por las que pasa a lo largo del tiempo, desde la fase de estudio y concepción hasta la de realización, explotación y mantenimiento.

Las fases o etapas de este ciclo de vida son definidas por diversas metodologías y existen diferentes variantes de las mismas.

En el siguiente diagrama (figura 3.3.1) mostramos las fases tradicionales de este ciclo de vida.

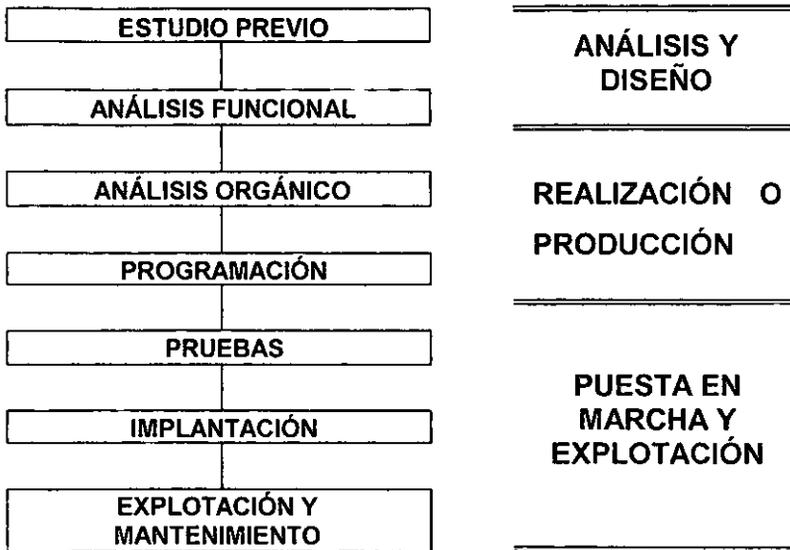


Figura 3.3.1. Fases de desarrollo del ciclo de vida.

Dentro del ciclo de vida del desarrollo de un sistema, se definen los alcances y las limitaciones de cualquier proyecto, estos en base a los requerimientos del usuario.

Las tareas en la definición de un sistema son enfocar y comprender el problema que se va a solucionar, para así poder emitir una justificación de la solución propuesta. Este proceso requiere de una intensa comunicación entre el usuario y el diseñador del sistema.

Es de vital importancia recopilar toda la información que se pueda. La factibilidad, alcance del proyecto, así como el control, los requerimientos para operar datos y los datos mismos, son los elementos que permiten al diseñador conocer la problemática a fondo, para posteriormente ofrecer una solución.



### 3.3.1. Planteamiento

Actualmente en la aseguradora, se necesita llevar el control de recibos en poder de los agentes esto para evitar posibles fraudes de parte de los agentes. Los agentes no cuentan con información sobre la cobranza pendiente, lo cual ocasiona un gran problema, porque al no tener esta información el agente no puede saber cuales son los clientes que adeudan a la compañía, por lo cual no puede hacer la cobranza. Y por lo tanto no se tiene actualizada la entrega de recibos a la compañía.

Es necesario un sistema de información que contenga la información de recibos, tanto los pendientes, como los recibos a dar de baja, y los que están por devolver.

Los requerimientos del sistema son:

- Autorización del proyecto.
- Contar con presupuesto para el desarrollo del sistema.
- Abrir el proyecto.
- Justificación del proyecto.
- Asignación de recursos de sistemas y del usuario.
- Responsable del proyecto.
- Participación del área de servicios técnicos y de comunicaciones.

---

### 3.3.2. Requerimientos Gerenciales

El sistema debe ser desarrollado en un tiempo relativamente corto, para que pueda estar a la vanguardia con los requerimientos de la compañía, y debe tener las siguientes características:

- Debe tener tecnología competitiva (retorno de inversión/tiempo en que estará vigente el sistema).
- De fácil manejo para cualquier usuario.
- Estar desarrollado en un sistema gráfico.
- Fácil de transportar, en caso de que los agentes de ventas tengan una computadora portátil, para llevarla con el cliente.
- Fácil manejo de recibos.
- Contenido de la información actualizada en cualquier momento.
- Respaldo de datos.
- Contar con la seguridad necesaria, para evitar posibles fraudes.
- Que tenga un alcance a nivel nacional, para así evitar inconsistencias con las demás sucursales.

Estos requerimientos se cubrirán con el sistema a desarrollar figura 3.3.2.1.

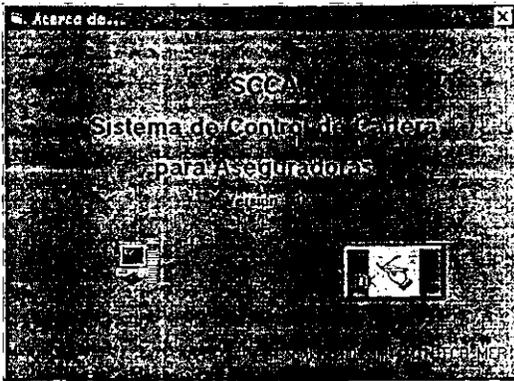


Figura 3.3.2.1. Sistema de cómputo.

### 3.3.3. Requerimientos de campo

Las necesidades anteriormente mencionadas, junto con las que se observan estando dentro del mismo campo de acción son las siguientes:

- Confiabilidad para los clientes.
- Tener un sistema que corresponda al estándar de lo que se usa en el mercado.
- Que contenga la ayuda necesaria dentro del mismo.
- Que sea agradable a la vista, tanto para el usuario, como para el cliente.
- Que tenga la facilidad de poder imprimir los recibos, las veces que sea necesario.
- Que tenga facilidad de leer los datos de los recibos pasados y actualizar registros nuevos.

Estos requerimientos son necesarios para poder optimizar el trabajo de los agentes llevando este sistema a los clientes (figura 3.3.3.1).



Figura 3.3.3.1. Facilidad de llevar el sistema a los clientes.



### 3.3.4. Análisis de requerimientos

El sistema debe cubrir los requerimientos solicitados tanto por la compañía aseguradora, como por los usuarios. Debemos considerar que el proceso de análisis no contempla únicamente estos dos tipos de requerimientos, sino también a los que tienen otros departamentos que comprenden a la aseguradora. El análisis debe optimizar los procesos y por lo tanto ofrecer un ahorro en todos los aspectos, como son:

- Optimizar el proceso que actualmente se realiza en forma manual, en diferentes áreas.
- Revisar y tener en cuenta la arquitectura actual del sistema de cómputo de la empresa, y expectativas de inversión.
- Ofrecer una línea de navegación con iconos y opciones en menú.
- Que el usuario teclee la menor cantidad de información.
- El usuario debe capturar su clave para tener acceso.
- Oportunidad y confiabilidad de la información.
- Atención amable por parte del sistema y del usuario.
- Mantener la información al día.
- Accesibilidad al sistema de recibos para su mantenimiento.
- Mantener un archivo histórico de las transacciones de los clientes.
- En la impresión debe aparecer la fecha de elaboración

### 3.3.5. Justificación del análisis de requerimientos

Los anteriores requerimientos conforman el análisis, esto en base a las encuestas con la gerencia, los agentes, otros departamentos, y los clientes, ahora veremos la justificación del sistema propiamente.

- Actualmente en la casa matriz aseguradora, el equipo de cómputo tiene procesadores 80486 y Pentium, pero en las sucursales solo cuentan solamente con los primeros, estas máquinas trabajan con Windows 95 y Windows 98, por lo tanto se puede cargar casi cualquier tipo de lenguaje que se pueda ejecutar bajo ambiente Windows.
- Como las sucursales regionales solo cuentan con procesadores 80486, tendremos que adaptarnos al equipo con el que se cuenta, para que no existan problemas de compatibilidad entre el software que utiliza la empresa, con el que se va a implementar el sistema.
- El usuario o agente deberá teclear su clave para saber en la aseguradora quien elaboró el recibo.
- Podremos tener todos los datos de los recibos pendientes y por cobrar, ya que se podrán cargar éstos en un tiempo muy corto, haciéndolo antes de dar la atención al cliente, sin afectar ni al usuario ni al cliente.



- El sistema, cuenta con ayuda en cada pantalla, para facilitar el manejo de éste en caso de que el usuario no tenga la experiencia para operarlo. Esta ayuda le dará una breve explicación del funcionamiento de los menús, iconos y botones que se encuentran en cada pantalla.
- El sistema estará desarrollado en Visual Basic 5.0, puesto que la aseguradora tiene la licencia de este lenguaje. Se desarrollará para trabajar en Windows 95, que se usa en la empresa. Se hará lo más amigable posible, para que prácticamente el agente no tenga que tomar un curso para operarlo.
- Para el desarrollo del Back end, la empresa también cuenta con la licencia y el software para trabajar con SQLServer 7.0, ya que está desarrollando otros proyectos con él.
- El contar con SQLServer 7.0 hace que nuestra relación costo-beneficio mejore, ya que este software tiene un costo muy elevado.

### 3.3.6. Beneficios

Los beneficios que se esperan tener con la implantación de este sistema son:

- Acceso al sistema a nivel nacional.
- Disminución de costos por la optimización del proceso.
- Control de recibos entregados y/o en poder del agente.
- Consultas de información actualizada.
- Disminución de trabajo en las divisiones.
- Control de las divisiones sobre las regionales.



### 3.4. RECOPIACIÓN Y ANÁLISIS DE LA INFORMACIÓN.

El análisis de la información es de gran importancia ya que se puede responder a preguntas esenciales como son:

- ¿Qué procesos integran el sistema?
- ¿Qué datos emplea cada proceso?
- ¿Qué datos son almacenados?
- ¿Qué datos ingresan y abandonan el sistema?

El análisis del sistema conoce el papel central que tienen los datos en las organizaciones. Seguir el flujo de datos por todos los procesos, es la finalidad del análisis de flujo de datos, ya que dice mucho sobre cómo se alcanza los objetivos de la organización; en el transcurso del manejo de transacciones y terminación de tareas los datos entran, son procesados, almacenados, recuperados, analizados, utilizados, cambiados y presentados como salidas.

El análisis de flujo de datos estudia el empleo de datos en cada actividad, documenta los hallazgos con diagramas de flujo de datos que muestra en forma gráfica la relación entre procesos y datos, y en los diccionarios de datos que describen de manera formal los datos del sistema y los sitios donde son utilizados.

El análisis de flujo de datos examina el empleo de los datos para llevar a cabo procesos específicos dentro del ámbito de una investigación de sistemas. El análisis puede pensarse de tal manera que se estudien actividades del sistema desde el punto de vista de los datos: dónde se originan, cómo se utilizan o cambian, hacia dónde van, incluyendo las paradas a lo largo del camino que siguen desde su origen hasta su destino.

Los componentes de la estrategia de flujo de datos abarcan tanto la determinación de los requerimientos como al diseño de sistemas. Una notación bien establecida facilita la documentación del sistema actual y su análisis por todos los participantes en el proceso de determinación de requerimientos.

A continuación definiremos cual es la información necesaria y específicamente donde la podemos encontrar, ya que se ven involucradas varias áreas, las cuales juegan un papel importante para la integración de dicha información.

En la Figura 3.4.1 se muestran las áreas que conforman la organización general y como se relacionan, pero para la recopilación de la información, nos basaremos en el área de Emisión de pólizas, Cobranza, Dirección y Auditoría de agentes mismas que definiremos a continuación:

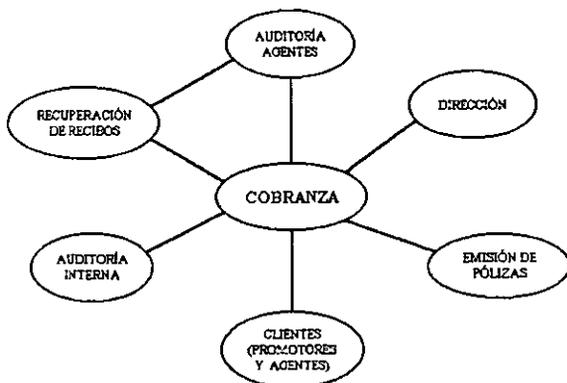


Figura 3.4.1 Áreas involucradas en la recopilación de información.

### Emisión de pólizas.

Actualmente es el área encargada de realizar las cotizaciones o propuestas de pólizas de seguros. De acuerdo con los datos que el agente de seguros proporcione del cliente o prospecto, esta área prepara la cotización que será entregada al cliente por medio del agente de seguros. Dicha cotización contiene:

- Razón social.
- Dirección.
- Tipo de seguro.
- Suma asegurada, cuota aplicada y prima. Cada uno de estos tres conceptos deberá aparecer por cada una de las coberturas incluidas en la propuesta de seguros.
- Una breve descripción del alcance que tiene cada una de las coberturas incluidas en la propuesta de seguros.
- Taza de financiamiento. Costo que tendría que pagar el cliente en caso de elegir una forma de pago diferente a la de contado (semestral, trimestral o mensual).
- Derecho de Póliza. Costo por la expedición de la póliza.
- IVA. (Impuesto al Valor Agregado).
- Prima total. Resultado de la suma aritmética de la Prima Neta, la Tasa de Financiamiento, el Derecho de Póliza y el IVA.
- Deducibles aplicables a cada cobertura en caso de siniestro.
- Fecha de la Cotización.
- Vigencia de la Cotización. Durante cuántos días respetarán los costos y las condiciones indicadas en la propuesta de seguros.
- Nombre y teléfono del agente de seguros.

Con la información antes mencionada se integra una propuesta de seguros que es entregada al cliente por medio de una carta la cual le da un sentido de formalidad a la misma.



**DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS**

Posteriormente se lleva a cabo la emisión de pólizas y se alimenta al sistema central con toda la información de éstas, para que posteriormente cobranza emita todos los recibos de pago. Se emiten dos juegos, un original que estará en poder del cliente y otra copia idéntica para el agente; estas son diferenciadas por una leyenda, para los asegurados "ORIGINAL - ASEGURADO" y para el agente "COPIA - AGENTE". Dichas pólizas tienen el un formato como el que se muestra en la Figura 3.4.2.

Figura 3.4.2 Póliza generada por el área de Emisión de Pólizas (Ej. Autos).

También junto con la póliza se genera un recibo de pago que es el comprobante de pago del asegurado, este viene como una parte desprendible el cual contiene el Registro Federal de Contribuyentes de la Aseguradora. Dicho recibo desprendible tiene el formato que se aprecia en la figura 3.4.3.

Figura 3.4.3 Recibo de pago del asegurado.

En esta área se debe tener un especial cuidado ya que la información que se indica en la propuesta de seguros se toma como base para generar la póliza.



- Si los datos indicados en la propuesta de seguros no son correctos, en la póliza también quedarán incorrectos.
- Si en la propuesta de seguros no se incluyó toda la información que requiere el área de emisión para generar la póliza, el proceso se tendría que detener hasta que se recopile la información faltante.
- Si las tarifas que tiene el sistema de emisión no son las mismas que se utilizaron en la elaboración de la propuesta de seguros, los costos van a ser diferentes y esto va a ocasionar un problema con el cliente.

### **Cobranza.**

En esta área se generan los reportes de cartera, para hacer esto se generan archivos planos con la información de los recibos generados, dichos archivos tienen los siguientes datos:

- Número de póliza
- Tipo de recibo
- Importe de la prima
- Fecha de vencimiento
- Clave de intermediario
- Tipo de intermediario
- Número de agente
- Numero de promotor
- Clave del ramo
- Tipo de moneda
- Nombre del asegurado
- Fecha de movimiento
- Clave de archivo

También una vez que el área de emisión termina con las pólizas esta área se encarga de emitir los recibos para los agentes de seguros, estos recibos pasan por los intermediarios hasta llegar al cliente o asegurado. Estos recibos son parecidos a los recibos que están en poder de los asegurados con la diferencia de que estos no sirven como comprobantes de pago, puesto que no contienen el Registro Federal de Contribuyentes.

Los recibos generados tienen el formato que se muestra en las Figura 3.4.4.



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

CONTROL AGENTE SERIE A No. 39000717867

Emitido en: MEXICO NACIONAL SO CO CO (C) (A) (P) (E) (D) (M) (O) (A) (S) (E)

Compañía: JEFERSON PEREZ SA		R.F.C.: 441028218
Domicilio: JEFERSON SA. VALLE DE LOS RINOS LA PAZ, ESTADO DE MEXICO		C.P.: 56400
Código postal: JEFERSON SA. VALLE DE LOS RINOS LA PAZ, ESTADO DE MEXICO		Teléfono: 55547411
Notas	Región de TRINIDAD GUERRERO	Poliza no. 10100000000000000000
	Poliza no. 10100000000000000000	Enlace no. 100000
	Vigencia desde fecha no. 01/01/2000 hasta: 31/12/2000	Creo. de expedición: 101.00
	Condición de crédito: MEXICO. Serie de recibos: 1000. Expedición: 101.00	Trans. pago trans. \$ 101.00
Nombre del agente: JEFERSON PEREZ SA		T.N.A. \$ 101.00
Efectivo <input type="checkbox"/> Cheque no. _____		Costo a pagar: 101.00
Fecha de depósito <input type="checkbox"/> Tarjeta de depósito <input type="checkbox"/>		Comisión del agente: 101.00
Estado: _____ No. de cuenta: _____		
Firma del agente y/o sello de la Cia.		

ESTE DOCUMENTO NO ES COMPROBANTE DE PAGO

Figura 3.4.4. Recibos emitidos en el área de cobranza en poder de los agentes.

Posteriormente se generan reportes de la cartera por agente y por promotoría que después son agrupados por promotoría. Esto reportes contienen los siguientes datos:

**Reporte por agente.**

- División
  - Subdirección
  - Región
  - Promotoría
  - Agente
  - Número de póliza
  - Ramo
  - Nombre del asegurado
  - Prima
  - Fecha de vencimiento
  - Estado del recibo
  - Forma de pago
  - Fecha de movimiento
- Reporte por promotoría

- División
- Subdirección
- Región
- Promotoría
- Clave de agente
- Nombre del agente
- Cantidad de recibos
- Tipos de recibos
- Total por agente en dinero
- Totales generales



Otro tipo de reportes que genera esta área son los reportes que se utilizan para darle seguimiento a un recibo y que es utilizado por algunas otras áreas para llevar a cabo tareas de auditoría. Este reporte contiene los siguientes datos:

- Número de póliza
- Tipo de recibo
- Prima
- Fecha de vencimiento
- Moneda
- Ramo
- Nombre del asegurado
- Forma de pago
- Causa del requerimiento
- Número de recibo
- Estado del recibo
- Fecha y hora de expedición
- Modificado por
- Seguimiento

#### **Auditoría de agentes.**

Esta área es la encargada de mantener un registro actualizado de todos los agentes de ventas dados de alta en la compañía de seguros, determinando que categoría tiene cada uno.

También se da un seguimiento detallado de cada agente para saber cuales son los recibos que ellos tienen y el estado de estos.

En esta área existe información en archivos referente a todos los datos de un agente, dicho archivo contiene la siguiente información:

- Datos personales del agente
- Categoría
- Promotoría
- División
- Subdirección
- Región

#### **Dirección.**

Esta área se encarga de crear, estandarizar e implantar todos los procedimientos internos en la compañía de seguros. Con el fin de que todas las áreas involucradas en alguna tarea cuenten con un método o procedimiento, ésta área unifica los pasos, criterios y caminos a seguir con la intención de no tener diferencias de apreciación de conceptos.



Esta área crea documentos en donde los involucrados en determinada tarea participan en elaborarlo, donde informan paso a paso como deberá ser realizada. Todos los participantes firman este documento donde se comprometen a respetar lo ahí estipulado y a la vez a informar o difundir dicho procedimiento con toda la gente involucrada.

### Información Adicional.

Existe además información en catálogos (carpetas) sobre las divisiones que integran a la Aseguradora, este catálogo se muestra en la figura 3.4.5.

DIVISIÓN	SUBDIRECCIÓN	CIUDAD	
MONTERREY	NOROESTE	CD. OBREGÓN	
		ENSENADA	
			HERMOSILLO
			MEXICALT
		NORTE	CD. JUÁREZ
			CHIHUAHUA
		NORESTE	DURANGO
			MONCLOVA
			SALTILLO
			TORREÓN
		GOLFO	MATAMOROS
			REYNOSA
			TAMPICO
	GUADALAJARA	BAJO	CD. VICTORIA
NVO. LAREDO			
			AGUASCALIENTES
			LEÓN
			SAN LUIS POTOSÍ
			ZACATECAS
			CELAYA
		PACÍFICO	CULIACÁN
			COLIMA
			MAZATLÁN
			TEPIC
			MORELIA
			QUAPÁN
MÉXICO		SUR	PUEBLA
	VERACRUZ		
			TLAXCALA
			OAXACA
		SURESTE	MÉRIDA
			VILLA HERMOSA
			TUXTLA GUTIÉRREZ
			CAMPECHE
		CENTRO	DISTRITO FEDERAL
			CUERNAVACA
			GUANAJUATO
			PACHUCA
			ACAPULCO

Figura 3.4.5 Estructura de la organización por regiones.



### 3.5. DESCOMPOSICIÓN FUNCIONAL.

El área de cobranza se encarga de recopilar toda la información de las pólizas generadas por la emisión y de generar los recibos de pago, por tanto es el área que concentra la información de la cartera. Se encarga de distribuir todos los reportes a las correspondientes áreas y conoce el estado financiero de la compañía, por lo cual es un área estratégica de la compañía que debe contar con la información correcta y a tiempo.

Todas las demás áreas indirectamente reportan el estado de los recibos al área de cobranza y esta es capaz de monitorear el estado de cualquier oficina regional o divisional o de cualquier agente o promotor.

Si no se tiene información de las cuentas de los agentes, se da el caso de que estos dejan de reportar los movimientos pendientes lo que se convierte en pérdidas para la empresa, la dirección debe planear las estrategias de recuperación de cartera y conocer el estado del negocio en todas las regionales.

Podemos detectar 6 áreas que están involucradas con la problemática del área de Cobranza y estas son:

- Dirección
- Recuperación de recibos.
- Auditoría de Agentes.
- Auditoría interna.
- Emisión de pólizas.
- Clientes (agentes y promotores).

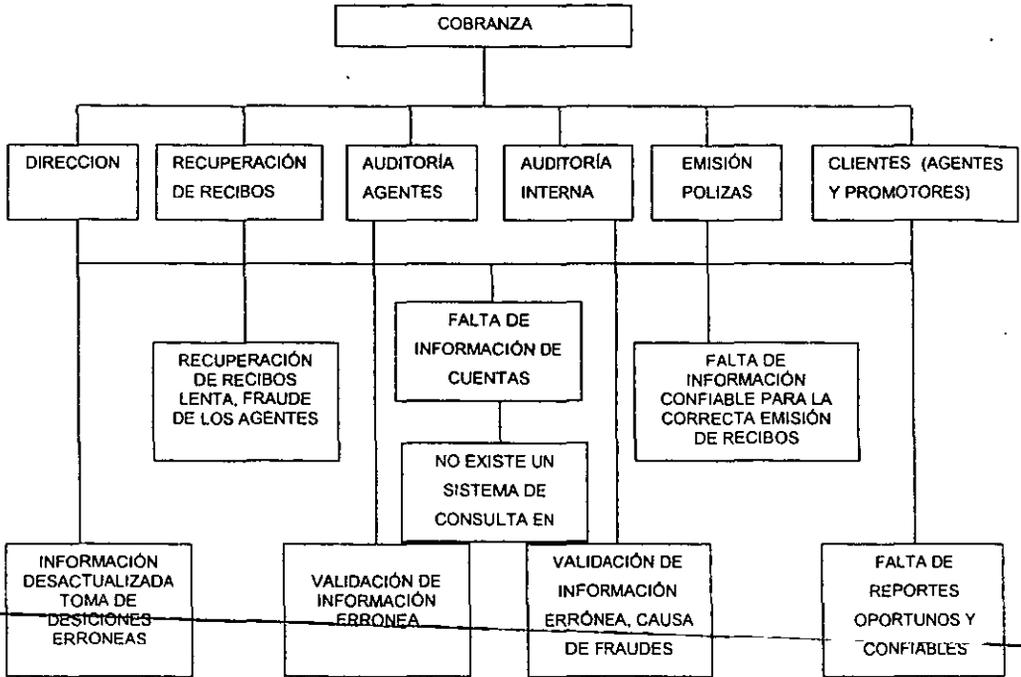


Figura 3.5 Diagrama de bloques de problemática por áreas.

Todas las áreas comparten puntos en común, por ejemplo la falta de información y de un sistema de consulta en línea, lo cual es muy importante para tener la mayor información en tiempo real.

### Dirección.

La toma de decisiones que realiza la dirección general de la compañía, en la mayoría de las veces se hace con información desactualizada, por lo que las decisiones tomadas pueden ser erróneas y las desviaciones en el funcionamiento de la compañía no pueden ser corregidas oportunamente, por lo que la corrección en la desviación en el rumbo de la compañía es más lenta.

### Recuperación de recibos.

Esta área solicita al área de cobranza los reportes de cartera vencida (recibos amparados) de los agentes, con el fin de dar seguimiento a la recuperación de la misma ó retirar estos recibos de la circulación, ya que cada uno de estos, es una factura, y esta factura no es cobrable.

La recuperación de estos recibos es muy importante para la compañía, ya que evita que los agentes cometan algún tipo de fraude en contra de la compañía, puesto que



pueden seguir cobrándole al asegurado y reportar este dinero ante la compañía, y si el asegurado tiene estos recibos, en caso de un siniestro, la compañía se obliga a pagar al asegurado.

### **Auditoría de agentes.**

Esta área monitorea a los agentes con cartera vigente elevada (vencida o no) y trata de recuperar los recibos no cobrables (vencidos o no), así como de los adeudos vencidos y que los agentes no entregan a la compañía. Por lo que esta área también depende de los reportes de cartera de los agentes, si esta área al momento de auditar no cuenta con toda la información, los agentes evaden su responsabilidad hasta que se les son solicitados estos recibos completos.

### **Auditoría Interna.**

En esta área se monitorean todos los recibos recuperados, ya que los agentes regresan recibos incompletos, y esto puede ser el principio de algún fraude.

Debido a que no existe un sistema de consulta en línea los recibos recuperados de las regionales pueden haber sufrido alguna anomalía, este es regresado a la oficina regional, por valija, con el fin de poder solicitarle al agente una explicación acerca de este recibo, el envío del recibo tarda 2 días y esto retrasa la operación de la empresa y alarga el tiempo de recuperación.

La problemática detectada en el área de auditoría, parte de hecho de que, como ya se mencionó, el flujo de información no sea el adecuado y no se cuente con un sistema de consulta en línea, genere una validación errónea, es muy importante que todos los procesos de las áreas estén centralizadas, para que dicha validación de esta información sea real y confiable.

### **Emisión de pólizas.**

Esta área se encarga de la emisión de pólizas, y es la base de la existencia del área de cobranza, es la encargada de alimentar al sistema central toda la información de las pólizas, y de esta forma poder emitir en el área de cobranza los recibos de pago que serán entregados a los intermediarios y posteriormente al usuario final.

### **Clientes (Agentes y Promotores).**

Los agentes dependen de los reportes oportunos y confiables proporcionados por el área de cobranza con el fin de conocer su cartera, aún y que la conocen, pero con el fin de evadir la responsabilidad de pago dicen desconocerla, por lo que la compañía debe proporcionar esta información oportuna, confiable y completa. El retraso en la entrega de estos reportes por el área de cobranza es grave, ya que esto ocasiona que nuestros agentes no cobren oportunamente a los asegurados y se incrementa el riesgo no recuperado a través de las primas.



Por último estos seis factores como se puede percibir están relacionados entre sí, para la toma de decisiones en cuanto a la cobranza, pero a su vez el flujo de información que se requiere no es hecha a tiempo, y la misma debe de recolectarse de diferentes áreas, después procesarla y complementarla de tal forma que permita realizar la labor más rápida y sistemática.



# **CAPÍTULO IV**

## **PROPUESTA DE SOLUCIÓN**



## 4.1. OPCIONES DE SOLUCIÓN.

A continuación se describen las posibles soluciones mediante las cuales se podría atacar la problemática planteada con anterioridad.

### **Adquisición de un producto.**

Esta opción implica investigar si existe algún producto en el mercado que cumpla con los requisitos que se deben cumplir para satisfacer las necesidades antes planteadas.

Existen en el mercado algunos productos que pueden cubrir parcialmente las necesidades, pero no van a funcionar al 100%, para lograr esto después de la compra se debe de pedir asesoría y consultoría directa de el fabricante para adaptarlo a las necesidades y cumpla ahora si al 100%. Esto implica un costo de compra, un costo de asesoría y consultoría y un costo de capacitación. Un aspecto importante que hay que considerar es, saber si el fabricante está dispuesto a hacer el trabajo de adaptación o en su defecto proporcionar los archivos fuentes para ser modificados.

### **Adaptar los sistemas actuales.**

Los sistemas que existen actualmente son aislados y lentos además muy poco seguros. Existen pequeños módulos que trabajan por separado, éstos operan de manera rudimentaria y no existe ninguna interrelación automática entre ellos por lo que los hace muy poco eficientes y prácticos. Existen algunos procesos que se llevan a cabo manualmente y otros a través de la alimentación de una hoja de calculo, los reportes también se generan manualmente a partir de algunos resultados obtenidos con las herramientas anteriores.

El adaptar los sistemas existentes significa estandarizar los procesos que se llevan manualmente y unificarlos mediante un sistema que se alimente de toda la información que se genera, además de aprovechar toda la información que se almacena en archivos que alimentan a la hoja de cálculo y generar los reportes de manera más automática, para el caso de los procesos que se hacen en forma manual se deberá buscar la manera de optimizarlos y en los posible automatizarlos con lo que ya existe.

Esta opción puede no ser muy efectiva puesto que la mayoría de los procesos que se realizan son hechos de forma manual.

### **Desarrollo a la medida.**

Esta opción significa desarrollar un sistema que cumpla totalmente con las expectativas y requisitos del usuario o usuarios, dicho sistema deberá contemplar todos y cada uno de los procesos que se llevan a cabo así como la salida de información, todo esto deberá ser de manera automática.

Para llevar a cabo esto se hará un análisis de algunas herramientas que existen en el



mercado y que son aptas para el desarrollo a la medida tanto para el Front-End como para el Back-End.

En esta parte se analizarán algunas de las opciones de software disponibles, específicamente se abordarán los siguientes casos: Manejadores de Bases de Datos Relacionales y Herramientas Visuales de Desarrollo.

### **Manejadores de bases de datos relacionales.**

A continuación realizaremos una evaluación y comparación de las características de los principales manejadores de bases de datos relacionales que existen en el mercado: ORACLE versión 7.3, SYBASE SQL SERVER 11, INFORMIX 7.2 y MICROSOFT SQL SERVER 7. Todos éstos se caracterizan por contar con optimizadores de consultas, manejo de triggers, vistas, integridad declarativa y soportan bases distribuidas.

Si solo observáramos superficialmente estos manejadores, podría escogerse cualquiera, sin embargo, se debe realizar una evaluación más a fondo para verificar tanto la calidad como la extensión de cada producto.

### **Modelo de datos relacional.**

~~Aunque todos los manejadores de bases de datos relacionales que estamos analizando~~ se denominan relacionales, su soporte del modelo de datos relacional debe ser revisado con mucho cuidado. Todos estos soportan los conceptos básicos relacionales, tales como datos almacenados en tablas y accesos a estos datos mediante operaciones de conjunto de alto nivel, la mayoría empleando SQL. Cada uno de estos manejadores aseguran soportar **constraints** (restricción que es agregada a la definición de una tabla para limitar los valores de los datos que van a ser insertados) para la integridad, solo Informix y Oracle soportan el borrado en cascada como una opción cuando se viola un constraint de integridad referencial. La implantación de los constraints debe analizarse también puesto que la mayoría de estos productos tienen mecanismos muy rudimentarios. Por ejemplo, casi todos crean índices ocultos para crear los constraints de llave primaria.

### **Consultas.**

Todos los manejadores evaluados aseguran soportar el estándar ANSI SQL-92 en el nivel más bajo de las operaciones de definición de las bases de datos y la manipulación de éstas. Los aspectos más importantes son cubiertos en los niveles superiores de este estándar.

Uno de los aspectos que también deben revisarse es el caso de la operación outer join, ésta es usada más de lo que generalmente se piensa, por tanto es extremadamente importante verificar que el manejador soporte todos los tipos necesarios de outer joins en los cuales pudiera estarse interesado, tales como el outer join completo, por la izquierda y por la derecha. Debe analizarse también la sintaxis utilizada, puesto que



todos los productos manejan alguna variación de la operación outer join, solo Microsoft SQL SERVER soporta la sintaxis especificada por el estándar ANSI SQL-92, lo cual indica que el outer join debe ser especificado explícitamente en la cláusula FROM de la instrucción SELECT. Los demás productos emplean una variación de la sintaxis de Oracle, en la cual el outer join es especificado utilizando símbolos, como un asterisco o un signo ' + ' en las condiciones del join, de la cláusula WHERE.

### **Objetos de la base de datos.**

Se debe observar la implantación o creación de triggers puesto que esto es importante. Algunos productos emplean sus mecanismos de triggers de forma oculta para implantar sus constraints de integridad declarativa. Esto es aún más importante si se desea implantar las reglas del negocio usando triggers. En cuanto a BLOB's (binary large lenguaje), todos los productos revisados tienen la capacidad de soportarlos, los cuales pueden ser utilizados para almacenar imágenes, texto, documentos, voz, audio y cualquier otro dato sin estructura. Sin embargo no todas las herramientas y lenguajes de desarrollo de aplicaciones pueden manejar con facilidad este tipo de datos. Algunos ambientes de desarrollo de cuarta generación pueden no ser capaces de procesar los BLOB's proporcionados por el manejador de bases de datos.

### **Distribución y conectividad.**

Algunos productos agregan también el uso de llamadas a procedimientos remotos, los cuales permiten acceder datos en bases de datos remotas como si estos datos estuvieran almacenados localmente. SYBASE en particular tiene una poderosa implantación de llamadas a procedimientos remotos. El soporte para base de datos distribuidas que ofrecen los proveedores varía considerablemente. Algunos productos como SYBASE ofrecen un shell de múltiples bases de datos que permiten acceder bases de datos remotas.

### **Tablas comparativas.**

Como hemos visto cada uno de los productos analizado tiene excelentes características en algunas áreas mientras su calidad es durable en otras, por lo cual es necesario hacer algunas tablas comparativas que incluyan un conjunto de características que se puedan ponderar de acuerdo a su importancia y finalmente comparar los productos de acuerdo a los requerimientos. Deben feerse reportes y suplementos de comparación que aparecen en revistas especializadas y así obtener buenos puntos de referencia.



**Modelo de Datos Relacional**

	Oracle 7	Sybase	Informix	Microsoft SQL Server
Dominios	No	No	No	No
Opiniones de violación de integridad referencial.	Restrictiva excepto el borrado en cascada.	Solo restrictiva	Restrictiva excepto el borrado en cascada.	Solo restrictiva
Personalizar mensajes referenciales.	No	No	No	No
Cláusula WHERE referencial	No	No	No	No

Tabla 4.1.1. Modelo de datos relacional

**Objetos de la Base de Datos.**

	Oracle 7	Sybase	Informix	Microsoft SQL Server
Vistas actualizables con opción de verificación	Si	Si	Si	Si
Tipos de datos definidos por el usuario	Si	Si	Si	Si
BLOB's	Si	Si	Si	Si
Tipos de datos soportados	Image, video, text, messaging, spatial, data types.	Binary, image, money bit, text, varbinary.	Byte, text hasta 2GB.	Image, video, text, Money, bit, binary
Estructuras de índices	B-Tree, bitmap y hash.	B-Tree	B+Tree y clustered.	Clustered
Estructuras de tablas.	Heap y clustered.	Heap y clustered.	Sin selección.	Sin selección.
Facilidades de optimización "Tunning"	Distribución de índices de tablas.	Index, prefetch, i/o buffer cache, block size.	Extents, fragmentación de tablas o roud robin.	Fill factors.
Nivel de triggers.	Basado en registro y basado en conjunto de registros.	Basado en conjunto de registros.	Basado en registro y basado en conjunto de registros.	Basado en conjunto de registros.
Ejecución del trigger.	Antes y después de la operación que disparó el trigger.	Después de la operación que disparó el trigger.	Antes y después de la operación que disparó el trigger.	Después de la operación que disparó el trigger.
Anidamiento del trigger	Si	Si	Si	Si
Lenguaje usado en stored procedure.	PL/Sql	Transact-sql	Sql	Transact-sql

Tabla 4.1.2. Objetos de Base de Datos.

**Herramientas Visuales de Desarrollo.**

La elección de la herramienta de desarrollo es un aspecto que debe revisarse con cuidado ya que muchas veces puede ser el aspecto que lleve a la realización exitosa de un proyecto o a su fracaso. De igual forma, las herramientas que no puedan reunir las expectativas de desempeño, llevarán al fracaso después de que ésta sea entregada a los usuarios finales.



## Visual C++

Visual C++ cuenta con herramientas de programación visuales impresionantes, pero todavía se tiene que saber programación de Windows. Sus herramientas visuales App Wizard, Class Wizard y el editor de recursos App Studio, de diseño excelente están disponibles dentro del WWB (Wizard Workbench). Parte de la integración se logró a través de un manejador de dispositivo virtual. Todas las herramientas de visualización de Visual C++ son para trabajos en Windows.

Puede generar aplicaciones para DOS si se tiene la edición profesional, pero no es en eso en lo que Visual C++ se destaca. Las herramientas App Wizard y Class Wizard dependen de la biblioteca Microsoft Foundation Class (MFC), que incluye la mayor parte del juego de programación de Windows (SDK) en una estructura de clase de C++. Se requiere programar solamente en C o escribir directamente a la API (Interfaz de Programación de Aplicaciones) de Windows, los Wizards no le serán útiles.

El App Wizard permite seleccionar funciones básicas como "Printing = impresión", "Previewing = Vista anticipada" y "Context Sensitiv Help = Ayuda sensible al contexto" y luego generar código basado en MFC.

Es fácil generar la estructura de una aplicación, personalizarla requiere un amplio conocimiento del lenguaje C++ y de MFC. Los programas Wizard son herramientas buenas para el aprendizaje y pueden aumentar considerablemente la productividad de aquellos que tienen una buena base de C++ y MFC, sin embargo no se protege al usuario de los peligros internos de Windows.

La herramienta más importante del C++ es App Studio, que es un editor de recursos maravilloso. Puede manejar cualquier recurso y es completamente visual. Con el App Studio la creación de un menú cambia totalmente.

## Visual Basic 5.

Visual Basic 5 concluye toda una generación en este lenguaje. Con su remozada interfaz (que puede regresarse a la de las versiones anteriores) incluye muchas más opciones que Visual Basic 4. La principal novedad, por la que es buscado, es la creación de controles ActiveX, que debió incluirse debido a las presiones comerciales recibidas por Delphi (versión visual de Pascal) de Borland, que captó muchos desarrolladores desilusionados con el atrasó de Visual Basic 4. También es muy promocionado el aumento en el performance, debido a la posibilidad de generar código compilado en vez de código interpretado como en las versiones anteriores. Aclaremos que a pesar de tratarse de código compilado, es necesario seguir incluyendo el "runtime". En aplicaciones de distribución normal (vía diskettes o CD-ROM) no se nota el tamaño de este archivo, pero si se desea crear controles ActiveX para incluir en páginas HTML, ocupando 1.5 MB a veces es necesario pensar en otras herramientas de desarrollo. Microsoft aclaró que se incluirá en la próxima versión de su explorador y en la actualización de Windows 95 y Windows NT.



Denominado "Dream release " por el equipo de desarrolladores, incluye muchas novedades, como la posibilidad de armar otra clase de componentes, ActiveX documents, que cierran la historia del MDI, permitiendo aplicaciones realmente centralizadas en los documentos OLE, que pueden incluirse en cualquier contenedor de objetos OLE.

Este lenguaje ha crecido conforme aparece una nueva versión. Se orienta, sobre todo al desarrollo empresarial y de redes, como Intranet, Extranet e Internet. Además incluye diversas utilerías que lo integran, incluso, una serie de herramientas de programación.

Cuenta con cuatro diferentes ediciones que se adaptan a los requerimientos del usuario:

- Working Model, que es gratuita y puede obtenerse desde la red internet. Esta edición es mas apta para conocer el entorno y ver algunas de sus características, que para utilizarse en un ambiente de producción.
- Aprendizaje (antes estándar), aquella que se consigue junto con una serie de libros y que sirve para aprender programación en Visual Basic. Incluye varias funciones completas (se pueden generar ejecutables), aunque carece de suficientes características para generar aplicaciones de tipo profesional, ésta es la más utilizada en el medio de los desarrolladores. Incluye gran parte de ~~herramientas necesarias para generar aplicaciones comerciales y~~ profesionales aptas para cubrir casi cualquier necesidad.
- Empresarial, es la más completa. Incluye herramientas que se utilizan en el entorno empresarial, como acceso a datos en redes LAN. También cuenta con la capacidad de crear grupos de desarrollo controlados y algunos otros aspectos del desarrollo empresarial, como la conexión directa a bases de datos Oracle y Microsoft SQL Server.

Todas estas versiones pueden encontrarse en el idioma español, así como la documentación, herramientas de ayuda, mensajes y utilerías diversas.

#### **Delphi 4.**

Este lenguaje nació grande y, más que crecer, se ha pulido en cada versión. Se apoya en Object Pascal, con lo que es posible acceder a estructuras de datos y a la programación orientada a objetos (OOP).

Permite, además, generar aplicaciones en modo de consola, así como aplicaciones visuales. Con algunas adaptaciones, se pueden ejecutar programas generados en versiones anteriores de pascal que no hagan uso de Turbo Vision.

Desde que apareció, ofreció la generación de ejecutables autónomos (que no requieren de archivos en tiempo de ejecución y un modelo de objetos basado en VCL (Virtual Component Library, Componente de Librerías Virtuales), aunque también soporta VBX, OCX y ActiveX sin mayor problema, así como los esquemas OLE y COM, y otras



propuestas como CORBA. Además permite el desarrollo de aplicaciones que tienen que ver con Internet.

Existen tres versiones que son:

- Estándar, que es un entorno básico orientado al aprendizaje y a su uso por parte de desarrolladores que no requieren de muchas características.
- Profesional, dirigida al grueso de los desarrolladores. Permite la generación de controles OCX, ActiveX y VCL, así como la emulación de entornos de datos distribuidos mediante el Interbase.
- Cliente / Servidor, dirigida a usuarios poderosos y de entorno empresarial. Incluye el control de código fuente y grupos de desarrollo, así como el acceso a bases de datos distribuidas, muy en especial de Oracle.

Entre las desventajas de Delphi se encuentra el no tan extenso soporte a nivel mundial y la relativa dificultad para encontrar desarrolladores especializados; así como tampoco existen versiones en español.

#### Tablas comparativas.

Producto	Compañía
Visual Basic 5.0	Microsoft Corporation
Visual C++	Microsoft Corporation
Delphi	Borland Inc.

Tabla 4.1.3. Compañía del producto.

Producto	Facilidad de Aprendizaje
Visual Basic 5.0	Bueno
Visual C++	Regular
Delphi	Regular

Tabla 4.1.4. Evaluación de Facilidad de uso.

Producto	Evaluación de Seguridad
Visual Basic 5.0	Bueno
Visual C++	Bueno
Delphi	Regular

Tabla 4.1.5. Evaluación de Seguridad.

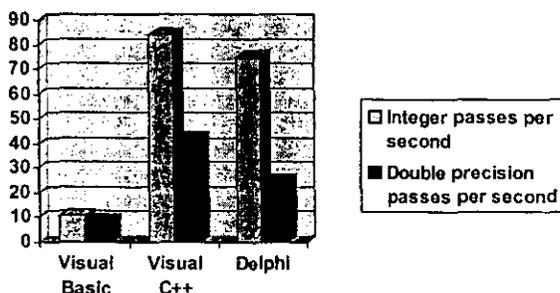


	Test	Visual Basic	Delphi	Visual C++
Display	Empty display	52.509	26.85	86.667
	Labels	0.316	1.295	4.748
	Bitmaps	1.226	3.296	8.925
Database	Overhead	9.75	7.838	6.935
	Small retrieval	0.34	0.612	0.393
	Large retrieval	0.25	0.11	0.161
	In process method	0.000295	0.082	N/A
	Out of process let	1.066	2.572	N/A
	Out of process method	1.039	2.572	N/A
Lenguaje	Sieve of Eratoshenes	360.625	3323.82	16940.12
	String manipulation	900.35	48.13	1941.36

Tabla 4.1.6. Fuente: Carnegie Technology Group, Inc. (los números pequeños son mejores).

	Integer passes per second	Double precision passes per second
Visual Basic 5.0	11.1	8.8
Visual C++	84	42.6
Delphi	74.9	24.9

Tabla 4.1.7. Fuente: Carnegie Technology Group, Inc. (los números pequeños son mejores).



Grafica 4.1.1. Interpretación grafica de la tabla 4.1.7.



## 4.2 ELECCIÓN DE LA SOLUCIÓN ÓPTIMA

Una opción para poder encontrar la solución óptima, es centrarse en el problema principal y a partir de él derivar la opción más adecuada para el desarrollo de la misma; no sin antes prevenir todos los requerimientos que debe tener un buen sistema, como son:

- La relación costo-beneficio.
- El tiempo de entrega.
- El tiempo para capacitar a los usuarios.
- El tiempo para adaptarse al sistema.
- La compatibilidad con otras plataformas.

Además de estos requerimientos, es necesario saber que la elección de una solución no es del todo libre, también se tiene que considerar que tal vez la empresa no desea o no puede comprar nuevos recursos, y quiere que se trabaje con la infraestructura actual de la empresa, el sistema debe ser capaz de adaptarse a estos recursos. O bien, si la empresa esta dispuesta a comprar nuevos recursos, se debe realizar un análisis tanto de las necesidades del sistema, como de las necesidades y políticas de la empresa, para que se proporcione un mayor rendimiento, con el mínimo de recursos, sin afectar los actuales sistemas o programas que maneja la empresa en otras áreas.

Para evitar problemas de inconsistencia en la información se evaluaron diferentes alternativas para desarrollar el sistema de control de cartera para una aseguradora, el cual elimine toda la problemática.

Para el desarrollo de este sistema se evaluaron tres diferentes herramientas de programación como ya se vio anteriormente, las cuales son:

Para el Back End:

- Oracle7
- Sybase
- Informix
- SQLServer

Para el Front End:

- Visual C++
- Delphi
- Visual Basic

### 4.2.1. Comparación de los programas para desarrollar Back Ends.

En los apartados anteriores se fueron presentando las características de los manejadores de bases de datos factibles para el desarrollo del back end. Para poder compararlos es necesario tomar en cuenta todas esas características importantes que buscamos en un sistema para desarrollar y manipular bases de datos.



La tabla 4.2.1.1 nos muestra la forma de acceso a la información, para poder comparar los tres manejadores de bases de datos.

ACCESO A LA INFORMACIÓN	VISUAL C++	DELPHI	VISUAL BASIC
Conectividad completa con todos los formatos de datos	√		
Crea nuevas tablas en cualquier formato		√	√
Habilidad para trabajar con varios formatos de datos	√	√	√
Salva las tablas en cualquier formato de base de datos.			√

Tabla 4.2.1.1. Acceso a información

La tabla 4.2.1.2, muestra las características de personalización de los programas.

CAPITULO 1	PERSONALIZACIÓN	VISUAL C++	DELPHI	VISUAL BASIC
	Asistentes para construir rápidamente formas, reportes y demás		√	√
	Un verdadero generador de reportes para la edición rápida y precisa de éstos.		√	√
	Soporte de lenguaje de programación visual.	√	√	√
	Variedad de validaciones de campos y valores predeterminados.	√		√

Tabla 4.2.1.2. Personalización

Ahora que ya se compararon cada una de las opciones, se puede generar la conclusión pertinente de selección.

Después de analizar cada una de las opciones que se tenían como factibles para desarrollar el sistema manejador de bases de datos, se concluyó que a pesar de las grandes ventajas que ofrece, Oracle, Sybase, e Informix la mejor solución es SQLServer, ya que nos ofrece la mejor relación costo-beneficio.

Se utilizará SQLServer, porque además de ser un manejador poderoso y fácil de usar, la empresa cuenta con este Software, además de que se sabe que se cuenta con la licencia para utilizar el SQLServer.

También se eligió SQLServer, porque como la empresa ya cuenta con este software, no será muy difícil la capacitación del personal, ya que han trabajado con él anteriormente.



#### 4.2.2. Comparación de los programas para desarrollar Front Ends.

En la tabla 4.2.2.1 observamos la comparación entre Visual Basic 5.0, Delphi 3.0 y Visual C++, con relación al despliegue en pantalla y a la base de datos.

		Visual Basic	Delphi	Visual C++
Despliegue en pantalla	Etiquetas	0.755	1.295	4.748
	Mapa de bits	2.42	3.296	8.925
Bases de datos	Desbordamiento	22.234	7.838	6.935
	Small retrieval	1.415	0.612	0.393
	Large retrieval	0.218	0.11	0.161

Tabla 4.2.2.1. Comparación de productos. Los números representan ciclos de reloj, por tanto los números pequeños son mejores.

En la siguiente tabla 4.2.2.2 se observa la comparación de los mismos productos, pero ahora con relación al número de ciclos de reloj en operaciones con enteros y dobles.

	Visual Basic	Delphi	Visual C++
Ciclos de reloj por segundo con enteros	3	79.4	75.2
Ciclos de reloj por segundo con dobles	2.6	24.9	34.8

Tabla 4.2.2.2. Comparación con respecto al número de ciclos de reloj en operaciones. Los números pequeños son mejores.

De acuerdo al análisis hecho anteriormente, como sucedió con el Back End, no existen suficientes razones para adquirir un nuevo Software para la interfaz gráfica, ya que la empresa cuenta con la licencia para utilizar Visual Basic 5.0, y el equipo para poder instalar la aplicación desarrollada con esta herramienta.

Además de que el 80% del personal con el que cuenta la empresa maneja, o conoce Visual Basic 5.0, esto hará que el sistema se implemente y se utilice más rápidamente, y por tanto también ofrecerá una mejor relación costo-beneficio.

Otra de las grandes ventajas de utilizar Visual Basic 5.0 sobre Delphi, es la facilidad de aprendizaje, y la seguridad que proporciona, lo que se ve reforzado por los comentarios de quienes opinan que aprender Delphi es casi tan complejo como aprender lenguajeC. Visual Basic es el lenguaje de programación visual con mayor soporte a nivel mundial. Se puede obtener ayuda e información por diversos medios y es relativamente sencillo.



encontrar desarrolladores especializados.

Al trabajar con Visual Basic 5.0 los programas se procesan con mayor velocidad.

La combinación de SQLServer con Visual Basic 5.0 par el manejo de un volumen de datos considerable (como ejemplo, una base de datos de un millón de registros) el rendimiento es muy bueno.



## **CAPÍTULO V**

### **DESARROLLO E IMPLANTACIÓN DEL SISTEMA**



## 5.1. BASE DE DATOS.

A continuación se presentaran el diagrama de contexto, los diagramas de flujo de datos, la base de datos y el proceso de normalización considerado para el sistema de control de cartera para aseguradoras.

### 5.1.1. Diagrama de contexto.

En la figura 5.1.1.1 mostramos el diagrama de contexto ó diagrama de flujo de datos de nivel cero, este diagrama presenta las entidades que se ven involucradas en las entradas y las salidas de información, la cual fluirá y será manipulada a través del sistema.

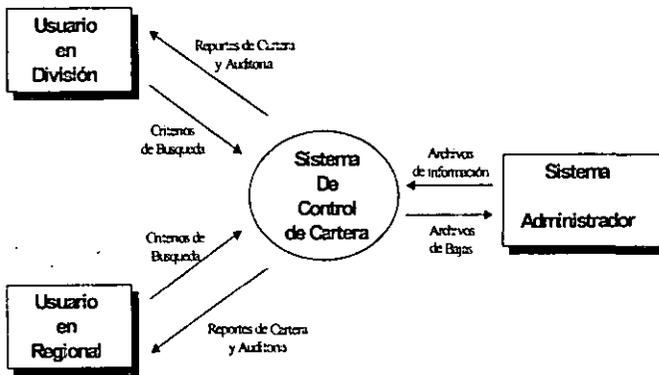


Figura 5.1.1.1 Diagrama de contexto.

Como podemos observar en el diagrama de la figura 5.1.1.1, las tres entidades actúan como entradas y salidas de información.

Los usuarios en las divisiones y en las oficinas regionales, tienen la misma funcionalidad, alimentan al sistema la información que están buscando y reciben su información, ya sea en pantalla o en papel, y estos reportes son los que se entregan a todas las áreas y a los clientes.

El sistema administrador enviará la información de los recibos de póliza generados, y también se le regresará al sistema administrador la información de los recibos recuperados.

### 5.1.2. Diagrama de flujo de datos.

En el presente capítulo se tratará el flujo de la información involucrada en el sistema, ejemplificando todos y cada uno de los procesos, partiendo de lo general a lo particular, diagramando el proceso desde el nivel uno hasta el nivel tres y cuatro en algún casos.



En la figura 5.1.2.1 se muestra el diagrama de flujo de datos de nivel 1, en el cual podemos observar las dos principales entradas de información del proceso central del sistema y la información que saldrá, a través de las consultas y reportes que se generarán a través del mismo.

En las figuras 5.1.2.2 y 5.1.2.3 se muestran los diagramas de flujo de datos de nivel 2; puesto que el proceso de clasificación y consulta de información que se muestra en el diagrama de nivel 1 (figura 5.1.2.1), el cual se dividirá en dos partes, la parte del amparado, que son todos los recibos vencidos por más de tres meses, y la parte del pendiente, que son todos los recibos vencidos con menos de tres meses ó que aún no han vencido.

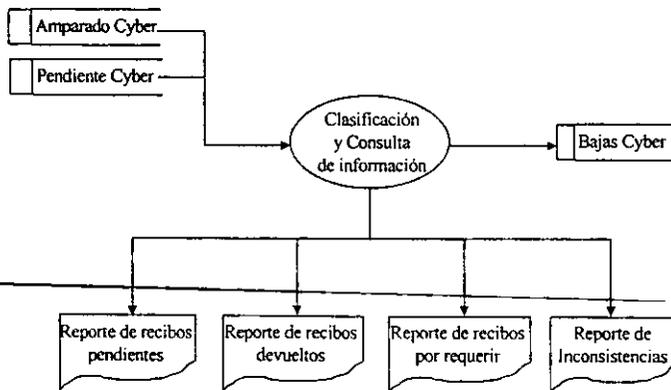


Figura 5.1.2.1 Diagrama de flujo de datos de nivel 1 (Proceso general).

Los recibos pendientes son todos aquellos que su fecha de vencimiento ya caducó pero tienen menos de tres meses de haber vencido y son recibos de consulta exclusivamente, los recibos del amparado son todos aquellos que ya vencieron y su fecha de vencimiento es de más de tres meses de antigüedad, estos recibos pasan por un proceso de auditoría y un proceso de corrección de información, ya que el agente que vendió la póliza algunas veces no es capturado y esto origina que el recibo no se pueda rastrear, pero se conoce a quien se le dio este, por lo que se le debe de asignar a alguien en este punto, así como también, la parte del amparado generará un archivo de información, el cual se regresará al sistema administrador con el fin de dar de baja los recibos recuperados o que su situación ya ha sido aclarada dentro del proceso de auditoría.

La explotación de cada una de las burbujas o procesos de los diagramas de flujo de datos de nivel 2, que requieran un nivel más de explotación mayor se mostrarán más adelante.

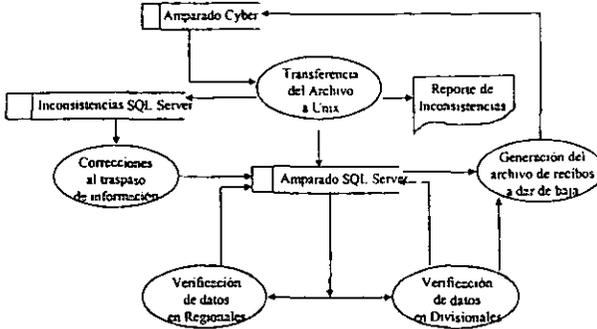


Figura 5.1.2.2 Diagrama de flujo de datos de nivel 2. (Amparado)

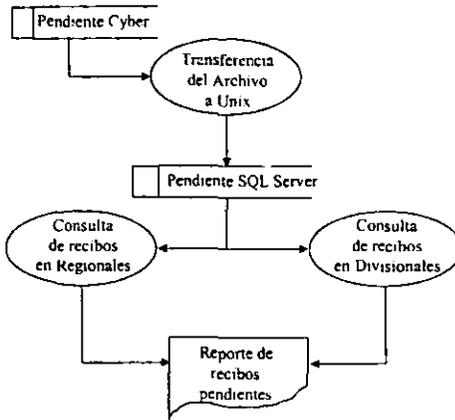


Figura 5.1.2.3 Diagrama de flujo de datos de nivel 2. (Pendiente)

En primera instancia analizaremos el nivel 2 de la parte del amparado ejemplificada en el diagrama de la figura 5.1.2.2, la cual tiene cinco diagramas de nivel 3 que son los que se detallan a continuación.

El primer diagrama (figura 5.1.2.4) plasma la transferencia del archivo del equipo Cyber a Unix, el cual nos muestra como el usuario dispara el proceso de transferencia y pasa por un proceso de validación, generando inconsistencias e información lista para consultar, así como un reporte de inconsistencias.

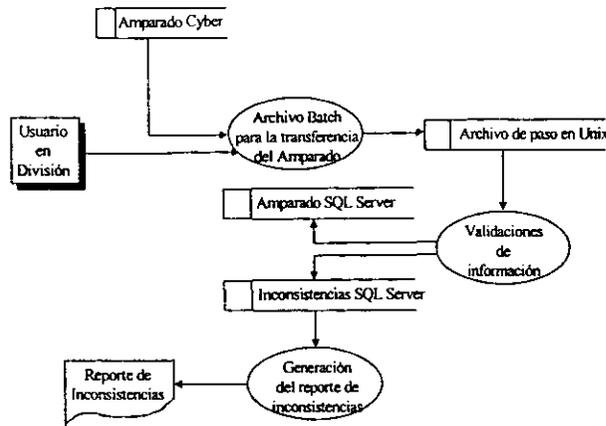


Figura 5.1.2.4 Diagrama de flujo de datos de nivel 3. (Transferencia del archivo del amparado)

La información que tiene una clave de agente invalida, debe pasar por un proceso de corrección de información, el cual se expone en el diagrama de la figura 5.1.2.5, y debe ser corregida para no alimentar información inconsistente o errónea en la nueva base de datos, esto nos ayudará a poder identificar las inconsistencias que existen en la información, tanto en la información correcta como en la información errónea. Ya que el área de emisión de pólizas registra en ocasiones agentes inexistentes, lo cual implica un fraude y se requiere rastrear.

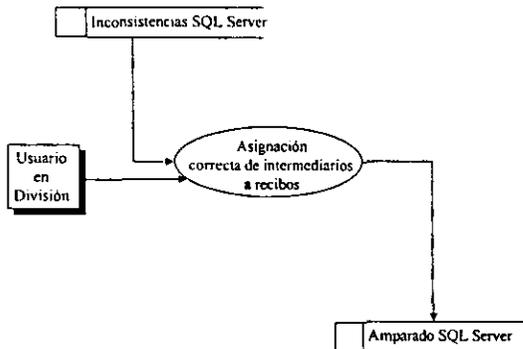


Figura 5.1.2.5 Diagrama de flujo de datos de nivel 3. (Corrección de información)

En los diagramas de las figuras 5.1.2.6 y 5.1.2.7 se muestran los procesos de verificación de información en oficinas regionales y en oficinas divisionales, este proceso eliminará el viaje del recibo auditando desde la oficina regional y la oficina divisional, ya que el objetivo será que la oficina regional hará el requerimiento del recibo al agente y la división valide las observaciones sin necesidad de trasladarse a la región y evitando también estos gastos, ya que ahora la oficina regional también auditará a los agentes y la oficina divisional auditara a la oficina regional.

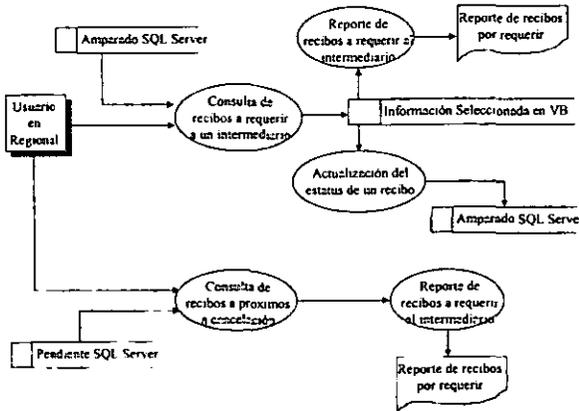


Figura 5.1.2.6 Diagrama de flujo de datos de nivel 3. (Proceso en oficina regional)

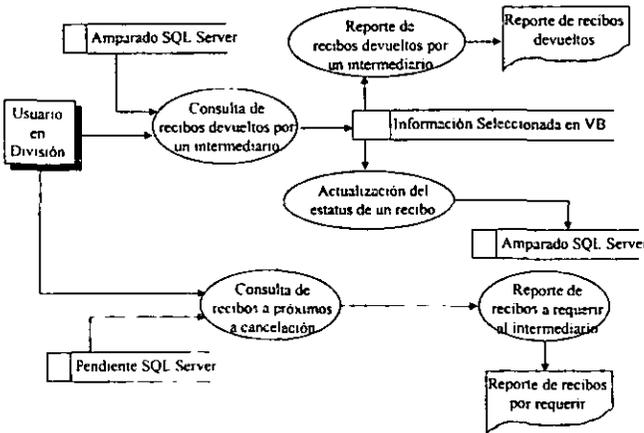


Figura 5.1.2.7 Diagrama de flujo de datos de nivel 3. (Proceso en oficina divisional)

A su vez, las oficinas regionales generarán directamente al usuario que lo requiera sus reportes de cartera pendiente y vencida (amparada), con información del día y completa y el usuario de la división generará reportes de cartera recuperada y ambos usuarios podrán emitir reportes de cartera pendiente.

Y por ultimo el proceso del amparado debe generar un archivo de recibos recuperados para su eliminación en el sistema administrador, este archivo lo genera la oficina divisional con la información obtenida y será enviada al equipo Cyber para que se eliminen estos datos del sistema como se muestra en la figura 5.1.2.8.

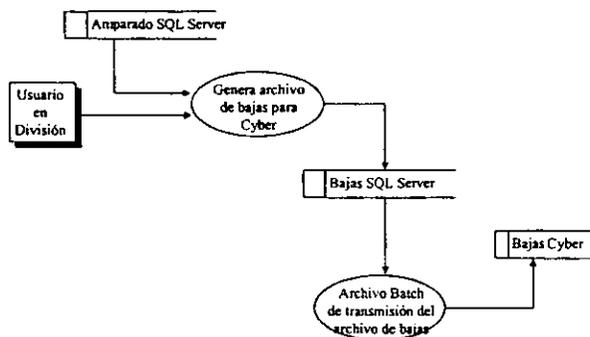


Figura 5.1.2.8 Diagrama de flujo de datos de nivel 3. (Baja de información)

Ahora toca el turno de analizar la parte de recibos pendientes, la cual tiene tres diagramas de nivel 3 que son los que se detallan a continuación.

El primer diagrama (figura 5.1.2.9) plasma la transferencia del archivo del equipo Cyber a Unix, el cual nos muestra como el usuario dispara el proceso de transferencia pasa por un proceso de validación, generándose inconsistencias e información lista para consultar, pero únicamente la información es importante, ya que las inconsistencias únicamente son consultadas dentro del proceso del amparado, y no se realiza ninguna corrección, solamente se informa a la parte de emisión para su corrección directamente en el sistema administrador.

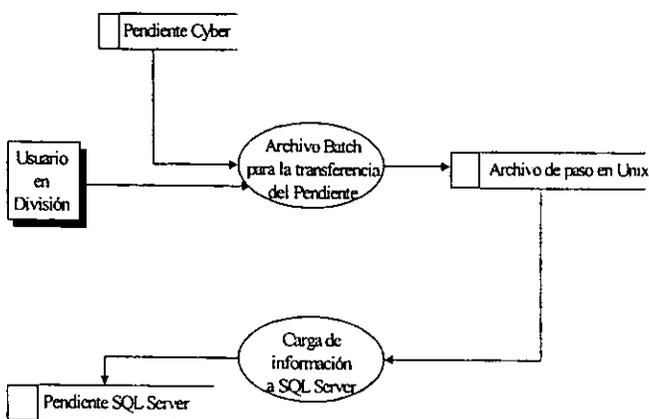


Figura 5.1.2.9 Diagrama de flujo de datos de nivel 3. (Transferencia del archivo de pendientes)



La consulta de información de los recibos pendientes, es igual en las oficinas regionales y en las oficinas divisionales, como se muestra en las figuras 5.1.2.10 y 5.1.2.11, ya que es información la cual no es requerida, y que puede ser consultada

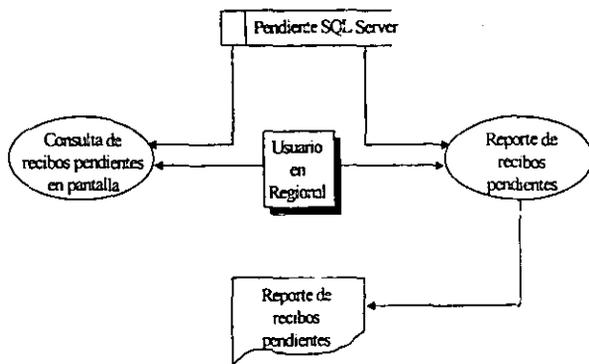


Figura 5.1.2.10 Diagrama de flujo de datos de nivel 3. (Consulta de pendientes en regionales)

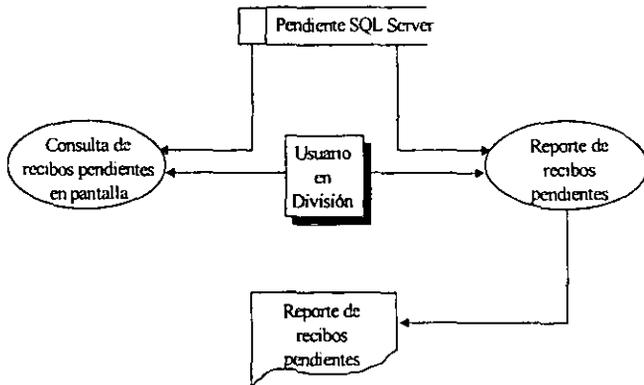


Figura 5.1.2.11 Diagrama de flujo de datos de nivel 3. (Consulta de pendientes en divisionales)

para ayudar al agente a la recuperación de estos pagos, estos reportes también nos ayudan a ver cual es la cartera que probablemente ingrese a la compañía, y con información oportuna podemos planear mejor a futuro, ya que el presupuesto que la dirección planea será mejor orientado.



### 5.1.3. Diccionario de datos.

El diccionario de datos es una herramienta que nos muestra los datos manejados en el sistema y sus características, en la lista que a continuación se presenta, mostramos el nombre de la tabla, el nombre del campo y su tipo de dato, así como un indicador de llave primaria (PK) y de llave foránea (FK) y una pequeña descripción del dato.

TABLA	COLUMNA	TIPO DE DATO	P. K.	F. K.	DESCRIPCIÓN
carchvo	clve_carchvo	smallint	X		Clave del archivo
	dscrpcion_carchvo	char(30)			Descripción del archivo
cerror	clve_cerror	smallint	X		Clave del Error
	dscrpcion_cerror	char(30)			Descripción del Error
cestdo_rcbo	clve_cestdo_rcbo	smallint	X		Clave del Estado del Recibo
	dscrpcion_cestdo_r	char(30)			Descripción del Estado del Recibo
	nvel_cestdo_rcbo	smallint			Nivel del Estado del Recibo
cfrma_pgo	clve_cfrma_pgo	smallint	X		Clave de la Forma de Pago
	dscrpcion_cfrma_p g	char(30)			Descripción de la Forma de Pago
cintrmdrio	clve_cofcna	int		X	Clave de la Oficina
	dscrpcion_cintrmdr	char(30)			Nombre del Intermediario
	nmro_prmtria_adm ns	int		X	Número de la Promotoría
	tpo_intrmdrio_spr	smallint		X	Tipo de la Promotoría
	nmro_prmtria_vnta	int			Promotoría de Venta
	clve_ctpo_intrmdri	smallint	X	X	Tipo de Intermediario
	clve_cintrmdrio	int	X		Clave del Intermediario
	clve_alfnmrco_nvo	char(5)			Clave Alfanumérica Nuevo
	clve_alfnmrco_cint	char(5)			Clave Alfanumérica
	clve_nmrco_cintrm d	int			Clave Numérica
cmnda	clve_cmnda	smallint	X		Clave de la Moneda
	dscrpcion_cmnda	char(30)			Descripción de la Moneda
cofcna	clve_cofcna	int	X		Clave de la Oficina
	dscrpcion_cofcna	char(30)			Descripción de la Oficina
	clve_ctpo_ofcna	smallint		X	Tipo de Oficina
	nmro_ofcna_sprior	int		X	Oficina Superior
	rsponsble_cofcna	char(30)			Responsable de la Oficina
	virtual_cofcna	smallint			Clave Virtual
corgen	clve_corgen	char(1)	X		Clave del Origen
	dscrpcion_corgen	char(30)			Descripción del Origen
crmo	rmo_vda_crmo	smallint			Indicador de Ramo de Vida
	clve_cuen	smallint		X	Clave de la UEN
	dscrpcion_crmo	char(30)			Descripción del Ramo



**DISEÑO E IMPLANTACIÓN DE UN  
SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS**

TABLA	COLUMNA	TIPO DE DATO	P K	F K	DESCRIPCIÓN
	clve_crmo	char(2)	X		Clave del Ramo
csstma	clve_csstma	smallint	X		Clave del Sistema
	dscrpccion_csstma	char(30)			Descripción del Sistema
	rspnsble_csstma	char(30)			Responsable del Sistema
ctpo_intrmdrio	clve_ctpo_intrmdri	smallint	X		Clave del Tipo de Intermediario
	dscrpcion_ctpo_int	char(30)			Descripción del Tipo de Intermediario
ctpo_ofcna	clve_ctpo_ofcna	smallint	X		Clave del Tipo de Oficina
	dscrpcion_ctpo_ofc	char(30)			Descripción del Tipo de Oficina
ctpo_rcbo	clve_ctpo_rcbo	smallint	X		Clave del Tipo de Recibo
	dscrpcion_ctpo_rcb	char(30)			Descripción del Tipo de Recibo
cuen	clve_cuen	smallint	X		Clave de la UEN
	dscrpcion_cuen	char(30)			Descripción de la UEN
	rspnsble_cuen	char(30)			Responsable de la UEN
cusrio	clve_cusrio	char(8)	X		Usuario
	pssword_cusrio	char(8)			Contraseña
	dscrpcion_cusrio	char(40)			Nombre del Usuario
	clve_cofcna	int		X	Oficina a la que pertenece
hrcbo_mvmtnt o	fcha_audtria_mvmt nt	datetime			Fecha de Auditoría
	clve_cestdo_rcbo	smallint		X	Estado del Recibo
	nmro_rchzo_mvmtnto	smallint	X		Número de Rechazo
	hra_audtria_mvmtnto	char(5)			Hora de Auditoría
	csa_rchzo_mvmtnto	varchar(100)			Causa de Rechazo
	clve_cusrio	char(8)		X	Usuario
	nmro_cmbio_amprdo	int	X	X	Número de Cambio
	nmro_plza_amprdo	char(13)	X	X	Número de Póliza
	fcha_vncmnto_amp rd	char(6)	X	X	Fecha de Vencimiento
	clve_ctpo_rcbo	smallint	X	X	Tipo de recibo
	imprte_prma_amp do	decimal(12, 2)	X	X	Importe de la Prima
tcrga_intrmdrio o	clve_cintrmdrio	int	X		Clave del Intermediario
	clve_ctpo_intrmdri	smallint	X		Tipo de Intermediario
	clve_alfnmrco_cint	char(5)			Clave Alfanumérica
	clve_nmrco_cintrm d	int			Clave Numérica
	clve_alfnmrco_nvo	char(5)			Clave Alfanumérica Nuevo



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

TABLA	COLUMNA	TIPO DE DATO	P K	F K	DESCRIPCIÓN
	dscrpcion_cintrmdr	char(30)			Nombre del Intermediario
	clve_cofcna	int			Clave de la Oficina
	nmro_prmtria_adms	int			Número de la Promotoría
	tpo_intrmdrio_spr	smallint			Tipo de la Promotoría
	nmro_prmtria_vnta	int			Promotoría de Venta
tcrga_rcbos	nmro_plza_crga	char(13)	X		Número de Póliza
	nmro_cmbio_crga	int	X		Número de Cambio
	clve_ctpo_rcbo	smallint	X		Tipo de Recibo
	imprte_prma_crga	decimal(12, 2)	X		Importe de la Prima
	fcha_vncmnto_crga	char(6)	X		Fecha de Vencimiento
	clve_cintrmdrio	int			Clave del Intermediario
	clve_ctpo_intrmdri	smallint			Tipo de Intermediario
	nmro_agnte_crga	char(5)			Número de Agente
	nmro_prmtor_crga	char(5)			Número de Promotor
	clve_crmo	char(2)			Clave del Ramo
	rcbo_rmo_vda_crga	smallint			Ramo de Vida
	clve_cfrma_pgo	smallint			Forma de Pago
	clve_cmnda	smallint			Moneda
	clve_cbrnza_crga	char(5)			Clave de Cobranza
	nmbre_asgrdo_crga	char(30)			Asegurado
	fcha_mvmento_crga	datetime			Fecha de Movimiento
	clve_carchvo	smallint			Archivo
	clve_corgen	char(1)			Origen
	clve_cerror	smallint			Error
tintrfces	nmbre_tintrfces	char(8)	X		Interface
	vrision_tintrfces	smallint			Versión Cargada
tprmso_sstma	clve_csstma	smallint	X	X	Clave del Sistema
	fcha_tprmso_sstma	datetime			Fecha de Creación
	clve_cusrío	char(8)	X	X	Usuario
trcbo_amprdo	clve_cbrnza_amprdo	char(5)			Clave de Cobranza
	nmbre_asgrdo_amprdo	char(30)			Asegurado
	clve_cmnda	smallint		X	Moneda
	clve_cfrma_pgo	smallint		X	Forma de Pago
	clve_cestdo_rcbo	smallint		X	Estado del Recibo
	fcha_mvmento_amp_rdo	datetime			Fecha de Movimiento
	clve_corgen	char(1)		X	Origen



**DISEÑO E IMPLANTACIÓN DE UN  
SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS**

TABLA	COLUMNA	TIPO DE DATO	P K	F K	DESCRIPCIÓN
	clve_carchvo	smallint		X	Archivo
	ultmo_rchzo_amprdo	smallint			Número del Ultimo Rechazo
	nmro_cmbio_amprdo	int	X		Número de Cambio
	clve_ctpo_rcbo	smallint	X	X	Tipo de Recibo
	nmro_plza_amprdo	char(13)	X		Número de Póliza
	clve_crmo	char(2)		X	Clave del Ramo
	clve_ctpo_intrmdri	smallint		X	Tipo de Intermediario
	imprte_prma_amprdo	decimal(12, 2)	X		Importe de la Prima
	clve_cintrmdrio	int		X	Clave del Intermediario
	fcha_vncmnto_ampord	char(6)	X		Fecha de Vencimiento
trcbo_incsstnte	nmro_plza_incsstntn	char(13)	X		Número de Póliza
	nmro_cmbio_incsstnt	int	X		Número de Cambio
	clve_ctpo_rcbo	smallint	X	X	Tipo de Recibo
	imprte_prma_incssts	decimal(12, 2)	X		Importe de la Prima
	fcha_vncmnto_incssts	char(6)	X		Fecha de Vencimiento
	clve_cintrmdrio	int		X	Clave del Intermediario
	clve_ctpo_intrmdri	smallint		X	Tipo de Intermediario
	nmro_agnte_incsstnt	char(5)			Número de Agente
	nmro_prmtor_incssts	char(5)			Número de Promotor
	clve_crmo	char(2)		X	Clave del Ramo
	rcbo_rmo_vda_incssts	char(2)			Ramo de Vida
	clve_cfrma_pgo	smallint		X	Forma de Pago
	clve_cmnda	smallint		X	Moneda
	clve_cbrnza_incssts	char(5)			Clave de Cobranza
	nmbre_asgrdo_incssts	char(30)			Asegurado
	fcha_mvmnto_incssts	datetime			Fecha de Movimiento
	clve_carchvo	smallint		X	Archivo
	clve_corgen	char(1)		X	Origen
	clve_cerror	smallint		X	Error
trcbo_pndnte	nmro_plza_pndnte	char(13)	X		Número de Póliza



TABLA	COLUMNA	TIPO DE DATO	P	F	DESCRIPCIÓN
	nmro_cmbio_pndnte	int	X		Número de Cambio
	clve_ctpo_rcbo	smallint	X	X	Tipo de Recibo
	imprte_prma_pndnte	decimal(12, 2)	X		Importe de la Prima
	fcha_vncmnto_pndnte	char(6)	X		Fecha de Vencimiento
	clve_cintrmdrio	int		X	Clave del Intermediario
	clve_ctpo_intrmdrio	smallint		X	Tipo de Intermediario
	clve_crmo	char(2)		X	Clave del Ramo
	clve_cfrma_pgo	smallint		X	Forma de Pago
	clve_cmnda	smallint		X	Moneda
	clve_cbrza_pndnte	char(5)			Clave de Cobranza
	nmbre_asgrdo_pndnte	char(30)			Asegurado
	fcha_mvmtnto_pndnte	datetime			Fecha de Movimiento
	clve_carchvo	Smallint		X	Archivo
	clve_cestdo_rcbo	Smallint		X	Estado del Recibo

#### 5.1.4. Diagrama entidad relación.

A continuación, presentamos el diagrama entidad - relación (figura 5.1.4.1) que nos permite el manejo de información para el sistema de control de cartera que se va a desarrollar, este diagrama presenta todas y cada una de las tablas (entidades) y las relaciones que surgieron del análisis de información.

En el diagrama entidad-relación se puede observar, que los datos principales de los recibos están catalogados en tres tablas (trcbo\_amprdo, trcbo\_pndnte y trcbo\_incsstnte), estas tablas separan los recibos en amparados, pendientes e inconsistentes, con el fin de no mezclar la naturaleza de los recibos, aún y que los datos son iguales o muy similares.

También vemos todas las tablas de los catálogos que utilizará el sistema, asociadas a estas tablas de recibos, como son: ramos, archivos, formas de pago, tipos de recibo, monedas, origen e intermediarios (crmo, carchvo, cfrma\_pgo, ctpo\_rcbo, cmnda, corgen y cintrmdrio), para cada recibo, se ve involucrada una clave de estos catálogos, por lo que son relaciones directas e independientes.

Como se puede observar la entidad de los ramos (crmo) esta asociada a la unidad empresarial de negocio (cuen), ya que cada ramo pertenece a una unidad empresarial de negocio, de la misma manera, podemos observar que las oficinas y los tipos de oficina tienen una relación similar (cofca y ctpo\_ofca) también vemos a los intermediarios con los tipos de intermediario (cintrmdrio y ctpo\_intrmdrio) en una



relación similar de pertenencia de las oficinas y los intermediarios a un tipo específico.

También podemos observar que los usuarios y los intermediarios están relacionados con una oficina, ya que esta oficina es la que rige las agrupaciones de las búsquedas y el acceso a una cierta cantidad de información.

Los permisos de acceso al sistema se encuentran relacionados con los usuarios y con los sistemas (`tprms_ostma`, `cusrio` y `csstma`), estas tablas con estas relaciones se crearon con el objetivo de que la base de datos que se generó pueda crecer hacia otros sistemas, integrando el nuevo sistema a la tabla de sistemas (`csstma`) y los usuarios a la tabla de usuarios (`cusrio`) en el caso de que no existan, e incluir los permisos para acceder al sistema que se incluya en la tabla de permisos (`tprms_ostma`).

También observamos que se encuentra una relación de los recibos amparados con la historia de cada uno de estos (`trcbo_amprdo` y `hrcbo_amprdo`), ya que cada recibo auditado (únicamente amparados) mantiene su historia de movimientos, y junto a estas tablas se incluye la de recibos pendientes (`trcbo_pndnte`), para poder relacionar la tabla de estado del recibo (`cestdo_rcbo`) con estas, la cual contiene, como su nombre lo indica, el estado del recibo, entregado, rechazado, etc.

De la misma manera, podemos observar la relación que guarda la tabla de intermediarios consigo misma (entidad recursiva), ya que en esta tabla (`cintrmdrio`) se encontrarán registrados los agentes y los promotores, y a todo agente se le relaciona con una promotoría, que puede ser él mismo, y como ambos (promotor y agente) tienen la misma información, se registran en una sola tabla relacionándose a sí misma con los campos (`nmro_prmtria_adms` y `tpo_intrmdrio_spri`) que son la llave del registro superior y están registrados en esta misma tabla.

También contamos con tres tablas de paso, las cuales nos ayudan a manejar la información que se alimentará al sistema como los intermediarios y los recibos (`tcrga_intrmdrio` y `tcrga_rcbos`) y la versión de la interface de salida que se está procesando (`tintrfces`) para el archivo de bajas, otra tabla que nos ayuda a detectar el archivo de origen de los recibos es la tabla del archivo (`carchvo`) la cual nos dice si es pendiente ó amparado y si es de México, Monterrey o Guadalajara.

Cuando se realiza una carga de recibos y el intermediario original es incompleto o inválido o alguna clave de catálogo importante es inválida, este recibo se envía a la tabla de inconsistentes, y se indica el error detectado, sea por la clave del intermediario, principalmente agente o promotor, o cualquier otro caso de catálogo.





### 5.1.5. Normalización.

Con el fin de normalizar la información del flujo de datos que se verán involucrados en el sistema, se aplicaron las tres primeras formas normales (1NF, 2NF y 3NF) de la siguiente manera.

En la figura 5.1.5.1 se muestra la información contenida para cada uno de los movimientos de auditoría de recibos, esta es una entidad que requiere de normalización, con el fin de poder utilizar su información dentro del sistema.

#### Movimientos del recibo

Número de póliza  
 Número de cambio  
 Tipo de recibo  
 Importe de prima  
 Fecha de vencimiento  
 Número de rechazo  
 Estado del recibo  
 Fecha de auditoría  
 Hora de auditoría  
 Causa de rechazo  
 Usuario del movimiento

#### recibo\_movmto

```

numero_piza_emprdo char(13) (FK)
numero_cmbio_emprdo integer (FK)
clave_tipo_recbo smallint (FK)
importe_prima_emprdo decimal(12,2) (FK)
fecha_vencimto_emprdo char(6) (FK)
numero_rchzo_movmto smallint

clave_estado_recbo smallint (FK)
fecha_audtria_movmto date
hra_audtria_movmto char(5)
caa_rchzo_movmto varchar(100)
clave_usuario char(8) (FK)
    
```

Figura 5.1.5.1 Campos que componen los movimientos de recibos.

Todos los campos de los movimientos tienen su propio dominio, por lo que para cada registro solo existe un valor para cada campo, como podemos observar en el siguiente ejemplo:

CAMPO	DESCRIPCIÓN
Número de póliza, número de cambio, tipo de recibo, importe, vencimiento.	Es la llave única del recibo.
Estado del recibo	Es el estatus actual del recibo.
Usuario	Es la clave del usuario que registro el movimiento de auditoría.

Viendo estos ejemplos, podemos concluir que esta tabla ya se encuentra en la primera forma normal (1NF), por lo que cumple con la definición de esta forma, que dice, "La primera forma normal (1NF) es cuando todas las relaciones de la base de datos cumplen con la propiedad de tener un valor atómico en cada uno de sus atributos".



Ahora, considerando la definición de la segunda forma normal (2FN), que dice, "Una relación esta en segunda forma normal (2NF) si, y solo si, está en 1NF y todos los atributos que no son llave, dependen completamente de la llave primaria". Por lo que se realiza la siguiente división:

Como cada recibo contiene ramo, moneda, forma de pago, intermediario y tipo de recibo. Por lo que, se realiza la división de los movimientos hechos a un recibo y los datos generales de los recibos en otra tabla con el fin de separar la información redundante de los movimientos.

Ahora si, todos los campos, como se menciona en la definición, dependen funcionalmente de la llave primaria que es la misma en cada tabla nueva; esto hace cumplir la definición, por lo que ya se encuentran en la 2NF, como se muestra en la figura 5.1.5.2.

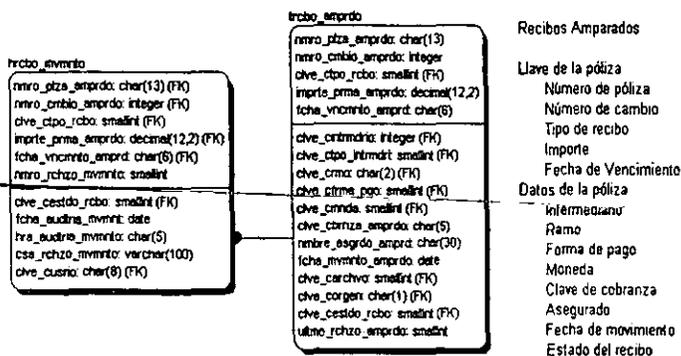


Figura 5.1.5.2 División de recibos y movimientos.

Continuando con la normalización de las tablas, el siguiente paso es llevarlas a la tercera forma normal (3NF), lo cual significa según la definición, "Una relación esta en la tercera forma normal (3NF) si, y solo si esta en 2NF y todos los atributos que no son llave principal, son mutuamente independientes".

Como podemos observar, se debe extraer de los datos generales de las pólizas todas las dependencias innecesarias, y convertir estas en tablas independientes, por lo que se deben crear tablas para ramos, monedas, formas de pago, intermediarios y tipos de recibo, descomponiendo la tabla de pólizas en datos generales y catálogos de la misma, quedando los datos de los recibos (clave de cobranza, nombre del asegurado, fecha de movimiento, y las claves que componen los catálogos asociados) y las tablas con los datos independientes entre si (catálogos asociados), como se muestra en la figura 5.1.5.3.









Una vez que seleccionamos la opción "Target Server" del menú, aparece la pantalla de la figura 5.2.3, en donde, para este caso, seleccionaremos SQL Server y la versión 7, que es la opción seleccionada para el desarrollo que se va a realizar.

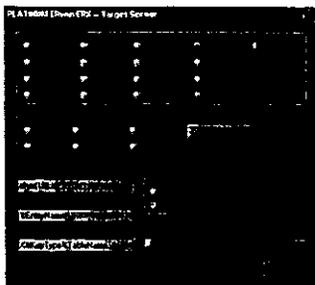


Figura 5.2.3 Pantalla para seleccionar la base de datos.

Se tomo la decisión de tener un estándar para la definición de los nombres que se utilizarán en la creación de la base de datos, este proceso se define a continuación:

Objeto	Observaciones
Tablas	El nombre de la tabla comenzará con una letra que indica: <ul style="list-style-type: none"><li>• c, si la tabla es un catálogo.</li><li>• t, si la tabla es una entidad de registros con múltiples catálogos.</li><li>• h, si la tabla es un almacén de registros históricos.</li></ul>
Campos	El nombre consta de dos partes: <ul style="list-style-type: none"><li>• La primera muestra el nombre o descripción del campo.</li><li>• La segunda refleja la tabla de origen, en caso de ser una llave foránea indica de donde proviene este campo y en caso de ser un campo propio de la tabla nos indica que pertenece a esta.</li></ul>
Tablas y campos	Para definir el nombre se utilizarán mnemónicos, eliminando todas las vocales intermedias de una palabra, manteniendo únicamente la primera en caso de iniciar con esta y la última vocal sin importar la posición de esta.

En ERWin contamos con una barra de herramientas con los objetos que se pueden incluir en la base de datos que se puede crear, estos objetos se muestran en la figura 5.2.4, y se detallan posteriormente.



Figura 5.2.4 Barra de herramientas de ERWin.



Objeto	Descripción
	Este botón nos ayuda a indicar los objetos que se han insertado dentro del modelo creado.
	Este botón nos sirve para insertar tablas independientes, tablas que sus campos no dependen de ninguna otra tabla, únicamente contienen su llave primaria y los campos propios de la tabla.
	Este botón, al contrario del anterior, nos sirve para insertar tablas dependientes, tablas que al menos uno de sus campos dependen de otra tabla, ósea que contienen su llave primaria, los campos propios de la tabla y una llave foránea.
	Este botón nos permite insertar un cuadro de texto en el modelo que se esta desarrollando, ya sea una nota o alguna observación
	Este botón nos permite manipular los atributos o campos de una tabla, ya sea entre posiciones dentro de la misma tabla, hasta un cambio de este atributo a otra tabla.
	Este botón nos permite crear las relaciones con una dependencia obligatoria entre la tabla padre y la tabla hija, por lo que la llave primaria de la tabla padre es parte importante de la llave primaria de la tabla hija.
	Este botón nos ayuda a crear vistas, que se extraen de una o más tablas hacia esta entidad.
	Este botón nos permite crear las relaciones con una dependencia independiente entre la tabla hija y la tabla padre, por lo que la llave primaria de la tabla padre es parte los atributos de la tabla hija y no forma parte de su llave primaria.

Ahora procederemos a describir el proceso de creación de todos y cada uno de los objetos de nuestra base de datos, los elementos creados son: tablas, atributos y relaciones.

En primera instancia crearemos todos los catálogos, como por ejemplo, el tipo de recibo, la forma de pago y la moneda, estas tablas contienen solamente una clave única que los identifica y una descripción que es el nombre del dato catalogado, las claves son de tipo numérico y las descripciones son de máximo treinta posiciones, por lo que seleccionamos de la barra de herramientas el botón de tablas independientes y creamos cada una de las tablas de la siguiente manera. Colocamos una tabla en el área de diseño y alimentamos el nombre de esta y de los campos que la formaran, esto para cada una de las tablas como se muestra en la figura 5.2.5.

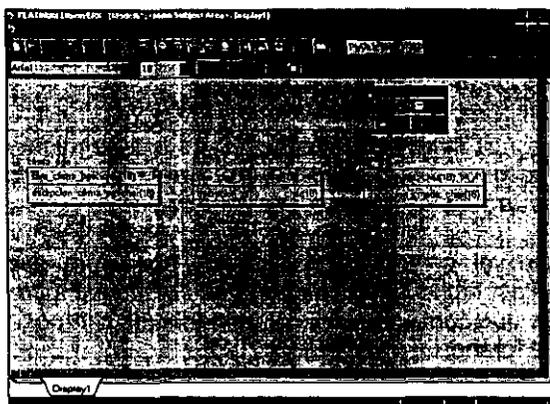


Figura 5.2.5 Dibujo de tablas y atributos.

Como podemos ver en la figura 5.2.5 el tipo de dato que les coloca a los campos es un carácter de longitud 18, lo cual debemos cambiar de acuerdo a las especificaciones de

la tabla que se esta creando, por lo que para cambiar el tipo de dato de las tablas seleccionamos de la barra de herramientas el primer botón ("flecha"), el cual nos ayuda a seleccionar el objeto que deseamos modificar y dando un doble click con el botón izquierdo del mouse sobre este objeto nos aparece la pantalla de la figura 5.2.6.

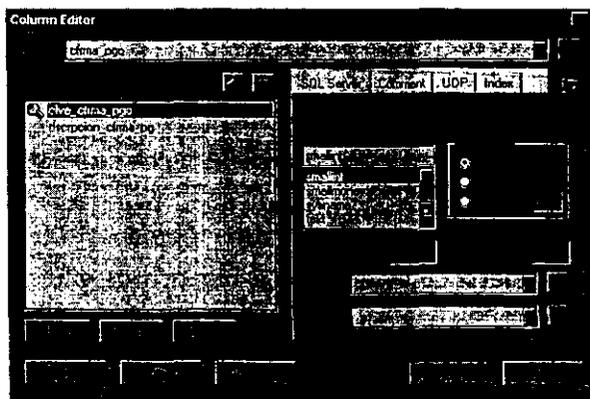


Figura 5.2.6 Pantalla de asignación de información a los atributos.

Esta pantalla nos permite al seleccionar la pestaña con el nombre "SQL Server" el tipo de dato del atributo seleccionado en el lado izquierdo, por lo que para el ejemplo, la clave la seleccionamos como tipo entero corto "smallint", y a la descripción del registro el tipo es texto de 30 posiciones "char(30)", y posteriormente, seleccionamos la pestaña con el nombre "index", con el fin de poder marcar la clave como llave primaria, ya que



este es nuestro indicador único de la tabla, como se muestra en la figura 5.2.7., una vez hecho todo, esto vemos ya nuestras tablas con un tipo de dato asignado, las llaves primarias marcadas, y una línea en la vista de la tabla, dividiendo la llave primaria del resto de los datos, como se muestra en la figura 5.2.8 con nuestras entidades terminadas.

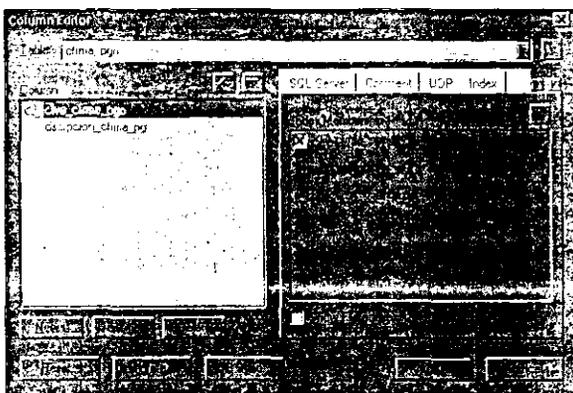


Figura 5.2.7 Pantalla de asignación de índices.

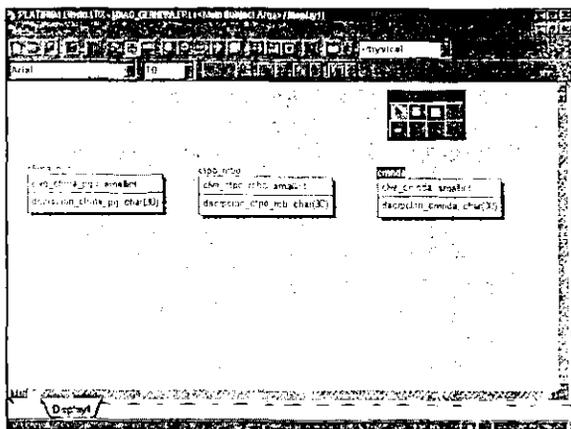


Figura 5.2.8 Tablas terminadas.

Ahora explicaremos la creación de la tabla de recibos amparados, tabla dependiente, y sus relaciones. Seleccionamos de la barra de herramientas el botón de tablas dependientes y como con las tablas dependientes colocamos una tabla en el área de diseño y alimentamos el nombre de esta y de los campos que la formaran como se muestra en la figura 5.2.9., y al igual que las tablas independientes, los tipos de dato y la definición de la llave primaria se define en la pantalla de las figuras 5.2.6 y 5.2.7.

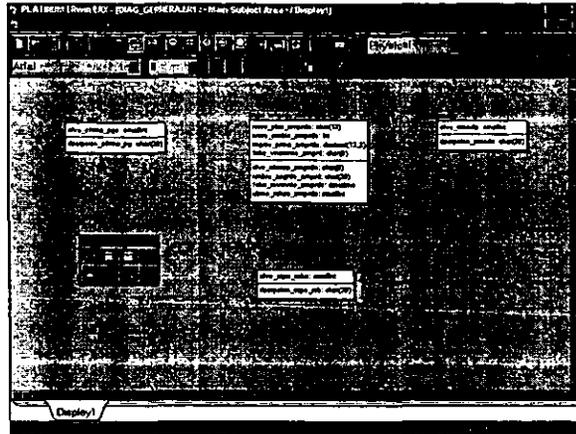


Figura 5.2.9 Tabla dependiente creada.

A continuación se integraran al modelo las relaciones entre los catálogos y la tabla de los recibos, como ninguna de las claves de los tres catálogos del ejemplo seleccionado son parte de la llave primaria de los recibos, seleccionamos de la barra de herramientas el botón de relaciones independientes, y procedemos a marcar la relación en el diseño, de la tabla padre a la tabla hija, y automáticamente se mostraran en la tabla hija (recibos) los atributos heredados de la tabla padre (catálogos). El modelo con los catálogos de ejemplo y la tabla de los recibos queda como se muestra en la figura 5.2.10.

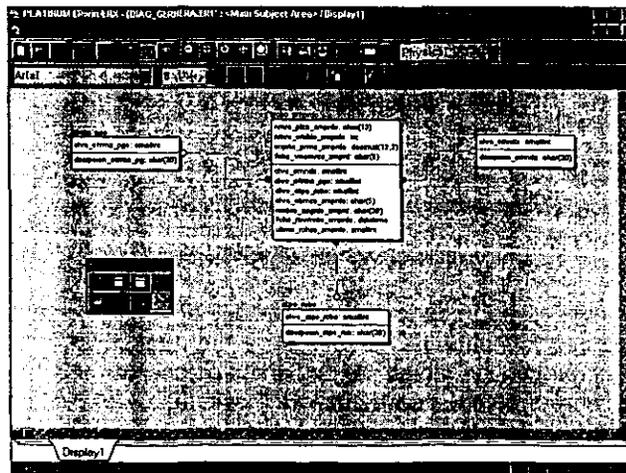


Figura 5.2.10 Tablas relacionadas.





```
grant all to dbo;

create table tintfces
(
  nmbre_tintfces char(8) not null,
  vrsion_tintfces smallint not null,
  primary key (nmbre_tintfces)
);
revoke all on tintfces from public;

create table csstma
(
  clve_csstma smallint not null,
  dscrpcion_csstma char(30) not null,
  rspnsble_csstma char(30) not null,
  primary key (clve_csstma)
);
revoke all on csstma from public;

create table cerror
(
  clve_cerror smallint not null,
  dscrpcion_cerror char(30) not null,
  primary key (clve_cerror)
);
revoke all on cerror from public;

create table ctpo_ofcna
(
  clve_ctpo_ofcna smallint not null ,
  dscrpcion_ctpo_ofcna char(30) not null,
  primary key (clve_ctpo_ofcna)
);
revoke all on ctpo_ofcna from public;

create table cofcna
(
  clve_cofcna integer not null,
  dscrpcion_cofcna char(30) not null ,
  clve_ctpo_ofcna smallint not null ,
  nmro_ofcna_sprio integer not null,
  rspnsble_cofcna char(30) not null,
  vrtual_cofcna smallint not null,
  primary key (clve_cofcna)
);
revoke all on cofcna from public;

create table cusrio
(
  clve_cusrio char(8) not null,
  pssword_cusrio char(8) not null,
  dscrpcion_cusrio char(40) not null,
  clve_cofcna integer not null,
  primary key (clve_cusrio)
);
revoke all on cusrio from public;

create table tprmso_sstma
(
  clve_cusrio char(8) not null ,
  clve_csstma smallint not null,
  fcha_tprmso_sstma datetime not null,
  primary key (clve_cusrio,clve_csstma)
);
revoke all on tprmso_sstma from public;

create table ctpo_intrmdrio
(
  clve_ctpo_intrmdri smallint not null ,
  dscrpcion_ctpo_int char(30) not null ,
  primary key (clve_ctpo_intrmdri)
);
revoke all on ctpo_intrmdrio from public;

create table cintmrdio
(
  clve_cintmrdio integer not null,
  clve_ctpo_intrmdri smallint not null ,
  clve_alfnmrco_cint char(5),
  clve_nmrcoc_cintrmd integer,
  clve_alfnmrco_nvo char(5),
  dscrpcion_cintrmdr char(30),
  clve_cofcna integer not null ,
  nmro_prmtria_admn integer,
  tpo_intrmdrio_spri smallint not null ,
  nmro_prmtria_vnta integer,
  primary key (clve_cintmrdio,clve_ctpo_intrmdri)
);
revoke all on cintmrdio from public;

create index i_alfnmrco on cintmrdio (clve_alfnmrco_cint,
  clve_ctpo_intrmdri);
create index i_nmrcoc on cintmrdio
(clve_nmrcoc_cintrmd,clve_ctpo_intrmdri);

create index i_alfa_nvo on cintmrdio (clve_alfnmrco_nvo,
  clve_ctpo_intrmdri);

create table cestdo_rcbo
(
  clve_cestdo_rcbo smallint not null ,
  dscrpcion_cestdo_rcbo char(30) not null,
  nvel_cestdo_rcbo smallint not null ,
  primary key (clve_cestdo_rcbo)
);
revoke all on cestdo_rcbo from public;

create table cuen
(
  clve_cuen smallint not null ,
  dscrpcion_cuen char(30) not null ,
  rspnsble_cuen char(30) not null ,
  primary key (clve_cuen)
);
revoke all on cuen from public;

create table crmo
(
  clve_crmo char(2) not null ,
  dscrpcion_crmo char(30) not null ,
  rmo_vda_crmo smallint not null ,
  clve_cuen smallint not null ,
  primary key (clve_crmo)
);
revoke all on crmo from public;

create table carchvo
(
  clve_carchvo smallint not null ,
  dscrpcion_carchvo char(30) not null ,
  primary key (clve_carchvo)
);
revoke all on carchvo from public;

create table cfrma_pgo
(
  clve_cfrma_pgo smallint not null ,
  dscrpcion_cfrma_pg char(30) not null ,
  primary key (clve_cfrma_pgo)
```



## DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
);  
revoke all on cfrma_pgo from public;
```

```
create table ctpo_rcbo  
(  
  dve_ctpo_rcbo smallint not null ,  
  dscrpcion_ctpo_rcb char(30) not null ,  
  primary key (dve_ctpo_rcbo)  
);
```

```
revoke all on ctpo_rcbo from public;
```

```
create table cmnda  
(  
  dve_cmnda smallint not null ,  
  dscrpcion_cmnda char(30) not null ,  
  primary key (dve_cmnda)  
);
```

```
revoke all on cmnda from public;
```

```
create table corgen  
(  
  dve_corgen char(1) not null ,  
  dscrpcion_corgen char(30) not null ,  
  primary key (dve_corgen)  
);
```

```
revoke all on corgen from public;
```

```
create table trcbo_amprdo  
(  
  nmro_plza_amprdo char(13) not null ,  
  nmro_cmbio_amprdo integer not null ,  
  dve_ctpo_rcbo smallint not null ,  
  imprte_prma_amprdo decimal(12,2) not null ,  
  fcha_vncmnto_amprdo char(6) not null ,  
  dve_cintrmdrio integer not null ,  
  dve_ctpo_intrmdri smallint not null ,  
  dve_crmo char(2) not null ,  
  dve_cfrma_pgo smallint not null ,  
  dve_cmnda smallint not null ,  
  dve_cbrnza_amprdo char(5) not null ,  
  nmbre_asgrdo_amprdo char(30),  
  fcha_mvmtnto_amprdo datetime not null ,  
  dve_carchvo smallint not null ,  
  dve_corgen char(1) not null ,  
  dve_cestdo_rcbo smallint not null ,  
  ultimo_rchzo_amprdo smallint not null ,  
  primary key  
(nmro_plza_amprdo,nmro_cmbio_amprdo,dve_ctpo_rcbo,i  
mprte_prma_amprdo,fcha_vncmnto_amprdo)  
);
```

```
revoke all on trcbo_amprdo from public;
```

```
create table hrcbo_mvmtnto  
(  
  nmro_plza_amprdo char(13) not null ,  
  nmro_cmbio_amprdo integer not null ,  
  dve_ctpo_rcbo smallint not null ,  
  imprte_prma_amprdo decimal(12,2) not null ,  
  fcha_vncmnto_amprdo char(6) not null ,  
  nmro_rchzo_mvmtnto smallint not null ,  
  dve_cestdo_rcbo smallint not null ,  
  fcha_audtria_mvmtnto datetime not null ,  
  hra_audtria_mvmtnto char(5) not null ,  
  csa_rchzo_mvmtnto varchar(100) not null ,  
  dve_cusrio char(8) not null ,  
  primary key  
(nmro_plza_amprdo,nmro_cmbio_amprdo,dve_ctpo_rcbo,i  
mprte_prma_amprdo,fcha_vncmnto_amprdo,nmro_rchzo_mv  
mtnto)  
);
```

```
revoke all on hrcbo_mvmtnto from public;
```

```
create table trcbo_incsstnte  
(  
  nmro_plza_incsstn char(13) not null ,  
  nmro_cmbio_incsst integer not null ,  
  dve_ctpo_rcbo smallint not null ,  
  imprte_prma_incsst decimal(12,2) not null ,  
  fcha_vncmnto_incs char(6) not null ,  
  dve_cintrmdrio integer ,  
  dve_ctpo_intrmdri smallint ,  
  nmro_agnte_incsst char(5) not null ,  
  nmro_prrtor_incsst char(5) not null ,  
  dve_crmo char(2),  
  rcbo_rmo_vda_incs char(2) not null ,  
  dve_cfrma_pgo smallint not null ,  
  dve_cmnda smallint not null ,  
  dve_cbrnza_incsst char(5) not null ,  
  nmbre_asgrdo_incs char(30),  
  fcha_mvmtnto_incsst datetime not null ,  
  dve_carchvo smallint not null ,  
  dve_corgen char(1) not null ,  
  dve_ceror smallint not null ,  
  primary key  
(nmro_plza_incsstn,nmro_cmbio_incsst,dve_ctpo_rcbo,i  
mprte_prma_incsst,fcha_vncmnto_incsst)  
);
```

```
revoke all on trcbo_incsstnte from public;
```

```
create table trcbo_pndnte  
(  
  nmro_plza_pndnte char(13) not null ,  
  nmro_cmbio_pndnte integer not null ,  
  dve_ctpo_rcbo smallint not null ,  
  imprte_prma_pndnte decimal(12,2) not null ,  
  fcha_vncmnto_pndnt char(6) not null ,  
  dve_cintrmdrio integer not null ,  
  dve_ctpo_intrmdri smallint not null ,  
  dve_crmo char(2) not null ,  
  dve_cfrma_pgo smallint not null ,  
  dve_cmnda smallint not null ,  
  dve_cbrnza_pndnte char(5) not null ,  
  nmbre_asgrdo_pndnt char(30),  
  fcha_mvmtnto_pndnte datetime not null ,  
  dve_carchvo smallint not null ,  
  dve_cestdo_rcbo smallint not null ,  
  primary key  
(nmro_plza_pndnte,nmro_cmbio_pndnte,dve_ctpo_rcbo,imp  
rte_prma_pndnte,fcha_vncmnto_pndnte)  
);
```

```
revoke all on trcbo_pndnte from public;
```

```
create table targa_rcbos  
(  
  nmro_plza_crga char(13) not null ,  
  nmro_cmbio_crga integer not null ,  
  dve_ctpo_rcbo smallint not null ,  
  imprte_prma_crga decimal(12,2) not null ,  
  fcha_vncmnto_crga char(6) not null ,  
  dve_cintrmdrio integer ,  
  dve_ctpo_intrmdri smallint ,  
  nmro_agnte_crga char(5) not null ,  
  nmro_prrtor_crga char(5) not null ,  
  dve_crmo char(2),  
  rcbo_rmo_vda_crga char(2) not null ,  
  dve_cfrma_pgo smallint not null ,  
  dve_cmnda smallint not null ,  
  dve_cbrnza_crga char(5) not null ,  
  nmbre_asgrdo_crga char(30),  
  fcha_mvmtnto_crga datetime not null ,
```



```
clve_carchvo smallint not null ,
clve_corgen char(1) not null ,
clve_cerror smallint not null,
primary key
(nmro_plza_crga,nmro_cmbio_crga,clve_ctpo_rcbo,imprte_p
ma_crga,fcha_vncmnto_crga)
);
revoke all on tcrga_rcbos from public;
create index i_agentes on tcrga_rcbos (nmro_agnte_crga);
```

```
create table tcrga_intrmdrio
(
clve_cintrmdrio integer not null ,
clve_ctpo_intrmdri smallint not null ,
clve_alfnmrco_cint char(5),
clve_nmrcoc_cintrmd integer,
clve_alfnmrco_nvo char(5),
dscrpcion_cintrmdr char(30),
clve_cofcna integer not null ,
nmro_prmtria_admsn integer,
tpo_intrmdrio_spr smallint not null ,
nmro_prmtria_vnta integer,
primary key (clve_cintrmdrio,clve_ctpo_intrmdri)
);
revoke all on tcrga_intrmdrio from public;
```

```
alter table cintmrdio add constraint fk1_cintrmdrio foreign
key (nmro_prmtria_admsn,
tpo_intrmdrio_spr) references cintmrdio ;
```

```
alter table cofcna add constraint fk1_cofcna foreign key
(nmro_ofcna_spr)
references cofcna ;
```

```
alter table cofcna add constraint fk2_cofcna foreign key
(clve_ctpo_ofcna)
references ctpo_ofcna ;
alter table cusrio add constraint fk1_cusrio foreign key
(clve_cofcna)
references cofcna ;
```

```
alter table tprmsostma add constraint fk1_tprmsostma
foreign key (clve_cusrio)
references cusrio ;
```

```
alter table tprmsostma add constraint fk2_tprmsostma
foreign key (clve_csstma)
references csstma ;
```

```
alter table cintmrdio add constraint fk2_cintrmdrio foreign
key (clve_cofcna)
references cofcna ;
```

```
alter table cintmrdio add constraint fk3_cintrmdrio foreign
key (clve_ctpo_intrmdri)
references ctpo_intrmdrio ;
```

```
alter table crmo add constraint fk1_crmo foreign key
(clve_cuen) references
cuen ;
```

```
alter table trcbo_amprdo add constraint fk1_trcbo_amprdo
foreign key (clve_crmo)
references crmo ;
```

```
alter table trcbo_amprdo add constraint fk2_trcbo_amprdo
foreign key (clve_cmnda)
references cmnda ;
```

```
alter table trcbo_amprdo add constraint fk3_trcbo_amprdo
foreign key (clve_ctpo_rcbo)
references ctpo_rcbo ;
```

```
alter table trcbo_amprdo add constraint fk4_trcbo_amprdo
foreign key (clve_cfma_pgo)
references cfma_pgo ;
```

```
alter table trcbo_amprdo add constraint fk5_trcbo_amprdo
foreign key (clve_carchvo)
references carchvo ;
```

```
alter table trcbo_amprdo add constraint fk6_trcbo_amprdo
foreign key (clve_corgen)
references corgen ;
```

```
alter table trcbo_amprdo add constraint fk7_trcbo_amprdo
foreign key (clve_cintrmdrio,
clve_ctpo_intrmdri) references cintmrdio ;
```

```
alter table trcbo_amprdo add constraint fk8_trcbo_amprdo
foreign key (clve_cestido_rcbo)
references cestido_rcbo ;
```

```
alter table hrcbo_mvnmto add constraint fk1_hrcbo_mvnmto
foreign key (clve_cusrio)
references cusrio ;
```

```
alter table hrcbo_mvnmto add constraint fk2_hrcbo_mvnmto
foreign key (clve_cestido_rcbo)
references cestido_rcbo ;
```

```
alter table hrcbo_mvnmto add constraint fk3_hrcbo_mvnmto
foreign key (nmro_plza_amprdo,
nmro_cmbio_amprdo,clve_ctpo_rcbo,imprte_pma_crga,
cha_vncmnto_amprdo)
references trcbo_amprdo ;
```

```
alter table trcbo_incsstnte add constraint
fk1_trcbo_incsstnte foreign key (clve_cerror)
references cerror ;
```

```
alter table trcbo_incsstnte add constraint
fk2_trcbo_incsstnte foreign key (clve_carchvo)
references carchvo ;
```

```
alter table trcbo_incsstnte add constraint
fk3_trcbo_incsstnte foreign key (clve_cfma_pgo)
references cfma_pgo ;
```

```
alter table trcbo_incsstnte add constraint
fk4_trcbo_incsstnte foreign key (clve_cmnda)
references cmnda ;
```

```
alter table trcbo_incsstnte add constraint
fk5_trcbo_incsstnte foreign key (clve_corgen)
references corgen ;
```

```
alter table trcbo_incsstnte add constraint
fk6_trcbo_incsstnte foreign key (clve_ctpo_rcbo)
references ctpo_rcbo ;
```

```
alter table trcbo_pndnte add constraint fk1_trcbo_pndnte
foreign key (clve_carchvo)
references carchvo ;
```

```
alter table trcbo_pndnte add constraint fk2_trcbo_pndnte
foreign key (clve_cintrmdrio,
clve_ctpo_intrmdri) references cintmrdio ;
```



```
alter table trcbo_pndnte add constraint fk3_trcbo_pndnte
foreign key (clve_corgen)
references crmo ;

alter table trcbo_pndnte add constraint fk4_trcbo_pndnte
foreign key (clve_cmnda)
references cmnda ;

alter table trcbo_pndnte add constraint fk5_trcbo_pndnte
foreign key (clve_ctpo_rcho)
references ctpo_rcho ;

alter table trcbo_pndnte add constraint fk6_trcbo_pndnte
foreign key (clve_cfma_pgo)
references cfma_pgo ;

alter table trcbo_pndnte add constraint fk7_trcbo_pndnte
foreign key (clve_cesdto_rcho)
references cesdto_rcho ;

alter table tcrga_rchos add constraint fk1_tcrga_rchos
foreign key (clve_cerror)
references cerror ;

alter table tcrga_rchos add constraint fk2_tcrga_rchos
foreign key (clve_carchvo)
references carchvo ;

alter table tcrga_rchos add constraint fk3_tcrga_rchos
foreign key (clve_cfma_pgo)
references cfma_pgo ;

alter table tcrga_rchos add constraint fk4_tcrga_rchos
foreign key (clve_cmnda)
references cmnda ;

alter table tcrga_rchos add constraint fk5_tcrga_rchos
```

```
foreign key (clve_corgen)
references corgen ;

alter table tcrga_rchos add constraint fk6_tcrga_rchos
foreign key (clve_ctpo_rcho)
references ctpo_rcho ;

alter table trcbo_incsnsinte add constraint
fk7_trcbo_incsnsinte foreign key (clve_corno)
references crmo ;

grant select, update, insert, delete on tntrfces to dbo;
grant select, update, insert, delete on csstma to dbo;
grant select, update, insert, delete on cerror to dbo;
grant select, update, insert, delete on ctpo_ofcna to dbo;
grant select, update, insert, delete on cofcna to dbo;
grant select, update, insert, delete on corno to dbo;
grant select, update, insert, delete on trmso_sstma to dbo;
grant select, update, insert, delete on ctpo_intrmdrio to dbo;
grant select, update, insert, delete on cintrmdrio to dbo;
grant select, update, insert, delete on cesdto_rcho to dbo;
grant select, update, insert, delete on cnen to dbo;
grant select, update, insert, delete on crmo to dbo;
grant select, update, insert, delete on carchvo to dbo;
grant select, update, insert, delete on cfma_pgo to dbo;
grant select, update, insert, delete on ctpo_rcho to dbo;
grant select, update, insert, delete on cmnda to dbo;
grant select, update, insert, delete on corgen to dbo;
grant select, update, insert, delete on trcbo_ampdro to dbo;
grant select, update, insert, delete on hrcbo_mvmento to dbo;
grant select, update, insert, delete on trcbo_incsnsinte to dbo;
grant select, update, insert, delete on trcbo_pndnte to dbo;
grant select, update, insert, delete on tcrga_rchos to dbo;
grant select, update, insert, delete on tcrga_intrmdrio to dbo;
```

Para crear la base de datos desde SQL Server, es necesario activar el "SERVICE MANAGER" o motor de datos, abriendo la aplicación y mostrándose la pantalla del lado izquierdo de la figura 5.2.12 y presionando la tecla de Star/Continue, quedando la pantalla del lado derecho de la misma figura, solo si se activo.

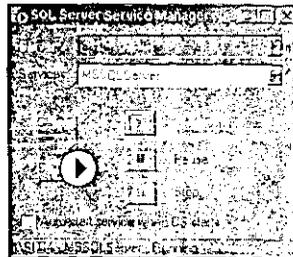
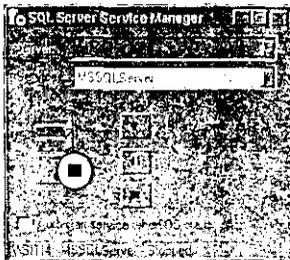


Figura 5.2.12 Pantallas activación del motor de datos.



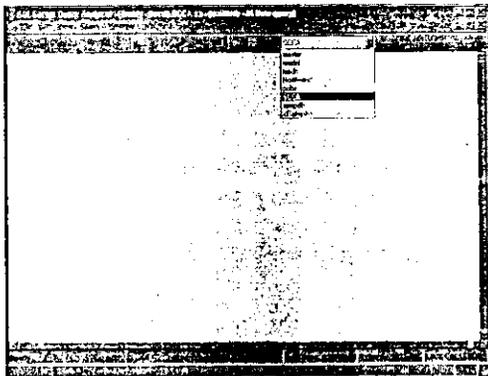


Figura 5.2.15 Pantalla del Query Analyzer donde se ejecutan las queries.

En las próximas páginas se encuentra el código de los procedimientos almacenados (store procedures o consultas precompiladas) que son funciones de código almacenadas a nivel base de datos, las cuales permiten hacer cambios en la estructura del negocio sin necesidad de modificar los clientes instalados en las PC's.

Este es un ejemplo de un procedimiento almacenado que se utilizará en este sistema.

```
IF EXISTS (SELECT name FROM sysobjects WHERE name='sp_usuario' AND type='P')
DROP PROCEDURE sp_usuario
GO
```

```
CREATE PROCEDURE sp_usuario
    @usua CHAR(8),
    @sist SMALLINT,
    @@cve_usrio CHAR(8) output,
    @@pas_usrio CHAR(8) output,
    @@des_usrio CHAR(40) output,
    @@cve_ofcna INTEGER output,
    @@des_ofcna CHAR(30) output,
    @@niv_ofcna SMALLINT output,
    @@ofi_sprior_ofcna INTEGER output AS
select u.clve_cusrio      = @cve_usrio ,
    u.pssword_cusrio    = @pas_usrio,
    u.dscrpcion_cusrio  = @des_usrio,
    u.clve_cofcna       = @cve_ofcna,
    o.dscrpcion_cofcna  = @des_ofcna,
    o.clve_ctpo_ofcna   = @niv_ofcna,
    o.nmro_ofcna_sprior = @ofi_sprior_ofcna
from   cusrio u, cofcna o, tprms_o_sstma p
where  o.clve_cofcna = u.clve_cofcna
and    u.clve_cusrio = p.clve_cusrio
and    u.clve_cusrio = @usua
and    p.clve_csstma = @sist;
```

El resto de los procedimientos almacenados se encuentra en el Apéndice C de este documento.



### 5.3. DISEÑO Y CONSTRUCCIÓN DEL FRONT END

El diseño del front end a utilizar tiene que estar muy ligado a los requerimientos del usuario, ya que en base a estos requerimientos el usuario podrá manipular la información a su conveniencia.

Para darle una mayor funcionalidad al sistema se creó un front end totalmente visual, intentando utilizar un lenguaje corto y conciso en cada uno de los botones de acción o comandos y de fácil operación para que los usuarios puedan adaptarse a él con mucha rapidez.

La interfaz se creó a base de formularios, los cuales se fueron manipulando para que el usuario agilice la entrada de información, además se pueden usar para agregar ventanas y cuadros de diálogo a la aplicación.

#### 5.3.1. Proyecto.

Un proyecto en Visual Basic, contiene todos los componentes de un sistema, como son barras de menús, cajas de herramientas, barras de herramientas, formularios, ventana de propiedades, etc.

La creación de un proyecto en Visual Basic consta de dos tipos de objetos, los cuales son:

- Formularios: Los formularios son ventanas sobre las cuales se dibujan los elementos que el usuario utiliza para comunicarse con la aplicación.
- Controles: Siempre se colocan en objetos contenedores, que tienen componentes visuales como los formularios (aunque a diferencia de los formularios, no pueden existir sin algún tipo de contenedor).

**Para la creación del sistema se utilizaron 19 formularios, y un módulo general**

El módulo general del sistema es el encargado de ejecutar las rutinas generales del programa. El procedimiento para la creación de la interfaz gráfica fue:

Seleccionar de la barra de menú "Archivo" y ordenar ejecutarse "Nuevo Proyecto.

Éste nos despegará una ventana, con la cual se tiene que elegir que tipo de proyecto se desea utilizar, como se ve en la figura 5.3.1.1.

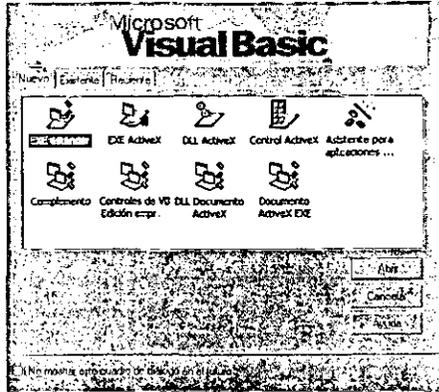


Figura 5.3.1.1. Ventana de nuevo proyecto.

Se elige la opción EXE estándar, y se desplegará una ventana de un formulario, en la cual se colocarán todos los controles, también se desplegará la ventana de proyectos, la cual contendrá todos los ficheros, como son formularios, clases y módulos, los cuales forman parte del proyecto. En esta ventana se pueden visualizar todos los formularios de ese proyecto, figura 5.3.1.2.

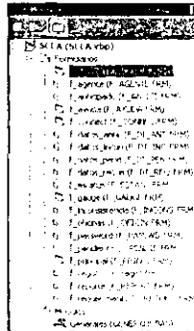


Figura 5.3.1.2. Ventana de proyectos con formularios y módulos.

Si se selecciona el menú "código", se observará el código usado en este caso para un formulario, también puede verse éste código para un módulo o clase.

Sobre el formulario se fueron colocando los controles necesarios para crear la interfaz gráfica, como son etiquetas, cajas de texto, marcos, casillas de verificación, botones de opción, botones de pulsación, etc., con la finalidad de ejecutar o visualizar los datos.

Una vez seleccionados y colocados los objetos necesarios para la creación de los formularios, se configuraron las propiedades de cada uno de ellos en la ventana de "propiedades" (cada objeto lleva asociado un conjunto de propiedades). En esta ventana se configuran las características específicas de los objetos, como son, tamaño del objeto, colores, tipo de letra, nombres, posiciones, etc.



Al finalizar la definición de las propiedades de los contrales, se escribió el código, que se uniría a un objeto, denominado procedimiento, conducido por un suceso o simplemente procedimiento.

A continuación se muestran las pantallas que conforman el sistema SCCA, totalmente terminadas.

La figura 5.3.1.3. muestra la pantalla de inicio del sistema SCCA. Esta primer pantalla contiene un menú, de opciones:

- Conexión
- Consultas
- Reportes
- Catálogo
- Ayuda

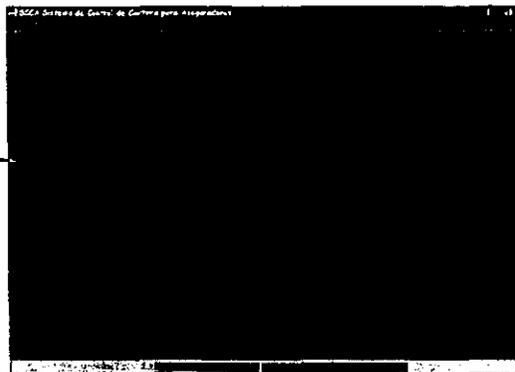


Figura 5.3.1.3. Pantalla de inicio.

### 5.3.2. Conexión.

En el menú "conexión" se selecciona el submenú "log in", con el cual accedera a otra ventana pidiendo el nombre de usuario y la clave (figura 5.3.1.4.).

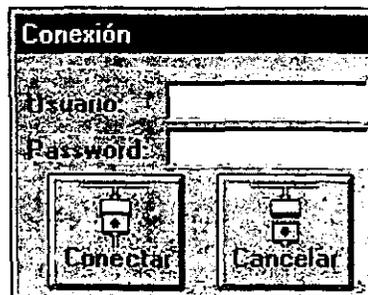


Figura 5.3.1.4. Ventana de conexión.



Esta pantalla está formada por los siguientes controles:

Cuadro de texto.

Botón de comando.

Etiquetas.

Si el nombre de usuario y la clave son correctos, podremos comenzar a ver o utilizar las consultas, reportes o catálogos del sistema. En el caso contrario, desplegará una ventana con un mensaje "EL PASSWORD NO COINCIDE CON SU REGISTRO", figura 5.3.2.1.

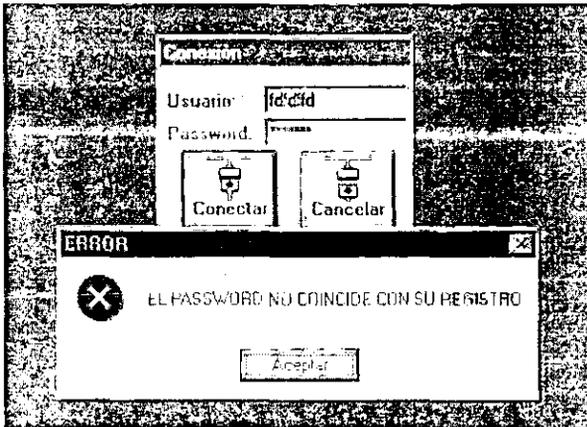


Figura 5.3.2.1 Mensaje que despliega en caso de error en el password

### 5.3.3. Consultas.

Si se selecciona el menú "consultas", del programa principal, mostrará cuatro submenús, en los cuales se selecciona que tipo de consulta se quiere realizar. Cada uno de estos submenús a su vez presenta una pantalla que al seleccionar el control deseado manda llamar y carga la pantalla de consulta para la respectiva selección:

- `f_antecipado.frm`: Consulta de pendientes entregados anticipadamente (figura 5.3.3.1.).



Figura 5.3.3.1. f\_anticipado.frm

- f\_inconsistencia.frm: Consulta de inconsistencias (figura 5.3.3.2).

Figura 5.3.3.2. f\_inconsistencia.frm



- f\_pendiente.frm: Consulta de pendientes de cobro (figura 5.3.3.3.).

Consulta de Pendientes de Cobro

Oficina: \_\_\_\_\_

División: \_\_\_\_\_

Subdivisión: \_\_\_\_\_

Región: \_\_\_\_\_

Intermediario: \_\_\_\_\_

Tipo: \_\_\_\_\_

Clave: \_\_\_\_\_ Promotor: \_\_\_\_\_

U.F.N.S. Ramos

Ambas  Todos

Personas  Uno

Empresas

Buscar

Salir

Figura 5.3.3.3. f\_pendiente.frm

- f\_requerido.frm: Consulta de requerimientos (figura 5.3.3.4.).

Consulta de Requerimientos

Oficina: \_\_\_\_\_

División: \_\_\_\_\_

Subdivisión: \_\_\_\_\_

Región: \_\_\_\_\_

Intermediario: \_\_\_\_\_

Tipo: \_\_\_\_\_

Clave: \_\_\_\_\_ Promotor: \_\_\_\_\_

U.F.N.S. Ramos

Ambas  Todos

Personas  Uno

Empresas

Buscar

Salir

Figura 5.3.3.4. f\_requerido.frm



#### 5.3.4. Reportes.

Seleccionando el menú "reportes", muestra tres submenús, se selecciona el tipo de reporte que se necesita. Dependiendo de la selección a elegir, nos enviará a una de las siguientes formas:

- `f_datos_antic.frm`: Son los recibos pendientes, entregados anticipadamente (figura 5.3.4.1.).



Figura 5.3.4.1. `f_datos_antic.frm`

- `f_datos_pend.frm`: Son los recibos pendientes de cobro (figura 5.3.4.2.).

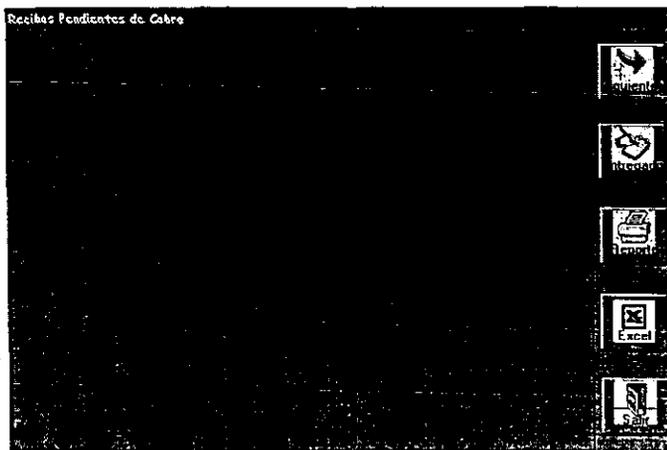


Figura 5.3.4.2. `f_datos_pend.frm`



- f\_datos\_reque.frm: Son los recibos de requerimientos (figura 5.3.4.3.).

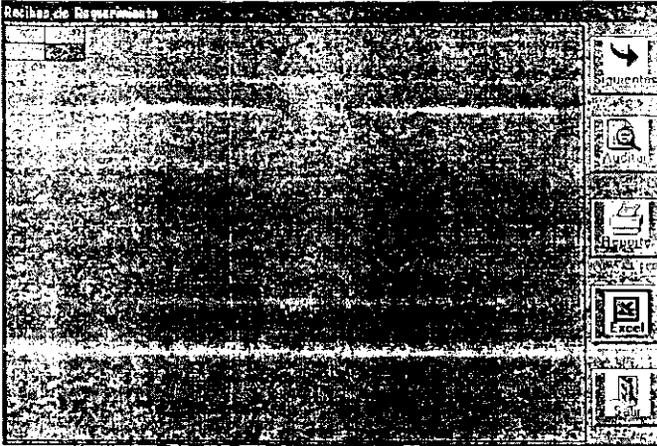


Figura 5.3.4.3. f\_datos\_reque.frm

Los reportes fueron creados con el generador de informes (Crystal Reports), utilizando la base de datos SCCA.

Para la elaboración de reportes con esta herramienta, se realizó el siguiente procedimiento:

- Se abrió el programa Crystal Reports desde la barra de programas de Windows.
- De la barra de menú se selecciona "archivo" y posteriormente el submenú "nuevo".
- Al seleccionar "nuevo", aparecerá una ventana la cual tiene varias opciones (figura 5.3.4.1.). Se selecciona la opción "informe" en la parte de la ventana que dice "Elija un tipo de informe y el tipo de dato", y en la parte derecha de la ventana se elige la opción "SQL/ODBC".

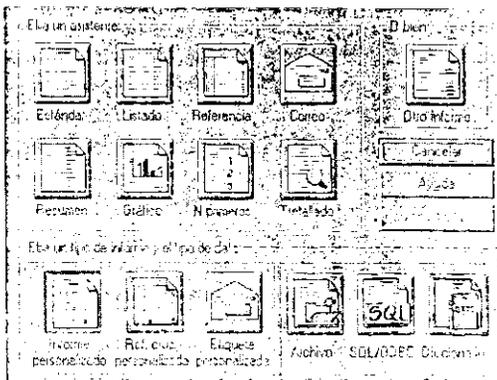


Figura 5.3.4.1. Ventana nuevo informe de Crystal Reports.



- Cuando se selecciona el tipo de informe que se desea generar y se da un click en aceptar, automáticamente aparecerá una nueva ventana donde pide proporcionar la ruta de donde va a tomar la base de datos a utilizar para generar los reportes.
- Una vez seleccionada la base de datos SCCA aparecerá una pequeña ventana donde pedira agregar los campos necesarios para generar los reportes.
- Al mismo tiempo se abrirá una ventana donde se van colocando los datos seleccionados en la ventana anterior (con un click sobre el campo que interesa insertar).

Una vez terminado de colocar todos los campos necesarios sobre el nuevo formato del reporte, solo resta escribir el nombre de cada uno de los registros colocados, ésto con la barra de menú seleccionando "insertar" y por último "etiqueta", con estos comandos se pueden escribir los nombres de los campos que se insertaron en el reporte.

### 5.3.5. Catálogo.

Si se selecciona el menú "catálogo", mostrará un submenú, el cual al elegirlo, nos enviará a la siguiente forma:

- f\_oficinas.frm: Catálogo de oficinas (figura 5.3.5.1.).

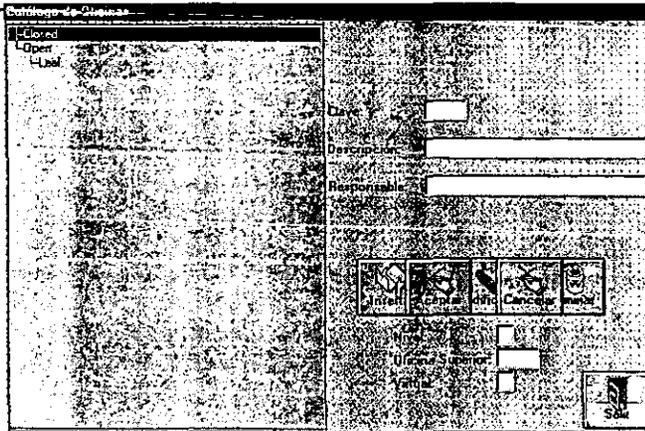


Figura 5.3.5.1. f\_oficinas.frm





### 5.3.8. Archivo ejecutable.

Una vez que se ha salvado la aplicación y se ha asignado un nombre, solo hace falta que esa aplicación sea un archivo ejecutable, es decir, la aplicación se podrá desplegar en cualquier otro ambiente.

Para crear este archivo hay que seleccionar la barra de menú "archivo" y elegir el submenú "hacer archivo EXE", el cual desplegará una ventana en donde se proporciona el nombre del archivo y la ruta donde se colocará.

En resumen, el front end es la presentación en Visual Basic de la información proporcionada en la base de datos de SQLServer.



## 5.4. DISEÑO E IMPLANTACIÓN DE RUTINAS DE DIAGNÓSTICO Y EVALUACIÓN DE LOS DATOS.

### 5.4.1. Introducción

En el ciclo de vida del desarrollo de sistemas la realización de las pruebas de software juegan un papel relevante, ya que la inversión de recursos económicos, humanos y de tiempo tienen un alto costo. La realización de dichas rutinas de diagnóstico no implica como finalidad principal la detección de inconsistencias por mal funcionamiento en el sistema, mas bien se fundamenta en el objetivo de entregar un producto de calidad, que minimice costos en tiempo, esfuerzo y dinero al conseguir reducir el mantenimiento del sistema al momento de ser implementado.

Si no se realiza eficientemente el análisis, diseño y la programación de pruebas, esta fase se puede extender en tiempo innecesario, así como si no se realiza un trabajo adecuado en esta fase se puede convertir en iterativo. Con las pruebas iniciales se detectan errores, las siguientes verifican si los programas corregidos funcionan correctamente.

Las interrogantes mas comunes al momento de implementar un sistema para ser probado por el usuario son: ¿El sistema hace lo que se pidió?, ¿Soportará la carga de trabajo planteada?, ¿Cuál es el porcentaje de falla del sistema?, etc. Es por todo lo anterior que se realiza el proceso de rutinas de diagnóstico y evaluación de los datos, las cuales disminuyen el riesgo de falla al operar el sistema.

La prueba de los programas es la técnica por la cuál se confirma el sistema, éstas se deben realizar antes de entregar el sistema al usuario. Las pruebas consisten en ejercitar el programa utilizando datos similares a los datos reales y observar los resultados interpretándolos para detectar errores o insuficiencias en el sistema. Cabe mencionar que las pruebas nunca demuestran que un programa esté 100% correcto, existe la probabilidad de fallas posteriores a la prueba mas compleja. Para obtener la presencia de errores, consiguiendo así su depuración es necesario elaborar un plan de pruebas que brinde dichos resultados. En dicho plan deberá indicar el nombre de la prueba, los datos de entrada, el objetivo de la prueba y los datos de salida recomendando se realicen estas pruebas con personal que no haya participado en la programación, pero que conozca la operación del negocio que utilizará el sistema.

#### 5.4.1.1. Flujo de información para las pruebas

En el esquema 5.4.1.1.1. podemos observar dos entradas, una es la configuración del software incluyendo especificación de requisitos, la especificación del diseño y el código fuente, y otra la configuración de prueba que incluye un plan y un procedimiento de pruebas evaluándose los resultados esperados contra los generados. Al descubrirse errores comienza el proceso de depuración, cuya finalidad es realizar las correcciones pertinentes.



Conforme se recompilan y se evalúan los resultados de la prueba se determina una medida cualitativa de calidad y confiabilidad del software, al encontrarse errores frecuentes que requieran modificaciones al diseño, entonces la calidad y fiabilidad del software está en entredicho, por lo tanto se continúan las pruebas, si por el contrario el funcionamiento del software parece ser el correcto detectándose errores en menor frecuencia, podemos acercarnos a una calidad y fiabilidad del software aceptables, o que las pruebas fueron inadecuadas, ya que no permitieron encontrar errores importantes. De la misma manera si en la prueba no se descubre ni un solo error queda la sospecha de que éstas no fueron las adecuadas y que el software puede estar defectuoso, para esto tendrá que descubrir dichos errores el usuario y deberán ser corregidos en la fase de mantenimiento o garantía.

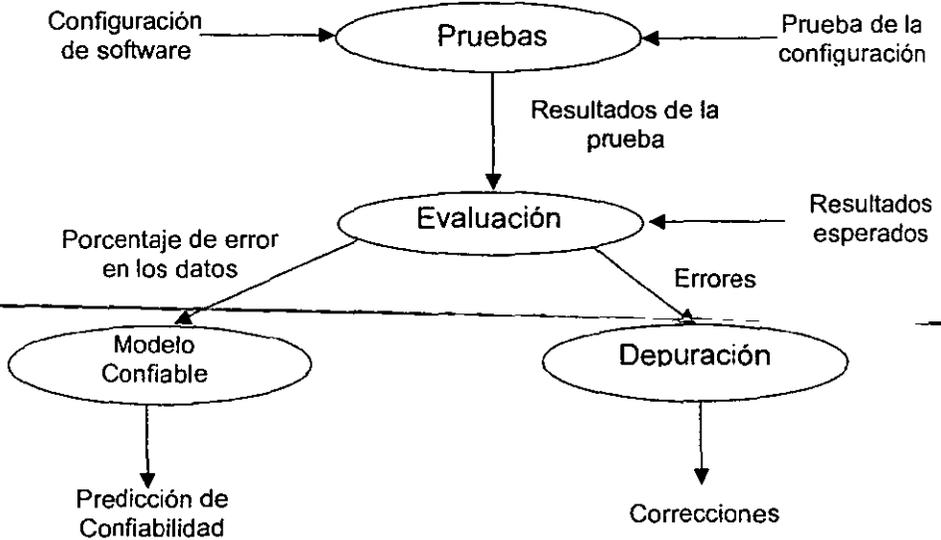


Figura 5.4.1.1.1. Flujo de información para las pruebas

## 5.4.2. Tipos de pruebas.

Analizaremos las pruebas más comunes que se han aplicado a este proyecto, las cuales englobaremos en cuatro distintos tipos:

- Prueba funcional.
- Prueba de recuperación.
- Prueba de desempeño.
- Prueba exhaustiva.

### 5.4.2.1. Prueba funcional.

Es con éste tipo de pruebas que se comenzó el análisis de los registros de cada cliente, y es con éste mismo tipo con los que el usuario validará el funcionamiento del sistema.



Se seleccionaron 5 oficinas regionales, se tomaron en cuenta todos los clientes de las 5 oficinas, se procedió a revisar los registros de los usuarios (agentes), como clave de nivel, clave de usuario, nombre, número de oficina, oficina y tipo. Lo que se revisó en esta parte de pruebas fue que los datos fueran los correctos, además de que se insertaron diferentes tipos de datos en los campos, (como numéricos, de texto y de fecha), para asegurarnos de que solo aceptara el tipo de dato correspondiente al campo elegido.

También se hizo esta prueba con los recibos pendientes de cobro, aquí se revisó la póliza, y el vencimiento de los recibos.

Además se revisó la pantalla de consulta de requerimientos, se analizaron los tipos de intermediarios, el tipo de usuario, y que ejecutara la consulta seleccionada, por ejemplo si se elige por división, solo tiene que mandarnos la consulta de entregados y revisados, y si se elige regional, entonces solo mandarnos la consulta de pendientes y rechazados.

Se enviaron estos datos a una hoja de Excel, se imprimieron estos datos y se compararon con los recibos con los que se cuenta en la empresa, presentándose algunas diferencias, las que fueron corregidas, dando como resultado que el sistema cumpliera con las especificaciones que dieron origen a su desarrollo, estas pruebas se mostrarán en el siguiente capítulo.

#### **5.4.2.2. Prueba de recuperación.**

En esta sección se realizaron pruebas tanto de la caída de energía eléctrica, como la caída del sistema de comunicación durante el proceso de registro de información comprobándose que éste tipo de fallas no afectan el proceso, es decir no hubo pérdida de información; cabe resaltar el apoyo de los sistemas “no break” (no ruptura de energía eléctrica) en las pérdidas de la planta de energía eléctrica.

#### **5.4.2.3. Prueba de desempeño.**

En esta etapa se buscó ejecutar la aplicación en situaciones extremas, que sería en una jornada normal de procesos diarios, obteniendo por resultado el que no se presentara degradación alguna del funcionamiento del sistema.

Esta prueba se llevó a cabo con 225,000 movimientos, tomando en cuenta que en un día normal se ejecutan alrededor de 150,000 movimientos.

#### **5.4.2.4. Prueba exhaustiva.**

En una segunda etapa se incrementó un 80% la carga de trabajo ordinaria, obligando al sistema a soportar una carga exhaustiva, obteniendo grandes resultados, ya que no hubo degradación mínima al sistema. Estas pruebas se presentan en el siguiente capítulo.



## 5.5. Integración y pruebas del Sistema

### Integración del Sistema.

Nos basamos en una integración ascendente para realizar la integración del sistema. Esta consiste en pruebas de unidad, seguidas por pruebas de subsistemas y luego por pruebas del sistema completo.

Las pruebas de unidad tienen como objetivo descubrir errores en los módulos individuales del sistema. Estos módulos se prueban aislados unos de otros en un ambiente artificial formado por los programas conductores y los datos necesarios para ejecutar los módulos. Las pruebas de unidad deben ser tan exhaustivas como sea posible, para garantizar que se ha probado cada caso representativo empleado en cada módulo. Dichas pruebas son fáciles si las estructuras se componen de módulos pequeños y débilmente acoplados.

En el caso de las pruebas a los subsistemas, su propósito es verificar la operación de las interfaces entre los módulos en el subsistema. Se deben probar tanto las interfaces de control como las de datos.

Las pruebas del sistema se relacionan con las interfaces, la lógica de decisión, el flujo de control, los procedimientos de recuperación, la eficiencia global, la capacidad y las características de integrado en su totalidad.

El objetivo a lograr es dar un seguimiento, orden y estabilidad al proceso de desarrollo de software, permitiéndonos obtener un producto de mejor calidad.

### Pruebas de Software

#### ¿Para qué probar el software?

- Dar a los programadores información para prevenir errores.
- Dar a los administradores información para evaluar el riesgo de utilizar un producto.
- Demostrar que el software sí funciona.
- Demostrar que el software NO funciona.
- Conseguir un producto libre de errores en lo posible.

Dar a los programadores información para prevenir errores.

- Los ingenieros de pruebas deben estar involucrados desde la fase de diseño.
- En estricta teoría, debe haber dos probadores por cada desarrollador.

Demostrar que el software sí funciona (*Pruebas limpias*).

- Se prueba para validar que el software cumpla con sus especificaciones funcionales y de interfaz.
- Se prueba para ver que el software sea acorde con su diseño.



Pero lo más importante:

Demostrar que el software NO funciona (*Pruebas sucias*).

- Todos tienden a probar que el software funciona.
- Los probadores deben intentar hacer que el sistema falle a toda costa.
- Por eso, donde sí los usan, los odian...
- Sin embargo, sólo así se puede garantizar la calidad de un software.
- El número de pruebas sucias debe superar por mucho al de las pruebas limpias.

Por ejemplo:

Si se tiene una variable "iEdad", que sólo debe trabajar con edades entre 18 y 60 años, se deben hacer las siguientes pruebas:

límite inferior - 1	iEdad = 17	Sucia
límite inferior	iEdad = 18	Limpia
promedio	iEdad = (18+60)/2	Limpia
límite superior	iEdad = 60	Limpia
límite superior + 1	iEdad = 61	Sucia
valor nulo	iEdad = Ø	Sucia
valor decimal	iEdad = 30.5	Sucia
valor alfanumérico	iEdad = "abc1@"	Realmente Sucia

### Comentario a las pruebas:

Los programadores casi siempre tienden a probar que la parte del sistema que desarrollan sí funciona, pues como ellos la han programado, se anticipan a su funcionamiento al probar, y suelen dar valores correctos intermedios. Esto deja descubiertas todas las posibilidades de error, que deben ser cubiertas por pruebas limpias y sucias:

**límite inferior - 1:** Esta prueba sucia es muy importante. Pudo ser que el programador haya escrito " $\geq 17$ " en lugar de " $> 17$ ", o que pensando escribir " $\geq 18$ ", haya escrito 17 en vez de 18: " $\geq 17$ ", y por ello, el programa acepte una edad igual a 17 años. El programa en este caso no debería calcular nada, y debería mostrar un mensaje indicando los parámetros correctos para que el usuario introduzca la edad adecuada. ¿Realmente lo hace? o más bien toma el 17 y lo procesa. Es por ello que esta prueba nunca debe omitirse.

**límite inferior y límite superior:** Estas pruebas limpias son también importantes, pues el programador pudo equivocarse y fijar un límite inferior mayor al correcto. Quizás escribió 19 ó 20 como edad mínima. Por ello, conviene siempre asegurarse que los límites fueron programados correctamente.

**promedio:** Con esta prueba limpia podemos asegurarnos que el programa funciona para un valor típico. Casi siempre, es ésta la única prueba que suelen hacer los



programadores, olvidando, o ni siquiera pensando, en todas las demás que aquí se mencionan.

**límite superior + 1:** La razón de aplicar esta prueba sucia es la misma que para probar límite inferior - 1.

**Siempre se debe probar un valor más allá de los límites.**

**valor nulo:** ¿Qué debe hacer el sistema si no se da valor alguno a esta variable? ¿Corre el riesgo de dividir entre cero? ¿Debe mostrar un mensaje de error al usuario y no dejarlo continuar? Con esta prueba se pretende encontrar un lugar donde falle el sistema.

**valor decimal:** Esta prueba sucia es muy importante. El valor esperado debe ser entero. Tal vez no suceda nada si se introduce un valor decimal, pero si después esta variable es reutilizada en el programa, quizás otra función no acepte su valor.

**valor alfanumérico:** Esta prueba sucia es una de las más importantes de todas, y una con la cual se encuentran muchos errores.

Sólo así se puede...

**Conseguir un producto libre de errores EN LO POSIBLE .**

**¡NO existen los programas libres de errores!**

Porque:

- No existen los programadores perfectos.
- Es imposible probar absolutamente todo.
- No se puede todo en esta vida.

El aseguramiento de calidad de software  
consiste en garantizar que un producto, **a pesar  
de sus errores**, funciona correctamente  
dentro de condiciones determinadas.

**¿Qué tan importante es un error?**

- Frecuencia
- Costo de Corrección
- Costo = Costo de descubrir + costo de corregir
- Costo por Instalación
- Depende del número de instalaciones que se hagan del sistema.
- Consecuencias
- Desde un error de ortografía hasta la explosión de un reactor nuclear.



DTS<sup>1</sup> indica la siguiente clasificación de prioridades, según sus consecuencias:

1. Mata personas.
2. Se pierden datos.
3. El sistema falla y se congela.
4. Errores irre recuperables.
5. Errores al interactuar con el entorno.
6. Errores de cosmética.

### Clasificación de errores por su ocurrencia

En la tabla 5.5.1 podemos ver una clasificación de errores de acuerdo con las estadísticas realizadas por los estudiosos de las pruebas de software. Con esta tabla, resulta sencillo identificar en qué orden buscar los errores que se esconden en un programa.

Familia de Errores	Porcentaje de ocurrencia
<b>Datos</b> Definición de datos, estructuras, declaración de variables, duplicación de variables, transformación de variables, flujo de datos	22.44%
<b>Funcionalidad</b> Funcionalidad mal entendida, faltante o duplicada.	16.19%
<b>Control de Flujo</b> Condiciones, ciclos, condiciones múltiples	12.82%
<b>Procesamiento</b> Expresiones aritméticas, evaluación de expresiones, signos (+/-), iniciación de variables, tiempo de procesamiento	12.36%
<b>Integración</b> Interfaces internas: invocación de componentes, parámetros equivocados, secuencias erróneas en parámetros, inicio de variables, estados incorrectos. Interfaces externas: interrupciones, dispositivos periféricos, manejadores ( <i>drivers</i> ), sincronización de entrada/salida, rendimiento.	8.98%
<b>Requerimientos</b> Requerimientos mal especificados, faltantes, ambiguos o inconsistentes.	8.12%
<b>Otros errores no especificados</b>	4.71%
<b>Definición de pruebas</b> Errores en el diseño de pruebas, requerimientos mal entendidos	2.76%
<b>Ejecución de pruebas</b> Base de datos iniciales mal definida, configuración, verificación	2.19%
<b>Codificación</b> Violación de estándares, documentación errónea, escasa documentación, documentación inconsistente, mal estilo	1.99%
<b>Arquitectura</b> Semáforos, manejo de recursos, multiprocesos.	1.74%
<b>Rendimiento</b> <b>Tiempo de respuesta, retrasos</b>	0.39%
<b>Casos de prueba completos</b>	0.39%
<b>Documentación de pruebas</b>	0.07%

Tabla 5.5.1 Familia de Errores



## Pruebas de Software

- Son procedimientos formales.
- Se deben preparar entradas y predecir salidas.
- Deben ser documentadas.
- Deben ejecutarse.
- Deben observarse los resultados.

## Niveles de Pruebas

- Pruebas de unidad.
- Pruebas de componentes.
- Pruebas de integración.
- Pruebas de sistema.

## Pruebas de Unidad

- La unidad es la parte más pequeña que puede ser probada en el software.
- Generalmente es escrita por un solo programador.
- Las pruebas de unidad buscan demostrar que la unidad no satisface sus especificaciones funcionales, o que su estructura no corresponde a su diseño.
- Al encontrar estos errores, se habla de errores de unidad.

## Pruebas de Componente

- El componente es la integración de varias unidades.
- Un componente puede ser una unidad, o incluso todo el sistema.
- Las pruebas de componente buscan demostrar que las unidades no satisfacen sus especificaciones funcionales al interactuar con otras unidades.
- Al encontrar estos errores, se habla de errores de componente.

## Pruebas de Integración

- La integración es el proceso en el que se juntan los componentes para generar componentes más grandes.
- Las pruebas de integración buscan demostrar que a pesar de que los componentes son satisfactorios individualmente, al combinarlos no funcionan o son inconsistentes.
- Ejemplos: Inconsistencias de validación de datos, parámetros equivocados, manejadores de funciones incorrectos.
- Al encontrar estos errores, se habla de errores de integración.



## Pruebas de Sistema

- El sistema es en realidad el componente más grande que se debe probar.
- Las pruebas de sistema buscan encontrar errores e inconsistencias que sólo pueden ser expuestos al integrar en su totalidad los componentes.
- Además las pruebas de sistema consisten en: pruebas de rendimiento, pruebas de seguridad, sensibilidad de la configuración, arranque del sistema y recuperación en los errores de sistema o de otros procesos que se ejecutan simultáneamente.

## Técnicas Para la Prueba de SW

- Grafos y rutas.
- Prueba de flujo de transacciones.
- Prueba de flujo de datos.
- Pruebas basadas en la lógica.
- Prueba de cajas negras.
- Pruebas automatizadas.

## Pruebas por Grafos y Rutas

- Piedra angular de las pruebas de software.
- Familia de técnicas que se basan en la selección de rutas para probar el flujo de control de los programas.
- Aplicable al SW moderno para pruebas de unidad
- En el flujo de control se encuentra la mayoría de los errores de un programa (25.18%)
- Requiere un conocimiento total de la estructura del programa.

## Pruebas de ciclos

La mayoría de los errores en el flujo de control se encuentran en los ciclos. Así, podemos afirmar que la mayoría de los errores de un sistema se deben a errores en la programación de los ciclos. Es por ello que conviene probarlos minuciosamente. Para ello, ya existen criterios que deben ser definitivamente tomados en cuenta, y nunca pasarlos por alto:

### Ciclos determinísticos

En los ciclos determinísticos, se conoce el número de veces que el ciclo será ejecutado, desde antes de entrar en su primera instrucción, y no existe forma alguna de interrumpir el ciclo y salir de él.



## Ciclos no determinísticos

Los ciclos no determinísticos difieren de los determinísticos en que el número de veces que el ciclo será ejecutado no es conocido antes de entrar en el ciclo. Puede suceder además que alguna condición dentro del ciclo lo interrumpa repentinamente y el programa salga de él y continúe con lo siguiente. Estos ciclos no determinísticos suelen contener muchos más errores que los ciclos determinísticos, por lo que conviene probarlos más minuciosamente.

## Ciclos anidados

Los ciclos anidados son problemáticos (es decir, suelen contener errores con facilidad). Incluyen por supuesto, los errores característicos de los ciclos sencillos (por ejemplo: terminar antes de tiempo, terminar después de tiempo, no terminar jamás, fallar en la primera condición y no ejecutarse), pero además contienen errores que surgen al ejecutar una y otra vez las condiciones de terminación en sendos ciclos.

## Ciclos no estructurados (o ciclos horribles)

Los ciclos horribles son aquellos en que se puede saltar a la mitad de ellos, sin pasar por el inicio del ciclo.

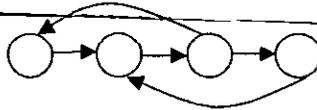


Figura 5.5.1. Ciclos no estructurados

Este ciclo resulta infame porque hay saltos espeluznantes hacia el interior del ciclo, y desde su interior hacia afuera. Actualmente no se suele programar así, pero muchos de los programas que los antiguos programadores nos legaron están contruidos de esta forma. No existe una buena técnica para probar los ciclos horribles (que peor nombre podrían tener). Sin embargo, este tipo de modelos se aplican al replicar el comportamiento humano al utilizar algunos sistemas. Por ejemplo, cuando se hace una llamada de larga distancia, de 11 dígitos, puede ser que al entrar en el ciclo de marcado, el usuario cuelgue antes de terminar de marcar si se da cuenta que marcó un dígito equivocado. Este tipo de ciclos requieren una mayor atención que los demás.

## Técnicas para probar ciclos

### Prueba de Valores Críticos

En un ciclo determinístico clásico, podemos considerar los siguientes números de prueba:

<b>valorInicial</b>	Valor inicial de la variable de control del ciclo <b>controlCiclo</b>
<b>valorSuperior</b>	Valor final de la variable de control del ciclo
<b>valorDeIncremento</b>	Cantidad en que la variable de control del ciclo debe ir aumentando



Las pruebas de valores críticos son la combinación de valores de estos tres números de prueba, que la experiencia muestra que son susceptibles a errores, además del caso típico o normal.

<b>Sobrepasar:</b>	Cualquier valor que provoque que se salga del ciclo inmediatamente.
<b>Una vez:</b>	Valores que hagan que el ciclo sea ejecutado exactamente una vez.
<b>Dos veces:</b>	Valores que hagan que el ciclo sea ejecutado exactamente dos veces.
<b>Típico:</b>	Un número típico de iteraciones.
<b>Máximo:</b>	El número máximo de iteraciones permitidas.
<b>Máximo + 1:</b>	Uno más que el máximo permitido.
<b>Máximo - 1:</b>	Uno menos que el máximo permitido.
<b>Mínimo:</b>	El mínimo requerido.
<b>Mínimo - 1:</b>	Uno menos que el mínimo requerido.
<b>Nulo:</b>	Comentado más adelante.
<b>Negativo:</b>	Comentado más adelante.

Muchas de las pruebas mencionadas pueden traslaparse, y esto ahorraría el número de pruebas que deben ejecutarse. Por ejemplo, si el valor mínimo es cero, entonces las siguientes identidades existen:

Mínimo - 1 = Negativo  
Mínimo = Sobrepasar  
Mínimo + 1 = Una vez

Similarmente, si el valor mínimo es 1:

Mínimo - 1 = Sobrepasar  
Mínimo = Una vez  
Mínimo + 1 = Dos veces

### Pruebas de Flujo de Datos

- Familia de estrategias de prueba basadas en la selección de rutas a través del programa para explorar secuencias de eventos relacionados con los estados de los datos.
- Se aplican para probar SW orientado a objetos, para realizar pruebas de integración y para probar hojas de cálculo.
- Detectan errores del tipo de variables no iniciadas, duplicación de definición de variables, tipos incorrectos, apuntadores erróneos y anomalías en el flujo de datos (v.gr. Cerrar antes de abrir un archivo).

### Pruebas de Flujo de Transacciones

- Una transacción es un conjunto de operaciones.
- Estas pruebas se utilizan para probar sistemas que funcionan en línea.



- Por ejemplo: ATMs, controladores de tráfico aéreo, reservación de vuelos.

### Pruebas Basadas en la Lógica

- Se utilizan para probar los requerimientos funcionales de un programa.
- Se basan en la construcción de tablas de decisión.
- Se emplean para analizar la consistencia de un sistema, y para verificar que esté completo.
- Aplicaciones: pruebas de lógica del hardware, pruebas de nuevos lenguajes, pruebas de sistemas expertos y de inteligencia artificial.

### Pruebas Automatizadas

Una de las cosas que resultan más tristes a la vista es ver que una persona hace a mano una tarea que podría ser automatizada. Es triste, pero al mismo tiempo causa risa. Nos encontramos en la quinta década de la industria de la computación y algunos de nosotros todavía nos vemos forzados a justificar el uso de computadoras (porque en el fondo, de eso se trata la automatización). Resulta divertido porque hace un siglo, la locomotora de vapor alcanzaba la cumbre. Cincuenta años después, carrozas viajaban de ciudad en ciudad jaladas por grupos de caballos que eran dirigidos por un chofer que los estabilizaba. Las pruebas manuales se parecen a ese chofer, pero sentado al frente de una locomotora: resulta peligroso, degradante, inútil, y además de todo, ridículo.

Si realmente se promoviera la dignidad humana, y la optimización de costos de trabajo se llevara a cabo como originalmente se pensó, los argumentos para promover las pruebas automatizadas caerían en oídos sordos. Esos no son los argumentos que usamos para justificar la inversión en automatización de pruebas, porque siempre habrá un ejecutivo "más inteligente" que nos los echará abajo. El argumento real es así de simple: **las pruebas de software hechas a mano, no sirven.** Punto. Nunca funcionaron bien en el pasado, no funcionan hoy, y no funcionarán en el futuro. Las pruebas manuales conducen a la autodecepción. Se confunde esfuerzo con logros. Y lo peor de todo, conducen a provocar una falsa confianza (aunque si alguno quiere provocarse una falsa confianza, existen fármacos baratos, legales e ilegales, más fáciles de ingerir que sentarse por meses a probar software a mano).

### Pruebas Dirigidas por el Riesgo

- Cómo encontrar los errores más importantes primero.
- Cómo puede ayudarnos a ello el concepto de Riesgo.
- Qué criterios seguir para asignar el Riesgo.

Cuando hay mucho por hacer y se cuenta con poco tiempo, es necesario priorizar de tal suerte que al menos lo más importante sea realizado.

En lo que se refiere a pruebas, nunca hay tiempo ni recursos suficientes. Además, las



consecuencias de omitir una prueba pueden ser severas. Por esta razón la priorización de pruebas ha merecido la atención de los estudiosos, que han propuesto el enfoque conocido como Pruebas Dirigidas por el Riesgo (*Risk Driven Testing*), donde el riesgo tiene un significado muy específico.

La premisa básica de este concepto de priorización consiste fundamentalmente en tomar las piezas del sistema que se han de probar –módulos, funciones, requerimientos– y evaluar cada pieza según dos variables: Impacto y Apariencia.

- El **impacto** consiste en aquello que pasaría si determinada pieza funcionara incorrectamente de cualquier manera. ¿Destruiría una base de datos? ¿o solamente implicaría que el encabezado de la columna de un reporte no se alineara correctamente?
- La **apariciencia** es un estimado de la probabilidad de que determinada pieza falle.

Juntos, la apariencia y el impacto determinan el Riesgo de la pieza:

$$\text{impacto} \cap \text{apariciencia} \Rightarrow \text{riesgo}$$

### Diseño de Casos de Prueba

- La importancia de concentrarse en una sola salida a la vez.
- Cómo encontrar las dependencias para dicha salida.
- Cómo verificar automáticamente que las dependencias encontradas son correctas.

Lo más importante al diseñar una prueba es seleccionar una sola salida para probar, y construir las pruebas para ella. La salida puede ser un dato, como una cantidad calculada, o puede ser una acción, como “detener el despliegue de información en un tablero electrónico”, o “guardar el archivo en un disco”.

Como habrán podido constatar quienes han realizado pruebas formales, resulta imposible por ejemplo hacer una prueba que verifique que todo un reporte sea generado correctamente.

Es muy importante entender que:

Un dato generado o una acción ejecutada por una computadora, sólo son correctos si **siempre** son correctos, bajo cualesquier circunstancias.

Existe una gran variedad de combinaciones posibles de valores de entrada para llegar a esta conclusión mediante una sola prueba para un reporte entero. Y aun para **una** sola salida, se necesitan varias pruebas para verificar que la salida es generada correctamente.



Para ejemplificar, necesitamos construir un "Generador de, que representa la pieza que estamos probando en el sistema, ya sea una función sometida a "pruebas de unidad", o incluso el sistema entero:



Figura 5.5.2 Generador de Salidas

Después, asignamos entradas posibles y salidas esperadas:

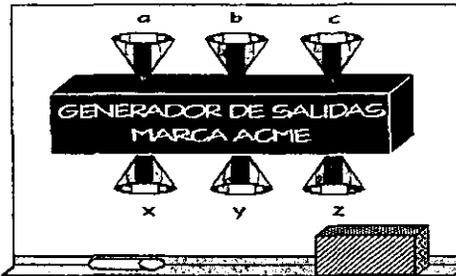


Figura 5.5.3.

En la figura anterior, podemos ver que el Generador de Salidas recibe las entradas a, b y c y genera las salidas x, y y z. Las salidas bien pueden ser datos o acciones, pero por ahora asumiremos que se trata de datos.

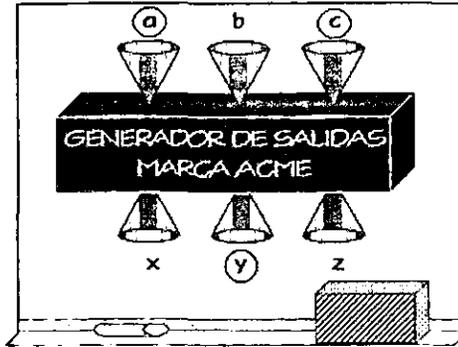
Supongamos que la salida y es la cantidad a pagar en una nómina, y que es más crítica que las salidas x o z. Así, concentramos las pruebas en z en primer lugar:

Viene ahora el paso crítico: apuntalar las dependencias. Para la salida que hemos escogido, necesitamos saber de cuáles entradas depende. Supongamos que la salida y depende de las entradas a y c, pero no depende de la entrada b. Si sabemos esto, podemos ver cómo construir los casos de prueba para la salida y.

Necesitamos entonces variar las entradas a y c a través de sus correctas combinaciones, pero no tenemos que preocuparnos por generar casos de prueba para



los diferentes valores de  $b$ .



Surgen las siguientes preguntas, muy importantes, acerca de las dependencias.

- ¿Qué tan buenas son?
- ¿Dónde encontramos la información acerca de las dependencias?
- ¿Cómo sabemos si las dependencias son correctas?

Analicemos cada pregunta a la vez:

### 1. ¿Qué tan buena es conocer las dependencias?

Primero, es importante conocer las dependencias de las salidas en las entradas, porque entonces podemos armar pruebas que **varíen sólo las entradas necesarias, y hacerlo de forma sistemática**. Esto significa que no tendremos casos de prueba extras, ni redundantes, y nos aseguraremos de que no

Cabe hacer notar que no es necesario entender cómo una salida depende de sus entradas. Sólo es necesario saber qué entradas son necesarias. Esto es mucho más fácil de determinar. De hecho, analizando exactamente cómo las salidas dependen de las entradas (en un sentido totalmente booleano, sí o no) conduce a un embrollo impráctico conocido como "Graficación de Causa y Efecto", en el cuál no penetraré en este documento, por considerarlo fuera de mi propósito y de mi alcance.

### 2. ¿Dónde encontramos la información acerca de las dependencias?

La mejor fuente de información al respecto de las dependencias son los usuarios. Basta con preguntarles ¿qué valores de entrada podrían afectar el valor de esta salida? Ellos sabrán la respuesta. El único problema aquí es que generalmente saben tanto que querrán decir incluso cómo dependen. Realmente, no es necesario entrar en tantos detalles.

Si somos lo suficientemente afortunados de contar con las especificaciones, también éstas contendrán información sobre la dependencia, pero usualmente resulta difícil



escarvar en las especificaciones para extraer esta información.

Finalmente, un lugar de donde no deseamos extraer esta información, es el código fuente. Es esto lo que estamos probando a fin de cuentas. No podemos basarnos en él con absoluta confianza.

Para cada salida que se pruebe, es necesario escribir las entradas de las cuales depende, y guardar esta información en un archivo o en una pequeña base de datos. Esto resulta clave para poder después construir casos de prueba de forma sistematizada.

### 3. ¿Cómo podemos saber si las dependencias son correctas?

Después de todo, si no son correctas, nuestros casos de prueba pueden no servir para nada. En los viejos sistemas podemos encontrar cantidades importantes de salidas que nos harían estremecer a la hora de diseñarles casos de prueba.

El truco consiste en lo siguiente:

Para verificar las dependencias de una salida en un conjunto de entradas debemos tener un breve programa o script (con una herramienta de automatización de pruebas se puede hacer fácilmente) que haga lo siguiente:

- Ejecutar el código y guardar los resultados.
- Hacer pruebas que varíen las entradas "que no importan" aleatoriamente dentro de sus límites válidos.
- Hacer que la computadora compare los resultados de las pruebas con los de las pruebas anteriores. Si estas entradas realmente no importan, se debe tener siempre el mismo resultado.
- De lo contrario, se trata de una dependencia que no se consideró. Hay que registrarla.
- Se puede dejar esta prueba automática ejecutándose digamos, mientras se sale a almorzar. Este puede ser un tiempo razonable para validar que las dependencias son correctas. Si se desea más precisión, se debe dejar ejecutando la prueba automática durante más tiempo.

### Pruebas aplicadas al sistema.

Se probó que cada uno de los menús de opciones y de las pantallas de captura cumplieran con los siguientes puntos:

**Ortografía correcta:** Todos los mensajes y pantallas deben estar escritos correctamente.

**Nombre del menú:** Cuando se selecciona un menú se mantiene el nombre de éste en la parte superior izquierda de la pantalla, con el fin de saber en todo momento la



opción en la que nos encontramos.

**Opciones del menú:** Se muestran todas las opciones que comprenden al menú seleccionado.



Figura 5.5.3 Menú de Consulta

**Descripción de la opción seleccionada:** Cuando el cursor está posicionado en un botón deberá aparecer una descripción en la barra de mensajes que nos indica la función de cada opción.

**Tecla entrar:** Durante la selección de una opción en los menús se podrá hacer uso de esta tecla para activar un botón de aceptación, a un botón de cancelación o una selección.

**Teclas de la primera letra:** Se podrá seleccionar una opción mediante la primera letra del nombre de la opción.

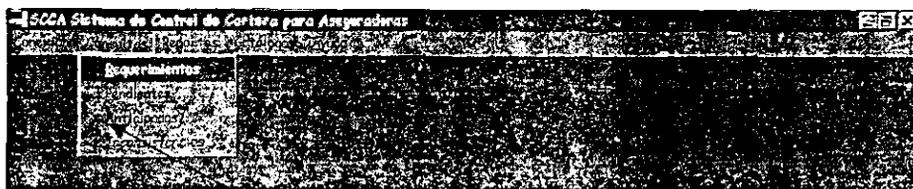


Figura 5.5.4 Selección de una opción mediante la primera letra.

**Uso del mouse:** Se podrá hacer uso del mouse oprimiendo el botón izquierdo con el fin de activar un botón de aceptación, o un botón de cancelación o una selección, o para ubicarse en un campo específico para introducir la información.



## Prueba de Integración

Durante esta prueba las actividades se centraron en el diseño y construcción de la arquitectura de software. Se verifica y se construye el programa como un todo.

Durante la integración, las técnicas que más prevalecen son las de diseño de casos de prueba de la caja negra, aunque se pueden llevar a cabo algunas pruebas de la caja blanca con el fin de asegurar que se cubran los principales caminos de control.

### Especificación de la prueba de integración.

**Alcance de la prueba.** Se probaron características tales como:

- Fácil uso por parte del usuario.
- Desplegado de ayuda.
- El fácil acceso utilizando los dispositivos de entrada: el mouse y el teclado.
- Además de que el número de módulos cumpliera con las especificaciones solicitadas.

**Plan de prueba.** La estrategia general para la integración se dividió en fases y subfases, dirigidas a características específicas funcionales del software:

- Interacción con el usuario:
  - Selección de ordenes.
  - Representación visual.
- Procesamiento y representación de errores.
- Manipulación y análisis de los datos.
- Procesamiento y generación de información visual.
  - Reportes y estadísticas.
- Estructura y contenido de la base de datos.

En cada fase se siguieron los criterios con sus correspondientes pruebas:

**Integridad de la interfase:** Se probaron las interfaces internas y externas a medida que se incorporaron los módulos.

**Validez funcional:** Se llevaron a cabo pruebas diseñadas para descubrir errores funcionales.

**Contenido de información:** Pruebas para descubrir errores asociados con las estructuras globales y locales.

**Rendimiento:** Pruebas para verificar los límites de rendimiento establecidos durante el diseño del software.



Procedimiento de prueba: Describimos el orden de integración y las pruebas correspondientes a cada fase de integración.

#### Prueba de validación.

En una prueba de validación se deben comprobar los criterios de validación establecidos durante la fase de definición del sistema. Proporciona una seguridad final de que el software satisface todos los requerimientos funcionales y de rendimiento. Se usan exclusivamente técnicas de prueba de la caja negra.

Después de encontrar y corregir los errores de interfases, comenzamos la prueba de validación.

Prueba alfa. Para llevar a cabo la prueba alfa se invitó a un usuario en el lugar del desarrollo y en presencia del desarrollador, éste registró errores y problemas de uso. Es decir, todo esto se llevó a cabo en un entorno controlado, en el área de desarrollo. Al principio al utilizar el sistema, el operador tenía cierto temor y no se presentaba seguro. Algunos de los mensajes que proporcionaba el sistema fueron modificados a petición del usuario por que les faltaba claridad.

Las principales deficiencias en esta prueba fue la falta de familiaridad en el uso de menús y el entender el procedimiento para manipular los datos.

Prueba beta. Para llevar a cabo la prueba beta se llevó el sistema a un lugar donde el encargado del desarrollo no estuvo presente. Sobre todo se presentaron problemas tales como la necesidad de una explicación sencilla de los errores y mensajes además dudas en el instructivo.

#### Prueba del sistema

Verifica que cada elemento del programa trabaja adecuadamente alcanzando la funcionalidad y el rendimiento del sistema total.

Entre los tipos de prueba del sistema que se llevaron a cabo fueron:

Prueba de recuperación: Con el fin de probar su tolerancia a fallos y el periodo de tiempo de corrección de fallas. Para lo que forzamos el fallo del software mediante:

- Caída de la red.
- Fallas de energía eléctrica.

En ambas la transacción en proceso era la que se perdía. La recuperación requiere la intervención humana, se evaluaron los tiempos medios de recuperación que no excedieron de 1 minuto y sólo se perdía el registro que estaba siendo procesado en ese momento, determinándose que se encontraba entre los límites aceptables.

Prueba de Seguridad: El mecanismo que evita que una persona no tenga acceso al



sistema pueda modificar cierta información, se evita con el uso de passwords, inclusive de diferentes niveles de acceso.

El campo password permite la combinación de cualquier carácter ASCII lo que aumenta la combinación de nombres y por lo tanto la seguridad.

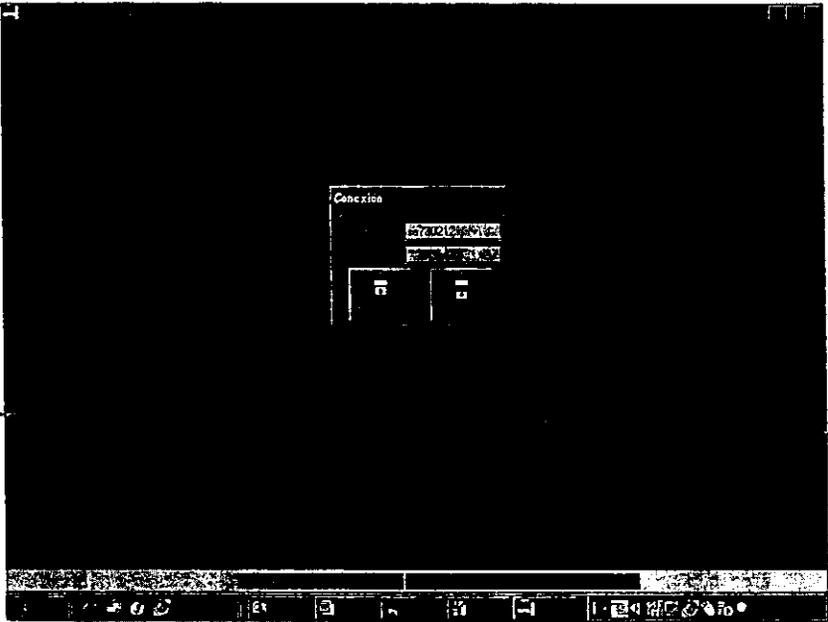


Figura 5.5.5 Prueba de Seguridad.

Prueba de resistencia y de rendimiento: Entre las pruebas que se sometieron fueron las siguientes:

Se realizaron de tal manera que se sometió al sistema a una demanda de recursos en cantidad y frecuencia anormales, por medio del aumento de interrupciones al sistema por medio de 15 operadores en red que demandaban al mismo tiempo recursos al sistema, de los cuales algunas eran de acceso normal al sistema.

Se probaron casos que produjeran excesivas búsquedas de datos residentes en disco, por medio de localizar a un cliente al mismo tiempo por 15 operadores.

Se probaron los tiempos de respuesta del sistema para los casos que requerían diferentes tamaños en la tabla de Clientes, desde 1 registro, pasando por 500 clientes hasta 5000.



## 5.6. FACTIBILIDAD TÉCNICA Y OPERATIVA.

Para poder garantizar que nuestro sistema sea viable de poner en producción es necesario llevar a cabo una serie de pruebas que permitan verificar que no existirán problemas de comunicación o problemas técnicos.

### Factibilidad Técnica.

Como primer paso, fue necesario verificar los requerimientos de equipo mínimos para poder ejecutar nuestro sistema en el servidor.

La infraestructura de la red dentro de la compañía y dentro de la red de área local de información, en donde se ubicará el servidor NT, ya se encuentra implementada y cumple con la normatividad establecida por la empresa para las redes que hagan uso del sistema operativo de red Windows NT dentro de la empresa.

Esta normatividad deberá seguirse por todas aquellas personas encargadas directamente de la instalación y puesta a punto del servidor NT.

El cumplimiento de esta normatividad garantiza la operación óptima del servidor y evitará cualquier problema con la interconexión, además de permitir la creación de una plataforma eficiente y estandarizada para proporcionar los servicios de red corporativos.

La parte más importante y más crítica de una red la constituye el servidor, su función principal es la de compartir recursos dentro de una red, como son los servicios de impresión, de disco y aplicaciones cliente/servidor. De la misma forma debe proporcionar una plataforma robusta para la ejecución de aplicaciones de misión crítica.

Por lo anterior, se hace necesario que la computadora destinada a ser servidor cumpla con los requisitos específicos para realizar de manera eficiente todas las funciones anteriormente mencionadas. Estos incluyen los concernientes a sus requerimientos mínimos de hardware y software, requerimientos eléctricos y ubicación física.

De acuerdo a las especificaciones del software, sistema operativo, aplicaciones y experiencia, los requerimientos mínimos necesarios fueron los siguientes:

- Servidor con Procesador Pentium II con velocidad de 350 MHz.  
Memoria RAM de 128 MB.  
Unidad de CD-ROM  
Unidad de Respaldo en cintas  
Disco Duro de 9 GB  
Floppy de 3.5"  
3 bahías disponibles (para un futuro crecimiento)  
3 ranuras de expansión PCI  
Tarjeta de red PCI



2 puertos seriales y 1 paralelo

Mouse

Monitor SVGA color

UPS con interface al servidor (1200 Watts). El servidor debe estar alimentado por un UPS, y a su vez contar con una interfaz al mismo para protegerlo de forma automática en caso de falla en el suministro de energía eléctrica; gracias a esta interface, es posible apagar en forma segura el servidor, evitándose así la pérdida de información y posibles daños al equipo.

- Sistema Operativo Windows NT.
- Microsoft SQL Server.
- Visual Basic.

El servidor debe estar colocado junto con los demás servidores de la compañía en un rack de servidores que se encuentre en un área exclusiva y libre del paso, así mismo, contar con aire acondicionado para mantener una temperatura constante de los equipos, todo esto con la finalidad de que el manejo físico de los equipos sea sin dificultad.

Para cumplir con estos requisitos no fue necesario solicitar nuevo equipo, ya que se contaba con el necesario para ello. Lo único que se tuvo que hacer fue realizar las configuraciones y adecuaciones necesarias para realizar las pruebas y verificar su funcionamiento.

---

Todo lo anterior se realizó con la ayuda de los administradores de los servidores, quienes dieron las facilidades para esta configuración.

De lo anterior podemos concluir que fue perfectamente viable la configuración del servidor, ya que se contaba con el equipo necesario.

Para poder definir los requerimientos mínimos necesarios en la parte de los clientes se instaló el sistema en 4 configuraciones diferentes con las siguientes características:

- Procesador 486 escalado a Pentium con velocidad de 133 MHz.  
Disco duro de 1GB.  
Windows 95.
- Procesador Pentium con velocidad de 133 MHz.  
Disco duro de 1.5 GB.  
Windows95.
- Procesador Pentium con velocidad de 200 MHz.  
Disco Duro de 1.2 GB.  
Windows 95.
- Procesador Pentium con velocidad de 333 MHz.  
Disco Duro de 2 GB.  
Windows 95.



En estos cuatro equipos, que son con los que cuentan actualmente las áreas en donde se instaló el sistema, se realizaron las siguientes pruebas:

- Verificación del espacio disponible en disco duro, ya que el sistema requiere de un espacio en disco de 50 MB como mínimo.
- Compatibilidad con el software existente en los equipos. Lo anterior es muy importante, ya que en algunas ocasiones se llegan a presentar problemas con versiones de algunas de las librerías del sistema, las cuales pueden entrar en conflictos con aplicaciones nuevas.
- Tiempo de respuesta. En este caso se hicieron pruebas para tratar de verificar los diferentes tiempos de respuesta entre las computadoras personales y el servidor en cuanto a tiempo entre consultas. Definitivamente no se contaba con un parámetro que nos pudiera guiar, el tiempo aceptado se definió en conjunto con las personas que operarían directamente el sistema, quienes indicaron que el tiempo de respuesta que se les mostró cumpliera con sus expectativas.

Todo lo anterior arrojó los resultados siguientes:

Como es fácil suponer, mientras más velocidad de procesamiento y mayor espacio en disco duro se tenga, menores son los tiempos de respuesta para el sistema, sin embargo, aún en las computadoras personales actualizadas de 486 a Pentium se obtuvieron tiempos de respuesta bastante aceptables. Lo anterior se debe a que al estar trabajando en un ambiente Cliente/Servidor, no importa mucho el tener un cliente delgado, lo importante es contar con un ancho de banda en la red que permita que la comunicación entre los clientes y el servidor sea lo bastante rápida para satisfacer las necesidades del cliente.

Con todo lo anterior pudimos definir como requerimientos mínimos para los clientes los siguientes:

- Procesador Pentium con velocidad de 133 MHz.  
Memoria RAM de 32 MB  
Floppy de 3.5"  
Tarjeta de red PCI  
2 puertos seriales y 1 paralelo  
Mouse  
Monitor SVGA color
- Espacio en Disco Duro de 50 MB.
- Windows 95 y software ya existente.

La última especificación se tuvo que estipular, porque la experiencia indica que cualquier software extra que se agregue al sistema es susceptible de generar conflictos con el software existente.



### **Administración del sistema.**

Para la administración del sistema se sugiere que el encargado de estas labores tenga la capacidad técnica y conocimientos de informática suficientes para cumplir con las siguientes funciones:

- Administración de usuarios.
- Administración de la seguridad.
- Realización de respaldos.
- Atención y aclaración de dudas.
- Soporte técnico.
- Control de cambios del sistema.

### **Control de cambios.**

Aún cuando el sistema está creado bajo los requerimientos del usuario, no están descartadas posibles modificaciones del sistema, con las que puedan aumentar su capacidad o agregar algún otro requerimiento.

Es por esta razón que deberá llevarse a cabo un control de cambios realizados en el sistema, indicando datos importantes como son: fecha, motivo de la modificación, tipo de modificación, persona que autorizó, impacto, ~~plan de contingencias y persona que realizó el cambio.~~

El control servirá para llevar un histórico del sistema y de cada una de sus modificaciones por fallas, limitaciones de diseño y/o cualquier aumento de necesidades.

### **Tipos de mantenimiento.**

Basándose en el crecimiento que el sistema pudiera tener, se dan las mejoras al sistema, esto es los diferentes tipos de mantenimiento, que se presentan a continuación. Existen cuatro tipos de mantenimiento: correctivo, preventivo, perfectivo y adaptativo dentro del cual se encuentra el aumentativo y el tecnológico.

- **Mantenimiento Perfectivo:** esta actividad de mantenimiento se da cuando un paquete de software tiene éxito. A medida que se usa el software, se reciben de los usuarios recomendaciones sobre nuevas posibilidades acerca de modificaciones a funciones ya existentes. Para satisfacer estas peticiones se lleva a cabo el mantenimiento perfectivo. Este mantenimiento comprende también los cambios solicitados al programador del sistema.
- **Mantenimiento Preventivo:** en este tipo de mantenimiento se previenen errores. Se da cuando cambia el software para mejorar una futura facilidad de mantenimiento. También se puede considerar el mantenimiento a la información que se maneja para garantizar que los resultados dados por el



sistema sean los correctos.

- **Mantenimiento Adaptativo:** en este tipo de mantenimiento se encuentran implícitos el aumentativo y el tecnológico. La vida útil estimada del software de aplicación puede fácilmente sobrepasar los diez años, pero considerando la evolución del ambiente, en la práctica éste puede volverse obsoleto. Por lo tanto, el mantenimiento adaptativo es una actividad que modifica el software para que las interacciones sean adecuadas en su entorno cambiante. El mantenimiento adaptativo se debe a cambios en el ambiente del programa y a la adaptación de nuevas unidades o módulos. De este mantenimiento puede derivarse los siguientes mantenimientos:
  - **Mantenimiento Aumentativo:** este se da cuando se incluyen nuevas funciones que fueron complementadas al inicio del desarrollo del sistema y que surgen como una necesidad del usuario.
  - **Mantenimiento Tecnológico:** se da debido a los cambios importantes en la informática, es decir, al avance de los sistemas operativos, lenguajes de programación y nuevas generaciones de hardware, con lo cual se mejoran las herramientas de trabajo, las versiones del software, equipos periféricos y otros elementos del sistema.
- **Mantenimiento Correctivo:** la primera actividad del mantenimiento se da ocasionalmente cuando la prueba del software no ha descubierto todos los errores latentes de un sistema. Durante el uso del sistema se encontrarán errores, los cuales deben ser informados al equipo de desarrollo. El proceso que incluye el diagnóstico y corrección de uno o más errores se denomina mantenimientos correctivo.



Figura 5.6.1 Tipos de mantenimiento.



### Análisis del costo del sistema.

El costo total del sistema puede realizarse basándose en lo siguiente:

- Número de líneas de código.
- Por hora – hombre. (Horas-Hombre de Analistas y Programadores).
- Por tiempo de desarrollo

El costo de una línea de código oscila entre los 20 y los 30 dólares. El sistema cuenta con aproximadamente 2200 líneas de código, por lo tanto el costo sería de 66000 dólares si tomamos el valor por línea de código de 30 dólares al tipo de cambio actual.

El tiempo de desarrollo está basado en las horas-hombre utilizadas para el análisis, desarrollo e implantación del sistema, por lo cual se tienen los siguientes datos:

El tiempo de desarrollo para el sistema fue de 4 1/2 meses, desde el análisis, desarrollo e implantación del mismo. Los analistas (que fueron dos) trabajaron durante cuatro semanas considerando sólo días hábiles y 8 horas por día, por lo tanto para el análisis se tienen 320 horas-hombre en total.

Los programadores (que fueron tres) trabajaron durante 3 1/2 meses 8 horas al día, solo se consideran los días hábiles, por lo tanto el número de horas por los 3 1/2 meses laborados es de horas-hombre por ambos. El costo por hora-hombre para los analistas y programadores de sistemas varía entre los 35 y los 55 dólares.

A continuación se presenta el análisis de precios realizado por horas-hombre:

Concepto	Costo unitario	Unidad de medición	Cantidad	Costo por concepto
Analistas(2)	50 dólares	Hora-hombre	320	16000 dólares
Programadores(3)	35 dólares	Hora-hombre	1680	58800 dólares
Total				74800 dls

Tabla 5.5.1 Análisis de costos por Hora-Hombre

Realizando un ajuste en los costos anteriores y basándose en el tiempo de desarrollo del sistema tenemos lo siguiente:



Concepto	Costo Unitario	Unidad de medición	Cantidad	Costo por concepto
Tiempo de desarrollo (Análisis, Diseño, Programación e Implantación)	\$21400	Mes	3.5	\$75000
Total				\$75000

Tabla 5.5.2 Análisis de costos por tiempo de desarrollo.

La empresa a la cual se le desarrollo el sistema ya cuenta con la infraestructura tecnológica (red) necesaria para la implantación del sistema, así como con el software y hardware necesarios, es por esta razón que estos conceptos no se incluyen en el análisis de costos.

Basándonos en el análisis de costos anterior, se determina que la operación de compra-venta del sistema puede realizarse de las siguientes formas:

- Venta del sistema por licencia.
- Venta total del sistema.
- El costo de venta por licencia sería de: Seiscientos dolares, al tipo de cambio actual.
- El costo del sistema será de 75000 dólares, al tipo de cambio actual.

La empresa para la cual se desarrollo el sistema tiene como política el adquirir totalmente los derechos de los programas realizados, es decir, requiere que le sea entregado la totalidad del código del sistema, manuales de usuario y manuales técnicos.

Por lo anterior la operación de compra y venta del sistema se realizará de forma total.

Al término del proyecto, la empresa que contrato el servicio, tendrá como producto final, las licencias y permisos respectivos sobre el producto desarrollado, así como los manuales, código fuente y todo aquello que involucre al sistema, así como un soporte de garantía sobre el producto, el cual se ofrece como valor agregado a la compañía.

### Factibilidad Operativa.

Para determinar si el sistema será benéfico para la empresa, se tiene que determinar los requerimientos operativos de ésta, por lo que hay que planear algunos pasos a seguir para asegurar el logro de los objetivos planteados, destacando los siguientes:

- Verificación cuidadosa de la calidad y la integración.
- Identificación de todos los elementos claves para las etapas de diseño y transición.
- Exactitud en la información de volúmenes de funciones y datos.



- Control del equipo para mantener un trabajo detallado, sin olvidar la planeación de tiempo establecido.
- Fuentes de información (datos a reunir, existentes en sistemas manuales y sistemas semiautomáticos).
- Estado de los datos, su preparación y conversión.
- Métodos de extracción de datos, su complejidad y esfuerzo necesario.
- Contribución del grupo de desarrollo.
- Recursos, tareas y calendarios.

Dentro de los beneficios que brindará la implantación del nuevo sistema se encuentran:

- Menores costos por concepto de operación y mantenimiento del equipo de cómputo en donde se ejecuta la aplicación.
- Mayor certidumbre en el manejo de los recibos así como de las finanzas de la empresa.
- Para el usuario final, la operación del sistema será amigable y bajo un ambiente gráfico.
- El mantenimiento del sistema no representará altos costos para la empresa debido a que parte del personal que desarrollo este proyecto labora en dicha empresa.
- ~~Se eliminarán muchos de los procesos manuales y por consiguiente se ahorrará mucho tiempo.~~

### Plan de desarrollo.

Expuesto lo anterior, estamos en posibilidad de generar un plan de desarrollo del sistema, el cual deberá establecer las tareas para llevar a cabo la implementación y utilización del sistema.

Las tareas pueden considerarse como las actividades definidas de un plan de desarrollo en donde cada actividad será enlistada en orden de ejecución, además deberá contemplar el tiempo estimado y los recursos necesarios para lograr el objetivo de la actividad. Cuando surja la necesidad, las tareas pueden dividirse en subtareas, con el fin de definir tareas específicas de desarrollo.

En este trabajo hemos dividido el plan de actividades en cinco tareas principales para el desarrollo del sistema.

- Requerimientos
- Análisis del desarrollo del sistema
- Selección de infraestructura
- Codificación
- Liberación

Cada una de estas actividades contiene subtareas que hacen que el sistema quede



totalmente definido. Las subtarefas están descritas en el plan de trabajo, y mencionan el tiempo de ejecución en días.

A continuación describiremos cada una de las tareas en las que se fundamenta el plan de desarrollo para la elaboración completa del sistema.

- **Definición.** En esta parte se define con claridad los objetivos y alcance del problema a resolver.
- **Análisis conceptual.** Aquí se realiza un análisis general de la solución en sus componentes principales.
- **Diseño conceptual.** Realizamos el diseño conceptual de la idea general de la propuesta a resolver tomando en cuenta todas las entidades posibles.
- **Modelo conceptual.** Se realiza el modelo conceptual del sistema completo basándose en los requerimientos del usuario.
- **Flujo de datos.** Se explican las entidades y el flujo de datos que aplicaría en el nuevo sistema.
- **Procedimientos.** Se establecen los procedimientos en los que el usuario se internará con el nuevo sistema.
- **Selección de infraestructura.** En esta etapa se realiza la selección de HW en el que se instalará el sistema y el SW con el que se realizará la programación.
- **Desarrollo de programas.** Aquí los programadores realizan todos los componentes del sistema.
- **Pruebas stand-alone.** Una vez desarrollado el sistema, este es probado en cada uno de sus componentes en una sola terminal haciendo una simulación en línea.
- **Pruebas en línea.** Ya que el sistema es trasladado al servidor éste es probado en cada uno de sus componentes ya en línea.



DISÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

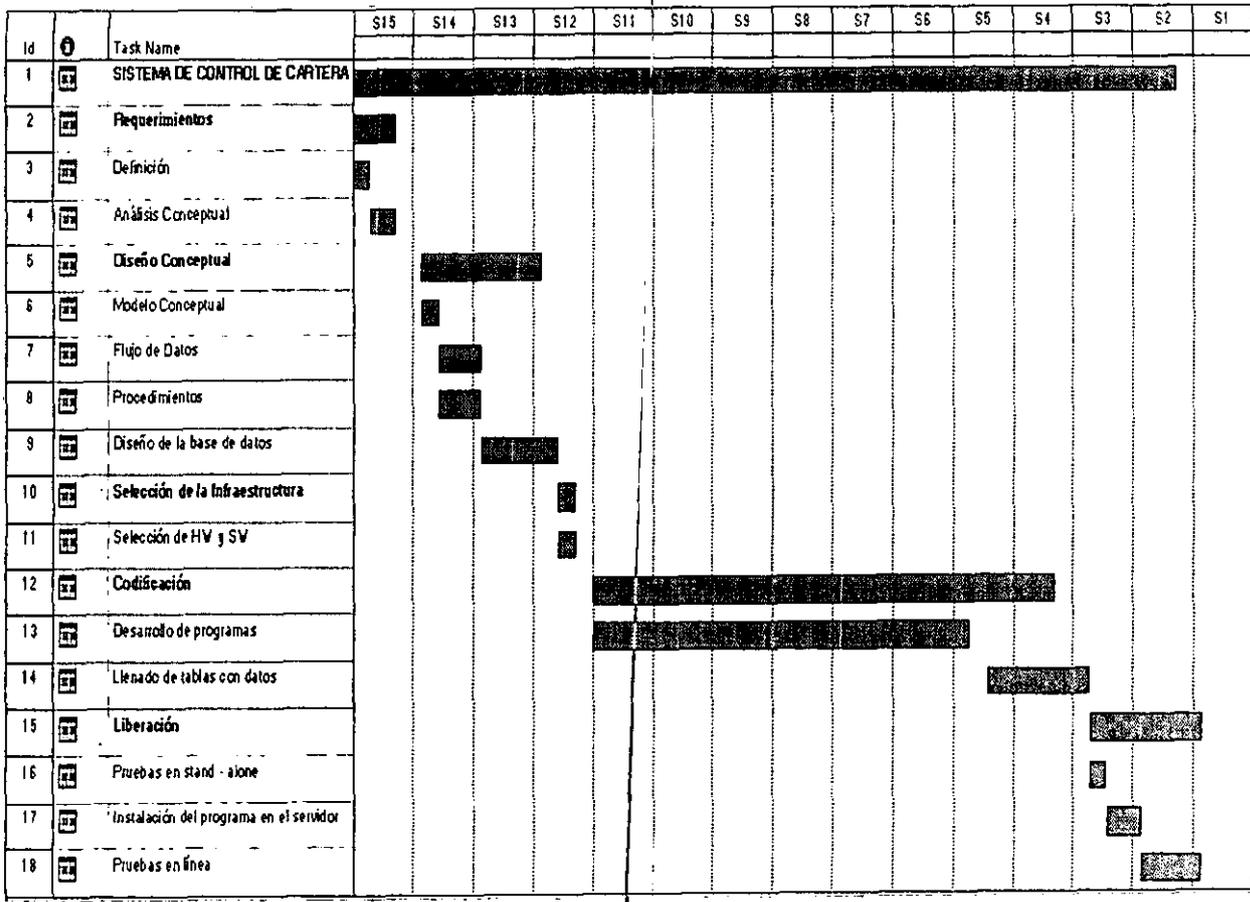


Figura 5.6.2 y 5.6.3 Diagramas de Gantt para el sistema.



# CONCLUSIONES



## CONCLUSIONES.

- Se cumplió con el objetivo principal de la tesis.
- El sistema de Recuperación de Cartera ha presentado un avance significativo en la agilización de la cobranza, y ha disminuido el fraude y las pérdidas de la compañía, por lo que esta cumpliendo con los objetivos.
- El sistema esta dividido en módulos, lo cual facilita su mantenimiento, se pueden identificar rutinas y procedimientos que necesiten ser modificados para implementar funciones que el usuario requiera en el futuro.
- El sistema es amigable y fácil de usar, ya que cuenta con un sistema de navegación entre menús y ventanas. Siempre se tiene toda la información que el usuario necesita en cualquier momento en la pantalla.
- La ingeniería de Software es el medio más adecuado para solucionar todos los problemas que resulten del desarrollo de los sistema.
- El sistema de recuperación de cartera resultó ser más exitoso de lo que se esperaba por lo que, en un futuro inmediato se harán necesarias mejores versiones.
- Con el desarrollo de este sistema se vera disminuida en gran proporción el crecimiento de la cartera vencida.
- El gasto que se emplea en el traslado de los recibos de pago será nulificado, porque ahora el viaje del recibo es electrónico y no por valija.
- La información es actualizada al día, lo que genera un beneficio económico muy importante.
- La interfase grafica generada es tan amigable que los usuarios reducen el tiempo de aprendizaje y la empresa el costo de capacitación.
- La reducción en tiempo de generación de información es tan importante que los resultados se verán reflejados inmediatamente.
- Es posible alcanzar un gran volumen de información en poco tiempo, lo cual no era factible con el procedimiento anterior.
- La dirección toma decisiones tan rápidamente, que el contar en la actualidad con la información en forma oportuna, ayudará a tomar decisiones inteligentes de negocio.



- Con la conclusión del presente trabajo observamos la importancia que están adquiriendo hoy en día los sistemas computacionales, ya que se han hecho básicos en todas las empresas e industrias, no importando cuan grandes o pequeñas sean estas, y la rama a la que se dediquen.
  - Es muy importante el diseño adecuado desde el inicio del sistema, ya que de no ser así, se tendrán que enfrentar una serie de problemas que se hubieran evitado con un buen diseño, esto en el mejor de los casos, ya que si es muy grande el problema, puede ser que se tenga que rediseñar todo el sistema, es por ello que es necesario el uso adecuado de las metodologías de desarrollo de sistemas, ya que nos llevarán a un diseño e implementación correctos.
  - El creciente desarrollo de sistemas computacionales, nos obligan a nosotros como ingenieros a buscar nuevas y mejores herramientas, con las cuales podamos estar a la vanguardia, ya que de no ser así se volviera obsoleto nuestro trabajo en unos cuantos meses.
  - El trabajo de un ingeniero no termina cuando finaliza el proyecto, se debe dar el mantenimiento adecuado para que este siga funcionando correctamente el mayor tiempo posible.
-



# APÉNDICES



## APÉNDICE A. MANUAL DE USUARIO.

### Introducción

El Sistema de Control de Cartera para Aseguradoras "SCCA" surge de la necesidad de llevar el control de todos los recibos que se encuentran en poder de un agente, ya sea que estos se encuentren cancelados, en periodo de gracia o vigentes y deben ser devueltos a la compañía, así como nos permite conocer la cartera pendiente de cualquier agente.

### Objetivos

- El presente manual dar a conocer las diferentes transacciones del sistema, distribuidas en 4 módulos principales que son:
  - Recibos por ser Requeridos.
  - Recibos Pendientes de Pago.
  - Recibos Entregados Anticipadamente a su Cancelación
  - Recibos Inconsistentes.
- Aprender las variantes de consulta de información.
- Explicar el uso de todas las formas de explotación de información mediante la utilización de un sistema más amigable y flexible.

### Arranque del Sistema

#### Inicio de la aplicación

Para poder dar inicio al sistema, desde el Explorador de programas de Windows seleccionamos el grupo de programas SCCA, dentro de este grupo de programas se encuentra el icono de la aplicación SCCA, se selecciona este icono con el mouse y se presiona doble click o se presiona ENTER desde el teclado.

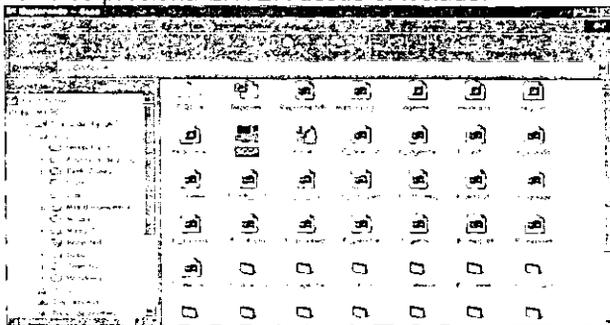


Figura 1. Grupo de Programas.



## Pantalla de Inicio

Cuando da inicio la aplicación nos muestra la pantalla inicial con las opciones principales del menú inactivas, ya que para acceder a estas se debe entablar la conexión con el servidor central de información.

Para poder entablar la conexión con el servidor central es necesario seleccionar del menú principal el modulo Conexión y la opción Log In..., la cual se explica detalladamente mas adelante.



Figura 2. Pantalla Principal.

## Menú Principal

Una vez conectados al sistema, el menú principal nos presenta los siguientes módulos:



Figura 3. Menú Principal.

A continuación se explica cada uno de los diferentes módulos del sistema.

## Conexión

El objetivo primordial de este modulo es poder entablar la comunicación con el servidor central de información

Este modulo cuenta con dos opciones:



Figura 4. Opciones del Modulo de Conexión.

A continuación se explica cada una de estas.

### Log In...

Esta opción nos muestra la pantalla de conexión que nos permite conectarnos con el servidor central de información para poder establecer comunicación con este.

Dentro de esta pantalla debemos colocar dentro del cuadro de Usuario nuestra clave personal y dentro del cuadro de Password nuestra contraseña, a continuación presionamos el botón de conectar de forma que si son correctos nuestros datos y tenemos acceso al sistema nos activa únicamente las opciones del menú principal a las que tenemos acceso, desplegando nuestro nombre en la parte inferior derecha de la pantalla principal, deshabilitando la opción de Log In..., puesto que ya estamos conectados.

En el caso de que no se nos autorice el acceso, el sistema nos permite repetir la operación, ya sea corrigiendo los datos capturados o cancelar el intento de conexión para intentarlo posteriormente.

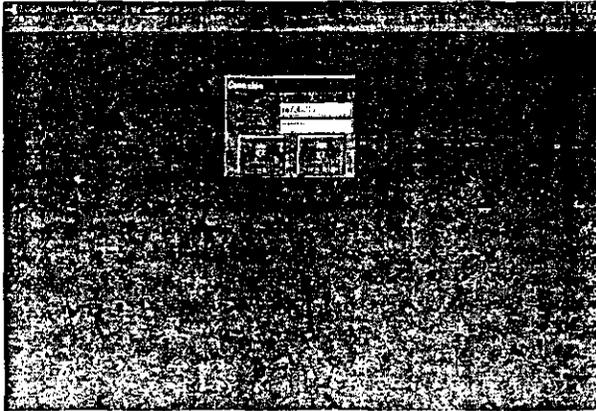


Fig. 5.- Pantalla de conexión.

### Salir

Esta opción nos permite salir del sistema perdiendo la conexión con el servidor central o en el caso de no habernos podido conectar también nos permite abandonar la aplicación.



## Consultas

Este es el modulo mas importante del sistema ya que nos permite el manejo de toda la información de recibos en poder de un intermediario y cuenta con las siguientes opciones:

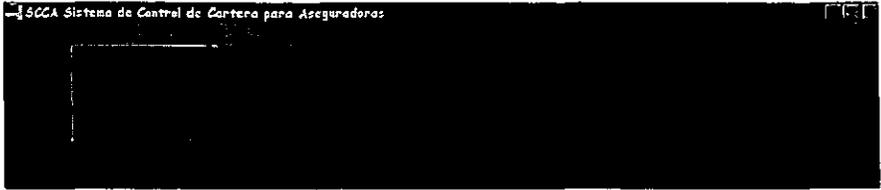


Figura 6. Opciones del Modulo de Consultas.

- Consulta de Requerimientos.
- Consulta de Pendientes de Cobro.
- Consulta de Entregados Anticipadamente a su Cancelación.
- Consulta de Inconsistentes.

A continuación se explica a detalle cada una de ellas.

### Consulta de Requerimientos

El objetivo de esta consulta es el de reportar los recibos que se encuentran cancelados o pagados con fondo y aun se encuentran en poder del intermediario, dichos recibos deben ser requeridos aunque ya no tengan validez, para evitar un mal manejo de estos en nombre de la compañía.

Esta consulta se accesa eligiendo la opción Requerimientos, la cual cuenta con distintas pantallas que se despliegan en forma secuencial y son: el criterio de selección, el despliegue de información y el cambio de estatus del recibo, ya sea para auditar a nivel divisional o para requerir a nivel regional.

#### Criterio de Selección

Esta pantalla nos permite establecer el criterio para la búsqueda de recibos, ya sea desde un nivel global, toda la compañía, hasta el nivel de agente, y poder seleccionar un ramo en específico o todos, así como una UEN específica o ambas.

Los recibos que nos va mostrar son de acuerdo al nivel que tenga el usuario que ingresó al sistema siendo:

- Recibos Pendientes y Rechazados para un usuario de regional.
- Recibos Entregados y Revisados para un usuario de división.
- Todos los recibos para usuarios de subdirección o globales (toda la compañía).



Figura 7. Criterio de Búsqueda de Requerimientos

El nivel mas alto es a nivel compañía (global), después sigue la división, la subdirección, la región, el promotor y el agente respectivamente.

Cuando nosotros solicitamos una consulta de nivel global y hasta promotor, el sistema agrupa todos los recibos y calcula totales del nivel inferior al que se solicitó, mostrando los datos en la pantalla de despliegue de información, así como también se muestra la información de cada recibo cuando solicitamos la consulta a nivel de agente.

La forma en que nosotros seleccionamos el nivel al que queremos la información, es la de llenar en forma descendente los campos de división, subdirección, región e intermediario. También se cuenta con niveles de acceso, ya que un usuario de regional únicamente puede consulta los agentes de su regional, ya que el sistema bloqueará los niveles a los que no tiene acceso un usuario, un usuario de división puede consultar desde cualquier subdirección de su división, hasta cualquier agente que se encuentra dentro de alguna región de esta.

En el bloque de intermediario, tenemos el tipo y clave de intermediario del que deseamos extraer la información, de tal forma, que cuando el tipo de intermediario es agente, del lado derecho del bloque de intermediario nos despliega la clave y descripción de la promotoría a la que pertenece este, en caso de ser cualquier otro tipo de intermediario este recuadro desaparece.

Figura 8. Bloque de agente.



En el bloque de UEN podemos seleccionar entre personas, empresas o ambas, esto nos sirve al momento de seleccionar la información únicamente nos despliega la información de los ramos que pertenecen a la UEN seleccionada.

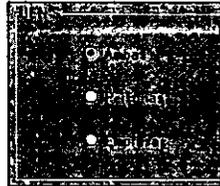


Figura 9. Bloque de UEN.

En el bloque del ramo se cuenta con dos opciones, todos los ramos o un ramo del catálogo en específico, esto con el fin de extraer una consulta de información mas exclusiva.

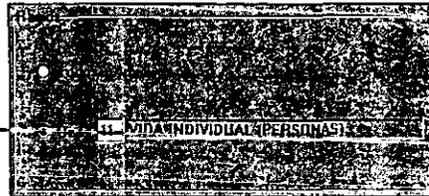


Figura 10. Bloque de ramo.

Dentro de la pantalla del criterio de selección, en el momento en que no conozcamos alguna clave ya sea de división, subdirección, región, tipo de intermediario, clave de intermediario o ramo, el sistema nos muestra una pantalla de ayuda para catálogos, la cual se activa al presionar doble click con el mouse sobre el campo del cual no conocemos la clave. El funcionamiento de esta pantalla se explica mas adelante dentro de este capítulo.

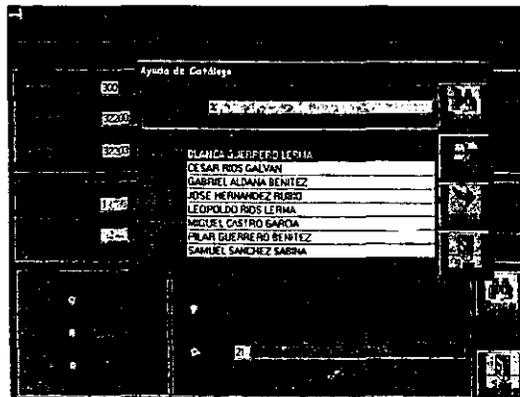


Figura 11. Pantalla de ayuda de catálogos.



Una vez que ya se estableció el criterio de selección se presiona el botón Buscar, dicho botón nos permite ir al servidor central por la información solicitada en el criterio, y nos muestra el resultado en la pantalla de despliegue de información en caso de existir.

En el momento en el que ya no se desea consultar mas información, el botón Salir nos permite cerrar la pantalla de criterio de búsqueda y regresar a la pantalla principal.

### Despliegue de Información

Esta pantalla nos muestra la información de todos los recibos que solicitamos dentro del criterio de búsqueda, la información es desplegada de dos formas distintas, la primera nos muestra el concentrado de recibos, agrupando a un nivel inferior del que se solicito la búsqueda, separando la información por UEN y desplegando un total para cada concentrado.

CLAVE	NOMBRE/CIUDAD	MONEDA	RECIBOS	UEN	PERSONAS	RECIBO
48259	ANA MARIA SALINAS CASTRO	PESOS	7	7420.12	0	0
61705	ANTONIO GARDUNO CANO	PESOS	10	51266.53	0	0
84416	AROLDI RIVERA CHAVERRI	PESOS	1	2344.31	0	0
41878	CARLOS ROSALES ASCANIO	PESOS	2	1250.75	0	0
55862	CARMEN H GUERRERO ROMERO	PESOS	9	6528.64	0	0
22113	CARMEN MILLAN BOJAILL	PESOS	9	55095.63	1	0
45249	DIEZ RICARDO DE CASTRO	PESOS	12	32411.40	3	0
54927	ELSA MA AMPUDIA SANTIBANEZ	PESOS	3	743.82	0	0
19306	FELIPE PEREZ GUERRERO	PESOS	2	283.48	4	0
40328	GENOVEVA GARDUNO CAMACHO	PESOS	8	7078.26	0	0
53054	GLORIA MORALES VAZQUEZ	PESOS	6	7522.39	0	0
54237	INGRID JEREZANO PEREZ	PESOS	21	37001.78	0	0
84442	JAIME MANRIQUE ROBLES	PESOS	4	1977.00	0	0
24657	JESUS G VALDIVIA GUTIERREZ	PESOS	14	35114.79	0	0
15165	JORGE NARCISO TORRES URIBE	PESOS	4	1956.36	0	0
4162	JOSE ANTONIO GUASCO YRIGOYEN	PESOS	6	11802.19	0	0
5502	JOSE RAUL GUERRA VARELA	PESOS	6	2059.47	0	0
23341	JUAN FERMIN PEREZ IRABIEN	PESOS	1	1894.67	0	0
84011	JULIO CESAR OBREGON RAMOS	PESOS	4	7695.92	0	0
22723	LA SERNA ANDRES GARDUNO DE	PESOS	8	2717.95	0	0
55425	LAURA RAMIREZ HERNANDEZ	PESOS	2	2583.18	0	0
54455	LOS SANTOS ANA E CANSINO DE	PESOS	6	32515.14	0	0

Figura 12. Despliegue de información por UEN.

Por ejemplo, si se solicito hasta región el sistema nos despliega por promotor y si se solicito por compañía el sistema nos despliega a nivel división, esta forma de despliegue de información se hace desde nivel compañía hasta promotor.

La segunda forma en que el sistema nos muestra la información se aplica cuando la solicitamos a nivel de agente, y nos despliega todos los datos de cada recibos que solicitamos en el criterio de búsqueda.



**DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS**

FECHA	NO CAMBIO	TIPO RECIBO	PRIMA	COMPLEMENTO	MOREA	RAMO
10/10/1998	0	INICIAL	1785	370302	PESOS	VIDA INDIVIDUAL
10/10/1998	0	INICIAL	3356.57	361211	PESOS	VIDA INDIVIDUAL
10/10/1998	42029	INICIAL	309.07	970108	PESOS	VIDA INDIVIDUAL
10/10/1998	42029	INICIAL	309.07	970308	PESOS	VIDA INDIVIDUAL
10/10/1998	42029	INICIAL	309.07	970408	PESOS	VIDA INDIVIDUAL
10/10/1998	42041	INICIAL	309.07	970108	PESOS	VIDA INDIVIDUAL
10/10/1998	42041	INICIAL	309.07	970308	PESOS	VIDA INDIVIDUAL
10/10/1998	42041	INICIAL	309.07	970408	PESOS	VIDA INDIVIDUAL
10/10/1998	0	INICIAL	1341.53	970219	PESOS	VIDA INDIVIDUAL
10/10/1998	48229	INICIAL	4104.98	361105	PESOS	VIDA INDIVIDUAL
10/10/1998	51141	INICIAL	1382.6	970302	PESOS	VIDA INDIVIDUAL
10/10/1998	454	INICIAL	1474.55	361104	PESOS	VIDA INDIVIDUAL
10/10/1998	454	SURSECUENTE	1356.13	361104	PESOS	VIDA INDIVIDUAL
10/10/1998	0	INICIAL	4373.42	970117	PESOS	VIDA INDIVIDUAL
10/10/1998	05305	INICIAL	1024.02	970328	PESOS	VIDA INDIVIDUAL
10/10/1998	0	INICIAL	1193.46	361108	PESOS	VIDA INDIVIDUAL
10/10/1998	0	INICIAL	2231.23	970102	PESOS	VIDA INDIVIDUAL
10/10/1998	62020	SURSECUENTE	11099.37	360228	PESOS	VIDA INDIVIDUAL
10/10/1998	0	INICIAL	2062.5	970212	PESOS	VIDA INDIVIDUAL
10/10/1998	0	INICIAL	4685.29	970415	PESOS	AUTONOMILES
10/10/1998	0	INICIAL	2873.68	361220	PESOS	AUTONOMILES

**Figura 13. Despliegue de información por recibo.**

Esta pantalla cuenta con varios botones de funciones los cuales se explican a continuación:



**Figura 14. Botones de pantalla**

La información que el sistema nos despliega es de 100 en 100 recibos por lo cual el botón de siguientes nos permite traer los próximos recibos (hasta 100) mientras este botón se encuentre activado.

El botón de requerir nos permite cambiar el estatus de un recibo a través de la pantalla de cambio de estatus, esta pantalla se explica detalladamente mas adelante, este botón puede tomar el nombre de auditar, dependiendo del nivel del usuario que ingrese al sistema.

El botón de reporte nos envía a impresión los datos que se tengan en pantalla con un formato definido, el formato de los reportes es de la siguiente manera:

**Formato de reporte de compañía a promotor**

Sistema de Control de Cartera para Aseguradoras						
Reporte de Recibos emitidos por Promotor						
CELULAR: 118 4100100000000000						
SUB-OFICINA: 14111111111111111111						
IMPRESORA: 11111111111111111111						
TEMPORAL: 11111111111111111111						
Oficina	Descripción	Cont.	VALOR	Com.	VALOR	Total
11111111111111111111	11111111111111111111	10	111.111.11	0	110	111.111.11
11111111111111111111	11111111111111111111	0	11.111.11	0	110	11.111.11
11111111111111111111	11111111111111111111	0	11.111.11	0	110	11.111.11
11111111111111111111	11111111111111111111	2	11.111.11	0	110	11.111.11
11111111111111111111	11111111111111111111	1	11.111.11	0	110	11.111.11
<b>Total:</b>		<b>12</b>	<b>111.111.11</b>	<b>0</b>	<b>110</b>	<b>111.111.11</b>

**Figura 15. Formato de reporte por oficinas o promotor.**





Esta pantalla también nos muestra la pantalla de ayuda de catálogo para cuando desconocemos la clave del nuevo estatus del recibo, mostrándonos únicamente las que tenemos acceso.

Una vez corregido el nuevo estatus y colocada una breve descripción se presiona el botón de modificar para que el cambio tenga efecto, esto nos elimina el recibo de la pantalla de despliegue de información y nos regresa a esta ya que en ese momento perdemos el control sobre este recibo.

En el caso de que no se desee modificar el estatus presionamos el botón de cancelar, esto nos regresa a la pantalla de despliegue de información sin modificar el estado del recibo antes de que se seleccionara para su modificación.

Esta pantalla también cuenta con una impresión definida de los movimientos que se han efectuado con este recibo, dicho formato se despliega a continuación.

**Sistema de Control de Cartera para Aseguradoras**  
Reporte de Seguimiento a un Recibo

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Asociación:</td><td>000000000</td></tr> <tr><td>Est. Control:</td><td>000000</td></tr> <tr><td>Módulo de Seguimiento:</td><td>000000</td></tr> <tr><td>Fecha de Inicio del Recibo:</td><td>00/00/00</td></tr> <tr><td>Fecha de Fin del Recibo:</td><td>00/00/00</td></tr> <tr><td>Estado:</td><td>0000</td></tr> <tr><td>Fecha de Inicio del Seguimiento:</td><td>00/00/00</td></tr> <tr><td>Fecha de Fin del Seguimiento:</td><td>00/00/00</td></tr> <tr><td>Operación de Seguimiento:</td><td>000000000</td></tr> </table>				Asociación:	000000000	Est. Control:	000000	Módulo de Seguimiento:	000000	Fecha de Inicio del Recibo:	00/00/00	Fecha de Fin del Recibo:	00/00/00	Estado:	0000	Fecha de Inicio del Seguimiento:	00/00/00	Fecha de Fin del Seguimiento:	00/00/00	Operación de Seguimiento:	000000000
Asociación:	000000000																				
Est. Control:	000000																				
Módulo de Seguimiento:	000000																				
Fecha de Inicio del Recibo:	00/00/00																				
Fecha de Fin del Recibo:	00/00/00																				
Estado:	0000																				
Fecha de Inicio del Seguimiento:	00/00/00																				
Fecha de Fin del Seguimiento:	00/00/00																				
Operación de Seguimiento:	000000000																				
No.	Descripción del Servicio	Fecha de Inicio	Fecha de Fin	Operación																	
1	SERVICIO	00/00/00	00/00/00	000000000																	
2	SERVICIO	00/00/00	00/00/00	000000000																	
3	SERVICIO	00/00/00	00/00/00	000000000																	

Figura 18. Formato del reporte de seguimiento de un recibo.

### Ayuda de Catálogos

Esta pantalla nos muestra la información contenida en el catálogo al que queremos acceder, el funcionamiento de esta pantalla es igual para cualquier catálogo y nos despliega la clave y la descripción de los elementos contenidos en este.

Esta pantalla cuenta de dos secciones principales:

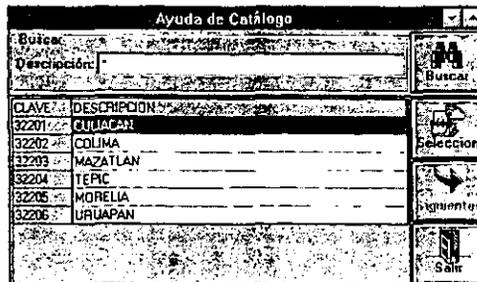


Figura 19. Pantalla de ayuda de catálogos.

La primera que es donde se puede seleccionar un grupo en específico de registros en base al criterio de selección que es una parte o toda la cadena de descripción, dentro



de la cadena de descripción podemos colocar un comodín que es el asterisco (\*), este comodín puede ir al principio y al final de una cadena, o en la parte media, este comodín significa cualquier cadena de cualquier longitud, y se presiona el botón Buscar que le indica al sistema que localice los primeros 50 registros del catálogo que su descripción coincida con la cadena que se ingreso en el criterio de búsqueda., por ejemplo:

Si solicitamos la ayuda en la clave de intermediario con el tipo de intermediario en agente y colocamos “\*PEREZ\*” dentro del criterio de búsqueda, al presionar el botón de Buscar el sistema nos nos retorna todos los agentes en los que existe la palabra PEREZ sin importar la posición de esta, pero si colocamos “\*PEREZ” dentro del criterio de búsqueda únicamente nos retornara a los agentes que su nombre termina con PEREZ.

La segunda parte es donde se despliegan los registros que se obtuvieron de la búsqueda, dentro de esta sección tenemos el botón de Selección que nos permite elegir el registro que tenemos marcado dentro de nuestros datos y colocar la clave y descripción en donde invocamos a la ayuda, también contamos con un botón de siguientes, el cual se activa en el momento que nuestro criterio de selección obtiene mas de 50 registros, el objetivo de este botón es el de traer los registros restantes de 50 en 50, hasta agotar nuestro criterio de selección, en la pantalla también contamos con un botón de salir el cual cancela nuestra ayuda de catálogo y nos regresa a la pantalla de donde la invocamos.

### **Consulta de Pendientes de Cobro**

El objetivo de esta consulta es el de reportar los recibos que se encuentran pendientes de cobro, con esta consulta podemos conocer la cartera proxima a vencer de un agente con el fin de poder presionarlo para su pronto pago, así como el poder marcar un recibo que aun se encuentra pendiente como entregado anticipadamente a su cancelación, esta consulta se accesa eligiendo la opción Pendientes, la cual cuenta con distintas pantallas que se despliegan en forma secuencial y son: el criterio de selección y el despliegue de información.

### **Criterio de Selección**

Esta pantalla nos permite establecer el criterio para la búsqueda de recibos, ya sea desde un nivel global, toda la compañía, hasta el nivel de agente, y poder seleccionar un ramo en específico o todos, así como una UEN específica o ambas.



Figura 20. Criterio de Búsqueda de Pendientes

El nivel mas alto es a nivel compañía (global), después sigue la división, la subdirección, la región, el promotor y el agente respectivamente.

Cuando nosotros solicitamos una consulta de nivel global y hasta promotor, el sistema agrupa todos los recibos y calcula totales del nivel inferior al que se solicitó, mostrando los datos en la pantalla de despliegue de información, así como también se muestra la información de cada recibo cuando solicitamos la consulta a nivel de agente.

La forma en que nosotros seleccionamos el nivel al que queremos la información, es la de llenar en forma descendente los campos de división, subdirección, región e intermediario. También se cuenta con niveles de acceso, ya que un usuario de regional únicamente puede consulta los agentes de su regional, ya que el sistema bloqueará los niveles a los que no tiene acceso un usuario, un usuario de división puede consultar desde cualquier subdirección de su división, hasta cualquier agente que se encuentra dentro de alguna región de esta.

En el bloque de intermediario, tenemos el tipo y clave de intermediario del que deseamos extraer la información, de tal forma, que cuando el tipo de intermediario es agente, del lado derecho del bloque de intermediario nos despliega la clave y descripción de la promotoría a la que pertenece este, en caso de ser cualquier otro tipo de intermediario este recuadro desaparece.

Figura 21. Bloque de agente.



En el bloque de UEN podemos seleccionar entre personas, empresas o ambas, esto nos sirve al momento de seleccionar la información únicamente nos despliega la información de los ramos que pertenecen a la UEN seleccionada.

UEN'S

Ambas

Personas

Empresas

Figura 22. Bloque de UEN.

En el bloque del ramo se cuenta con dos opciones, todos los ramos o un ramo del catálogo en específico, esto con el fin de extraer una consulta de información mas exclusiva.

Ramos

Todos

Uno

Figura 23. Bloque de ramo.

Dentro de la pantalla del criterio de selección, en el momento en que no conozcamos alguna clave ya sea de división, subdirección, región, tipo de intermediario, clave de intermediario o ramo, el sistema nos muestra una pantalla de ayuda para catálogos, la cual se activa al presionar doble click con el mouse sobre el campo del cual no conocemos la clave. El funcionamiento de esta pantalla se explica mas adelante.

**Ayuda de Catálogo**

CLAVE	DESCRIPCIÓN
63	DIANA CASTRO ESPEJO
74	CESAR OSORIO BERNAL
117	ERNESTO SANCHEZ PADILLA
156	JOSE DIAZ MARES
210	LEOPOLDO LOPEZ ESPEJO
283	MARIO SUAREZ CONTRERAS
303	PILAR CASTRO GUEDA
353	SAMUEL RAMIREZ MEZA
358	VICTOR RODRIGUEZ ESPARZA

Inicio Selección Búsqueda Salir

Figura 24. Pantalla de ayuda de catálogos.



Una vez que ya se estableció el criterio de selección se presiona el botón Buscar, dicho botón nos permite ir al servidor central por la información solicitada en el criterio, y nos muestra el resultado en la pantalla de despliegue de información en caso de existir.

En el momento en el que ya no se desea consultar mas información, el botón Salir nos permite cerrar la pantalla de criterio de búsqueda y regresar a la pantalla principal.

### Despliegue de Información

Esta pantalla nos muestra la información de todos los recibos que solicitamos dentro del criterio de búsqueda, la información es desplegada de dos formas distintas, la primera nos muestra el concentrado de recibos, agrupando a un nivel inferior del que se solicito la búsqueda, separando la información por UEN y desplegando un total para cada concentrado.

UEN	Nombre	Recibos	Saldo	Personas	Recibo
102	ARONDO GARZA ELIZONDO	FESOS	4	10588.25	0
103	ARNULFO A. AGUIRRE GUTIERREZ	FESOS	486	304492.33	6
104	ARNULFO AGUIRRE MORENO	FESOS	2	39.23	0
105	CUATRO RODRIGUEZ DE LEON	FESOS	2	39.23	0
106	DONATO GUTIERREZ HERNANDEZ	FESOS	29	77580.24	2
107	FELIPE GARCIA OSORIO	FESOS	34	33909.53	5
108	GABRIEL FLORES LOPEZ	FESOS	30	40388.60	1
109	GASPAR HUICOCHE ALSINA	FESOS	115	268.280	2
110	JESUS SANDOVAL MARTINEZ	FESOS	3	697.10	0
111	JOSE BENITO AGUIRRE MORENO	FESOS	61	73553.10	2
112	JULIAN ANTONIO CARRIZALEZ LOPEZ	FESOS	4	7294.94	0
113	LUIS FIGUEROA AFROYO	FESOS	40	2153.36	0
114	MARCO A GONZALEZ ESCOBEDO	FESOS	26	47052.93	0
115	MARIA ANGELETA FRIAS MANDIYA	FESOS	136	130184.51	0
116	MIGUEL ANGEL AGUIRRE MORENO	FESOS	4	33807.55	0
117	NANCY GUADALUPE AGUIRRE GARZA	FESOS	52	57790.96	1
118	NICOLAS VILLARREAL VILLARREAL	FESOS	2	5966.79	0
119	RAMIRO TUERINA CANTU	FESOS	56	27860.00	0
120	RICARDO FADY ALLENDE	FESOS	14	14699.93	2
121	ROBERTO HERNANDEZ GONZALEZ	FESOS	1	12.10	0
122	ROBERTO ROMERO HERNANDEZ	FESOS	2	117.82	0
123	ARNULFO A. AGUIRRE GUTIERREZ	COLARES	23	3838.18	0

Figura 25. Despliegue de información por UEN.

Por ejemplo, si se solicito hasta región el sistema nos despliega por promotor y si se solicito por compañía el sistema nos despliega a nivel división, esta forma de despliegue de información lo hace desde nivel compañía hasta promotor.

La segunda forma en que el sistema nos muestra la información se aplica cuando la solicitamos a nivel de agente, y nos despliega todos los datos de cada recibos que solicitamos en el criterio de búsqueda.



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

POLIZA	No CAMBIO	TIPO REC.	PRIMA PESOS	VENCIMIENTO	MONEDA	RAMO
1000796350	0	INICIAL	7.06	970517	PESOS	VIDA INDIVIDUAL
1000A09272	0	INICIAL	8.6	970527	PESOS	VIDA INDIVIDUAL
1000A21560	0	INICIAL	7.74	970517	PESOS	VIDA INDIVIDUAL
1000A71729	0	INICIAL	19.23	970525	PESOS	VIDA INDIVIDUAL
1000A72498	0	INICIAL	10.94	970511	PESOS	VIDA INDIVIDUAL
1000A9759	0	INICIAL	51.23	970520	PESOS	VIDA INDIVIDUAL
1000K6492	0	INICIAL	514.13	970501	PESOS	VIDA INDIVIDUAL
1000K6507	0	INICIAL	535.13	970501	PESOS	VIDA INDIVIDUAL
1000L1954	0	INICIAL	751.87	970524	PESOS	VIDA INDIVIDUAL
1000K6136	0	INICIAL	813.13	970519	PESOS	VIDA INDIVIDUAL
1000L1542	0	INICIAL	1345.71	970522	PESOS	VIDA INDIVIDUAL
1000K55811	0	INICIAL	12.44	970513	PESOS	VIDA INDIVIDUAL
1000P4973	0	INICIAL	1165.92	970507	PESOS	VIDA INDIVIDUAL
1000Y00206	0	INICIAL	31.18	970617	PESOS	VIDA INDIVIDUAL
1478800014	0	INICIAL	546.79	971011	PESOS	INCENDIO
8178850003	0	INICIAL	3816.97	970630	PESOS	AUTOMOVILES INC
8178859971	27390	DEVOLUCION	2588.5	970603	PESOS	AUTOMOVILES INC
8478850589	0	INICIAL	6660.06	970506	PESOS	AUTOMOVILES INC
8578850298	0	INICIAL	814.52	970615	PESOS	AUTOMOVILES INC
8578850564	0	INICIAL	4736.71	970529	PESOS	AUTOMOVILES INC
8578850620	0	INICIAL	747.3	970531	PESOS	AUTOMOVILES INC
8578859995	0	INICIAL	1099.58	970511	PESOS	AUTOMOVILES INC

Figura 26. Despliegue de información por recibo.

Esta pantalla cuenta con varios botones de funciones los cuales se explican a continuación:



Figura 27. Botones de pantalla

La información que el sistema nos despliega es de 100 en 100 recibos por lo cual el botón de Sigüientes nos permite traer los próximos recibos (hasta 100) mientras este botón se encuentre activado.

El botón de entregado nos permite cambiar el estatus de un recibo de pendiente a entregado anticipadamente a través del siguiente mensaje de confirmación, este boton solamente se encuentra activado cuando el nivel del usuario que accesor al sistema es de regional.



Figura 28. Mensaje de confirmación.

Si se acepta el cambio se corrige el nuevo estatus y nos elimina el recibo de la pantalla de despliegue de información y nos regresa a esta ya que en ese momento perdemos el control sobre este recibo.

En el caso de que no se desee modificar el estatus presionamos el botón de No, esto nos regresa a la pantalla de despliegue de información sin modificar el estado del recibo antes de que se seleccionara para su modificación.

El botón de reporte nos envía a impresión los datos que se tengan en pantalla con un formato definido, el formato de los reportes es de la siguiente manera:

### Formato de reporte de compañía a promotor

Sistema de Control de Cartera para Aseguradoras

Reporte de RECIBOS PAGADOS por PROMOTORIA

Clasificación: 010 AUTOMOVIL INC  
 Subclases: 0110 MEXICO (MEXI)  
 Región: 0101 MEXICO  
 Compañía: 01 PROMOTORA Y ASESORIA S.A.

Clase	Descripción	CANT.	VALOR PAGADO	CANT.	VALOR PAGADO	Com.	Total
01	0101 MEXICO (MEXI)	10	10,000.00	0	0.00	10	10,000.00
02	0202 MEXICO (MEXI)	1	25,000.00	0	0.00	2	25,000.00
03	0303 MEXICO (MEXI)	2	15,000.00	0	0.00	2	15,000.00
04	0404 MEXICO (MEXI)	1	20,000.00	0	0.00	1	20,000.00
05	0505 MEXICO (MEXI)	1	25,000.00	0	0.00	1	25,000.00

Figura 29. Formato de reporte por oficinas o promotor.





## APÉNDICE B. MANUAL TÉCNICO.

En este manual se da una visión sobre el funcionamiento interno del sistema y la forma como se le debe dar mantenimiento y provee soluciones en caso de que se presente alguna contingencia.

### SQL Server.

La base de datos del sistema fue creada en SQL Server y recibe el nombre de SCCA, esta base de datos requiere de ciertos cuidados para mantenerla activa de manera satisfactoria, es conveniente hacer un respaldo de la base datos todos los días pues en casos extremos podría perderse la información que en ella se almacena. A continuación se mencionará como se hace un respaldo en cinta, como se paran y corren los servicios y como se calendarizan tareas en el Scheduled Task.

### Servicios de SQL Server.

En algunas ocasiones será necesario parar los servicios de SQL Server, ya que por fallas del sistema o bien por otras causas. Los servicios de SQL Server se pueden parar de maneras diferentes, ya sea por medio del icono servicios ubicado en el panel de control, otra forma es a través de la herramienta SQL Enterprise Manager o bien por medio del SQL Service Manager.

Si optamos por la primera opción debemos hacer lo siguiente:

- Abrir el icono Servicios en el panel de control.
- Detener los servicios de MSSQL Server (figura 1)

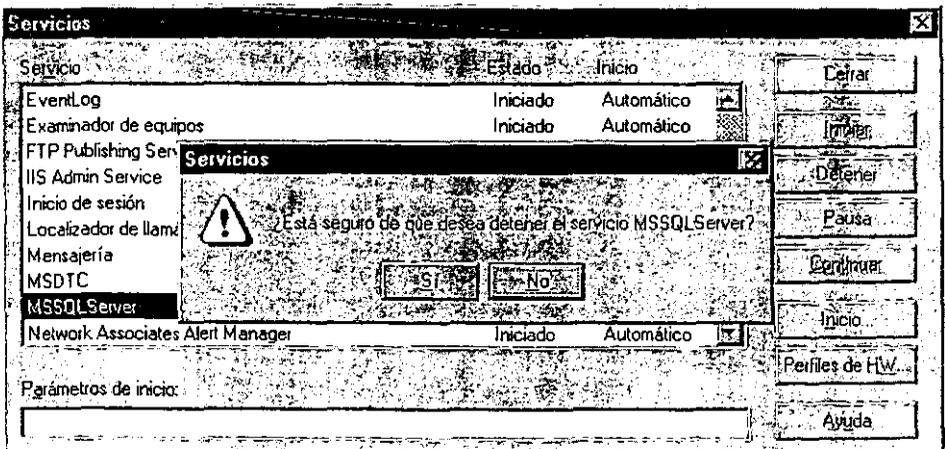


Figura 1. Ilustra como detener los servicios de SQL Server desde el panel de control.



Para detener los servicios de SQL Server por medio de la herramienta SQL Enterprise Manager se deben seguir los siguientes pasos:

- Abrir la herramienta SQL Enterprise Manager (figura 2).

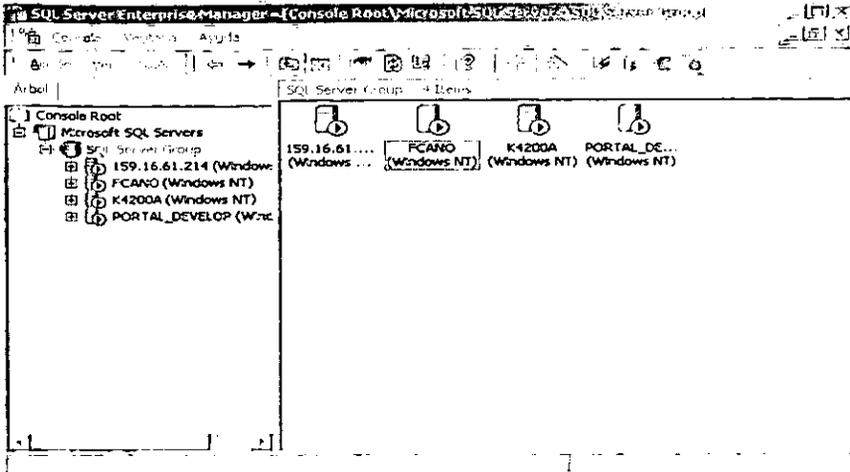


Figura 2. SQL Enterprise Manager.

- Detener los servicios en el servidor deseado (figura3).

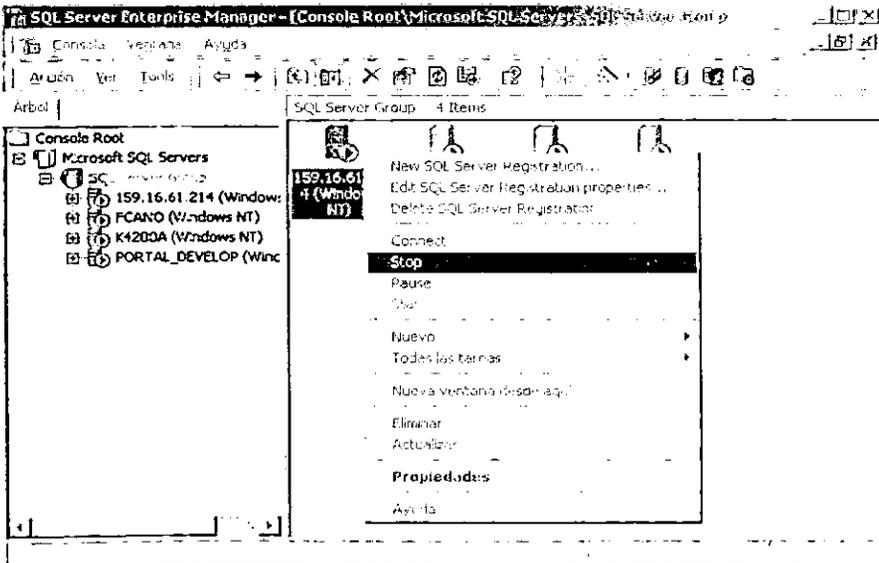


Figura 3. Ilustra como se detienen los servicios de un servidor en particular.



Si se desea detener los servicios a través del Service Manager se debe hacer lo siguiente:

- Abrir el Service Manager (figura 4).

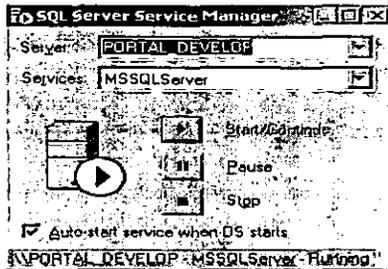


Figura 4. Service Manager.

- Seleccionar el servidor que deseamos detener y oprimir el botón Stop.

### Respaldo una base de datos.

El realizar el respaldo de una base de datos será de suma importancia, pues cuando se presente alguna contingencia en el sistema, tendremos una copia disponible para recuperar la base de datos. Para hacer un respaldo de una base de datos debe hacerse lo siguiente:

- Abrir el SQL Enterprise Manager.
- Crear un dispositivo de respaldo (figura 5).

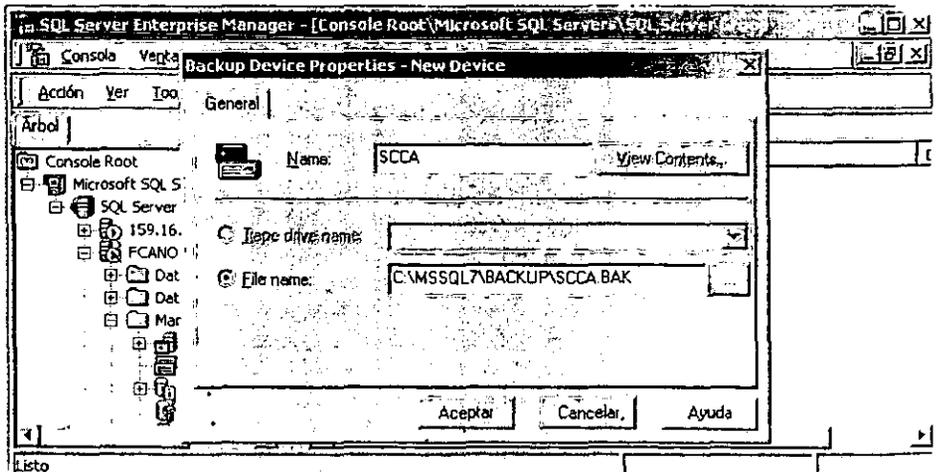


Figura 5. Creación de un dispositivo de respaldo.



- Una vez creado el dispositivo de respaldo, se procede a realizar el respaldo de la base de datos. Un dispositivo de respaldo se puede dirigir hacia un disco o hacia una unidad de cinta magnética.
- Debemos seleccionar la base de datos que deseamos respaldar, para este caso es la base llamada SCCA.
- Seleccionar la opción Backup Database... en el menú Tools. (figura 6).

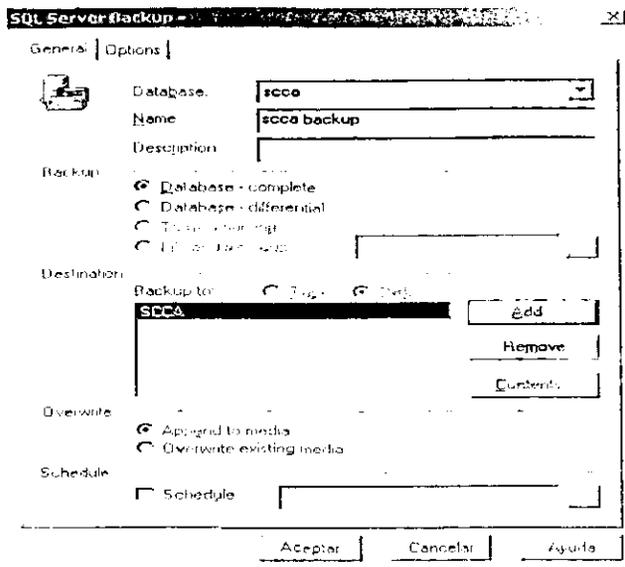


Figura 6. SQL Server Backup.

- Seleccionar la base de datos a respaldar
- Seleccionar el dispositivo en el cuál se va a respaldar.
- Oprimir el botón Aceptar.

### Importación y exportación de una Base de Datos.

La importación y exportación de una base de datos es muy importante, gracias a esta herramienta se pueden transferir los datos de una base de datos a otra, la transferencia puede ser tabla por tabla, sin índices o bien la transferencia de una base completa, claro esta que entre más opciones se elijan más tardado será el proceso. Para llevar a cabo la importación o exportación de una base de datos se deben realizar los siguientes pasos:

- Abrir el SQL Enterprise Manager.
- Seleccionar la opción Import Data... o Export Data... como se muestra en la figura 7.

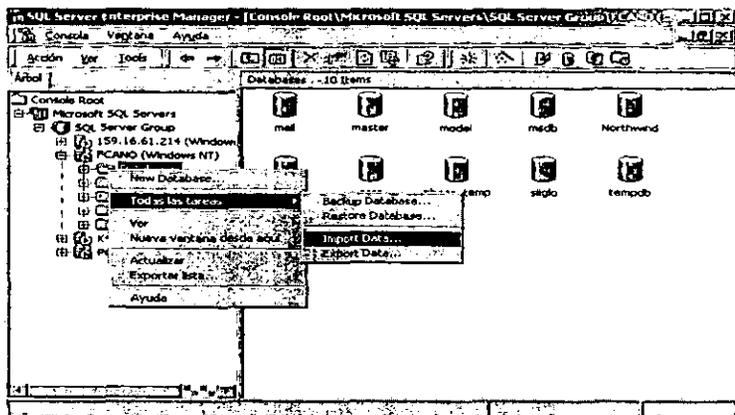


Figura 7. Proceso para importar o exportar datos.

- Posteriormente debemos seguir los pasos que nos indica el Wizard que tiene incluido SQL Server para hacer este tipo de operaciones, como se muestra en la Figura(8).

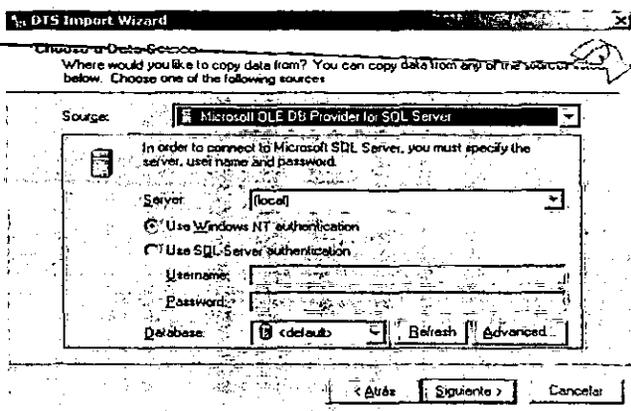


Figura 8. Una de las pantallas del Wizard de importaci3n de datos.

- Al finalizar el Wizard nos indicar3 que la operaci3n fue llevada a cabo con 3xito o si existieron algunos errores.

### Restauraci3n de una base de datos.

La restauraci3n de una base de datos consiste en restablecer una base de datos de la cual se hizo con anterioridad un respaldo a partir de un archivo que se encuentre en un medio magn3tico ya sea un disco duro o un cinta. Para llevar a cabo la restauraci3n de una base de datos se deben seguir los siguientes pasos:



- Abrir el SQL Enterprise Manager.
- Seleccionar la base de datos a restaurar y elegir Restore Database... (Figura 9).

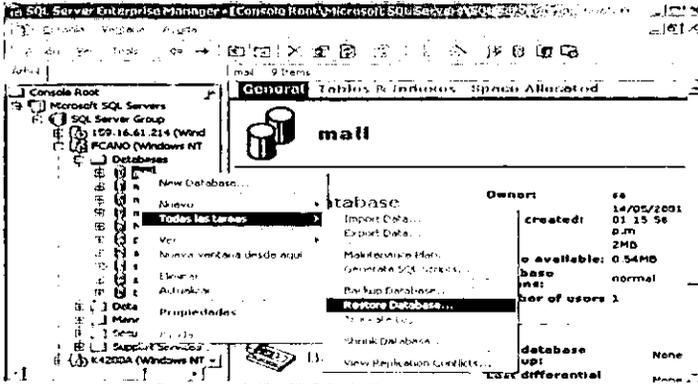


Figura 9. Selección de la opción Restore Database.

- Seleccionar el tipo de dispositivo así como la ubicación del dispositivo (figura 10).

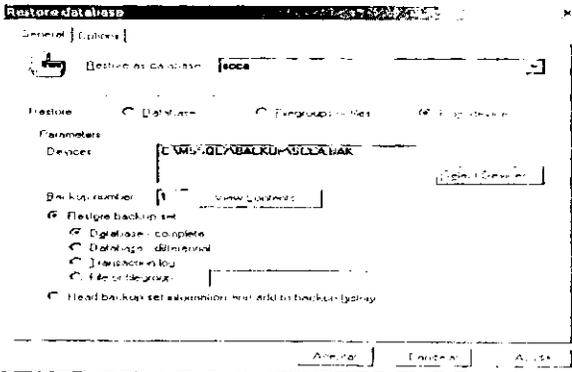


Figura 10. Ventana que nos permite restaurar una base de datos.

- Por último debemos oprimir el botón Aceptar y esperar los resultados, estos pueden ser satisfactorios o erróneos en caso de existir alguna dificultad para llevar a cabo la restauración.

### Calendarización de tareas mediante el Job Scheduling.

En algunas ocasiones es conveniente calendarizar tareas repetitivas o bien tareas que en el día no puedan llevarse a cabo para no saturar los recursos del sistema.

La opción Job Scheduling nos permite calendarizar las tareas requeridas por el administrador del sistema. Para calendarizar una tarea se debe hacer lo siguiente:



- Seleccionar la opción Job Scheduling... en el menú Tools (figura 11).

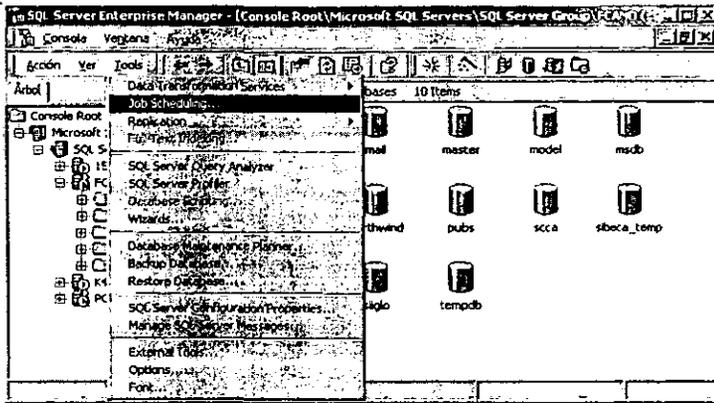


Figura 11. Selección del Wizard para calendarizar tareas.

- Seleccionar el tipo de tarea a calendarizar.
- Definir la acción a realizar.
- Definir la frecuencia de la tarea (figura 12).

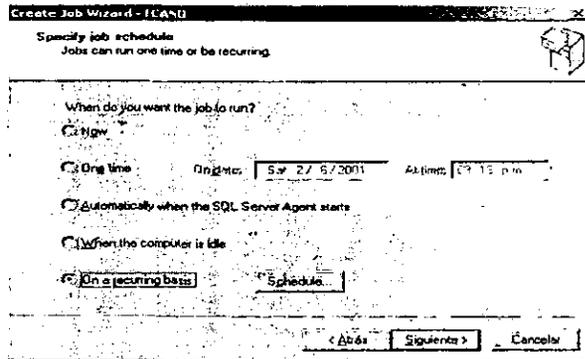


Figura 12. Selección de la frecuencia de ejecución de una tarea.

- Por ultimo guardar la tarea calendarizada oprimiendo el botón Finalizar (figura 13).

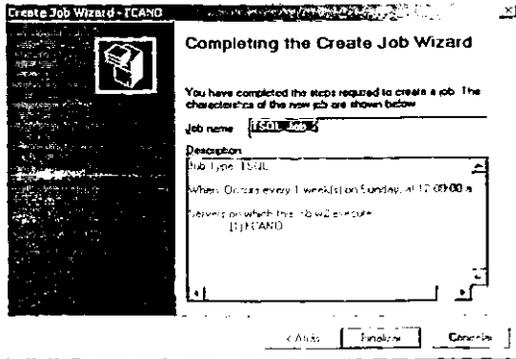


Figura 13.



## APÉNDICE C. CÓDIGO DEL SISTEMA

```
IF EXISTS (SELECT name FROM
sysobjects
WHERE name =
'sp_passwd' AND type = 'P')
DROP PROCEDURE sp_passwd
GO

CREATE PROCEDURE sp_passwd
@user CHAR (8),

@pass CHAR (8)

AS
BEGIN

UPDATE cursio
SET pssword_cursio = @pass
WHERE cve_cursio = @user

IF @@ERROR <> 0 GOTO
errores
errores:
ROLLBACK
TRANSACTION
SELECT 0
END
GO
GRANT ALL ON PROCEDURE TO
PUBLIC
GO

IF EXISTS (SELECT name FROM
sysobjects
WHERE name =
'sp_status' AND type = 'P')
DROP PROCEDURE sp_status
GO

CREATE PROCEDURE sp_status
@poliza CHAR(13),

@endoso INTEGER,

@tpo_rec SMALLINT,

@prima DECIMAL(12,2),

@fecha CHAR(6),

@cve_edo SMALLINT,

@causa VARCHAR(100),

@cve_usrio CHAR (8)
AS
DECLARE @rechazo SMALLINT

BEGIN
SELECT @rechazo=
trcbo_amprdo.ultimo_rchzo_amprdo
FROM trcbo_amprdo
WHERE
trcbo_amprdo.nmro_plza_amprdo =
@poliza

AND
trcbo_amprdo.nmro_cmbio_amprdo
= @endoso

AND
trcbo_amprdo.cve_ctpo_rcbo =
@tpo_rec

AND
trcbo_amprdo.imprte_prma_amprdo
= @prima

AND
trcbo_amprdo.fcha_vncmnto_amprd
= @fecha

AND
trcbo_amprdo.ultimo_rchzo_amprdo
= @rechazo)
GOTO errores
END
INSERT
INTO hrcbo_mvnmnto
VALUES (@poliza,
@endoso, @tpo_rec, @prima,
@fecha, @rechazo,
@cve_edo,
GETDATE(), DATEPART('hh','mi'),
@causa, @cve_usrio)

IF NOT EXISTS(SELECT
trcbo_amprdo.ultimo_rchzo_amprdo
FROM trcbo_amprdo
WHERE
trcbo_amprdo.nmro_plza_amprdo =
@poliza

AND
trcbo_amprdo.nmro_cmbio_amprdo
= @endoso

AND
trcbo_amprdo.cve_ctpo_rcbo =
@tpo_rec

AND
trcbo_amprdo.imprte_prma_amprdo =
@prima

AND
trcbo_amprdo.fcha_vncmnto_amprd =
@fecha)

IF @@ERROR <> 0 GOTO
errores
errores:
ROLLBACK
TRANSACTION
SELECT 0
END
GO
GRANT ALL ON PROCEDURE TO
PUBLIC
GO

CREATE PROCEDURE
sp_amprdo_reg @n1 SMALLINT:
@n2
SMALLINT,
@n3
SMALLINT,
@n4
SMALLINT,
@otiana
INTEGER,
@ramo CHAR(2),
@uen SMALLINT,
@sigtes SMALLINT,
@moneda SMALLINT,
@descrip CHAR(30),
@clave INTEGER
AS
DECLARE @cve_inter INTEGER,
@cve_moneda SMALLINT,
@des_inter CHAR (30),
@des_moneda CHAR (30),
@rec_personas INTEGER,
@tot_personas DECIMAL(20,2),
@rec_empresas INTEGER,
@tot_empresas DECIMAL(20,2),
@rec_total INTEGER,
@tot_total DECIMAL(20,2),
@i SMALLINT
BEGIN
SET @i = 0
IF @sigtes = 0
BEGIN
IF @uen IS NULL
BEGIN
IF @ramo IS NULL
BEGIN
DECLARE c_amprdo
CURSOR
```



DISEÑO E IMPLANTACIÓN DE UN  
SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
FOR select
a.clve_cmnda,p.clve_cintrmdrio,
p.dscrpcion_cintrmdr,m.dscrpcion_c
mnda,
count(*) ,sum(a.imprte_prma_amprdo
)
from cintrmdrio
l,cintrmdrio p, cmnda m,
cofcna o,
trcbo_amprdo a
where l.clve_cintrmdrio
= a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and
p.clve_cintrmdrio =
i.nmro_pmrtria_admns
and
p.clve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.clve_cmnda
= a.clve_cmnda
and o.clve_cofcna
= p.clve_cofcna
and o.clve_cofcna
= @oficina
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
group by
a.clve_cmnda,p.clve_cintrmdrio,
p.dscrpcion_cintrmdr,m.dscrpcion_c
mnda
order by
a.clve_cmnda,p.dscrpcion_cintrmdr,
p.clve_cintrmdrio

OPEN c_amprdo
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
WHILE
@@FETCH_STATUS=0 AND @i <=
100
BEGIN
select @rec_personas =
count(*) , @tot_personas =
sum(a.imprte_prma_amprdo)
from cintrmdrio i,
cintrmdrio p, crmo r,
trcbo_amprdo
a
where i.clve_cintrmdrio =
a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and p.clve_cintrmdrio
= i.nmro_pmrtria_admns
and
p.clve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.clve_cmnda
= a.clve_cmnda
and o.clve_cofcna
= p.clve_cofcna
and r.clve_crmo
= @oficina
and
o.clve_cofcna
= @ramo
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
group by
a.clve_cmnda,p.clve_cintrmdrio,
p.dscrpcion_cintrmdr,m.dscrpcion_c
mnda
order by
a.clve_cmnda,p.dscrpcion_cintrmdr,
p.clve_cintrmdrio
OPEN c_pndnte
FETCH NEXT FROM
c_pndnte
into @cve_moneda,
@cve_inter, @dcs_inter,
@des_moneda,
@rec_total,
@tot_total
WHILE
@@FETCH_STATUS=0 AND @i <=
100
BEGIN
select @rec_personas =
count(*) , @tot_personas =
sum(a.imprte_prma_amprdo)
from cintrmdrio i,
cintrmdrio p, crmo r,
trcbo_amprdo
a
where i.clve_cintrmdrio =
a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and p.clve_cintrmdrio
= i.nmro_pmrtria_admns
and
p.clve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.clve_cmnda
= a.clve_cmnda
and r.clve_crmo
= @oficina
and
o.clve_cofcna
= @ramo
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
group by
a.clve_cmnda,p.clve_cintrmdrio,
p.dscrpcion_cintrmdr,m.dscrpcion_c
mnda
order by
a.clve_cmnda,p.dscrpcion_cintrmdr,
p.clve_cintrmdrio
END
CLOSE c_amprdo
DEALLOCATE
c_amprdo
END
ELSE
DECLARE c_pndnte CURSOR
FOR select
a.clve_cmnda,p.clve_cintrmdrio,
p.dscrpcion_cintrmdr,m.dscrpcion_c
mnda,
count(*) ,sum(a.imprte_prma_amprdo
)
from cintrmdrio
l,cintrmdrio p, cmnda m,
cofcna o,
trcbo_amprdo a
where i.clve_cintrmdrio
= a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and
p.clve_cintrmdrio =
i.nmro_pmrtria_admns
and
p.clve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.clve_cmnda
= a.clve_cmnda
and o.clve_cofcna
= p.clve_cofcna
and r.clve_crmo
= @oficina
and
o.clve_cofcna
= @ramo
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
group by
a.clve_cmnda,p.clve_cintrmdrio,
p.dscrpcion_cintrmdr,m.dscrpcion_c
mnda
order by
a.clve_cmnda,p.dscrpcion_cintrmdr,
p.clve_cintrmdrio
END
CLOSE c_pndnte
DEALLOCATE
c_pndnte
END
```



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
@rec_empresas, @tot_empresas,
@rec_total,
@tot_total
SET @i = @i + 1
FETCH NEXT FROM
c_pndnte
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
END
CLOSE c_pndnte
DEALLOCATE c_pndnte
END
ELSE
IF @ramo IS NULL
begin
DECLARE c_amprdo
CURSOR
FOR select
a.cve_cmnda, p.cve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda,
count(*) sum(a.imprte_prma_amprdo
)
from cintrmdrio
i, cintrmdrio p, cmnda m,
cofcna o, crmo r,
trcbo_amprdo a
where i.cve_cintrmdrio
= a.cve_cintrmdrio
and
i.cve_ctpo_intrmdri =
a.cve_ctpo_intrmdri
and
p.cve_cintrmdrio =
i.nmro_prmtria_adms
and
p.cve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.cve_cmnda
= a.cve_cmnda
and
r.cve_crmo = a.cve_crmo
and o.cve_cofcna
= p.cve_cofcna
and
r.cve_crmo = @ramo
and
r.cve_cuen = @uen
and
a.cve_cestdo_rcho in (@n1, @n2,
@n3, @n4)
group by
a.cve_cmnda, p.cve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda
order by
a.cve_cmnda, p.dscrpcion_cintrmdr,
p.cve_cintrmdrio
OPEN c_amprdo
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total, @tot_total
WHILE
@@FETCH_STATUS = 0 AND @i
<= 100
BEGIN
if @uen = 0
begin
select @rec_personas =
count(*), @tot_personas =
sum(a.imprte_prma_amprdo)
from cintrmdrio p,
cintrmdrio i, crmo r,
trcbo_amprdo a
where i.cve_cintrmdrio =
a.cve_cintrmdrio
and
i.cve_ctpo_intrmdri =
a.cve_ctpo_intrmdri
and
p.cve_cintrmdrio =
i.nmro_prmtria_adms
and
p.cve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and r.cve_crmo
= a.cve_crmo
and r.cve_cuen
= 0
and a.cve_cmnda
= @cve_moneda
and
p.cve_cintrmdrio = @cve_inter
and
a.cve_cestdo_rcho in (@n1, @n2,
@n3, @n4)
end
else
set @rec_personas =
0
if @rec_personas = 0
set @tot_personas = 0
set @rec_empresas =
@rec_total - @rec_personas
set @tot_empresas =
@tot_total - @tot_personas
select @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_personas, @tot_personas,
@rec_empresas, @tot_empresas,
@rec_total,
@tot_total
SET @i = @i + 1
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
END
CLOSE c_amprdo
DEALLOCATE c_amprdo
```

```
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
WHILE
@@FETCH_STATUS = 0 AND @i
<= 100
BEGIN
if @uen = 0
begin
select @rec_personas =
count(*), @tot_personas =
sum(a.imprte_prma_amprdo)
from cintrmdrio p,
cintrmdrio i, crmo r,
trcbo_amprdo a
where i.cve_cintrmdrio =
a.cve_cintrmdrio
and
i.cve_ctpo_intrmdri =
a.cve_ctpo_intrmdri
and
p.cve_cintrmdrio =
i.nmro_prmtria_adms
and
p.cve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and r.cve_crmo
= a.cve_crmo
and r.cve_cuen
= 0
and a.cve_cmnda
= @cve_moneda
and
p.cve_cintrmdrio = @cve_inter
and
a.cve_cestdo_rcho in (@n1, @n2,
@n3, @n4)
end
else
set @rec_personas =
0
if @rec_personas = 0
set @tot_personas = 0
set @rec_empresas =
@rec_total - @rec_personas
set @tot_empresas =
@tot_total - @tot_personas
select @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_personas, @tot_personas,
@rec_empresas, @tot_empresas,
@rec_total,
@tot_total
SET @i = @i + 1
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
END
CLOSE c_amprdo
DEALLOCATE c_amprdo
```

```
END
END
ELSE
DECLARE c_amprdo
CURSOR
FOR select
a.cve_cmnda, p.cve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda,
count(*) sum(a.imprte_prma_amprdo
)
from cintrmdrio
i, cintrmdrio p, cmnda m,
cofcna o, crmo r,
trcbo_amprdo a
where i.cve_cintrmdrio
= a.cve_cintrmdrio
and
i.cve_ctpo_intrmdri =
a.cve_ctpo_intrmdri
and
p.cve_cintrmdrio =
i.nmro_prmtria_adms
and
p.cve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.cve_cmnda
= a.cve_cmnda
and
r.cve_crmo = a.cve_crmo
and o.cve_cofcna
= p.cve_cofcna
and
r.cve_crmo = @ramo
and
r.cve_cuen = @uen
and
a.cve_cestdo_rcho in (@n1, @n2,
@n3, @n4)
group by
a.cve_cmnda, p.cve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda
order by
a.cve_cmnda, p.dscrpcion_cintrmdr,
p.cve_cintrmdrio
OPEN c_amprdo
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total, @tot_total
WHILE
@@FETCH_STATUS = 0 AND @i
<= 100
BEGIN
if @uen = 0
begin
select @rec_personas =
count(*), @tot_personas =
sum(a.imprte_prma_amprdo)
from cintrmdrio p,
cintrmdrio i, crmo r,
```





DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
and o.clve_cofcna
= p.clve_cofcna
and o.clve_cofcna
= @oficina
and
a.clve_crho = @ramo
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
and ((a.clve_cmnda > @moneda)
or(a.clve_cmnda = @moneda and
p.dscrpcion_cintrmdr > @descrip)
or(a.clve_cmnda = @moneda and
p.dscrpcion_cintrmdr = @descrip
and
p.clve_cintrmdrio > @clave))
group by
a.clve_cmnda, p.clve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda
order by
a.clve_cmnda, p.dscrpcion_cintrmdr,
p.clve_cintrmdrio
OPEN c_amprdo
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
WHILE
@@FETCH_STATUS = 0 AND @i
<= 100
BEGIN
select @rec_personas =
count(*), @tot_personas =
sum(a.imprte_prma_amprdo)
from cintmrdio i,
cintrmdrio p, crmo r,
trcbo_amprdo
a
where i.clve_cintrmdrio =
a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and p.clve_cintrmdrio
= i.nmro_prmtria_adms
and
p.clve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and r.clve_crho
= a.clve_crho
and r.clve_cuen
= 0
and a.clve_cmnda
= @cve_moneda
and p.clve_cintrmdrio
= @cve_inter
and a.clve_crho
= @ramo
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
if @rec_personas = 0
Set @tot_personas = 0
```

```
set @rec_empresas =
@rec_total - @rec_personas
set @tot_empresas =
@tot_total - @tot_personas
select @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_personas, @tot_personas,
@rec_empresas, @tot_empresas,
@rec_total,
@tot_total
SET @i = @i + 1
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
END
CLOSE c_amprdo
DEALLOCATE
c_amprdo
END
ELSE
IF @ramo IS NULL
BEGIN
DECLARE c_amprdo
CURSOR
FOR select
a.clve_cmnda, p.clve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda,
count(*), sum(a.imprte_prma_amprdo
)
from cintmrdio
i, cintmrdio p, cmnda m,
cofcna o, crmo r,
trcbo_amprdo a
where i.clve_cintrmdrio
= a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and
p.clve_cintrmdrio =
i.nmro_prmtria_adms
and
p.clve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.clve_cmnda
= a.clve_cmnda
and
r.clve_crho = a.clve_crho
and o.clve_cofcna
= p.clve_cofcna
and o.clve_cofcna
= @oficina
and
r.clve_cuen = @uen
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
and ((a.clve_cmnda > @moneda)
or(a.clve_cmnda = @moneda and
p.dscrpcion_cintrmdr > @descrip)
```

```
or(a.clve_cmnda = @moneda and
p.dscrpcion_cintrmdr = @descrip
and
p.clve_cintrmdrio > @clave))
group by
a.clve_cmnda, p.clve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda
order by
a.clve_cmnda, p.dscrpcion_cintrmdr,
p.clve_cintrmdrio
OPEN c_amprdo
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
WHILE
@@FETCH_STATUS = 0 AND @i
<= 100
BEGIN
if @uen = 0
begin
select @rec_personas =
count(*), @tot_personas =
sum(a.imprte_prma_amprdo)
from cintmrdio p,
cintrmdrio i, crmo r,
trcbo_amprdo a
where i.clve_cintrmdrio
= a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and
p.clve_cintrmdrio =
i.nmro_prmtria_adms
and
p.clve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and r.clve_crho
= a.clve_crho
and r.clve_cuen
= 0
and a.clve_cmnda
= @cve_moneda
and
p.clve_cintrmdrio = @cve_inter
and
a.clve_cestdo_rcbo in (@n1, @n2,
@n3, @n4)
end
else
set @rec_personas =
0
if @rec_personas = 0
set @tot_personas = 0
set @rec_empresas =
@rec_total - @rec_personas
set @tot_empresas =
@tot_total - @tot_personas
select @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_personas, @tot_personas,
```



## DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
@rec_empresas, @tot_empresas,
@rec_total,
@tot_total
SET @i = @i + 1
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
END
CLOSE c_amprdo
DEALLOCATE c_amprdo
ELSE
DECLARE c_amprdo
CURSOR
FOR select
a.cve_cmnda, p.cve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda,
count(*) sum(a.imprte_prma_amprdo
)
from cintrmdrio
r, cintrmdrio p, cmnda m,
cofcna o, crmo r,
trcbo_amprdo a
where i.cve_cintrmdrio
= a.cve_cintrmdrio
and
i.cve_ctpo_intrmdri =
a.cve_ctpo_intrmdri
p.cve_cintrmdrio =
i.nmro_pmrtria_admsn
and
p.cve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and m.cve_cmnda
= a.cve_cmnda
and
r.cve_crmo = a.cve_crmo
and o.cve_cofcna
= p.cve_cofcna
and o.cve_cofcna
= @oficina
and
r.cve_crmo = @ramo
and
r.cve_cuen = @uen
and
a.cve_cestdo_rcho in (@n1, @n2,
@n3, @n4)
and ((a.cve_cmnda > @moneda)
or (a.cve_cmnda = @moneda and
p.dscrpcion_cintrmdr > @descrip)
or (a.cve_cmnda = @moneda and
p.dscrpcion_cintrmdr = @descrip
and p.cve_cintrmdrio > @clave))
group by
a.cve_cmnda, p.cve_cintrmdrio,
p.dscrpcion_cintrmdr, m.dscrpcion_c
mnda
```

```
order by
a.cve_cmnda, p.dscrpcion_cintrmdr,
p.cve_cintrmdrio
OPEN c_amprdo
FETCH NEXT FROM
c_amprdo
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
WHILE
@@FETCH_STATUS = 0 AND @i
<= 100
BEGIN
if @uen = 0
begin
select @rec_personas =
count(*), @tot_personas =
sum(a.imprte_prma_amprdo)
from cintrmdrio p,
cintrmdrio i, crmo r,
trcbo_amprdo a
where i.cve_cintrmdrio =
a.cve_cintrmdrio
and
i.cve_ctpo_intrmdri =
a.cve_ctpo_intrmdri
and
p.cve_cintrmdrio =
i.nmro_pmrtria_admsn
and
p.cve_ctpo_intrmdri =
i.tpo_intrmdrio_spr
and r.cve_crmo
= a.cve_crmo
and r.cve_cuen
= 0
and r.cve_cuen
= 0
and a.cve_cmnda
= @cve_moneda
and
p.cve_cintrmdrio = @cve_inter
and
a.cve_cestdo_rcho in (@n1, @n2,
@n3, @n4)
end
else
set @rec_personas = 0
if @rec_personas = 0
set @tot_personas = 0
set @rec_empresas =
@rec_total - @rec_personas
set @tot_empresas =
@tot_total - @tot_personas
select @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_personas, @tot_personas,
@rec_empresas, @tot_empresas,
@rec_total,
@tot_total
SET @i = @i + 1
FETCH NEXT FROM
c_amprdo
```

```
into @cve_moneda,
@cve_inter, @des_inter,
@des_moneda,
@rec_total,
@tot_total
END
CLOSE c_amprdo
DEALLOCATE
c_amprdo
end
GO
GRANT ALL ON PROCEDURE TO
PUBLIC
GO
IF EXISTS (SELECT name FROM
sysobjects
WHERE name =
'sp_incon_dcr' AND type = 'P')
DROP PROCEDURE
sp_incon_dcr
GO
CREATE PROCEDURE
sp_incon_dcr @error smallint,
@ramo char (2),
@uen smallint,
@sigtes smallint,
@moneda smallint,
@poliza char (13),
@endoso integer,
@tipo smallint,
@prima decimal (12,2),
@fecha char (6)
as
DECLARE @cve_moneda
smallint,
@cve_tpo_rec smallint,
@cve_ramo char (2),
@cve_archivo smallint,
@cve_error smallint,
@cve_for_pago smallint,
@cve_ongen char (1),
@num_poliza char (13),
@cve_region integer,
@des_region char (30),
@num_promo char (5),
@des_promo char (30),
@num_agente char (5),
@num_cambio integer,
@des_tpo_rec char (30),
@imp_prima decimal (12,2),
@fec_vento char (6),
@des_moneda char (30),
@des_ramo char (30),
@nom_aseg char (30),
@des_for_pago char (30),
@des_ongen char (30),
@fec_movto datetime,
```



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```

@des_error char (30),
@des_archivo char (30),
@i SMALLINT
SET @i = 0
IF @sigtes = 0
begin
  IF @uen IS NULL
  begin
    IF @ramo IS NULL
    begin
      DECLARE c_incon CURSOR
      FOR select m.cive_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
t.cive_ctpo_rcbo,
a.imprte_prma_incsst,
a.fcha_vncmnto_incs,
r.cive_crmo,
x.cive_carchvo, e.cive_cerror,
f.cive_cfma_pgo,
o.cive_corgen,
a.nmro_pmtor_incsst,
a.nmro_agnte_incsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmo,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_mvmtnto_incsst,
a.nmbre_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda m,
crmo r,cfma_pgo f,ctpo_rcbo t,
corgen o,
cerror
e. carchvo x, trcbo_incsstnte a
where
m.cive_cmnda = a.cive_cmnda
and
r.cive_crmo = a.cive_crmo
and
f.cive_cfma_pgo =
a.cive_cfma_pgo
and
t.cive_ctpo_rcbo =
a.cive_ctpo_rcbo
and
o.cive_corgen = a.cive_corgen
and
e.cive_cerror = a.cive_cerror
and
x.cive_carchvo = a.cive_carchvo
and
e.cive_cerror = 2
order by
m.cive_cmnda,a.nmro_plza_incsstn

```

```

a.nmro_cmbio_incsst,t.cive_ctpo_rc
bo,
a.imprte_prma_incsst,a.fcha_vncmnt
to_incsnt
SET @cve_region =
NULL
SET @des_region =
NULL
SET @des_promo =
NULL
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
WHILE
@@FETCH_STATUS=0 and
@i<=100
BEGIN
select @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza,
@des_region,
@num_promo,
@des_promo,
@num_agente, @num_cambio,
@des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,

```

```

@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE c_incon
IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.cive_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
t.cive_ctpo_rcbo,
a.imprte_prma_incsst,
a.fcha_vncmnto_incs,
r.cive_crmo, x.cive_carchvo,
e.cive_cerror,
f.cive_cfma_pgo, o.cive_corgen,
l.cive_cofcna, l.dscrpcion_cofcna,
i.cive_cintrmdrio,
i.dscrpcion_cintrmdr,
a.nmro_agnte_incsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmo,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_mvmtnto_incsst,
a.nmbre_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda m, crmo
r,cfma_pgo f,ctpo_rcbo t, corgen o,
cofcna l, cintrmdrio i, cerror e,
carchvo x,
trcbo_incsstnte a
where
m.cive_cmnda = a.cive_cmnda
and
r.cive_crmo = a.cive_crmo
and
f.cive_cfma_pgo =
a.cive_cfma_pgo
and
t.cive_ctpo_rcbo =
a.cive_ctpo_rcbo
and
o.cive_corgen = a.cive_corgen
and
e.cive_cerror = a.cive_cerror
and
x.cive_carchvo = a.cive_carchvo

```



## DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```

i.clve_cintmdrio = and
a.clve_cintmdrio

i.clve_ctpo_intmdrio =
a.clve_ctpo_intmdrio

l.clve_cofcna = i.clve_cofcna
and
e.clve_cerror = 1
order by
m.clve_cmnda,a.nmro_pliza_incsstn

a.nmro_cmbio_incsst,t.clve_ctpo_rc
bo,

a.imprte_prma_incsst,a.fcha_vncmn
to_incs

OPEN c_incon
FETCH NEXT FROM
c_incon
into
@clve_moneda, @num_poliza,
@num_cambio, @cve_tpo_rec,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo, @des_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE
c_incon
END
ELSE
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_pliza_incsstn,
t.clve_ctpo_rcbo,
a.imprte_prma_incsst,
a.fcha_vncmnto_incs,
r.clve_crmo,
x.clve_carchvo, e.clve_cerror,
f.clve_cfrma_pgo,
o.clve_corgen,
a.nmro_prmtor_incsst,
a.nmro_agnte_incsst,
l.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmo,
f.dscrpcion_cfrma_pg,
o.dscrpcion_corgen,
a.fcha_mvnmnto_incsst,
a.nmbre_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda m,
crmo r, cfrma_pgo f, ctpo_rcbo t,
corgen o,
cerror
e. carchvo x, trcbo_incsstnte a
where
m.clve_cmnda = a.clve_cmnda
and
f.clve_cfrma_pgo =
a.clve_cfrma_pgo

into
@clve_moneda, @num_poliza,
@num_cambio, @cve_tpo_rec,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo, @des_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE
c_incon
END
ELSE
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_pliza_incsstn,
t.clve_ctpo_rcbo,
a.imprte_prma_incsst,
a.fcha_vncmnto_incs,
r.clve_crmo,
x.clve_carchvo, e.clve_cerror,
f.clve_cfrma_pgo,
o.clve_corgen,
a.nmro_prmtor_incsst,
a.nmro_agnte_incsst,
l.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmo,
f.dscrpcion_cfrma_pg,
o.dscrpcion_corgen,
a.fcha_mvnmnto_incsst,
a.nmbre_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda m,
crmo r, cfrma_pgo f, ctpo_rcbo t,
corgen o,
cerror
e. carchvo x, trcbo_incsstnte a
where
m.clve_cmnda = a.clve_cmnda
and
f.clve_cfrma_pgo =
a.clve_cfrma_pgo

t.clve_ctpo_rcbo = and
a.clve_ctpo_rcbo
and
o.clve_corgen = a.clve_corgen
and
e.clve_cerror = a.clve_cerror
and
x.clve_carchvo = a.clve_carchvo
and
r.clve_crmo = a.clve_crmo
and
r.clve_crmo = @ramo
and
e.clve_cerror = 2
order by
m.clve_cmnda,a.nmro_pliza_incsstn

a.nmro_cmbio_incsst,t.clve_ctpo_rc
bo,

a.imprte_prma_incsst,a.fcha_vncmn
to_incs
SET @cve_region = NULL
SET @des_region =
NULL
SET @des_promo =
NULL

OPEN c_incon
FETCH NEXT FROM
c_incon
into
@clve_moneda, @num_poliza,
@num_cambio, @cve_tpo_rec,
@imp_prima, @fec_vento
@cve_ramo, @cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@des_region,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error,
@des_archivo
WHILE
@i=@FETCH_STATUS=0 and
@i<=100
BEGIN
select @clve_moneda,
@clve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza, @cve_region,
@des_region, @num_promo,
@des_promo, @num_agente,
@num_cambio, @des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon

```





DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```

a.nmro_pmrtr_incnss,
a.nmro_agnte_incnss,

l.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,

r.dscrpcion_cmno,
f.dscrpcion_cfma_pg,

o.dscrpcion_corgen,
a.fcha_mvmento_incnss,

a.nmbre_asgrdo_incn,
e.dscrpcion_cerror,

x.dscrpcion_carchvo
from cmnda m,
cmno r,cfma_pgo f,ctpo_rcb t,
corgen o,

cerror
e,carchvo x, trcbo, incnss into a
where
m.clve_cmnda = a.clve_cmnda
and
f.clve_cfma_pgo =
a.clve_cfma_pgo
and
t.clve_ctpo_rcb =
a.clve_ctpo_rcb
and
o.clve_corgen = a.clve_corgen
and
e.clve_cerror = a.clve_cerror
and
x.clve_carchvo = a.clve_carchvo
and
r.clve_cmno = a.clve_cmno
and
f.clve_cfma_pgo = @uen
and
e.clve_cerror = 2
order by
m.clve_cmnda,a.nmro_plza_incnss
,
a.nmro_cmbio_incnss,t.clve_ctpo_rc
bo,

a.imprte_prma_incnss,a.fcha_vncmto
to_incn
SET @cve_region = NULL
SET @des_region =
NULL
SET @des_promo =
NULL
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE c_incon
IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_plza_incnss,
a.nmro_cmbio_incnss,
t.clve_ctpo_rcb,
a.imprte_prma_incnss,
a.fcha_vncmto_incn,
r.clve_cmno, x.clve_carchvo,
e.clve_cerror,
f.clve_cfma_pgo, o.clve_corgen,
l.clve_cofcna, l.dscrpcion_cofcna,

```

```

@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
WHILE
@@FETCH_STATUS = 0 and @i <=
100
BEGIN
SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza,
@cve_region, @des_region,
@num_promo,
@des_promo,
@num_agente, @num_cambio,
@des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE c_incon
IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_plza_incnss,
a.nmro_cmbio_incnss,
t.clve_ctpo_rcb,
a.imprte_prma_incnss,
a.fcha_vncmto_incn,
r.clve_cmno, x.clve_carchvo,
e.clve_cerror,
f.clve_cfma_pgo, o.clve_corgen,
l.clve_cofcna, l.dscrpcion_cofcna,

```

```

i.clve_cintrmdrio,
i.dscrpcion_cintrmdrio,

a.nmro_agnte_incnss,

t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,

r.dscrpcion_cmno,
f.dscrpcion_cfma_pg,

o.dscrpcion_corgen,
a.fcha_mvmento_incnss,

a.nmbre_asgrdo_incn,
e.dscrpcion_cerror,

x.dscrpcion_carchvo
from cmnda
m,cmno r,cfma_pgo f,ctpo_rcb t,
corgen o,

cofcna l, cintrmdrio i, cerror e,
carchvo x,

trcbo_incnss into a
where
m.clve_cmnda = a.clve_cmnda
and
f.clve_cfma_pgo =
a.clve_cfma_pgo
and
t.clve_ctpo_rcb =
a.clve_ctpo_rcb
and
o.clve_corgen = a.clve_corgen
and
e.clve_cerror = a.clve_cerror
and
x.clve_carchvo = a.clve_carchvo
and
i.clve_cintrmdrio =
a.clve_cintrmdrio
and
i.clve_ctpo_intrmdrio =
a.clve_ctpo_intrmdrio
and
l.clve_cofcna = i.clve_cofcna
and
r.clve_cmno = a.clve_cmno
and
f.clve_cfma_pgo = @uen
and
e.clve_cerror = 1
order by
m.clve_cmnda,a.nmro_plza_incnss
,
a.nmro_cmbio_incnss,t.clve_ctpo_rc
bo,

a.imprte_prma_incnss,a.fcha_vncmto
to_incn
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,

```



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_for_pago, @cve_origen,
@des_region, @cve_region,
@num_promo, @des_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@des_error, @des_archivo
@nom_aseg,
@des_error, @des_archivo
WHILE
@@FETCH_STATUS = 0 and @i
<=100
BEGIN
/*BEGIN 14*/
SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza, @cve_region,
@des_region, @num_promo,
@des_promo, @num_agente,
@num_cambio, @des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo, @des_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE c_incon
ELSE
DECLARE c_incon
CURSOR
FOR select m.cve_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmblo_incsst,
t.cve_ctpo_rcbo,
a.imprte_prma_incsst,
a.fcha_vncmnto_incs,
r.cve_crmo,
x.cve_carchvo, e.cve_cerror,
f.cve_cfma_pgo,
o.cve_corgen,
a.nmro_prmtor_incsst,
a.nmro_agnte_incsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmo,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_mvmtnto_incsst,
a.nmbre_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda m,
ctpo_rcbo t,
cerror
where
a.cve_cmnda = a.cve_cmnda
and
a.cve_cfma_pgo =
a.cve_cfma_pgo
and
t.cve_ctpo_rcbo =
a.cve_ctpo_rcbo
and
o.cve_corgen = a.cve_corgen
and
e.cve_cerror = a.cve_cerror
and
x.cve_carchvo = a.cve_carchvo
and
r.cve_crmo = a.cve_crmo
and
@ramo
and
r.cve_cuen = @uen
and
e.cve_cerror = 2
order by
m.cve_cmnda, a.nmro_plza_incsstn,
a.nmro_cmblo_incsst, t.cve_ctpo_rc
bo,
a.imprte_prma_incsst, a.fcha_vncmnt
to_incs
SET @cve_region = NULL
SET @des_region =
NULL
SET @des_promo =
NULL
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
WHILE
@@FETCH_STATUS = 0 and @i <=
100
BEGIN
SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza,
@cve_region, @des_region,
@num_promo,
@des_promo,
@num_agente, @num_cambio,
@des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo, @des_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE c_incon
```



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```

IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
l.clve_ctpo_rcbo,
a.imprte_prma_incss,
a.fcha_vncmnto_incs,
r.clve_crho, x.clve_carchvo,
e.clve_cerror,
f.clve_cfma_pgo, o.clve_corgen,
l.clve_cofcna, l.dscrpcion_cofcna,
i.clve_cintrmdrio,
i.dscrpcion_cintrmdr,
a.nmro_agnte_incsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crho,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_mvmtnto_incss,
a.nmbr_e_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda
m, crmo r, cfma_pgo f, ctpo_rcbo t,
corgen o,
cofcna l, cintrmdrio i, cerror e,
carchvo x,
trcbo_incsstnte a
where
m.clve_cmnda = a.clve_cmnda
and
f.clve_cfma_pgo = a.clve_cfma_pgo
and
l.clve_ctpo_rcbo = a.clve_ctpo_rcbo
and
o.clve_corgen = a.clve_corgen
and
e.clve_cerror = a.clve_cerror
and
x.clve_carchvo = a.clve_carchvo
and
i.clve_cintrmdrio = a.clve_cintrmdrio
and
l.clve_cofcna = i.clve_cofcna
and
r.clve_crho = a.clve_crho

```

```

and
r.clve_crho = @ramo
and
r.clve_cuen = @uen
and
e.clve_cerror = 1
order by
m.clve_cmnda, a.nmro_plza_incsstn
a.nmro_cmbio_incsst, t.clve_ctpo_rcbo,
a.imprte_prma_incss, a.fcha_vncmnto_incs
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prma,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo,
@des_promo,
@num_agente,
@des_tpo_rec, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prma,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo,
@des_promo,
@num_agente,
@des_tpo_rec, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
WHILE
@@FETCH_STATUS = 0 and @@i <=
100
BEGIN
SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza, @cve_region,
@des_region, @num_promo,
@des_promo, @num_agente,
@num_cambio, @des_tpo_rec,
@imp_prma,
@fec_vento, @des_moneda,
@des_ramo,
@nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @@i = @@i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prma,
@fec_vento, @cve_ramo,
@cve_archivo,

```

```

@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo,
@des_promo,
@num_agente,
@des_tpo_rec, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE c_incon
END
END
ELSE
IF @uen IS NULL
BEGIN
IF @ramo IS NULL
BEGIN
IF @error = 2
BEGIN
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
l.clve_ctpo_rcbo,
a.imprte_prma_incss,
a.fcha_vncmnto_incs,
r.clve_crho, x.clve_carchvo,
e.clve_cerror,
f.clve_cfma_pgo, o.clve_corgen,
a.nmro_prmtor_incss,
a.nmro_agnte_incsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crho,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_mvmtnto_incss,
a.nmbr_e_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda
m, crmo r, cfma_pgo f, ctpo_rcbo t,
corgen o,
cerror e, carchvo x, trcbo_incsstnte a
where
m.clve_cmnda = a.clve_cmnda
and
r.clve_crho = a.clve_crho
and
f.clve_cfma_pgo = a.clve_cfma_pgo

```



## DISÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
and
t.clve_ctpo_rcbo =
a.clve_ctpo_rcbo

and
o.clve_corgen = a.clve_corgen
and
e.clve_cerror = a.clve_cerror
and
x.clve_carchvo = a.clve_carchvo
and
e.clve_cerror = 2
and ((m.clve_cmnda > @moneda)
or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn >
@poliza) or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst >
@endoso) or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.clve_ctpo_rcbo >
@tipo) or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.clve_ctpo_rcbo =
@tipo and
a.imprte_prma_incss >
@prima) or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.clve_ctpo_rcbo =
@tipo and
a.imprte_prma_incss =
@prima and a.fcha_vncmnto_incs
> @fecha))
order by
m.clve_cmnda, a.nmro_plza_incsstn
,
a.nmro_cmbio_incsst, t.clve_ctpo_rc
bo,
a.imprte_prma_incss, a.fcha_vncmnt
to_incs
SET @cve_region = NULL
SET @des_region =
NULL
SET @des_promo =
NULL
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@num_promo, @num_agente,
@des_tpo_rec, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
DEALLOCATE c_incon
SET @moneda = 0
IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
t.clve_ctpo_rcbo,
a.imprte_prma_incss,
a.fcha_vncmnto_incs,
r.clve_crmo, x.clve_carchvo,
e.clve_cerror,
f.clve_cfrma_pgo, o.clve_corgen,
```

```
@num_promo, @num_agente,
@des_tpo_rec, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
WHILE
@@FETCH_STATUS = 0 and @i
<= 100
BEGIN
SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza, @cve_region,
@des_region, @num_promo,
@des_promo, @num_agente,
@num_cambio, @des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@num_promo, @num_agente,
@des_tpo_rec, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
DEALLOCATE c_incon
SET @moneda = 0
IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.clve_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
t.clve_ctpo_rcbo,
a.imprte_prma_incss,
a.fcha_vncmnto_incs,
r.clve_crmo, x.clve_carchvo,
e.clve_cerror,
f.clve_cfrma_pgo, o.clve_corgen,
```

```
l.clve_cofcna, l.dscrpcion_cofcna,
i.clve_cintrmdrio,
i.dscrpcion_cintrmdr,
a.nmro_agnte_incsst,
l.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmo,
f.dscrpcion_cfrma_pgo,
o.dscrpcion_corgen,
a.fcha_mvmtnto_incss,
a.nmbre_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda
m, crmo r, cfrma_pgo f, ctpo_rcbo t,
corgen o,
cofcna l, cintrmdrio i, cerror e,
carchvo x,
trbo_incsstnte a
where
m.clve_cmnda = a.clve_cmnda
and
r.clve_crmo = a.clve_crmo
and
f.clve_cfrma_pgo =
a.clve_cfrma_pgo
and
t.clve_ctpo_rcbo =
a.clve_ctpo_rcbo
and
o.clve_corgen = a.clve_corgen
and
e.clve_cerror = a.clve_cerror
and
x.clve_carchvo = a.clve_carchvo
and
i.clve_cintrmdrio =
a.clve_cintrmdrio
and
i.clve_ctpo_intrmdri =
a.clve_ctpo_intrmdri
and
l.clve_cofcna = i.clve_cofcna
and
e.clve_cerror = 1
and ((m.clve_cmnda > @moneda)
or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn >
@poliza) or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst >
@endoso) or
(m.clve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
```



DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```

a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo >
@tipo) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incsst >
@prima) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incsst =
@prima and a.fcha_vncmnto_incs
> @fecha))
order by
m.cve_cmnda,a.nmro_plza_incsstn
,
a.nmro_cmbio_incsst,t.cve_ctpo_rc
bo,
a.imprte_prma_incsst,a.fcha_vncmnt
to_incs
OPEN c_incon
FETCH NEXT
FROM c_incon
into @cve_moneda,
@num_poliza,@num_cambio,
@cve_tpo_rec,
@imp_prima,@fec_vento,
@cve_ramo,@cve_archivo,
@cve_error,
@cve_for_pago,@cve_origen,
@cve_region,@des_region,
@num_promo,@des_promo,
@num_agente,@des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago,@des_origen,
@fec_movto,
@nom_aseg,@des_error,
@des_archivo
END
CLOSE c_incon
DEALLOCATE
c_incon
END
ELSE
IF @error = 2
BEGIN
DECLARE c_incon CURSOR
FOR select m.cve_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
t.cve_ctpo_rcbo,
a.imprte_prma_incsst,
a.fcha_vncmnto_incs,
r.cve_cmno,x.cve_carchvo,
e.cve_cerro,
f.cve_cfma_pgo,o.cve_corgen,
a.nmro_prrtor_incsst,
a.nmro_agnte_incsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_cmno,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_mvnmnto_incsst,
a.nmbre_asgrdo_incs,
e.dscrpcion_cerro,
x.dscrpcion_carchvo
from cmnda
m,cmno r,cfma_pgo f,ctpo_rcbo t,
corgen o,
cerro e,carchvo x,trcbo_incsstnte
a
where
m.cve_cmnda = a.cve_cmnda
and
f.cve_cfma_pgo =
a.cve_cfma_pgo
and
t.cve_ctpo_rcbo =
a.cve_ctpo_rcbo
and
o.cve_corgen = a.cve_corgen
and
e.cve_cerro = a.cve_cerro
and
x.cve_carchvo = a.cve_carchvo
and
r.cve_cmno = a.cve_cmno
and
r.cve_cmno = @ramo
and
e.cve_cerro = 2
and ((m.cve_cmnda > @moneda)
or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn >
@poliza) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst >
@endoso) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo >
@tipo) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incsst >
@prima) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incsst =
@prima and a.fcha_vncmnto_incs
> @fecha))
order by
m.cve_cmnda,a.nmro_plza_incsstn
,
a.nmro_cmbio_incsst,t.cve_ctpo_rc
bo,

```



DESEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
a.imprte_prma_incns,a.fcha_vncmnto_incs
to_incs
SET @cve_region = NULL
SET @des_region = NULL
NULL
SET @des_promo =
NULL
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@cve_vento, @imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_for_pago, @cve_error,
@cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@des_error, @des_archivo
@des_error, @des_archivo
WHILE
@@FETCH_STATUS = 0 and @i <=
100
BEGIN
SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza,
@cve_region, @des_region,
@num_promo,
@des_promo,
@num_agente, @num_cambio,
@des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_for_pago, @cve_error,
@cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
a.imprte_prma_incns,a.fcha_vncmnto_incs
to_incs
SET @cve_region = NULL
SET @des_region = NULL
NULL
SET @des_promo =
NULL
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@cve_vento, @imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_for_pago, @cve_error,
@cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE
c_incon
SET @moneda = 0
IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.cve_cmnda,
a.nmro_plza_incnsstn,
a.nmro_cmbio_incnsst,
t.cve_ctpo_rcbo,
a.imprte_prma_incns,
a.fcha_vncmnto_incs,
r.cve_crmno, x.cve_carchvo,
e.cve_cerror,
f.cve_cfma_pgo, o.cve_corgen,
l.cve_cofcna, l.dscrpcion_cofcna,
l.cve_cintrmdrio,
l.dscrpcion_cintrmdr,
a.nmro_agnte_incnsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmno,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_vncmnto_incns,
a.nmbre_asgrdo_incns,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda
m, crmo r, cfma_pgo f, ctpo_rcbo t,
corgen o,
cofcna l, cintrmdrio l, cerror e,
carchvo x,
trcbo_incnsstnte a
where
m.cve_cmnda = a.cve_cmnda
and
f.cve_cfma_pgo =
a.cve_cfma_pgo
and
t.cve_ctpo_rcbo =
a.cve_ctpo_rcbo
and
o.cve_corgen = a.cve_corgen
and
e.cve_cerror = a.cve_cerror
and
x.cve_carchvo = a.cve_carchvo
and
i.cve_cintrmdrio =
a.cve_cintrmdrio
and
i.cve_ctpo_intrmdrio =
a.cve_ctpo_intrmdrio
and
l.cve_cofcna = i.cve_cofcna
and
r.cve_crmno = a.cve_crmno
and
r.cve_crmno = @ramo
and
e.cve_cerror = 1
and ((m.cve_cmnda > @moneda)
or
(m.cve_cmnda = @moneda
and a.nmro_plza_incnsstn >
@poliza) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst >
@endoso) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst =
@endoso and t.cve_ctpo_rcbo >
@tpo) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst =
@endoso and l.cve_ctpo_rcbo =
@tpo and
a.imprte_prma_incns >
@prima) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incns =
@prima and a.fcha_vncmnto_incs
> @fecha))
order by
m.cve_cmnda,a.nmro_plza_incnsstn
a.nmro_cmbio_incnsst,t.cve_ctpo_rcbo,
a.imprte_prma_incns,a.fcha_vncmnto_incs
to_incs
OPEN c_incon
FETCH NEXT
FROM c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima, @fec_vento,
@cve_ramo, @cve_archivo,
@cve_for_pago, @cve_error,
@cve_origen,
@num_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
```



DISEÑO E IMPLANTACIÓN DE UN  
SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
@num_agente, @des_tpo_rec,
@des_moneda,
    @des_ramo,
@des_for_pago, @des_origen,
@fec_movto,

@nom_aseg, @des_error,
@des_archivo
    WHILE
@i=@FETCH_STATUS = 0 and @i <=
100
    BEGIN
        SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,

@num_poliza, @cve_region,
@des_region, @num_promo,

@des_promo, @num_agente,
@num_cambio, @des_tpo_rec,
    @imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
    @nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
    @des_error,
@des_archivo
    SET @i = @i + 1
    FETCH NEXT
FROM c_incon
    into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,

@imp_prima, @fec_vento,
@cve_ramo, @cve_archivo,

@cve_error, @cve_for_pago,
@cve_origen,

@cve_region, @des_region,

@num_promo, @des_promo,

@num_agente, @des_tpo_rec,
@des_moneda,
    @des_ramo,
@des_for_pago, @des_origen,
@fec_movto,

@nom_aseg, @des_error,
@des_archivo
    END
    CLOSE c_incon
    DEALLOCATE
c_incon
    END
    ELSE
    IF @ramo IS NULL
    BEGIN
        IF @error = 2
        BEGIN
            DECLARE c_incon CURSOR
                FOR select m.cve_cmnda,
a.nmro_plza_incsnstn,

a.nmro_cmbio_incsnst,
t.cve_ctpo_rcbo,

a.imprte_prma_incsns,
a.fcha_vncmnto_incsn,

r.cve_crmo, x.cve_carchvo,
e.cve_cerror,

f.cve_cfrma_pgo, o.cve_corgen,

a.nmro_prmtor_incsns,
a.nmro_agnte_incsnst,

t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,

r.dscrpcion_crmo,
f.dscrpcion_cfrma_pg,

o.dscrpcion_corgen,
a.fcha_mvmtnto_incsns,

a.nmbre_asgrdo_incsn,
e.dscrpcion_cerror,

x.dscrpcion_carchvo
from cmnda
m, crmo r, cfrma_pgo f, ctpo_rcbo l,
corgen o,

error e, carchvo x, trcbo_incsnstntc
a
where
m.cve_cmnda = a.cve_cmnda
and
f.cve_cfrma_pgo =
a.cve_cfrma_pgo
and
t.cve_ctpo_rcbo =
a.cve_ctpo_rcbo
and
o.cve_corgen = a.cve_corgen
and
e.cve_cerror = a.cve_cerror
and
x.cve_carchvo = a.cve_carchvo
and
r.cve_crmo = a.cve_crmo
and
r.cve_cuen = @uen
and
e.cve_cerror = 2
and ((m.cve_cmnda > @moneda)
or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsnstn >
@poliza) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsnstn = @poliza
and
a.nmro_cmbio_incsnst >
@endoso) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsnstn = @poliza
and
a.nmro_cmbio_incsnst =
@endoso and t.cve_ctpo_rcbo =
@tipo) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsnstn = @poliza
and
a.nmro_cmbio_incsnst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incsns >
@prima) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsnstn = @poliza
and
a.nmro_cmbio_incsnst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incsns =
@prima and a.fcha_vncmnto_incsn
> @fecha))
order by
m.cve_cmnda, a.nmro_plza_incsnstn
a.nmro_cmbio_incsnst, t.cve_ctpo_rc
bo,
a.imprte_prma_incsns, a.fcha_vncmnto
lo_incsn
    SET @cve_region = NULL
    SET @des_region =
NULL
    SET @des_promo =
NULL
    OPEN c_incon
    FETCH NEXT FROM
c_incon
    into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima, @fec_vento,
@cve_ramo, @cve_archivo,

@cve_error, @cve_for_pago,
@cve_origen,

@num_promo, @num_agente,
@des_tpo_rec, @des_moneda,
@cve_ramo,
@des_for_pago, @des_origen,
@fec_movto,

@nom_aseg, @des_error,
@des_archivo
    WHILE
@i=@FETCH_STATUS = 0 and @i <=
100
    BEGIN
        /*BEGIN 30*/
        SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,

@num_poliza, @cve_region,
@des_region, @num_promo,
@des_moneda, @num_agente,
@des_tpo_rec, @des_moneda,
@des_for_pago, @des_origen,
@fec_movto,

@nom_aseg, @des_error,
@des_archivo
    END
    CLOSE c_incon
    DEALLOCATE
c_incon
    END
    ELSE
    IF @ramo IS NULL
    BEGIN
        IF @error = 2
        BEGIN
            DECLARE c_incon CURSOR
```



DISEÑO E IMPLANTACIÓN DE UN  
SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
@des_promo, @num_agente,
@num_cambio, @des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@nom_aseg,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima, @fec_vento,
@cve_ramo, @cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_promo, @num_agente,
@des_tpo_rec, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg, @des_error,
@des_archivo
END /PEND 307
CLOSE c_incon
DEALLOCATE c_incon
SET @moneda = 0
IF @i < 100
BEGIN
DECLARE c_incon CURSOR
FOR select m.cve_cmnda,
a.nmro_plza_incsstn,
a.nmro_cmbio_incsst,
t.cve_ctpo_rcbo,
a.imprte_prma_incss,
a.fcha_vncmnto_incs,
r.cve_crmo, x.cve_carchvo,
e.cve_cerror,
f.cve_cfma_pgo, o.cve_corgen,
l.cve_cofcna, l.dscrpcion_cofcna,
i.cve_cintrmdrio,
i.dscrpcion_cintrmdr,
a.nmro_agnte_incsst,
t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,
r.dscrpcion_crmo,
f.dscrpcion_cfma_pg,
o.dscrpcion_corgen,
a.fcha_mvnmnto_incss.
```

```
a.nmbre_asgrdo_incs,
e.dscrpcion_cerror,
x.dscrpcion_carchvo
from cmnda
m, crmo r, cfma_pgo f, ctpo_rcbo t,
corgen o,
cofca l, cintrmdrio i, cerror e,
carchvo x,
trcbo_incsstnta a
where
m.cve_cmnda = a.cve_cmnda
and
f.cve_cfma_pgo =
a.cve_cfma_pgo
and
t.cve_ctpo_rcbo =
a.cve_ctpo_rcbo
and
o.cve_corgen = a.cve_corgen
and
e.cve_cerror = a.cve_cerror
and
x.cve_carchvo = a.cve_carchvo
and
i.cve_cintrmdrio =
a.cve_cintrmdrio
and
l.cve_cofcna = i.cve_cofcna
and
r.cve_crmo = a.cve_crmo
and
r.cve_cuen = @cuen
and
e.cve_cerror = 1
and ((m.cve_cmnda > @moneda)
or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn >
@poliza) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst >
@endoso) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo >
@tipo) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incss >
@prima) or
(m.cve_cmnda = @moneda
and a.nmro_plza_incsstn = @poliza
and
```

```
a.nmro_cmbio_incsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incss =
@prima and a.fcha_vncmnto_incs
> @fcha))
order by
m.cve_cmnda, a.nmro_plza_incsstn
a.nmro_cmbio_incsst, t.cve_ctpo_rc
bo,
a.imprte_prma_incss, a.fcha_vncmn
to_incs
OPEN c_incon
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima, @fec_vento,
@cve_ramo, @cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region, @des_region,
@num_promo, @des_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg, @des_error,
@des_archivo
WHILE
@GET_STATUS = 0 and @i <=
100
BEGIN
/PEND 327
SELECT @cve_moneda,
@cve_tpo_rec, @cve_ramo,
@cve_archivo,
@cve_error, @cve_for_pago,
@cve_origen,
@num_poliza, @cve_region,
@des_region, @num_promo,
@des_promo, @num_agente,
@num_cambio, @des_tpo_rec,
@imp_prima,
@fec_vento, @des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@des_error,
@des_archivo
SET @i = @i + 1
FETCH NEXT FROM
c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
```





## DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE CARTERA PARA ASEGURADORAS

```
a.imprte_prma_incns,
a.fcha_vncmnto_incns,

r.cve_cmno, x.cve_carchvo,
e.cve_cerror,

f.cve_cfrma_pgo, o.cve_corgen,

l.cve_cofcna, l.dscrpcion_cofcna,

i.cve_cintrmdrio,
i.dscrpcion_cintrmdr,

a.nmro_agnte_incnsst,

t.dscrpcion_ctpo_rcb,
m.dscrpcion_cmnda,

r.dscrpcion_cmno,
f.dscrpcion_cfrma_pg,

o.dscrpcion_corgen,
a.fcha_mvnmnto_incns,

a.nmbre_asgrdo_incns,
a.dscrpcion_cerror,

x.dscrpcion_carchvo
from cmnda
m, cmno r.cfrma_pgo f, ctpo_rcbo l,
corgen o,

cofcna l, intrmdrio i, cerror e,
carchvo x,

trcbo_incnsstnte a
where
m.cve_cmnda = a.cve_cmnda
and
f.cve_cfrma_pgo =
a.cve_cfrma_pgo
and
t.cve_ctpo_rcbo =
a.cve_ctpo_rcbo
and
o.cve_corgen = a.cve_corgen
and
e.cve_cerror = a.cve_cerror
and
x.cve_carchvo = a.cve_carchvo
and
l.cve_cintrmdrio =
a.cve_cintrmdrio
and
i.cve_ctpo_intrmdri =
a.cve_ctpo_intrmdri
and
l.cve_cofcna = i.cve_cofcna
and
r.cve_cmno = a.cve_cmno
and
r.cve_cmno = @ramo

r.cve_cuen = @uen
and
e.cve_cerror = 1
and ((m.cve_cmnda > @moneda)
or
(m.cve_cmnda = @moneda
and a.nmro_piza_incnsstn >
@poliza) or
(m.cve_cmnda = @moneda
and a.nmro_piza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst >
@endoso) or
(m.cve_cmnda = @moneda
and a.nmro_piza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst =
@endoso and t.cve_ctpo_rcbo >
@tipo) or
(m.cve_cmnda = @moneda
and a.nmro_piza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incns >
@prima) or
(m.cve_cmnda = @moneda
and a.nmro_piza_incnsstn = @poliza
and
a.nmro_cmbio_incnsst =
@endoso and t.cve_ctpo_rcbo =
@tipo and
a.imprte_prma_incns =
@prima and a.fcha_vncmnto_incns
> @fecha))
order by
m.cve_cmnda, a.nmro_piza_incnsstn
,
a.nmro_cmbio_incnsst, t.cve_ctpo_rc
bo,
a.imprte_prma_incns, a.fcha_vncmnto
to_incns

OPEN c_incon
FETCH NEXT
FROM c_incon
into @cve_moneda,
@num_poliza, @num_cambio,
@cve_tpo_rec,
@imp_prima,
@fec_vento, @cve_ramo,
@cve_archivo,
@cve_error,
@cve_for_pago, @cve_origen,
@cve_region,
@des_region,
@num_promo, @des_promo,
@num_agente, @des_tpo_rec,
@des_moneda,
@des_ramo,
@des_for_pago, @des_origen,
@fec_movto,
@nom_aseg,
@des_error, @des_archivo
END
CLOSE c_incon
DEALLOCATE c_incon
END
END
END
END
END
END
GO
GRANT ALL ON PROCEDURE TO
PUBLIC
GO
```



## BIBLIOGRAFÍA

- MILAN MILENKOVIC  
Sistemas Operativos. Conceptos y diseño  
McGraw Hill  
2ª Edición
- JEFFREY D. ULLMAN  
Principles of Database Systems  
Computer Science Press  
2a Edición
- HENRY F. KORTH, ABRAHAM SILBERSCHATZ  
Database System Concepts  
McGraw Hill, Inc.  
2a Edición
- GEORGE KOCH  
Oracle: The complete reference  
Osborne McGraw Hill
- JAMES MARTIN  
Organización de las Bases de Datos  
Prentice Hall  
Traducido de la 1ª Edición
- JAMES R. GROFF & PAUL N. WEINBERG  
Using SQL  
Osborne McGraw Hill
- JENS HERBERS, JÖRG HERBERS, JÖRG RENSMANN  
El gran libro de Delphi2  
Marcombo Boixareu Editores
- HAWRYSZKIEWYCZ, IGOR TITUS  
Database analysis and design  
Macmillan  
2a Edición
- GREEN PAUL ELIOT  
Network interconnection and protocol conversion  
IEEE Communications Society  
1998



- KAUFFELS, FRANZ-JOACHIM  
Network management: Problems, standars and strategies  
Addison – Wesley  
1992
- Kunreuther, Howard  
International trade in insurance  
University of Pennsylvania  
1991
- Raynes, Harold Ernest  
Insurance  
Oxford University  
1960
- VAUGHAN, EMMETT J.  
Fundamentals of risk and insurance  
J. Wiley  
1996
- FINNEY, HARRY ANSON  
Curso de contabilidad  
Uteha
- GARY CORNELL  
Visual Basic 6.0  
Prentice Hall  
2a. Edición
- MICROSOFT PRESS  
Microsoft Visual Basic 6.0  
Referencia del Lenguaje  
Mc Graw Hill
- LUIS JOYANES AGUILAR  
Microsoft Visual Basic 6.0  
Osborne Mc Graw Hill  
1a. Edición
- ANDREW S. TANENBAUM  
Operating Systems  
Prentice Hall  
2a. Edición



- **RON SOUKUP**  
A fondo Microsoft SQL Server 7.0  
Mc Graw Hill – Microsoft Press  
1a. Edición
- **Implementing a database with Microsoft SQL Server 7.0**  
Microsoft Certificate.
- **Microsoft SQL Server 7 al descubierto**  
Sharon Bjeletich, Greg Mable, et al.  
Prentice Hall, Sams