



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

97
29

FACULTAD DE INGENIERÍA

DESARROLLO DE UN SISTEMA PARA EL CONTROL Y
REASIGNACIÓN DE AGENTES DE LA CARTERA
HUÉRFANA DE GRUPO NACIONAL PROVINCIAL

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTAN:

ELVIA PÉREZ ESCUTIA
EMILIO SILVA DRITRIT
JESÚS FRANCISCO VALENZUELA RODRÍGUEZ
LORENA ISABEL MARTÍNEZ HERRERA
MARÍA GUZMÁN SÁNCHEZ

TESIS CON
FALLA DE ORIGEN



DIRECTOR DE TESIS: M. EN I. LAURO SANTIAGO CRUZ

MÉXICO, D.F.

JUNIO, 1996

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicamos el presente trabajo a nuestros seres queridos

A Dios

*Por iluminarnos el camino día con día y
fortalecer nuestra voluntad.*

*Por darnos la oportunidad de respirar,
de amar y de luchar por un ideal.*

*Por acompañarnos siempre, y en los
momentos difíciles ser nuestra paz.*

A nuestros Padres

*Por su amor infinito, por su fortaleza,
por su tenacidad.*

Por ser aliento y esperanza en todo momento.

Por ser protección y confianza.

Por regalarnos su vida sin reservas.

A nuestros Hermanos

*Por su apoyo, porque con nadie más podríamos
compartir nuestros triunfos y nuestras alegrías.*

**A nuestra Universidad, nuestra Facultad y nuestros
Profesores.**

Nuestro más sincero agradecimiento y respeto.

GRACIAS POR SIEMPRE...

*Un agradecimiento especial,
por su apoyo en la realización de
este trabajo a:*

M. en Ing. Lauro Santiago Cruz

Ing. Arturo Gudiño Chong

Lic. Jesús Espíndola Cuevas

Elvia, Emilio, J. Francisco, Lorena, María.

ÍNDICE

INTRODUCCIÓN	1
Capítulo I. ANTECEDENTES GENERALES	4
1.1 Historia del seguro.....	4
Primera Etapa.....	5
Segunda Etapa.....	6
Tercera Etapa.....	6
1.2 Antecedentes de Grupo Nacional Provincial.....	7
1.3 Aspectos generales de GNP.....	11
Actividades principales de GNP.....	12
1.4 Ramos de seguros en GNP.....	12
1.5 Plaza GNP.....	14
1.6 Principios fundamentales del seguro.....	16
El seguro.....	16
Póliza.....	17
Agente.....	17
Póliza Huérfana.....	17
Prima.....	17
Clasificación del seguro.....	17
Capítulo II. ANÁLISIS	19
2.1 Conceptos básicos.....	19
2.2 Situación actual.....	26
Problemática.....	39
2.3 Sistema propuesto.....	40
Requerimientos del sistema.....	48
Alternativas para la selección de una base de datos.....	48
2.4 Planeación de actividades.....	50
2.5 Análisis del sistema.....	51

Capítulo III. DISEÑO	55
3.1 Conceptos básicos	56
Almacenamiento de datos	56
Base de datos y archivos convencionales	56
Abstracción de la información	58
Modelo de datos	58
Instancias, esquemas e independencias de los datos	61
Diseño de software	61
Motivos para realizar el diseño estructurado	62
Consideraciones generales	62
Metas del diseño estructurado	63
Acoplamiento y cohesividad	63
Técnicas del diseño estructurado	65
Diseño orientado al flujo de datos	65
Diagrama de flujo de datos	66
Diccionario de datos	67
Diagrama entidad-relación	68
3.2 Diagrama de flujo de datos (DFD) del sistema POHUER	69
3.3 Diccionario de datos	74
3.4 Diagrama entidad-relación del sistema POHUER	76
3.5 Normalización	77
3.6 Creación de prototipos	84
Capítulo IV. DESARROLLO	89
4.1 Conceptos básicos	89
Codificación	89
Características de los lenguajes de programación	90
Cuestiones importantes en programación	91
Puntos clave para la programación	91
Programación orientada a objetos	92
PARADOX	92
Enfoque por procedimientos	93
Enfoque por objetos	93
4.2 Panorama general del sistema POHUER	93
4.3 Operación	95
Entrada al sistema	95
4.4 Módulo de consulta y reasignación	97
4.5 Módulo de impresión y reportes	106
4.6 Módulo de mantenimiento	122
4.7 Módulo de datos adicionales	124

Capítulo V. PRUEBAS E IMPLANTACIÓN	128
5.1 Conceptos básicos	128
Pruebas.....	128
El proceso de prueba.....	129
Tipos de prueba.....	130
Implantación.....	130
5.2 Pruebas del sistema POHUER.....	130
5.3 Implantación del sistema POHUER.....	132
Instalación.....	132
Manual de Usuario.....	133
Capacitación.....	133
Mantenimiento del sistema POHUER	134
 RESULTADOS Y CONCLUSIONES	 136
 BIBLIOGRAFÍA	 137
 ÍNDICE DE MATERIAS	 139
 APÉNDICE A. Glosario	 A1
 APÉNDICE B. Código de POHUER	 B1

INTRODUCCIÓN

En la actualidad el manejo de información es un factor determinante en el desarrollo de una organización, de ello depende su crecimiento y competitividad para asegurar su permanencia en el mercado. La necesidad de contar con información oportuna y veraz se deriva de la obligación de una empresa de proporcionar atención rápida, confiable y segura, manteniendo una comunicación estrecha con sus clientes.

Los grandes volúmenes de información y la tecnología que evoluciona vertiginosamente han originado la creación de sistemas de cómputo y comunicaciones, que facilitan la integración de actividades dentro de las empresas así como el intercambio de datos en lugares distantes. Es imposible pensar que pueda existir una organización sin la utilización de la computadora como herramienta de apoyo en el procesamiento de información.

Aunque existen grandes firmas desarrolladoras de software, es importante hacer notar que éstas, producen sistemas que ayudan a una empresa sólo en algunas tareas. Sin embargo, debido a la complejidad de sus actividades, es necesario automatizar completamente el manejo de información y englobar cada detalle dentro de un sistema de cómputo, siendo necesario desarrollar software a la medida, dejando el software comercial ya existente en el mercado para actividades rutinarias no específicas como son: procesadores de texto, hojas de cálculo, gráficos, etcétera.

Dentro del territorio nacional existe un gran número de empresas que por el enorme volumen de información que manejan han adoptado sistemas de cómputo y comunicaciones para tener un control óptimo de su información. Como ejemplo de éstas podemos mencionar: bancos, tiendas de autoservicio, agencias de viajes y compañías aseguradoras. Estas últimas han tenido un gran auge en la última década, debido a la inestabilidad económica y social en la que se encuentra nuestro país.

El amplio mercado de estos seguros no se limita únicamente a empresas, sino que, su campo de acción se ha extendido a otro tipo de instituciones y colectividades, que aun no teniendo obligaciones de índole laboral o legal, requieren de mecanismos de tipo motivacional hacia sus miembros.

Entre las compañías de Seguros más importantes del país se encuentra Grupo Nacional Provincial (GNP) que se ha preocupado cada vez más por estar a la vanguardia en lo que a tecnología de cómputo se refiere, con la finalidad de ofrecer un mejor servicio a sus asegurados, basado en alta calidad y eficiencia, así como para brindar apoyo a su Fuerza Productora a través de herramientas ágiles y prácticas que faciliten el logro de sus objetivos.

Actualmente Grupo Nacional Provincial (GNP) detectó que no existe un control eficiente en el manejo de las pólizas de los agentes cancelados (Cartera Huérfana). Cuando es cancelado un agente, la Gerencia de Cartera Huérfana es notificada e inmediatamente procede a verificar las pólizas que estaban asignadas a éste agente. Sin embargo para llevar a cabo el proceso anterior se tiene que extraer la información requerida, de varios sistemas desarrollados en diversos lenguajes y plataformas que se encuentran en diferentes host. Los sistemas anteriores no tienen forma de comunicarse entre sí y para las necesidades actuales de GNP se requiere consultarlos simultáneamente, lo que ha convertido el sistema actual, en un proceso lento e inoperable, además de que los costos de capacitación y entrenamiento del personal son cada vez mayores y el manejo de información es más complicado.

El objetivo de esta tesis es desarrollar una herramienta de consulta que permita integrar datos de diferentes plataformas o sistemas, en un ambiente multiusuario que use **interface** gráfica, para el control y reasignación de agentes de la Cartera Huérfana de GNP, del Ramo de Vida.

Para lograr el objetivo propuesto, el presente trabajo se estructuró de tal manera que en el primer capítulo se describe la historia del seguro, las actividades generales de GNP y los diversos ramos de seguros (vida y daños) que maneja la compañía. En el segundo capítulo se plantea la situación actual de la problemática de la Cartera Huérfana y se hace un análisis para darle solución mediante la creación de un sistema. En el tercer capítulo se lleva a cabo el diseño del sistema denominado POHUER, mediante el uso de la Ingeniería de Software, aplicando la metodología de Chris Gane & Thrish Sarson. Además se realiza la creación de prototipos del sistema POHUER. En el cuarto capítulo se efectúa el desarrollo del sistema, presentando una visión completa del sistema en general, de los módulos y de los programas en el ambiente en que se va a trabajar. En el capítulo quinto se mencionan las actividades que se llevan a cabo en las etapas de pruebas e implantación del sistema. Se habla del mantenimiento del sistema, se explican

los tipos de mantenimiento que existen, y se menciona el mantenimiento que podrá darse al sistema POHUER. Por último, se presentan los resultados y conclusiones obtenidos en base a los objetivos propuestos y se anexa la bibliografía consultada, glosario, índice de materias y los apéndices complementarios.

CAPÍTULO I

ANTECEDENTES GENERALES

En el presente capítulo se hará una reseña histórica de los seguros y GNP. Se mostrará el lugar que ocupa GNP en el mercado asegurador tanto en México como en el extranjero. También se contemplarán diversos aspectos de GNP como son recursos humanos y tecnológicos. Se mencionarán los tipos de seguros que maneja GNP dentro de sus ramos Vida y Daños. Se dará una visión general de GNP con el fin de identificar la Gerencia de Cartera Huérfana, para la cual se desarrollará el sistema motivo de esta tesis. Por último, se dará un panorama de las actividades que realiza esta Gerencia.

1.1 HISTORIA DEL SEGURO

Ante todos los peligros por lo que se ha visto amenazado el hombre desde sus orígenes, éste ha tratado de encontrar soluciones adoptando distintas actitudes. Entre todas estas actitudes está la de transferir el peligro o riesgo al que se encuentra expuesto, apareciendo para llevar a cabo esta operación el Seguro.

El Seguro responde a una necesidad de previsión para amortiguar o compensar las consecuencias económicas de acontecimientos dañinos. Dado el carácter masivo que reviste la industria aseguradora, se ha convertido en una función de interés público.

El Seguro surge como un tipo de financiamiento, mediante el cual no se traspasa el peligro en sí, sino parte de sus efectos financieros, disminuyendo así las pérdidas, que pudiera sufrir el asegurado en sus bienes o personas en caso de suceder un accidente.

La historia del Seguro se puede dividir en tres etapas:

Primera: Antigüedad y Edad Media hasta el siglo XIV.

Segunda: Del siglo XIV al XVII.

Tercera: Del siglo XVII hasta nuestros días.

Primera Etapa.

Durante este tiempo aparecen los primeros sistemas de ayuda mutua. En esta época tenemos a diferentes civilizaciones que tuvieron aportaciones importantes, entre éstas se encuentran:

- a) *BABILONIA: "Código Hamurabi", por medio de éste, si en una caravana uno de los integrantes perdía algo, los demás pagaban proporcionalmente la pérdida. Lo mismo con pérdidas sufridas en la ciudad.*
- b) *INDIA: "Código de Manú", regulaba básicamente la navegación, pero en cuanto al Seguro no se encuentran vestigios de su regulación.*
- c) *EGIPTO: Se maneja la idea de ayuda mutua entre los socios de una institución, para ayudar en los ritos funerarios del socio que falleciera.*
- d) *GRECIA: "Ley Rhodia de Jactu", regulaba el sector marítimo. Una pérdida se repartía entre todos los propietarios de la mercancía transportada en el barco. Constituye la base del Derecho Mercantil Marítimo. Los griegos tenían una asociación llamada "ERANOI", por la que se daban asistencia a los necesitados a través de un fondo común constituido por todos los agremiados.*
- e) *ROMA: Existía una asociación de militares que aportaban una cuota con la que tenían derecho a una indemnización para gastos de viaje por cambio de guarnición en caso de retiro o muerte. Los artesanos formaban el "Collegia Tenuiorum, Collegia Funeraticia", con el que los participantes contaban con gastos por muerte, seguridad constituida por un fondo formado por el estado y los beneficios obtenidos por las herencias dejadas por socios muertos.*

f) *EDAD MEDIA*: Existían ayudas mutuas como: "Las Gildes", que proporcionaban ayuda mutua de carácter religioso. Los monasterios daban socorro y caridad a huérfanos, viudas y desempleados por medio de limosnas. "Los Juras", daban protección por medio de rentas y pensiones que daban los reyes por gracia o merced. Finalmente podemos mencionar "Las Tontinas", de origen italiano, consistían en aportar sumas fijas de dinero cuyo total se divide entre el número de supervivientes a una fecha dada.

Segunda Etapa.

Del siglo XIV al XVII aparecen las primeras instituciones de Seguros en los ramos Marítimo, Vida e Incendio. El primer contrato de Seguro conocido, relativo al Seguro Marítimo, data de 1347, suscrito en Génova. Amparaba tanto los accidentes del transporte como la tardanza en la llegada del buque a su destino.

Las primeras pólizas de Seguro de Vida se expidieron en Londres en The Royal Exchange por comisionistas que distribuían el riesgo entre grupos de comerciantes que se reunían, siendo emitida la primera póliza de este ramo en 1583.

Con el incendio de Londres en 1666, el Seguro de Daños toma fuerza. En 1677, en Hamburgo, se funda la primera caja general pública de incendios, formada por varios propietarios que reunían cierta cantidad para socorrerse entre ellos en caso de incendio.

Tomando en cuenta el incendio de Hamburgo, surge la compañía de seguros contra incendio en 1681, en Londres llamada, "Fire Office".

En 1686 surge en Londres la más poderosa empresa de seguros, "Lloyd's".

En cuanto a la regulación jurídica del Seguro, ésta se inicia con las ordenanzas de Barcelona, y posteriormente las de Burgos, Sevilla, y Bilbao.

Tercera Etapa.

Esta etapa comienza con la aparición de las primeras empresas de Seguro sobre bases más técnicas, basadas en la ley de los grandes números y de las probabilidades. La aparición de estas empresas se debió principalmente al gran florecimiento de la industria en esa época, los grandes descubrimientos, y por la regulación jurídica a través del control administrativo. Las bases técnicas del Seguro aparecen desde 1654 con el cálculo de probabilidades y la ley de los grandes números iniciada por De Mére. En 1693, en Londres, se presentó un estudio sobre mortalidad humana, y así, una serie de estudios que beneficiaron a la empresa del Seguro.

Todas estas aportaciones ayudaron al surgimiento de las empresas aseguradoras con estas bases. Finalmente en 1699 nace la primera compañía de seguros sobre la vida "Society of Assurance's of Widows and Orphans".

1.2 ANTECEDENTES DE GRUPO NACIONAL PROVINCIAL

Durante los primeros meses del siglo XX, William B. Woodrow y varios hombres de negocios establecieron la primera empresa de seguros de vida en México: La Nacional. El entorno social comenzaba a vivir el preludio de acontecimientos que presagiaba la lucha armada de una revolución. Debido al conflicto bélico, se paralizaron las actividades económicas del país; sin embargo, La Nacional mantuvo firmes los compromisos contraídos con la sociedad mexicana. A medida que la Revolución avanzaba los problemas nacionales se hicieron patentes, pese a esto La Nacional continuó con el desarrollo y crecimiento de su empresa.

Con Venustiano Carranza se inicia un importante período de recuperación económica en México y la metrópoli comienza nuevamente la marcha de su crecimiento remodelando su imagen, se trazan nuevas avenidas y el asfalto suplente al empedrado.

En la figura 1.1. se puede observar claramente los avances y la transformación de las avenidas.

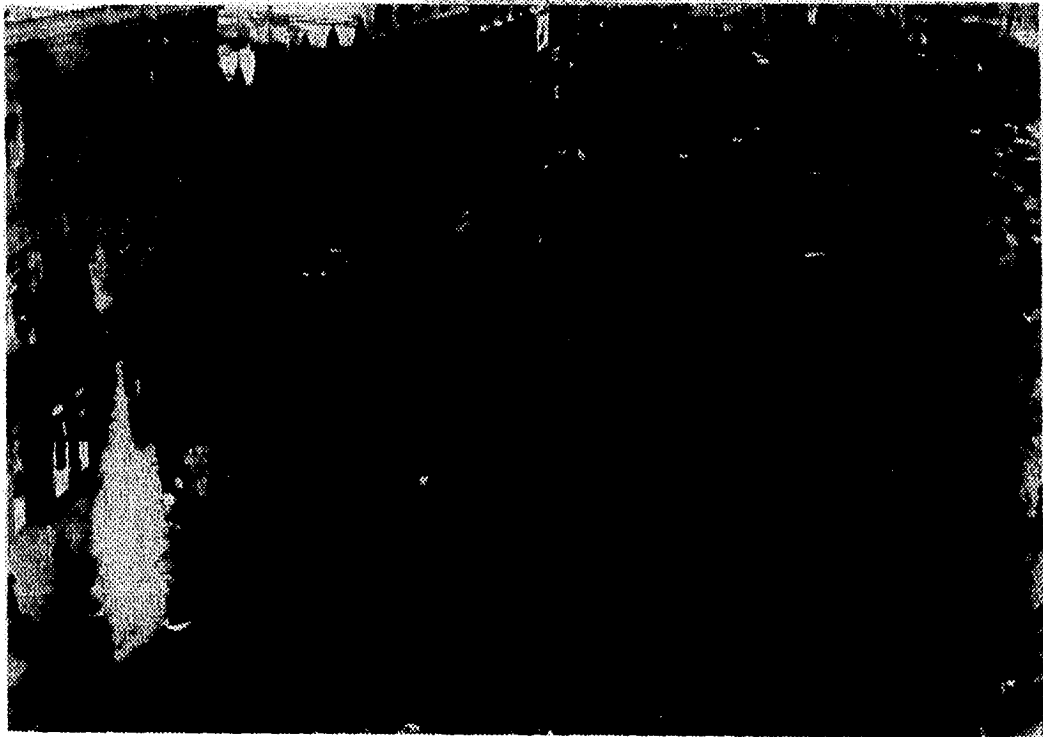


Fig. 1.1. El asfalto suplente al empedrado.

Para fines de 1928, la empresa era dueña de casas y terrenos en una superficie rectangular de 634 metros cuadrados. En 1929, contabilizados los recursos para la inversión y su rentabilidad, la Nacional decidió iniciar la construcción del "mejor edificio en México para el uso a que se le destinaba": oficinas y negocios. La obra empezó en julio de 1931 y en octubre de ese mismo año, se colocó la cimentación y un mes después se empezaron a colocar los soportes principales de la estructura de acero. Finalmente en 1932 cobró vida el primer rascacielos de la Ciudad de México en donde se encontraban las oficinas de La Nacional. En la figura 1.2. podemos apreciar el inicio y término del edificio de la Nacional.

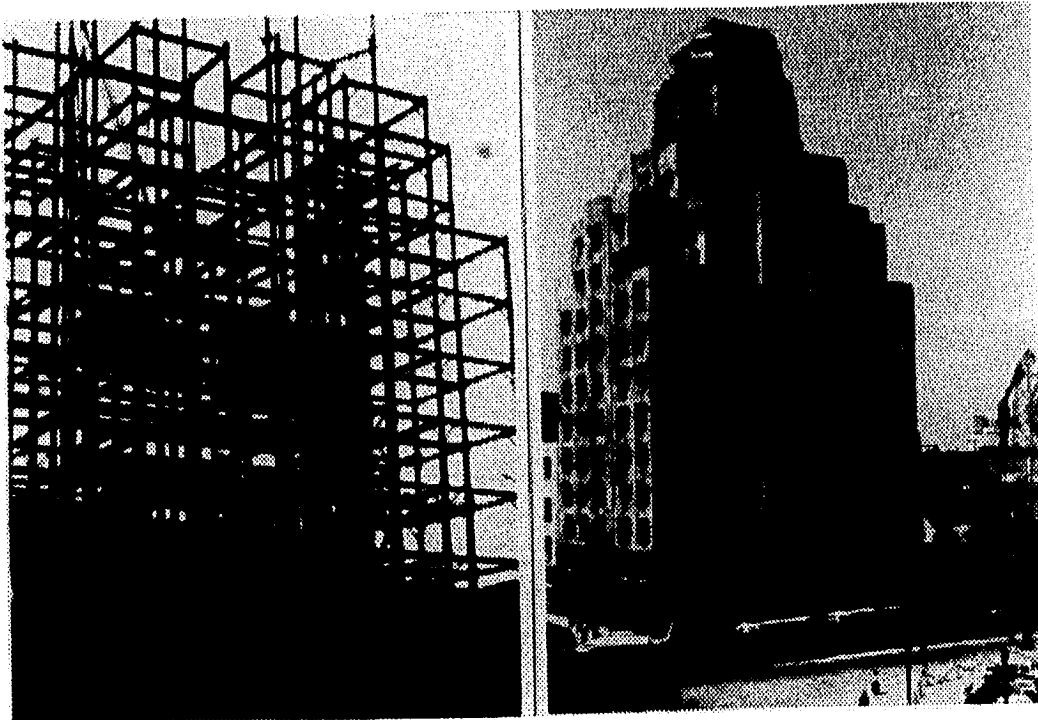


Fig. 1.2. Construcción del edificio de la Nacional.

En 1936 se obtiene la autorización de la Secretaría de Hacienda y Crédito Público para constituir una nueva empresa: Seguros la Provincial, que surge porque México aun no contaba con un medio que respaldara las inversiones de empresarios, puesto que La Nacional sólo se dedicaba al ramo de vida. En el año de 1968 deciden unir fuerzas ambas empresas, culminando así, una etapa de relaciones y esfuerzos conjuntos que dió lugar a un hecho novedoso dentro de la industria de seguros: La constitución de Grupo Nacional Provincial (GNP).

Desde 1968 GNP buscó colocarse a la vanguardia del mercado mexicano de seguros. En 1972 Grupo Nacional Provincial forma parte del Grupo Bal, uno de los conglomerados empresariales más importantes del país y ocupa, invariablemente, un lugar destacado dentro de las veinte compañías más grandes. En la lista de las quinientas empresas más sobresalientes de México, publicada por la revista *Expansión*. En 1975 GNP logró ubicarse en el primer lugar en ventas del sector asegurador, tanto en México como en Latinoamérica, posición que ha mantenido hasta el momento a pesar de que el mercado de seguros en México ha tenido una creciente competencia.

Este grupo empresarial se ha caracterizado permanentemente por su interés en programas de carácter social, entre los que se destacan aquellos que tienen relación con la capacitación y desarrollo de personas.

En la figura 1.3. se muestra el monto de las primas directas de GNP en 1995, y en la figura 1.4. se observa la participación que tiene GNP en el mercado, también en 1995, de las aseguradoras que operan en México.

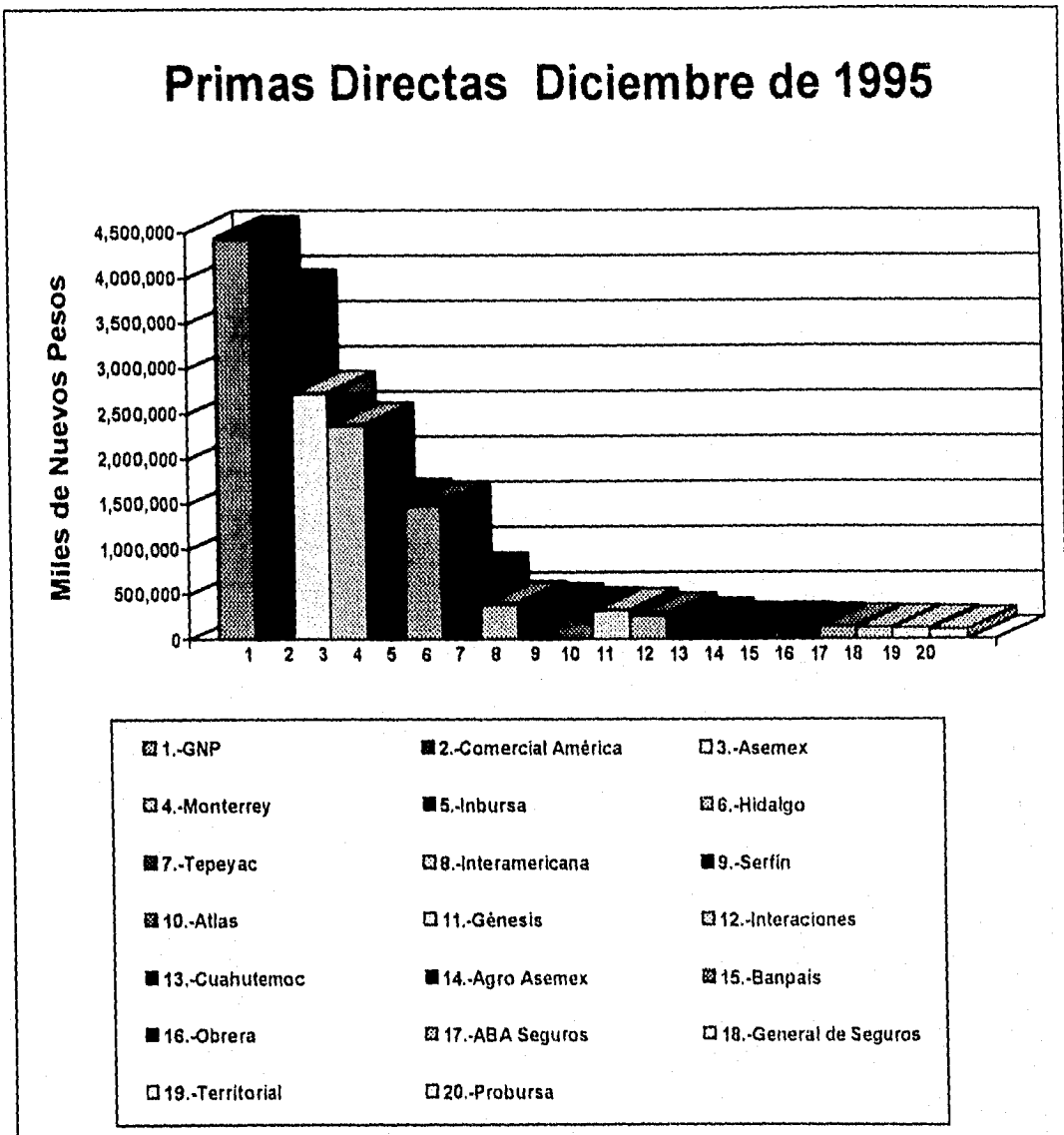


Fig. 1.3. Gráfica de monto de primas.

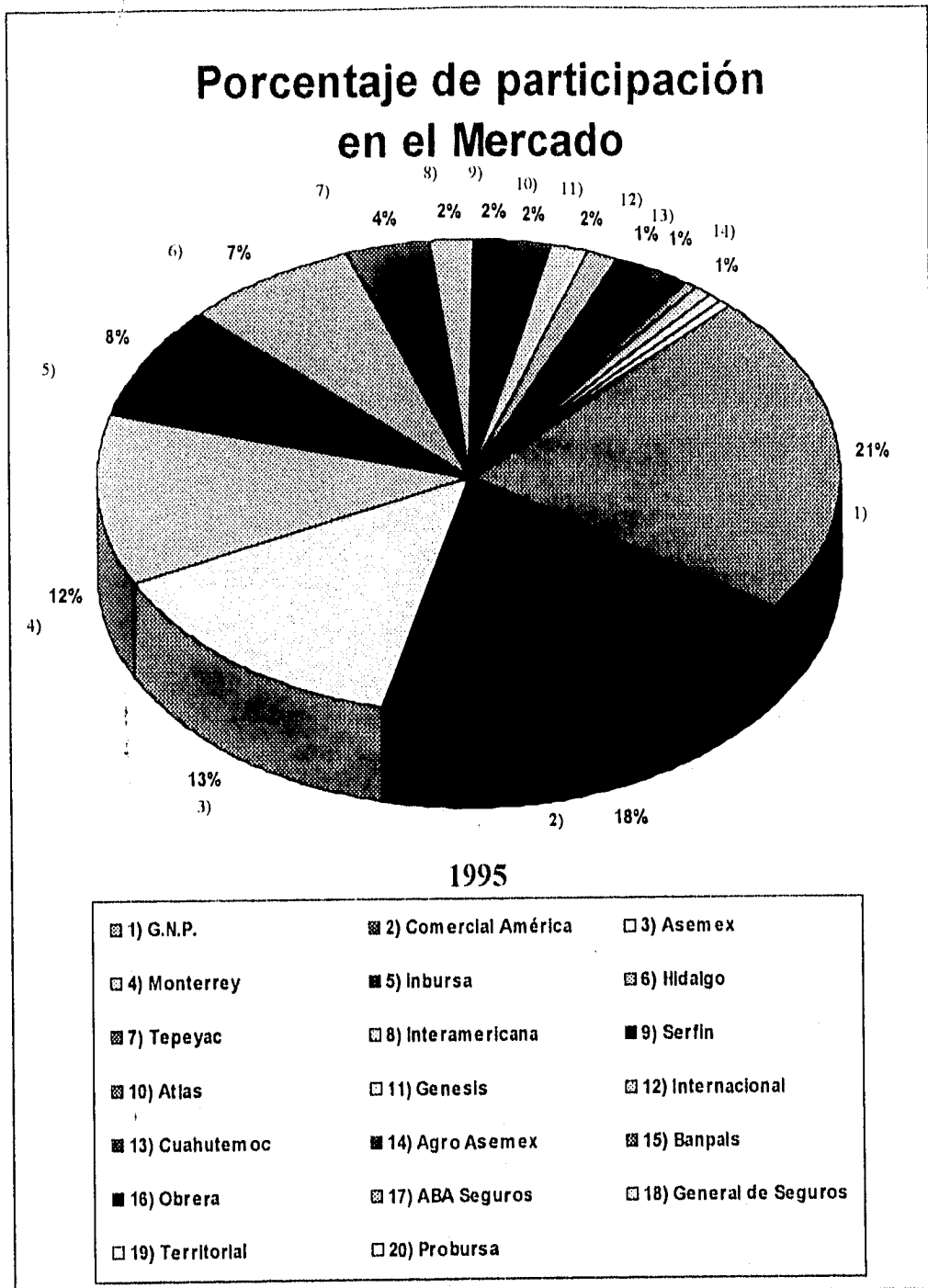


Fig. 1.4. En la gráfica se puede observar el porcentaje de mercado que ocupan las veinte aseguradoras más importantes del país. Se puede notar que grupo Nacional Provincial tuvo el veintiún por ciento del mercado asegurador nacional en 1995.

1.3 ASPECTOS GENERALES DE GNP

El recurso más valioso de GNP es su gente, desde la planta ejecutiva y sus empleados administrativos, aproximadamente 3200 personas, hasta su fuerza productora, integrada por poco más de 4000 agentes de ventas y cerca de 200 gerentes de zona, a lo que se agregan alrededor de 150 despachos de agentes especiales y firmas de corredores. Con lo anterior se conforma un equipo de más de 8000 colaboradores.

En cuanto a tecnología, GNP ha establecido una de las redes de informática más grandes de México; dos macrocomputadoras instaladas en el edificio de cómputo, que es reconocido como uno de los más avanzados del continente, son enlazadas a más de 4200 estaciones de trabajo distribuidas en todo el país. Esta conexión se realiza a través de una red privada de telecomunicaciones que utiliza los medios más avanzados, destacando un sistema de videoconferencia basado en un circuito cerrado de televisión entre la capital y las oficinas regionales del interior del país, en la figura 1.5, se muestran las instalaciones del edificio de cómputo.



Fig. 1.5. Instalaciones del edificio de cómputo

Otros servicios ofrecidos por la red de informática con que cuenta GNP incluye, enlaces con su fuerza productora, los sistemas de automatización de oficinas, correo electrónico, enlaces entre conmutadores telefónicos para formar una red de más de 2,500 extensiones y otros servicios adicionales. También son importantes los desarrollos realizados con sistemas expertos, capacitación multimedia y manejo de expedientes por medio de proceso de imágenes, entre otros.

La estructura de Servicio de GNP está formada por oficinas regionales en la Ciudad de México y en provincia; oficinas de servicio y de ventas que cubren todo el territorio nacional; centrales de ajuste en la capital, Monterrey y Guadalajara; Supervisorías de Reclamaciones a nivel Nacional; oficinas de representación en Nueva York y despachos de agentes especializados en pólizas, en California, Arizona, Nuevo México y Texas.

Las oficinas regionales de GNP, llamadas centros regionales, tienen funciones descentralizadas, esto es para ofrecer un servicio a toda la República Mexicana. Sin embargo, esta descentralización propicia que cada centro regional tenga su propia metodología para llevar a cabo su control de documentos, de volúmenes de trabajo, de personal, etc.

ACTIVIDADES PRINCIPALES DE GNP

Para lograr su desarrollo, GNP ha basado sus actividades en los siguientes principios fundamentales:

- En una profunda vocación aseguradora, basada en un historial de seriedad y en un claro sentido de responsabilidad social.
- En el diseño de productos y servicios que nos permita mantener una posición de vanguardia en todas las líneas de negocios.
- En el desarrollo tecnológico, especialmente en informática y comunicaciones.
- En mantener características distintivas, la solvencia, la oportunidad en el cumplimiento de sus obligaciones y el apego a los valores fundamentales que guían sus acciones.
- En fortalecer sus alianzas estratégicas con compañías nacionales y extranjeras en las diferentes líneas de negocio que opera, incluyendo las reaseguradoras que le dan respaldo.

1.4 RAMOS DE SEGUROS EN GNP

GNP tiene como misión fundamental satisfacer las necesidades de protección y previsión financiera de sus clientes frente a los riesgos a que están expuestos. Esto se logra mediante la venta de seguros que realiza su fuerza productora (agentes de ventas). Los seguros se dividen en:

- a) RAMO DE VIDA
- b) RAMO DE DAÑOS

Dentro del **RAMO DE VIDA** encontramos tres divisiones: *Vida, Pensiones, Accidentes y Enfermedades.*

En la división Vida se cuenta con los siguientes tipos de seguros:

- *Línea Universal: Seguro flexible de protección y ahorro.*
- *Línea Diamante: Coberturas en dólares.*
- *Beneficios y cláusulas adicionales: Muerte y pérdidas orgánicas por accidente, coberturas por incapacidad total y permanente.*
- *Seguros de Grupo y Colectivo: Para cubrir a los colaboradores de las empresas.*
- *Productos de Venta y Mercadeo Masivo: Descuento por nómina, módulos en punto de venta y campañas por correo directo.*

En la división Pensiones se cuenta con los siguientes tipos de seguros:

- *Grupo inversión y contratos de depósitos en administración: Planes de pensiones para empresas.*
- *Programa óptimo de retiro: Pensiones individuales.*

En la división Accidentes y Enfermedades se cuenta con los siguientes tipos de seguros:

- *Línea Azul: Pólizas de gastos médicos mayores para empresas e individuos.*
- *Pago directo y cirugías programadas: Sistemas de atención de siniestros.*
- *Red de hospitales, médicos y otros prestadores de servicios: Sólida infraestructura que permite un mejor control de costos y óptimos niveles de servicio.*
- *Médica Móvil: Servicios de emergencia.*

Dentro del **RAMO DE DAÑOS** se encuentran dos divisiones: *Automóviles y Daños.*

En la división Automóviles se cuenta con los siguientes tipos de seguros:

- *Con una extensa gama de coberturas: Amplias, limitadas y de responsabilidad civil.*
- *Fianza garantizada y servicios legales: Coberturas adicionales que aseguran la inmediata liberación del vehículo afectado y evitan problemas jurídicos.*
- *Línea Naranja: Permite a nuestros asegurados la compra de refacciones y servicios para sus autos a precios preferenciales.*

- *Red de ajustadores, abogados, talleres y hospitales: Infraestructura de servicio para la atención a siniestros, única en el mercado.*

En la división Daños se cuenta con los siguientes tipos de seguros:

- *Polifam a todo riesgo: Sistema simplificado de aseguramiento de casas habitación.*
- *Pólizas múltiples para individuos y empresas: Incendio y riesgos alicados, robo, dinero y valores, responsabilidad civil y otros.*
- *Pólizas especiales para grandes empresas: Especialmente diseñadas para cubrir sus necesidades específicas.*
- *Otras pólizas: Transportes (cascos y mercancías), equipo electrónico, montaje, obra civil, equipo de contratista y otras.*
- *Servicios de valuación: Para determinar sumas aseguradas correctas en edificios, maquinaria y equipo.*
- *Prevención y administración de riesgos: Estudios para prevenir y reducir pérdidas.*

1.5 PLAZA GNP

En lo que respecta a instalaciones físicas, Plaza GNP ocupa un lugar especial. En la figura 1.6. se puede observar el emblema de GNP y en la figura 1.7. se muestran las oficinas centrales de Grupo Nacional Provincial, en un terreno en la Ciudad de México de 105,000 m² de extensión, en la que se han construido diversos edificios de oficinas, así como el centro de cómputo, salón de usos múltiples e instalaciones deportivas.

Dichas instalaciones, que se hicieron acreedoras en 1994 a un premio en la tercera Bienal de Arquitectura, se han venido construyendo por etapas y el plan maestro contempla la realización de otras subsecuentes a mediano y largo plazo.

En la figura 1.8. se muestra el organigrama general de Plaza GNP, para identificar las diferentes dependencias que la constituyen.



Fig. 1.6. Emblema de GNP.

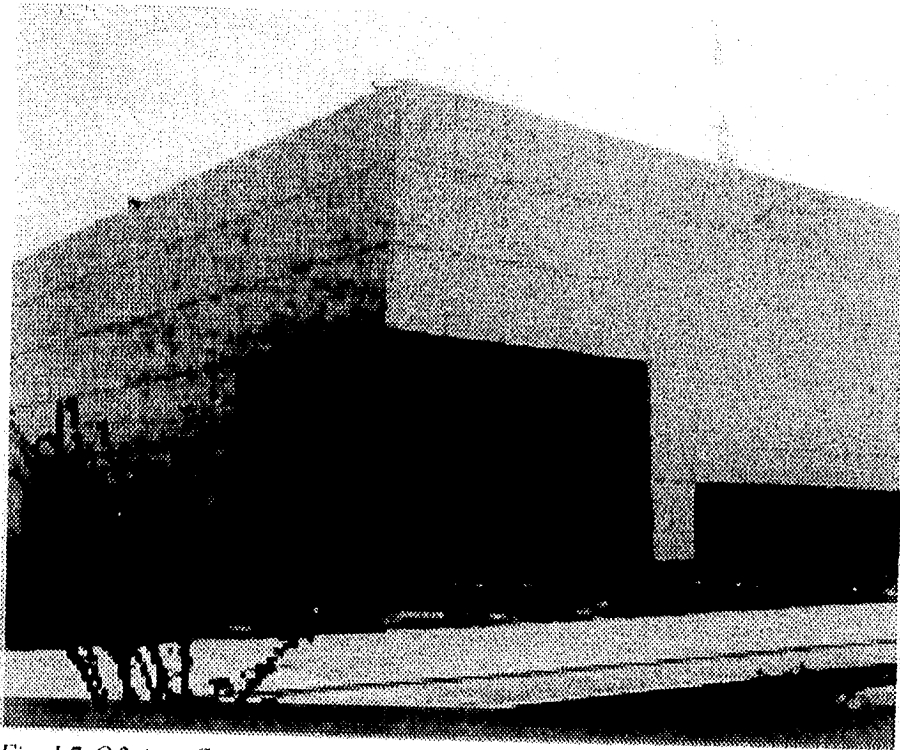


Fig. 1.7. Oficinas Centrales de GNP.

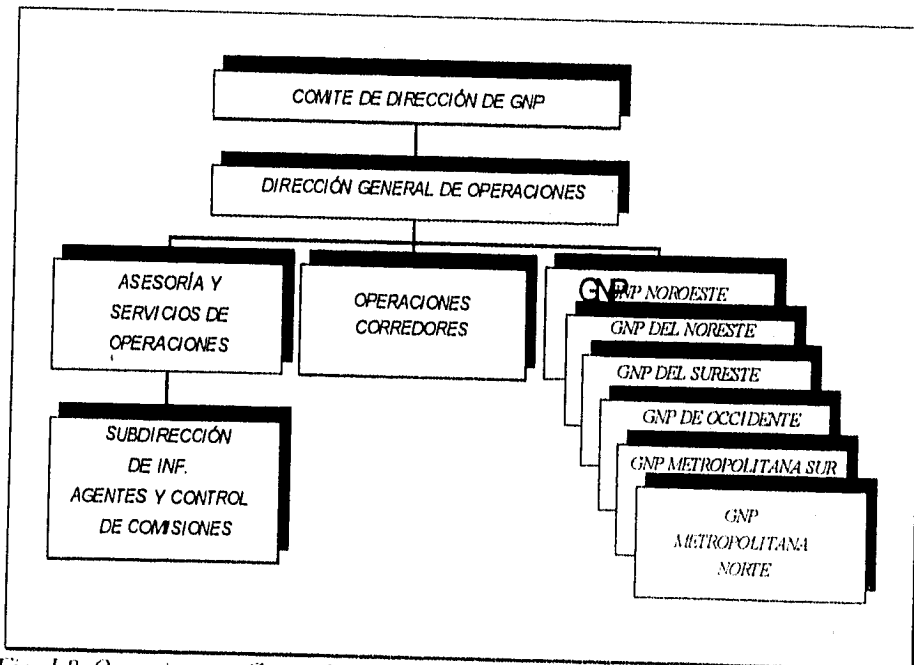


Fig. 1.8. Organigrama General de GNP.

1.6 PRINCIPIOS FUNDAMENTALES DEL SEGURO

Antes de continuar es importante comprender algunos aspectos fundamentales del Seguro:

El riesgo es una eventualidad que tiene cierta probabilidad de ocurrir y que puede acarrear, como consecuencia, una pérdida económica. Existen dos tipos de riesgos :

- a) Riesgos Especulativos: Aquéllos cuyas consecuencias pueden ser favorables o adversas.*
- b) Riesgos Puros: A diferencia de los especulativos acarrear siempre consecuencias adversas.*

La mayoría de los riesgos son desfavorables para unos cuantos individuos y no forzosamente para la sociedad en su conjunto. Estos riesgos son imposibles de medir, no se cuenta con la información que permita el cálculo certero de su gravedad y probabilidad de ocurrencia.

Sólo se puede disponer de la protección contra esos riesgos y aprovecharla hasta cierto punto y en casos aislados, porque: (1) la eliminación de consecuencias adversas puede costar demasiado con relación a la ganancia posible y hasta eliminar la posibilidad de ganancias; (2) muchas personas prefieren aceptar tales riesgos; (3) hay poca información en que basar la prima del seguro; y (4) el seguro contra esas contingencias implica un riesgo moral demasiado grande para quien lo proporciona. De suceder, no genera ganancias, por consiguiente es muy importante que la gente busque medios para reducirlos, transferirlos o evitarlos.

En principio, lo que se debe hacer es identificar los riesgos posibles, dentro de una organización; evaluarlos, y en base a esto tomar medidas de acción. La ocurrencia o realización del riesgo se le conoce como siniestro.

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 aquellos que hayan sufrido pérdidas.

PÓLIZA

El contrato de Seguro se hace constar en el documento legal que se llama "PÓLIZA" en el que se establecen los derechos y obligaciones de la empresa aseguradora y del asegurado. La póliza debe contener datos como:

- *Nombres, domicilios de los contratantes y firma de la empresa aseguradora*
- *Designación de las personas o cosas aseguradas*
- *Naturaleza de los riesgos asegurados*
- *Monto de la garantía*
- *Cuota o prima del seguro*
- *Tiempo de vigencia*

AGENTE

Es la fuerza productora de una compañía aseguradora, del agente depende la venta de seguros en cualquiera de sus ramas, la presentación de la pólizas tanto para el asegurado como para la compañía de seguros y la administración de las mismas. Se puede decir que el agente es la conexión directa entre la compañía de seguros y el asegurado.

PÓLIZA HUÉRFANA

Se puede entender por póliza huérfana a aquella que se ha quedado sin ningún agente que la represente ante una compañía de seguros.

PRIMA

La prima es el precio del Seguro. Representa una cantidad de dinero que aporta cada asegurado formando un fondo de reservas que servirá para el pago de siniestros que sufra alguno de los asegurados. Para el establecimiento de la prima se aplican cuotas ya fijas, según sea cada caso. Las cuotas generalmente se expresan al por ciento y están calculadas de acuerdo a la peligrosidad del riesgo y a la frecuencia con que pueda ocurrir el siniestro.

CLASIFICACIÓN DEL SEGURO

El seguro se divide en dos grandes ramos dando respuesta a situaciones diferentes muy bien definidas:

- Vida
- Daños

Una vez descritas las actividades principales de GNP, y los conceptos básicos de una empresa de seguros, se procede a plantear la problemática de la Cartera Huérfana y darle una solución por medio de un sistema de cómputo. Para llevar a cabo esta tarea se hace indispensable realizar un análisis detallado de todos los requerimientos del sistema utilizando los métodos de la Ingeniería de Software, como se hará en el capítulo siguiente.

CAPÍTULO II

ANÁLISIS

En el presente capítulo se mencionarán brevemente algunos conceptos básicos sobre la metodología del ciclo de vida de un sistema. Se mostrará la situación actual por la que atraviesa GNP en cuanto al manejo de la Cartera Huérfanas. Se analizarán las causas por las cuales GNP decidió crear una nueva gerencia, la "Gerencia de Cartera Huérfana", dentro de la Subdirección de Información de Agentes y Control de Comisiones. Se dará una propuesta de solución para resolver los problemas que se derivan de la Cartera Huérfana mediante la creación del Sistema de Control y Reasignación de la Cartera Huérfana en GNP, tomando en cuenta los requerimientos de hardware y software que son necesarios para la implantación del sistema. Finalmente se analizarán los procesos que llevará a cabo el sistema propuesto.

2.1 CONCEPTOS BÁSICOS

El software, o mejor dicho, la habilidad para desarrollarlo, se reconoce ahora como la clave para obtener una ventaja competitiva en un negocio de alta tecnología. El principio del software estructurado referente a que cada parte es fácil de comprender y modificar resulta crucial, esto sólo puede lograrse si el proceso mediante el cual se construye está bien fundado.

La Ingeniería como disciplina, que busca siempre obtener el mayor beneficio posible con una suma dada de medios en cualquier área en la que se aplica, no podía dejar de ser indispensable dentro del área de software. Por ende, la "Ingeniería de Software" surgió como una disciplina específica dentro de la Ingeniería

La Ingeniería de Software está compuesta por una serie de pasos que abarcan, métodos, herramientas y procedimientos. Estos pasos se denominan frecuentemente "Modelos del proceso del desarrollo de software".

Existen varios modelos del proceso de desarrollo de software disponibles en la actualidad. El modelo de desarrollo de software utilizado con mayor frecuencia, a menudo se denomina el "Modelo V o cascada" (quizá por su similitud a un conjunto de cascadas). El modelo representa el ciclo de vida del software como un conjunto de actividades ligadas, pero separadas, con entradas descendentes a etapas sucesivas y retroalimentación ascendente para proporcionar verificación contra etapas previas, así como una validación final de los requerimientos. El modelo del ciclo de vida de un sistema se ilustra en la figura 2.1.

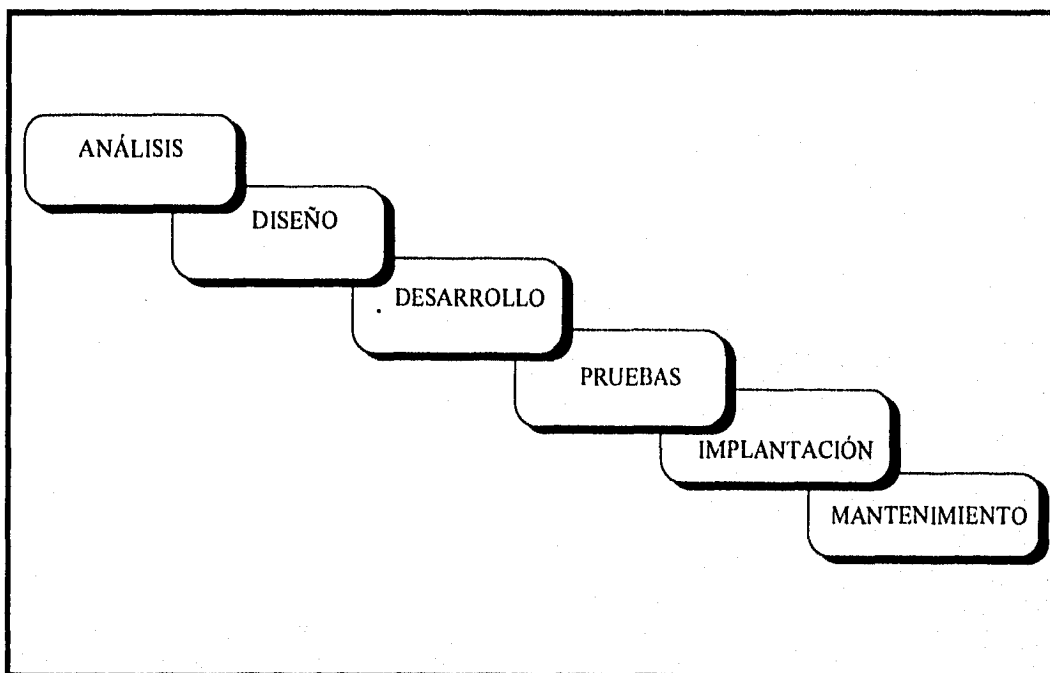


Fig. 2.1. "Modelo V o cascada" del ciclo de vida de un sistema.

Las etapas del ciclo de vida de un sistema son las siguientes:

ANÁLISIS: Es la primera etapa del ciclo de vida, donde se identifican los requerimientos del sistema a desarrollar mediante la recopilación de información acerca de las necesidades de la

empresa. Además se obtiene una definición del sistema, indicando sus alcances y restricciones. En esta etapa se deben de tener claros los objetivos que se pretenden alcanzar con el sistema y se debe realizar una planeación de las actividades que se desarrollarán en las etapas posteriores.

En el análisis se deben realizar las siguientes actividades:

- Describir la **Situación Actual** con el fin de identificar los requerimientos del sistema. Generalmente se realizan entrevistas para investigar la situación actual de la empresa, ver figura 2.2.

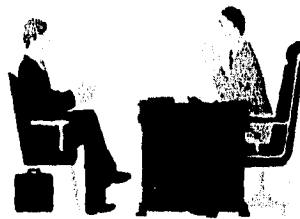


Fig. 2.2. Se investiga la situación actual.

- Definir una **Propuesta de Solución** que sea óptima para satisfacer las necesidades de la empresa, considerando los recursos con los que cuenta la misma y especificando claramente los objetivos del sistema. La propuesta generalmente se entrega en un documento por escrito para que la empresa que solicita el sistema le de el visto bueno, ver figura 2.3.



Fig. 2.3. Se proporciona una propuesta de solución.

- *Definición de Requerimientos y Alcances del Sistema. Se deben especificar los requerimientos de software y hardware necesarios para implantar el sistema, ver figura 2.4.*

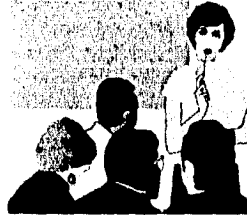


Fig. 2.4. Se deben definir requerimientos.

- *Hacer una Planeación de las Actividades que se realizarán para la consecución de los objetivos antes planteados. La planeación consiste en utilizar ciertas herramientas para definir las actividades a realizar, su prioridad y el tiempo en que se ejecutarán, además de dividir el trabajo, ver figura 2.5.*



Fig. 2.5. Se planean actividades, tiempos y división de trabajo.

- *Realizar propiamente el Análisis de los Procesos del Sistema Propuesto, ver figura 2.6.*



Fig. 2.6. Realizar análisis de procesos.

DISEÑO: Es el proceso de transformar los requerimientos del sistema a representaciones gráficas con el fin de facilitar la transcripción de estas representaciones a programas de computación. El diseño no es una tarea sencilla, el diseñador debe ser una persona creativa que logre una representación completa de los requerimientos del sistema identificados en el análisis, ver figura 2.7.



Fig. 2.7. Se transforman los requerimientos a representaciones gráficas.

Las notaciones de diseño más utilizadas son las siguientes:

- **Diagrama de flujo de datos (DFD):** Es un diagrama que se utiliza para describir un diseño de sistemas de alto nivel; muestra los diversos elementos de procesamiento en un sistema, los flujos de datos entre estos elementos de procesamiento y el almacenamiento de datos principal dentro del sistema. En la figura 2.8. se muestra un ejemplo de un DFD.

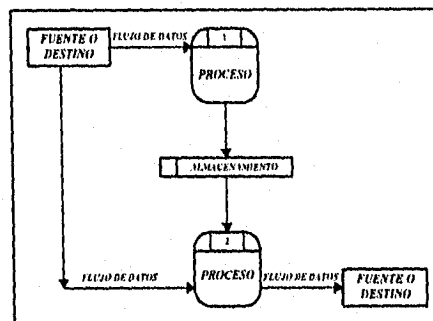


Fig. 2.8. Ejemplo de un DFD.

- **Diagrama de estructura de datos (DED):** Es una gráfica jerárquica que muestra la relación estructural de los componentes de un sistema de software. Esta notación describe cómo puede descomponerse un ítem de datos en ítems de datos más pequeños. En el nivel más alto, debe representarse un objeto completo, que puede después descomponerse en campos separados. La figura 2.9. muestra un ejemplo de un DED.

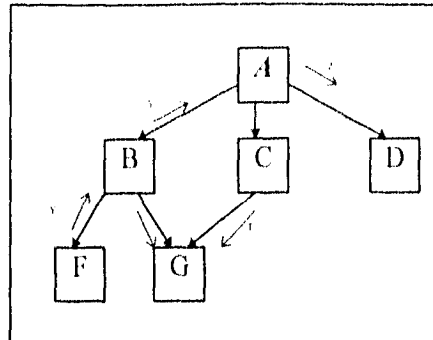


Fig. 2.9. Ejemplo de un DED.

- **Diagramas de relaciones de entidades (E-R):** Un diagrama de relaciones de entidades se utiliza para mostrar el tipo de relación existente entre entidades diferentes de un sistema. Una entidad es un objeto que existe y que puede distinguirse de cualquier otro por medio de sus características o atributos. Las relaciones que se dan entre entidades pueden ser "uno a uno", "uno a muchos", "muchos a uno" y "muchos a muchos". Es común que las entidades se representen por medio de cuadros, y las relaciones por diferentes tipos de flechas en líneas que interconectan los cuadros. Los cuadros por lo general contendrán el nombre de la entidad en lenguaje natural, que será descrito de manera más completa en un diccionario de datos. La figura 2.10. muestra un ejemplo de un diagrama E-R.

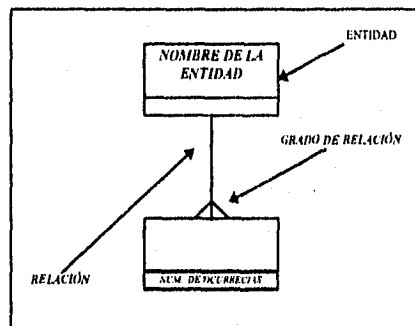


Fig. 2.10. Ejemplo de un diagrama E-R.

DESARROLLO: Se considera desarrollo a la etapa del ciclo de vida del sistema en donde se traduce una representación del software dada por un diseño detallado, a un lenguaje de programación, ver figura 2.11.

La producción de programas confiables y fáciles de mantener es un proceso independiente del lenguaje de programación. Los lenguajes de alto nivel facilitan el proceso de convertir un diseño en una aplicación, pero no hay razón para que no se puedan construir buenos programas en cualquier otro lenguaje.

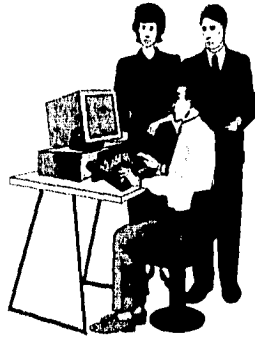


Fig. 2.11. Desarrollo de un sistema.

PRUEBAS: Una prueba es un proceso de confirmación que suele realizarse durante la aplicación. La prueba consiste en ejercitar el programa utilizando datos similares a los datos reales que habrán de ser ejecutados por el programa, observar los resultados y deducir la existencia de errores o insuficiencias del programa a partir de las anomalías de ese resultado, ver figura 2.12.

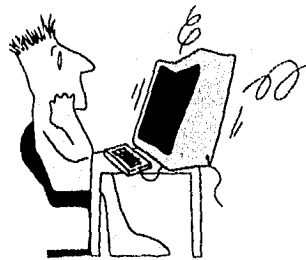


Fig. 2.12. Detectar errores es el objetivo principal de las pruebas.

IMPLANTACIÓN: Una vez confirmado que el sistema no tiene errores y cumple con las especificaciones o requerimientos descritos desde el análisis, se procede a implantarlo, es decir instalarlo en su ambiente real de trabajo. En esta etapa se realizan otras actividades como lo son: crear archivos que requiera el sistema, capturar información, entregar manuales, capacitar al personal, etc. Ver figura 2.13.



Fig. 2.13. Implantación del sistema.

MANTENIMIENTO: El mantenimiento es el proceso de modificar un programa cuando ya se ha entregado y está en uso, ver figura 2.14. Existen cuatro tipos de mantenimiento estos son:

- **Mantenimiento de perfeccionamiento:** Este mantenimiento comprende los cambios solicitados por el usuario o por el programador del sistema.
- **Mantenimiento adaptativo:** Este tipo de mantenimiento se debe a los cambios en el ambiente del programa.
- **Mantenimiento correctivo:** Es la corrección de errores del sistema no descubiertos.
- **Mantenimiento preventivo:** Esta categoría final dirige sus beneficios hacia el proceso general del mantenimiento mismo. Implica hacer cambios al software que por sí mismos, no mejoran el desempeño, pero provocan que las actividades futuras de mantenimiento sean más fáciles (más baratas) de llevar a cabo.

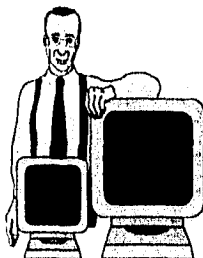


Fig. 2.14. El personal deberá estar capacitado para dar mantenimiento al sistema.

2.2 SITUACIÓN ACTUAL

La Cartera Huérfana había recibido poca atención desde la creación de GNP. Sin embargo, en las últimas estadísticas que realizó la Subdirección de Información de Agentes y Control de Comisiones se detectó que el monto de las primas de las pólizas huérfanas aumentó considerablemente. Lo anterior ha dado origen a una pérdida cuantiosa para GNP que se ha reflejado en una disminución sustancial de la cartera de clientes, que al no tener contacto directo con la compañía desconoce el estado actual de su póliza. Esta pérdida de clientela podría dar lugar a que GNP llegara a tener, en determinado momento, una mala imagen para futuros clientes.

Las causas que se pueden atribuir a los resultados arrojados por las estadísticas radican principalmente en los siguientes puntos.

- *Las pólizas al quedar huérfanas no tienen conducto de cobro debido a que el asegurado sólo tiene contacto con GNP a través de su agente.*
- *Existe una baja conservación de la cartera de agentes cancelados ya que son muy pocos los clientes que recurren a la compañía para checar el estado de sus pólizas y efectuar sus pagos.*
- *El asegurado, al no tener un agente que lo atienda, pierde los servicios de asesoría y atención personalizada, propiciando una mala imagen para la compañía.*
- *Se tienen saldos deudores irrecuperables.*
- *Se cobran comisiones sin dar atención a los asegurados.*
- *Existe la posibilidad de que algunos agentes cancelados actúen con dolo o de mala fe ya que los asegurados no reciben notificación sobre la cancelación del agente.*

Para dar atención a la Cartera Huérfana, la Subdirección de Información de Agentes y Control de Comisiones decidió crear una Gerencia exclusiva, la Gerencia de Cartera Huérfana, con el fin de recuperar las pólizas de los agentes cancelados contactando a los clientes vía telefónica o por telegrama, para brindarles el servicio de cobro y mantenimiento a sus pólizas. De esta forma, Grupo Nacional Provincial muestra nuevamente la preocupación que tiene por que sus clientes estén siempre bien atendidos, otorgándoles un mejor servicio. En la figura 2.15. se presenta el organigrama de la Subdirección para identificar el lugar que ocupa la Gerencia de Cartera Huérfana.

La Gerencia de Cartera Huérfana, teniendo como objetivo controlar y recuperar la Cartera Huérfana, debe disponer de información actualizada referente a los agentes cancelados y las pólizas asignadas a éstos. Lo anterior con el propósito de dar un seguimiento adecuado a la Cartera Huérfana.

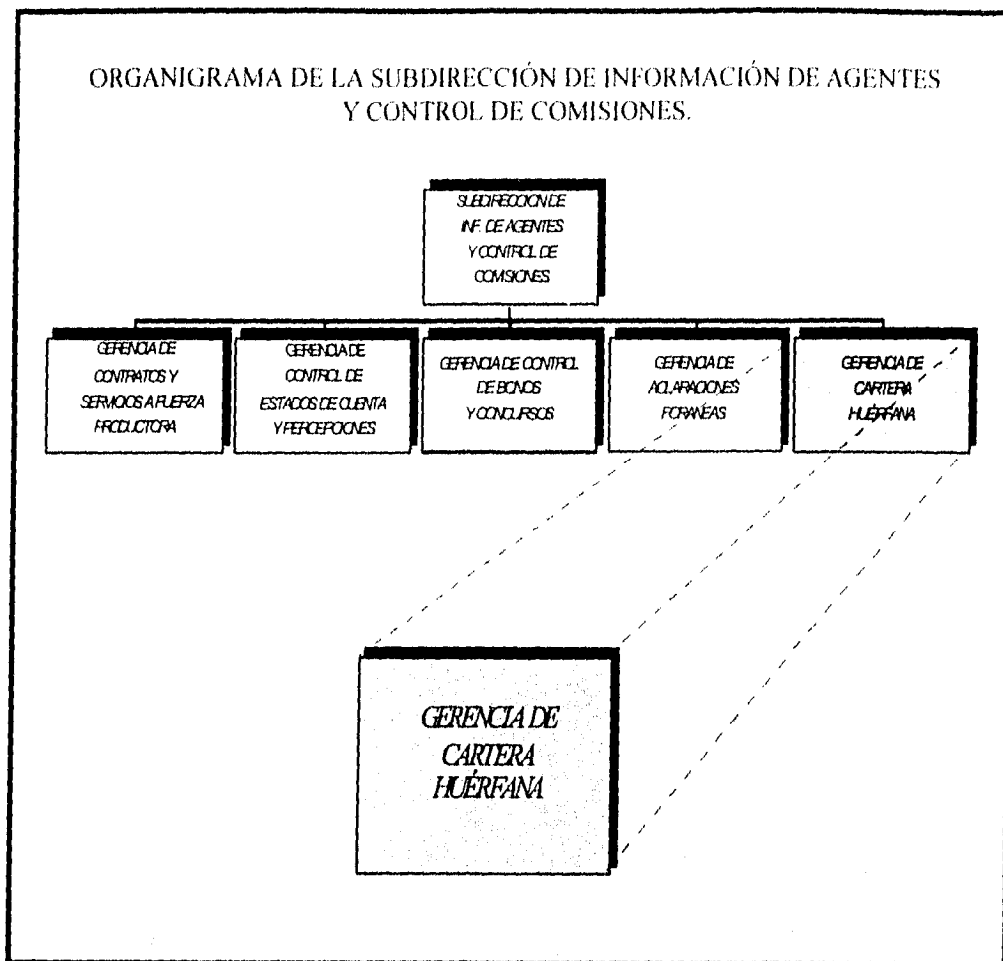


Fig. 2.15. Organigrama de la Subdirección de Información de Agentes y Control de Comisiones.

La información que requiere la Gerencia de Cartera Huérfana, para llevar a cabo sus objetivos propuestos, se encuentra particionada en diferentes sistemas. Estos sistemas existen desde hace aproximadamente veinte años y se han ido actualizando conforme las necesidades de GNP lo han exigido, teniendo desarrollados los sistemas en diferentes lenguajes y plataformas.

GNP a nivel nacional dispone de dos host (IBM/9121-320), en el primero se tiene un sistema que maneja el Ramo de Daños (SO-1) y en el segundo se almacena la información referente al Ramo de Vida (SO-2), es decir polizario de daños y polizario de vida. Estos sistemas están en diferentes plataformas y diversos lenguajes. Los sistemas se enlazan con una VAX6000-420 y para consultarlos se requiere de terminales tontas y emulación de terminal con equipos PC's conectados a servidores. Existen 51 servidores y hasta 250 terminales por servidor. En la figura 2.16. se muestra claramente como está interconectado el equipo de cómputo.

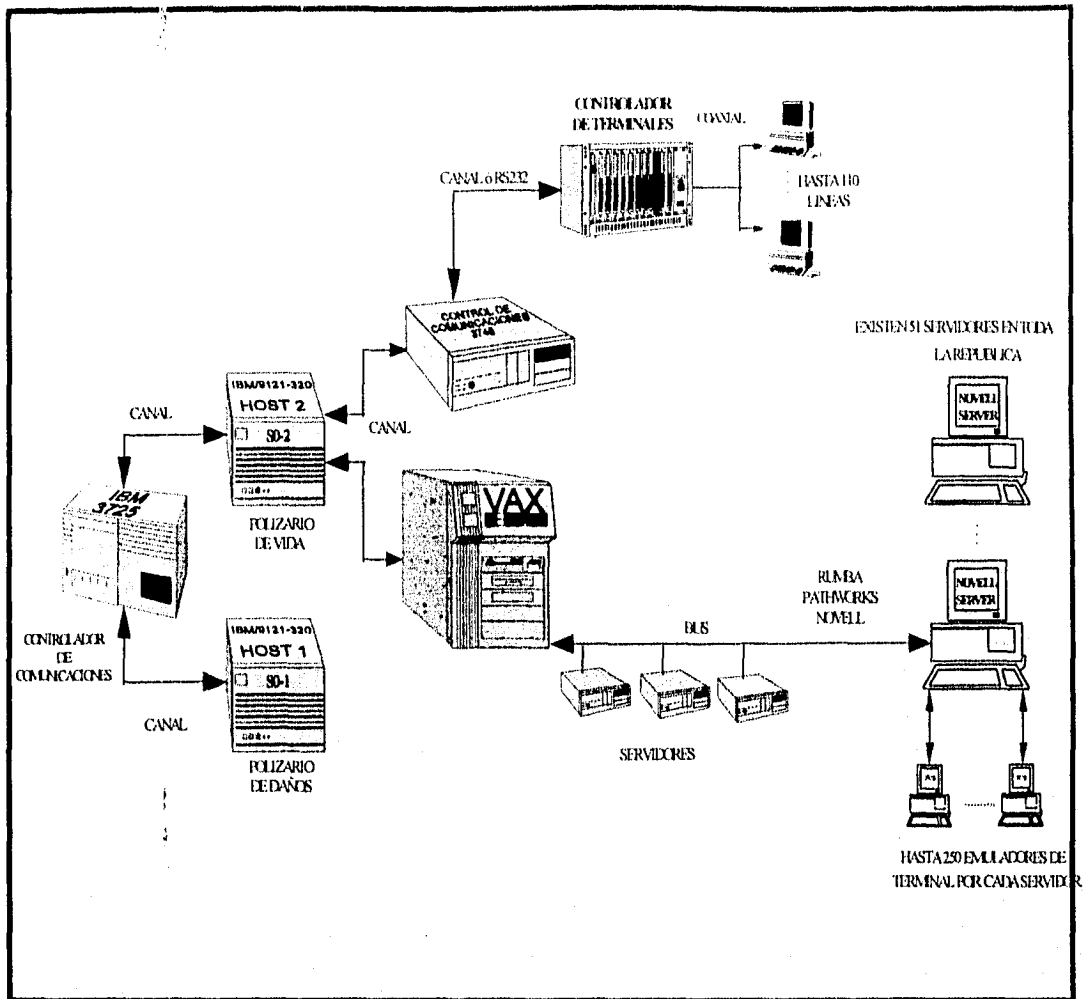


Fig. 2.16. Diagrama de conexión del equipo de cómputo.

Los sistemas DAÑOS (SO-1) y VIDA (SO-2) están desarrollados en lenguaje ensamblador y cuentan con diferentes módulos en los cuales se encuentra toda la información referente a las pólizas de GNP. La figura 2.17. muestra la pantalla principal para acceder a los sistemas SO-1 y SO-2, que se puede observar en una terminal.

Los módulos que integran al sistema (SO-1 y SO-2) pueden a su vez disponer de sistemas satélites los cuales están desarrollados en lenguaje COBOL, COBOL NATIVO y NATURAL, soportados por el manejador de base de datos IDMSDC. Los sistemas satélites corren en el equipo VAX que se utiliza como **interface** entre las PC's con emulación de terminal.

Debido a que la Gerencia de Cartera Huérfana tiene como prioridad principal atacar la Cartera Huérfana del Ramo de Vida sólo se requerirá de la información que se encuentra en los módulos V2 (VIDA) y V4 (IDMSDC) para el presente trabajo.

Esta información es desplegada generalmente en 14 pantallas numeradas secuencialmente de la S1 a S14. Dichas pantallas, excepto S1A, pueden contener pantallas adicionales que se designan alfabéticamente, esto ocurre cuando la información no cabe en una sola pantalla.

A continuación se muestra en las figuras 2.18. a 2.31., un ejemplo de las pantallas de una sola póliza huérfana, para ilustrar el número de pantallas que se deben recorrer para obtener la información requerida para controlar y reasignar una póliza.

La póliza de este ejemplo cuenta con 14 pantallas, sin embargo en el caso de algunas pólizas el número de pantallas puede ser mayor.

Es importante mencionar que para la reasignación y control de la Cartera Huérfana sólo se requieren ciertos datos de algunas de las pantallas, pero para encontrar los datos necesarios es menester navegar por todas las pantallas.

En algunos casos se deben realizar operaciones mecánicas con los datos para obtener cierta información. Un ejemplo de lo anterior, es cuando se necesita determinar la suma asegurada de la póliza. Si existen subpantallas de esta información, entonces será imprescindible utilizar una sumadora o calculadora para obtener el monto total.

Lo anterior ocasiona que se puedan cometer equivocaciones en los cálculos y que el tiempo de consulta se incremente excesivamente.

IIMG 00000226,CO=N1,S1A	VIGOR
ASEGURADO	CONTRATANTE
FREGOSO G., ALFONSO AV. MARINA NAL # 329 EDIF B 2 COL. ANAHUAC DELEG. MIGUEL HIDALGO 11320 MÉXICO, D.F.	

Fig. 2.18. Pantalla S1A.

IIMG 00000226,CO=N1,S2		SECCION DE CONTROL		VIGOR
CONTRATO	N100000226 0	FECHA ULT CONT	04/30/95	
FECHA EXTRAC.	04/01/96	COD. ULT CONT	1	
PAGADO HASTA	04/01/96	FECHA ULT OTROS	02/29/96	
COBRADO HASTA	04/01/97	COD. ULT OTROS	1	
DIA VENCIMIENTO	11	REGLA F. DE P.	04	
COD. VENCIM.	2	RECARGO	\$.00	
FORMA DE PAGO	12	REGLA RECARGO	1	
COND. COBRO	A1	PRIMA ATRASADA	0	
ESTADO	22	DIVISION	1	
SUSPENSO	0	COD. RECORDATORIO		
COD. MANEJO	4	RESTRICCION	00	
COD. COMISION	0	COD. DE REAS.	0	
PARCIALIDAD	\$.33	EDO OFNA. CD.	32 020	
CONTROL DE COB	0			
GARANTIZADA				
TRAILER DE PRESTAMO				
TIP INTPAG.A	TASA COD	INT ACUM	PRESTAMO	ULT ACT.
1 04/01/96	8.000 1	\$.00	\$4.04	04/01/95 0

Fig. 2.19. Pantalla S2.

IIMG 00000226,CO=N1,S3		VIGOR
CONTRL COBRANZA		MENSAJE ERROR
** NINGUN **		** NINGUN **
		LINEAS CTRL COBRANZA
		** NINGUN **

Fig. 2.20. Pantalla S3.

```

IIMG 00000226,CO=N1,S3A                                VIGOR
                TRAILER DE SUSP.
                **NINGUNO**
                PRIMAS MISCELANEAS EN DEPOSITO
                **NINGUNO**
                CONTROL DE DEP. ESP.
                **NINGUNO**
                PRIMA DESCONTADA
                **NINGUNO**
                AUM. SUMA ASEG.
                **NINGUNO**

```

Fig. 2.21. Pantalla S3A.

```

IIMG 00000226,CO=N1,S4                                VIGOR
                . 403 (B) TAX INFORMATION
                **NINGUNO**

```

Fig. 2.22. Pantalla S4.

```
IIMG 00000226,CO=N1,S5          VIGOR
      CAMBIO PRIMA                EXPIRAC. DE SEG. PRORR
      ** NINGUNO**                ** NINGUNO**

      MENSAJE GENERAL              CAMBIO BENEFICIO
      TYP PH NTFY DAT MESSAGE      ** NINGUNO**
      4 01 04/01/02 SAGB
      APL COUNT                     EXTRAPRIMAS DE VIDA
      TYP PH NTFY DAT NON-APL #NON-APL
      4 00 12/01/99 1 12           ** NINGUNO**
```

Fig. 2.23. Pantalla S5.

```
IIMG 00000226,CO=N1,S6          VIGOR
      NET COST TO POLICYHOLDER
      *** NONE ***

      ETI/RPU RESERVE
      *** NONE ***
```

Fig. 2.24. Pantalla S6.

IIMG 00000226,CO=N1,S7		BENEFICIOS BAS	VIGOR
FASE		01	
TARIFA		70	
CLS,BASE,SUB	1AVA07		
EDAD EMIS,VERD	34,34		
FECHA EMIS	04/01/41		
CTL EDAD,SEXO	0,1		
FECHA NAC	04/01/07		
UNIDADES	.013		
VALOR/UNIDAD	\$1,000.00		
SUMA ASEG	\$13.00		
PRIMA ANUAL	\$25.74		
VG,DIV,OTRA	3,0,0		
PART	1		
CNTL SEG P	6		
DURACION	55		
EV,BEN,COMIS	A,1,1		
TBL SNORMAL,COB	,1		
MRDM	0		
SALDAR,VENCER	04/03,04/03		
EDO DE EMIS	32		
DEV PRIMA			
COD DEV PRIMA			
RENTA MENSUAL			

Fig. 2.25. Pantalla S7.

IIMG 00000226,CO=N1,S8		BENEFICIOS ADICIONALE	VIGOR
*** NONE ***			
VALORES TEMPORAL			
*** NONE ***			
DEPOSITOS Y AUMENTOS			
*** NONE ***			
REMP. RENOVABLE-BENEFICIOS BASICOS			
*** NONE ***			
TEMP. RENOVABLE-BENEFICIOS ADICIONALES			
*** NONE ***			

Fig. 2.26. Pantalla S8.

```
IMG 00000226,CO=N1,S8Z                                VIGOR
                                AGENT COMMISSION CONTROL
PHSE,GP,SQ  1,1,1
START       04/41
STOP       04/03
COMM %     100.00
WRITING AGT 00001
COMM RATE  70.00
TYP,SIT,CD 1,1,DP
1ST AGENT
COMM RATE
LV,ST,CD
2ND AGENT
COMM RATE
LV,ST,CD
3RD AGENT
COMM RATE
LV,ST,CD
MARKET
COV,USER   1,
SR,ENT,EX  5,1,9
EFF DATE 04/01/41
ADV,PR,CON 0,0,0
ORIG MD,FM 012,A
```

Fig. 2.27. Pantalla S8Z.

```
IIMG 00000226,CO=N1,S12                                VIGOR
                                NOMBRE Y DIRECCION DEL AGENTE
AGT: 00001
NVL: 01  STAT: TERM
CLAVES ZONAS VACANTES
AV. CERRO DE LAS TORRES NO. 395
MEXICO
```

Fig. 2.28. Pantalla S12.

```

IMG 00000226,CO=N1,S12A                                VIGOR
      CLIENTE
-PR--NO.CLIENTE-
01 000000001281                                ROLES-C B NV P R
                                                01 00 01 00
-----UNIONES DE LA FASE-----
1
SEX  FECHA NAC  OCUPACION  ESTADO  PEO
1   04 01 1907
NOMB: FREGOSO G.,ALFONSO                                FEC. CAMBIO ESC.
RFC: 0000000000000000  IND/CIA I  RETENC. FED.: 1  MUERTE :
                                                03/01/1995  OCD
DIRECCION                                                DETALLE POLIZA
AV MARINA NAL # 329 EDIF B 2                            SISTEMA L70  NO. POLIZA: N100000226
COL.ANAHUAC                                                PERSONA 01  F. EMISION: 04/01/1941
DELEG. MIGUEL HIDALGO                                    SEC.TRLR.CLIENTE  R.IMPTO.: 0
11320 MEXICO, D.F.                                       SECD: 001 VIGENCIA ACT MEC/ANN:
CONTR:Y PAGAD:Y RECIBO :N ASEGUR.:Y
*** TRAILER HISTORICO DE LA DECLARACION ANUAL M-9 ***

* NINGUNO **
    
```

Fig. 2.29. Pantalla S12A.

```

IMG 00000226,CO=N1,S12B                                VIGOR
*****GUARANTEED PREMIUMS*****
* NINGUNO **
** TRAILER HISTORICO DEL PRESTAMO **
FASE  FECHA EFECTIVA  TIPO DE  TOTAL DEL  CREDITO
NO.   MES DIA  A|O  PRESTAMO  PRESTAMO  DISPONIBLE
01    04 01 1995    D          $ .32-    $11.17-
CREDITO  IMPUESTO  IMPUESTO  CANTIDAD
REPORTADO FEDERAL  ESTATAL  PENADA
$ .00    $ .00    $ .00    $ .00
** U-S SEVEN-PAY PREMIUM UTILITY TRAILER **
* NINGUNO **
    
```

Fig. 2.30. Pantalla S12B.

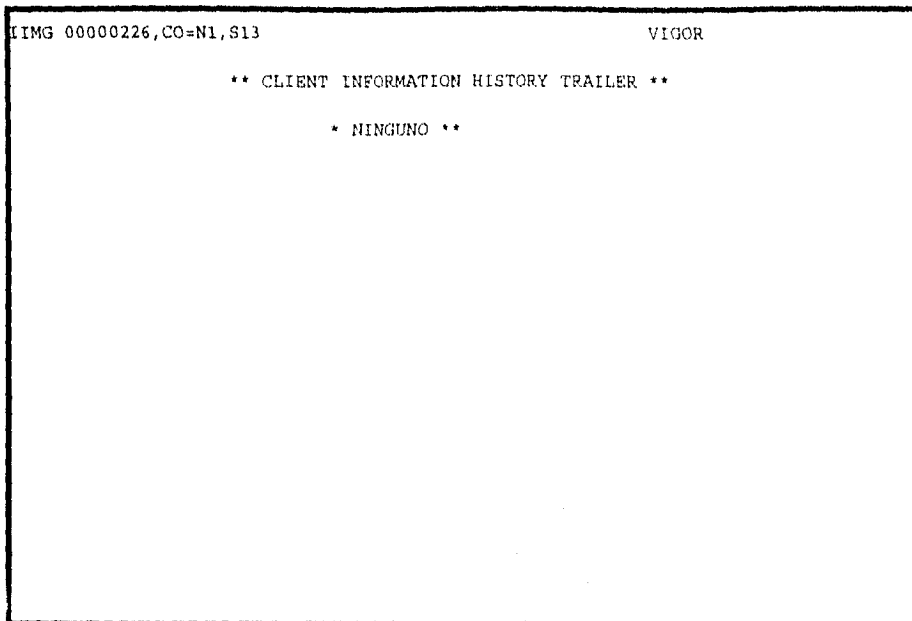


Fig. 2.31. Pantalla S13.

Dentro del módulo V4 (IDMSDC) se tienen varios sistemas satélites, de los cuales nos interesan para el control y reasignación de la Cartera Huérfana del Ramo de Vida sólo dos. **SISI** (Sistema Integral de Seguros de Inversión), **SAETA** (Sistema Automatizado de Estadísticas y Transacciones de Agentes) y **COCOA** (Control y Contratación de Agentes).

En SISI, se encuentran los siguientes datos:

- Monto del seguro de inversión
- Saldo disponible en efectivo
- Movimiento de pólizas de Línea Universal
- Número de retiros
- Pago de renovación

En SAETA, se tienen que consultar los siguientes datos:

- Nombre del agente
- Status del agente

- *Clave del agente*
- *Fecha de conexión*
- *Fecha de cancelación*
- *Zona*
- *Documentos por cobrar*
- *Informe de producción*
- *Informe de percepciones*
- *Informe de conservación*
- *Estados de cuenta*
- *Trasposos de cartera*

El sistema SAETA se actualiza con el sistema COCOA. Este sistema fue creado en PARADOX.

PROBLEMÁTICA

El tener la información referente a cada póliza en diferentes sistemas, que contienen además diversas pantallas, origina que la obtención de la información que se requiere para el manejo de la Cartera Huérfana sea muy complicada. Los sistemas están desarrollados en diversos lenguajes y plataformas y se ubican en diferentes host, lo que origina que se tenga que estar accedando (entrando con clave de usuario) constantemente a cada sistema, volviendo sumamente lenta y tediosa la consulta masiva, haciendo imposible la asesoría con el asegurado vía telefónica.

El no contar con un sistema que integre la información relativa a la Cartera Huérfana, ha generado los siguientes problemas:

- *Duplicidad de información.*
- *Pérdida de tiempo para recabar información.*
- *Es imposible dar atención al asegurado vía telefónica.*
- *Se puede dar información errónea al asegurado.*
- *Altos costos de capacitación al personal.*

Lacónicamente podemos decir que el problema principal de la Gerencia de Cartera Huérfana es no contar con una herramienta eficaz que integre de manera expedita la información de la Cartera Huérfana para permitir su control y la reasignación de agentes a las mismas.

2.3 SISTEMA PROPUESTO

El objetivo de esta tesis es proporcionar a la Gerencia de la Cartera Huérfana una herramienta de consulta, generada por un Sistema de Programación para el control y reasignación de agentes de la Cartera Huérfana de GNP en el Ramo de Vida, sistema al que nos referiremos desde este momento como **POHUER** (pólizas huérfanas). Este sistema permitirá integrar datos de diferentes plataformas en un ambiente multiusuario utilizando una **interface** gráfica, con el fin de ser más amigable al usuario.

Se propone generar el sistema por medio de emulación de terminal, que permita la integración de datos de las diferentes plataformas. El Sistema **POHUER**, hará uso de los sistemas en los que se encuentra toda la información del Ramo de Vida de GNP, es decir **SO-2**, y tomará sólo la información que sea de interés para el control y la reasignación de la Cartera Huérfana.

Para generar el sistema de emulación descrito será necesario utilizar una **interface** de comunicación que nos permita enlazar los diferentes sistemas que se utilizan para llevar el control de la Cartera Huérfana.

El sistema **POHUER** contará con cuatro secciones que se mencionan a continuación:

➤ **Consulta y Reasignación:** Esta sección permitirá consultar los agentes cancelados y las pólizas asignadas a éstos con el fin de reasignarlas. Por último, podrá llevar el control de las llamadas telefónicas que se hagan a los clientes para contactarlos. Lo anterior se dividirá en los siguientes subsecciones:

- a) Datos Generales de Póliza.
- b) Datos Generales de Agente.
- c) Resultado o Control de Llamadas.
- d) Datos Principales.

Los datos que se manejarán en las subsección de Datos Generales de una Póliza son los siguientes:

- ⇒ Clave de póliza
- ⇒ Nombre del asegurado
- ⇒ Fecha de emisión de la póliza

- ⇒ Fecha de vencimiento de la póliza
- ⇒ Fecha límite de cobro
- ⇒ Suma asegurada
- ⇒ Prima sin forma de pago
- ⇒ Monto de préstamo
- ⇒ Status de póliza
- ⇒ Forma de pago
- ⇒ Edad del asegurado al adquirir la póliza
- ⇒ Edad actual del asegurado
- ⇒ Beneficios adicionales

Los datos que se manejarán en la subsección de Datos Generales del Agente son:

- ⇒ Clave de póliza
- ⇒ Nombre del asegurado
- ⇒ Número de agente
- ⇒ Status del agente
- ⇒ Fecha de conexión
- ⇒ Fecha de cancelación
- ⇒ Zona a la que pertenecen

Los datos que se manejarán en la subsección de Resultados y Control de Llamadas son:

- ⇒ Clave de póliza
- ⇒ Nombre del asegurado
- ⇒ Teléfono del asegurado (casa y oficina)
- ⇒ Tipo de llamada (Tipología)
- ⇒ Fecha de llamada
- ⇒ Hora de llamada
- ⇒ Duración de llamada (minutos)
- ⇒ Resultado de la llamada (Se encontró al asegurado o no)
- ⇒ Fecha de próxima llamada
- ⇒ Hora de próxima llamada
- ⇒ Envío de telegrama (en caso de que el asegurado no cuente con teléfono)
- ⇒ Observaciones
- ⇒ Número de zona asignada

- ⇒ Número de agente asignado
- ⇒ Fecha de asignación del nuevo agente
- ⇒ Nombre del nuevo agente

La información requerida para la subsección de los Datos Principales son:

- ⇒ Clave de la póliza
- ⇒ Nombre del asegurado
- ⇒ Domicilio del asegurado
- ⇒ Teléfono del asegurado (casa y oficina)
- ⇒ Nombre del contratante
- ⇒ Domicilio del contratante
- ⇒ Teléfono del contratante (casa y oficina)
- ⇒ Número del agente
- ⇒ Tiempo acumulado de llamadas contactadas con los asegurados

➤ **Captura de Datos Adicionales:** Esta sección permitirá capturar datos de las referencias de los asegurados que no se encuentren en el sistemas de donde se extrae la información (SO-2). Se podrán capturar o corregir los datos del asegurado y contratante en caso de estar incompletos o incorrectos. Los datos que se manejarán en esta sección son los siguientes:

- ⇒ Clave de póliza
- ⇒ Nombre del asegurado
- ⇒ Domicilio del asegurado
- ⇒ Teléfono del asegurado (casa u oficina)
- ⇒ Nombre del contratante
- ⇒ Domicilio del contratante
- ⇒ Teléfono del contratante (casa u oficina)
- ⇒ Nombres de las referencias (pueden existir hasta 3)
- ⇒ Teléfonos de las referencias

➤ **Mantenimiento:** Esta sección servirá como módulo de actualización de datos de usuario y/o la tipología. Los submódulos en que se dividirán estas actividades son los siguientes:

- a) Usuarios
- b) Tipologías

La información a la que se puede dar mantenimiento para la subsección de Usuarios es la siguiente:

- ⇒ Clave de usuario
- ⇒ Password de usuario para entrar a la red
- ⇒ Clave de acceso para el sistema SISl
- ⇒ Clave de acceso para el sistema V4
- ⇒ Clave de acceso para el sistema V2
- ⇒ Teléfono del usuario

La información a la que se puede dar mantenimiento para la subsección de Tipologías es la siguiente:

- ⇒ Clave de llamada o resultado
- ⇒ Resultado de la llamada

Impresión de reportes: Las funciones de esta sección serán: generar telegramas (para el contacto con el asegurado en caso de no haber teléfono), generar consolidado de operación y un macroreporte en donde se puedan obtener datos específicos que se requieran en determinado momento. La segunda parte de este módulo tendrá la posibilidad de imprimir los telegramas previamente creados, impresión de estadísticas, un macroreporte previamente generado, listas de agentes cancelados y dispositivos. Estas actividades estarán divididas entre las siguientes subsecciones:

- a) Creación de Telegramas.
- b) Impresión de Telegramas.
- c) Consolidado de Operación.
- d) Histórico de Telegramas.
- e) Megareporte.
- f) Estadísticas del Universo de Datos.
- g) Reporte de Agentes Dispositores.
- h) Reporte de Agentes Cancelados.

La información requerida para la subsección de Creación de Telegramas es la siguiente:

- ⇒ Fecha de generación de telegrama
- ⇒ Número de centro regional
- ⇒ Número de zona
- ⇒ Status
- ⇒ Número de póliza
- ⇒ Número de agente
- ⇒ Rango de primas

La información requerida para la subsección de Impresión de Telegramas es la siguiente:

- ⇒ Fecha de impresión del telegrama
- ⇒ Nombre del asegurado
- ⇒ Dirección del asegurado
- ⇒ Nombre del usuario
- ⇒ Teléfono del usuario donde pueda ser contactado

En el caso de la subsección de Consolidado de Operación se llevará un control de las llamadas realizadas y el tiempo de duración de cada llamada, por cada usuario. Por último se llevará un control sobre la tipología de cada llamada.

- ⇒ Clave de usuario
- ⇒ Número de llamadas realizadas por cada usuario
- ⇒ Tiempo total de llamadas por usuario
- ⇒ Tipología de llamada
- ⇒ Contestación
- ⇒ Totales generales de la información antes mencionada

En la subsección de Histórico de Telegramas se registrarán el número de telegramas enviados a cada asegurado, la fecha en que se crearon y el usuario que los envió. En este reporte se manejarán los siguientes datos:

- ⇒ Fecha de creación del telegrama
- ⇒ Fecha de envío del telegrama
- ⇒ Usuario que envió el telegrama
- ⇒ Número de telegramas enviados a un mismo asegurado

Para la subsección de Macroreporte se podrá obtener información específica de las pólizas. La información consultada quedará conjuntada según lo que se requiera en un determinado momento. Esta subsección podrá generar reportes con datos del asegurado, datos del contratante, datos del agente actual, datos de la póliza, datos de los resultados de las llamadas telefónicas, y datos de los agentes nuevos. Se debe recalcar que el usuario puede obtener los datos que quiera y no necesariamente obtendrá todos los datos que a continuación se mencionan:

- ⇒ Clave de póliza
- ⇒ Nombre del asegurado
- ⇒ Fecha de emisión de la póliza
- ⇒ Fecha de vencimiento de la póliza
- ⇒ Fecha límite de cobro
- ⇒ Suma asegurada
- ⇒ Prima sin forma de pago
- ⇒ Monto del préstamo
- ⇒ Status de póliza
- ⇒ Forma de pago
- ⇒ Edad del asegurado al adquirir la póliza
- ⇒ Edad actual del asegurado
- ⇒ Beneficios adicionales
- ⇒ Número de agente
- ⇒ Status del agente
- ⇒ Fecha de conexión
- ⇒ Fecha de cancelación
- ⇒ Zona a la que pertenece la póliza
- ⇒ Teléfono del asegurado (casa y oficina)
- ⇒ Tipo de llamada (Tipología)
- ⇒ Fecha de llamada
- ⇒ Hora de llamada
- ⇒ Duración de llamada (minutos)
- ⇒ Resultado de la llamada (Se encontró al asegurado o no)
- ⇒ Fecha de próxima llamada
- ⇒ Hora de próxima llamada
- ⇒ Envío de telegrama (en caso de que el asegurado no cuente con teléfono)
- ⇒ Observaciones
- ⇒ Número de zona asignada
- ⇒ Número de agente asignado
- ⇒ Fecha de asignación del nuevo agente
- ⇒ Nombre del nuevo agente

- ⇒ Clave del nuevo agente
- ⇒ Zona del nuevo agente

La subsección de Estadísticas del Universo Total de datos, emitirá reportes de estadísticas de todos los agentes cancelados. La información que se manejará en esta subsección es la siguiente:

- ⇒ Centro regional
- ⇒ Rangos de primas
- ⇒ Zona
- ⇒ Status
- ⇒ Número total de pólizas de una zona determinada
- ⇒ Suma total de primas (año prima)

La subsección de Agentes Dispositores creará un listado de todos los agentes que se encuentren con status 06. Los datos que se requerirán para realizar esta actividad son:

- ⇒ Fecha de conexión del agente
- ⇒ Fecha de cancelación del agente
- ⇒ Clave del agente
- ⇒ Clave provincial

En la subsección de Reporte de Agente Cancelado se creará un listado de todos los agentes que tienen un status de tipo cancelatorio. Los datos requeridos para este proceso son:

- ⇒ Fecha de conexión del agente
- ⇒ Fecha de cancelación del agente
- ⇒ Clave del agente
- ⇒ Clave provincial

En la figura 2.320. se pueden apreciar los diferentes módulos que formarán parte del sistema POHUEP.

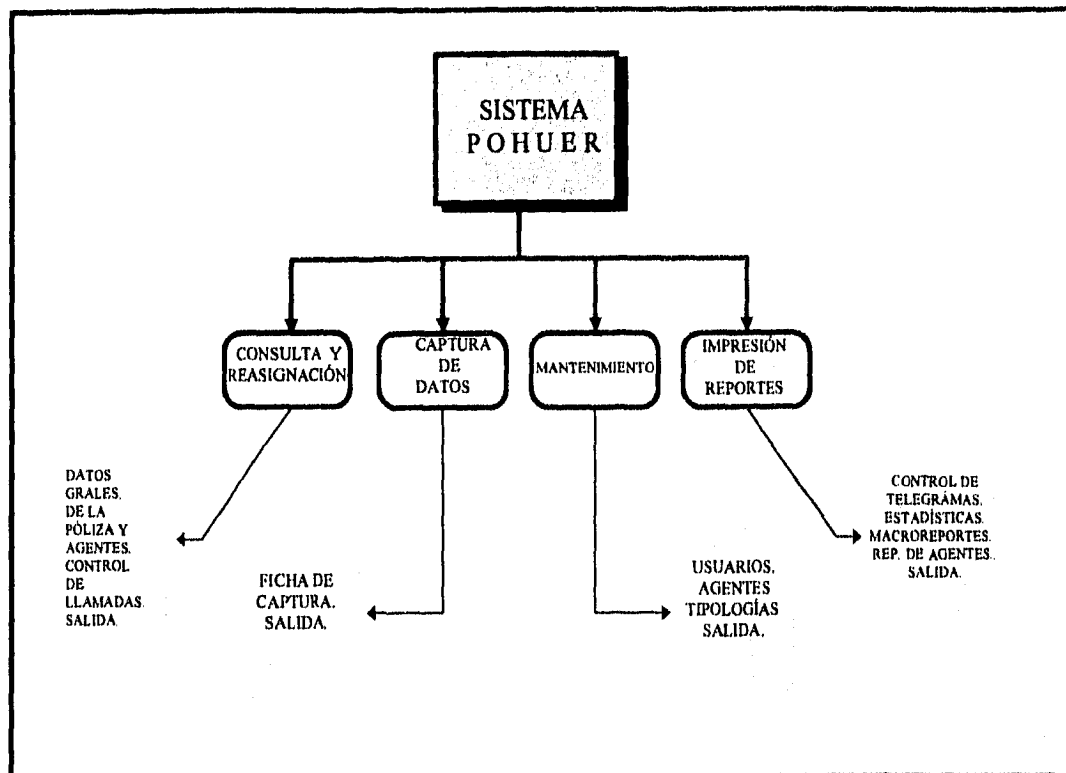


Fig. 2.32. Módulos del sistema POHUER.

Al desarrollar el sistema POHUER y enlazar los diferentes sistemas que operan en terminales y computadoras por medio de una red, se obtendrán grandes beneficios, entre éstos tenemos:

- Tener acceso al mismo tiempo a varios sistemas de información sin tener que abandonar el sistema en uso.
- Presentar la información de forma organizada en pantallas de fácil y raudo acceso.
- Minimizar el tiempo de capacitación de usuarios.
- Proporcionar ayuda en línea.
- Agilizar la obtención de información.

El sistema que se acaba de describir pretende aprovechar al máximo los recursos con los que cuenta GNP, tanto de software como de hardware, con el fin de disminuir los costos de su implementación.

REQUERIMIENTOS DEL SISTEMA

A continuación se describirá el software y hardware requerido para el funcionamiento del Sistema de Control y Reasignación de Agentes para la Cartera Huérfana de GNP.

*Hardware: PC/486 con emulación de terminal con acceso a los diferentes sistemas (V2, SISI, SAETA), con monitor SVGA, 8 Mb de RAM.
Impresora laser.
FAX.
Tarjeta HP de 16 bits para red.
Módem 2400 Mbps.*

*Software: Sistema Operativo DOS V.6.
Sistema Operativo Netware Novell.
Windows V.3.2.
Sistemas V2, SISI, SAETA.
Sistema de comunicaciones RUMBA.*

ALTERNATIVAS PARA LA SELECCIÓN DE UNA BASE DE DATOS

La selección de la base de datos adecuada con la que se llevará a cabo el desarrollo del sistema POHUER, deberá considerar las responsabilidades con las que cumplirá el sistema. Para este efecto se analizaron las características principales de algunas bases de datos. En este análisis se consideró a las bases de INFORMIX, DBASE, PARADOX, ORACLE, PROGRESS, Y SYBASE. Las características principales que se evaluaron fueron las siguientes: requerimientos de hardware, clientes que soporta, sistema operativo de red, interface gráfica del sistema operativo de red, sistema manejador de la base de datos y/o lenguaje, herramientas para desarrollo de aplicaciones y los módulos de conectividad preestablecidos. En la figura 2.33. se presenta una tabla donde se comparan las bases de datos mencionadas.

No obstante que el Sistema POHUER puede ser implementado en algunas de las bases de datos que se presentan en la tabla de la figura 2.33., se decidió utilizar PARADOX, debido a que la Gerencia de Normatividad y Control de GNP tiene, en estos momentos, estándares de desarrollo de software que deben de seguir todos los departamentos dentro de GNP, y uno de los estándares que maneja es el lenguaje de base de datos PARADOX. Cabe mencionar también, que la Gerencia de Cartera Huérfana sólo cuenta con PARADOX como herramienta de desarrollo para el sistema y en caso de escoger otra herramienta, tendría que adquirirla. Otra razón para elegir PARADOX es que el sistema POHUER requerirá la información de los agentes que se encuentra contenida dentro del sistema COCOA, el cuál se encuentra desarrollado en PARADOX.

CARACTERÍSTICAS	INFORMIX	DBASE	PARADOX	ORACLE	PROGRESS	SYBASE
Requerimientos de hardware	Mainframes, Sistemas abiertos. Prácticamente se puede instalar en cualquier plataforma y los requerimientos varían de acuerdo al equipo.	9 Mb en disco. IBM PC o compatible con un procesador 386 o mayor.	6-24 Mb en disco. IBM PC o compatible con un procesador 386 o mayor.	HP 9000, HP 3000 MPE/IX, DEC Ultrix, VAX MVS, NCR Tower, IBM RS/600, PC's con procesador 386 o mayor.	Procesador 386 o mayor de 25 a 18 Mb en disco. Estaciones de trabajo basadas en RISC de 500Kb a 4 Mb en memoria para usuario.	72-111 Mb en disco, 16 Mb RAM. Este producto corre en las plataformas más utilizadas en el mundo.
Cientes que soporta	UNIX, WINDOWS, Macintosh	DOS, WINDOWS	MS-DOS, WINDOWS	MS-WINDOWS, MACINTOSH, OS-2, MS-DOS, DG-UX, Open, VMS, UNIX.	IBM-AIX, IIP-UX, UNIX, NCRO, SUN OS, ULTRIX, SCO-UNIX, MS-DOS, OS-2	UNIX, VMS, OS-2, WINDOWS NT, MS-DOS, SOLARIS, DG-UX
Sistema operativo de red	NOVELL, WINDOWS NT	NOVELL, WINDOWS NT, DECNET, PATHWORKS, IBM LAN SERVERS6000, MICROSOFT LAN MANAGER	NOVELL, WINDOWS NT, VINES, DECNET, Pathworks, IBM LAN SERVER, MICROSOFT LAN MANAGER, UNIX WARE	NOVELL, WINDOWS NT, SCO-UNIX, OS-2	MS-WINDOWS, Motif, Character, UNIX	UNIX, VMS, OS-2, WINDOWS NT, NETWARE
Interface gráfica del sistema operativo en red	WINDOWS, MOTIF, MACINTOSH, CHARACTER.	WINDOWS	WINDOWS	WINDOWS, Motif, MACINTOSH	RDBMS, HCL, WSQ/C, SQL, ODBS, DDE, C	WINDOWS, Motif, MACINTOSH, OS-2
Sistema manejador de la base de datos y/o lenguaje	RDBMS, ODBC, C, COBOL.	RDBMS, DBASE, DDE, DLL, C, C++, Pascal, SQL, ODBC	RDSMS, DBASE, DDE, DDL, C, C++, Pascal, SQL, ODBC	RDBMS, SQL, PL/SQL, C, C++, FORTRAN, COBOL,	AMBIENTE DE DESARROLLO DE APLICACIONES (ADE), 4GL ADE.	RDBMS, ANSI SQL, Transact SQL.
Herramientas para desarrollo de aplicación	INFORMIX - NEW ERA, INFORMIX-4GL, INFORMIX-MENUS, INFORMIX-SQL, INFORMIX-ESQL, PAR, C O COBOL	Two way tools, Form Expert, Crystal Reports for Dbase, Borland Database Engine 2.0, Report Smith 2.5	Two way tools, Form Expert, Crystal Reports for Dbase, Borland Database Engine 2.0, Report Smith 2.5	Two way tools, Form Expert, Crystal Reports for Dbase, Borland Database Engine 2.0, Report Smith 2.5	ORACLE, RMS, Rdb/VMS, C-ISAM, CT-ISAM, DB2, OBJECT ACCESS, OBJECT STORE, SYBASE	Las aplicaciones se pueden desarrollar en más de 125 herramientas de desarrollo distintas
Módulos de conectividad preestablecidos	ORACLE, RMS, DB2, OBJECT ACCESS, OBJECT STORE, SYBASE.	DBASE, PARADOX, ORACLE, SYBASE, MS/SQL SERVER, INFORMIX, INTERBASE	DBASE, PARADOX, ORACLE, SYBASE, MS/SQL SERVER, INFORMIX, INTERBASE	DBASE, PARADOX, ORACLE, SYBASE	DB2, SQL, Rdb	Tiene más de 1000 asociados en aplic.

Fig. 2.33. Comparativo de bases de datos relacionales.

Esquema Centralizado

El sistema POHUER utilizará el esquema centralizado, en donde toda la información se almacena en un mismo equipo, y de acuerdo a este tipo de esquema los accesos a la base de datos se harán por medio de terminales remotas, lo que mantiene a la base de datos totalmente actualizada y por ende, la fiabilidad de encontrar a todos los agentes que han sido cancelados.

Ventajas

- *Todos los procesos de administración del equipo y la base de datos son centralizados.*
- *No se necesita capacitar a personal adicional para la administración y operación del equipo.*
- *El sistema está totalmente actualizado.*
- *Veracidad en encontrar a todos los agentes que han sido cancelados.*
- *Reasignación rápida de agentes a las pólizas huérfanas.*
- *Aviso oportuno a los asegurados del cambio de agente.*

Desventajas

- *Se deberá tener un sistema robusto, el cual pueda atender todas las necesidades de procesamiento y almacenamiento.*
- *Es indispensable implantar un sistema de seguridad paralelo o de respaldo para evitar cualquier tipo de eventualidades que pudieran surgir durante la operación del sistema.*

2.4 PLANEACIÓN DE ACTIVIDADES

La administración efectiva de un proyecto de software depende de la planeación detallada de sus actividades y avance, previniendo problemas que puedan surgir y preparando por anticipado las soluciones tentativas a ellos.

La planeación de un sistema contempla desde la definición de requisitos hasta la entrega del sistema terminado.

Los diagramas de barras son notaciones gráficas que se pueden utilizar en la programación del tiempo de un proyecto. La programación del tiempo y la estimación de costos

del software están íntimamente relacionados. La mayor parte de los costos de un proyecto de software grande son sólo aquellos relativos al pago de la gente que escribe el software.

El tamaño del sistema se calcula por medio de la definición de requisitos, para establecer las partes que componen el sistema.

Una vez identificadas las necesidades de la Gerencia de Cartera Huérfana y habiendo planteado una propuesta, haremos uso de un diagrama de barras para definir un plan de trabajo y hacer una programación de tiempo de desarrollo del sistema POHUER.

A continuación se presenta en la figura 2.34. el plan de trabajo a seguir para la implementación del sistema POHUER.

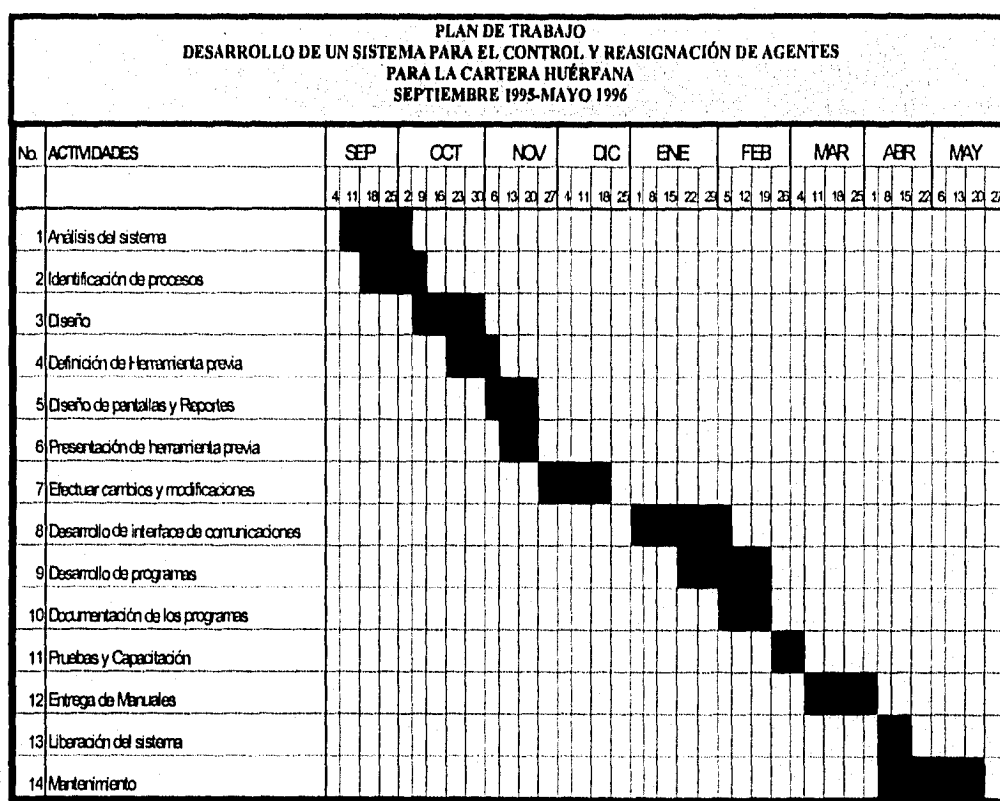


Fig. 2.34. Plan de trabajo para el sistema POHUER.

2.5 ANÁLISIS DEL SISTEMA

Para llevar a cabo el análisis del sistema es indispensable identificar los procesos que se deben realizar para reasignar agentes a la Cartera Huérfana de GNP. Estos procesos son los siguientes:

- a) Identificar los agentes que han sido cancelados.
- b) Detectar pólizas vigentes de los agentes cancelados.
- c) Solicitar autorización a la gerencia de zona.
- d) Establecer contacto con el asegurado.
- e) Cobrar, reasignar y notificar a la gerencia de zona.

a) *Identificar los agentes que han sido cancelados.*

Para identificar a los agentes cancelados, será necesario acceder a determinados datos del sistema COCOA para recabar la información referente al agente (nombre, status, clave, zona, etc.). Se debe mencionar que COCOA maneja indistintamente a todos los agentes. Por lo anterior será necesario primero, identificar a los agentes cancelados mediante su status y clasificarlos de acuerdo a su centro regional y zona. Posteriormente se clasificará al agente dependiendo del status al que pertenezca para tomar una decisión sobre el manejo que se dará a las pólizas que éstos tenían asignadas.

El motivo de cancelación de un agente (status) esta clasificado como se observa en la figura 2.35.

STATUS	DESCRIPCION
04	BAJA PRODUCCIÓN
05	CONVENIR A LA COMPAÑIA
06	DISPOSICIÓN DE PRIMA
07	FALLECIMIENTO
08	FALTA DE REFRENDO
09	CÉDULA PROVISIONAL VENCIDA
12	CANCELACIÓN POR DOCTOS. FALTAN
13	REINCIDENCIA CAMPAÑA
14	SIN CERTIFICADO DE CURSOS
15	DOCTOS. EN TRAMITE
41	CANC. CAMPAÑA

Fig. 2.35. Tabla de Status.

b) *Detectar pólizas vigentes de los agentes cancelados.*

Una vez que se clasifiquen los agentes cancelados será necesario identificar cuales son las pólizas que tenían asignadas. Para investigar la información correspondiente a la pólizas se procederá a verificar la información del sistema V2 que se encuentra en uno de los host. La información de la póliza que contiene V2 es la siguiente:

- *Fecha de emisión y cancelación de la póliza.*
- *Vigencia y duración.*
- *Datos del asegurado y del contratante.*
- *Cobertura de la póliza.*
- *Suma asegurada.*
- *Monto de la prima.*

Será necesario acceder al sistema SISI (Sistema Integral de Seguro de Inversiones) para investigar el monto del seguro de inversión (no todas las pólizas tienen seguro de inversión hay que validarlo). Con toda esta información se podrá integrar la información de pólizas huérfanas que ha generado cada agente.

c) Solicitar autorización a la gerencia de zona.

Una vez reconocida la Cartera Huérfana, se considerarán primero aquellas pólizas que tienen un monto de prima superior, posteriormente dependiendo del centro regional se pedirá autorización a la gerencia de zona correspondiente para reasignar un nuevo agente a la póliza huérfana.

d) Establecer contacto con el asegurado.

Cuando la Gerencia de Zona autorice la reasignación de un agente a una póliza huérfana el siguiente paso será indagar los datos del asegurado que se encuentran en el sistema V2 para contactarlo por vía telefónica ó telegrama. Lo anterior con el propósito de restablecer comunicación con el asegurado para informarle acerca del estado actual de su póliza y proponerle la reasignación de un nuevo agente.

Esta actividad no corresponde propiamente al sistema, pero será necesario contemplarla para el proceso de reasignación de un agente a la póliza huérfana.

e) Reasignación de agente y cobro a las pólizas huérfanas.

Si cuando se contacta al asegurado se tiene una respuesta positiva, es decir, que al informarle al asegurado que su agente ya no trabaja para la empresa, éste acepta que se le asigne uno nuevo, entonces se actualizarán tanto los datos del agente como los de la póliza, por lo tanto, la póliza deja de pertenecer a la Cartera Huérfana.

Para la mejor comprensión de los procesos que se seguirán para el control y reasignación de la Cartera Huérfana, en la figura 2.36. se mostrará un esquema que ilustra con claridad la secuencia de estos procesos.

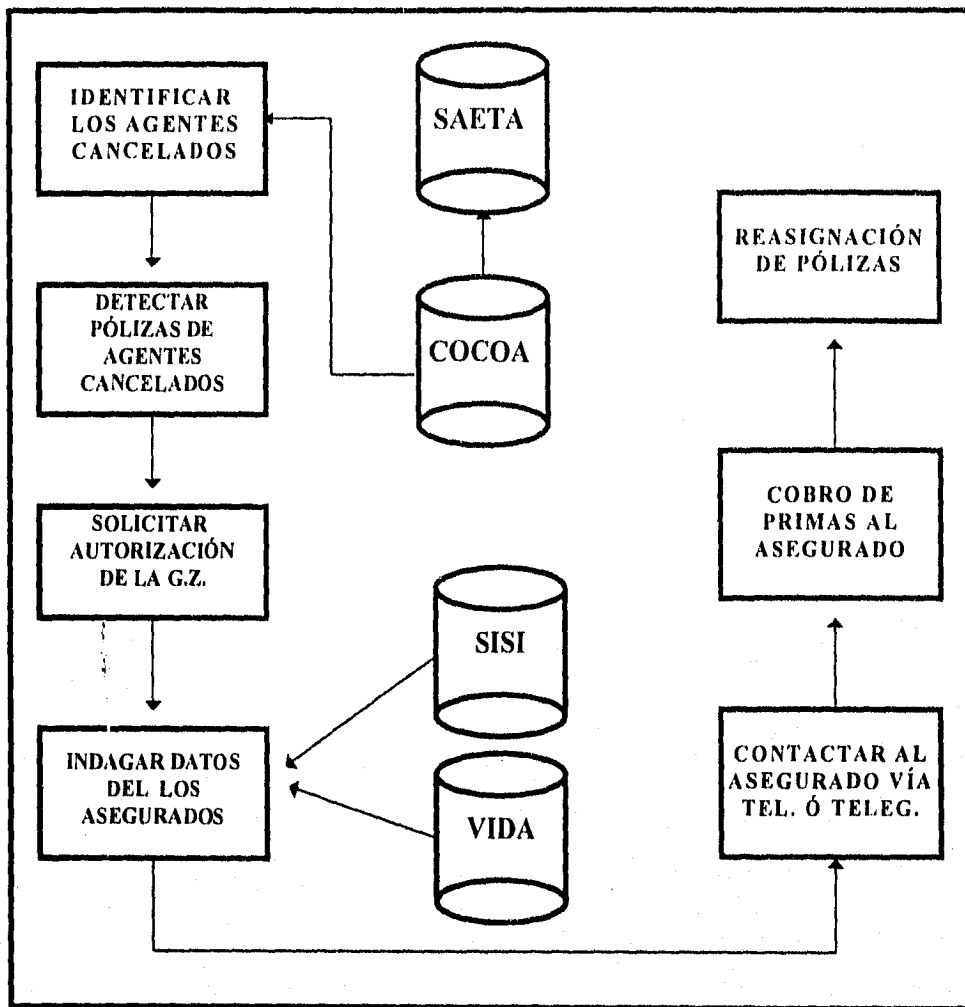


Fig. 2.36. Diagrama de procesos para reasignar la Cartera Huérfana.

La forma en que se piensa facilitar las actividades a la Gerencia de Cartera Huérfana es mediante un Sistema de Cómputo para el Control y Reasignación de la Cartera Huérfana de GNP. Para desarrollar dicho sistema se hará uso del Ciclo de Vida del "Modelo V o cascada" antes descrito. Tomando lo anterior en cuenta, en este capítulo se analizó la situación actual por la que atraviesa la Gerencia de Cartera Huérfana, para así definir los objetivos del sistema y especificar sus requerimientos.

Por último, una vez analizados los objetivos del sistema, se establecieron los procedimientos para el control y la reasignación de agentes a la Cartera Huérfana de GNP.

CAPÍTULO III

DISEÑO

En este capítulo se mencionarán algunos conceptos básicos sobre diseño y bases de datos. Se analizarán con lujo de detalle las características que deberá tener el sistema POHUER para llevar a cabo posteriormente su desarrollo. Se hará uso de metodologías (integradas por una serie de herramientas y métodos) comúnmente utilizadas en el diseño, como son: los diagramas de flujos de datos (DFD), diccionario de datos (DD), el modelo entidad relación (E-R) y la normalización.

La metodología que se utilizará para el diseño del sistema POHUER es la de Chris Gane & Trish Sarson, quienes han llegado a ser muy populares como ingenieros de software en los últimos tiempos, gracias a la fama de esta metodología. La metodología consta de siete pasos:

- 1. Dibujar el diagrama de flujo de datos.*
- 2. Construir el diccionario de datos.*
- 3. Dibujar el diagrama entidad relación.*
- 4. Aplicar normalización.*
- 5. Revisar los diagramas de flujo de datos con precisión.*
- 6. Partir el modelo lógico en procesos y datos en procedimientos simples.*
- 7. Especificar detalles de cada procedimiento que puedan ser requeridos para la implantación del sistema.*

3.1 CONCEPTOS BÁSICOS

ALMACENAMIENTO DE DATOS

Generalmente la unidad básica de almacenamiento es el carácter, que puede ser un número o una letra. El carácter ocupa el nivel inferior de almacenamiento y casi nunca se trabaja directamente con ellos aisladamente, sino con una agrupación que por sí misma posee una significación, estos grupos de caracteres se denominan campos.

Los campos se agrupan para formar lo que se conoce como registro. Cada registro contiene la información completa de un ente en especial, a su vez el conjunto de todos los registros constituye un archivo.

Una clave primaria (por lo general, denominada llave o clave solamente) es un campo (o grupo de campos) que contiene datos, los cuales sirven para identificar cada uno de los registros de una tabla. Las claves necesitan un valor único de cada registro (fila) de la tabla, esto evita que haya registros duplicados en una misma tabla. Las tablas que disponen de claves identificadoras reciben el nombre de tablas con clave.

Es posible crear una clave a partir de un campo o grupo de campos; cuando se especifica un grupo de campos, éste recibe el nombre de clave compuesta. Estas se crean cuando ninguno de los campos por separados son suficientes para identificar un registro como único, por lo tanto la clave se crea por la combinación de dos o más campos. Estas claves son creadas para tener la certeza de que cada registro de una tabla es único, deben incluirse en la clave compuesta los campos que sean necesarios para su identificación.

BASE DE DATOS Y ARCHIVOS CONVENCIONALES

Los Sistemas Manejadores de Base de Datos son un conjunto de archivos independientes y relacionados entre sí (Bases de Datos) y una variedad de programas para acceder la información. El objetivo de estos sistemas es crear una interface transparente al usuario para guardar, modificar y recuperar la información de la base de datos en forma eficiente.

Durante la década de los 60's los Sistemas de Base de Datos toman gran importancia teórica y comercial. En los siguientes años este campo de la computación sufre grandes cambios conceptuales y tecnológicos. Los Sistemas de Base de Datos se crean para mantenimiento y explotación de un gran volumen de información. Para el análisis y comprensión de los Sistemas de Base de Datos se describen a continuación los conceptos fundamentales.

En un sistema de procesamiento de archivos convencional, los datos del sistema se guardan en diversos archivos permanentes, creados durante el tiempo con el fin de satisfacer distintas necesidades de la organización, y se escriben varios programas de aplicaciones para agregar y extraer información. Estos sistemas tienen ciertas desventajas importantes.

Redundancia e inconsistencia de los datos: Es posible que un mismo dato esté repetido en varios sitios (archivos). Esta redundancia aumenta el costo de almacenamiento y acceso, además de incrementar la posibilidad de que exista inconsistencia en la información, es decir, que las distintas copias de la misma información no concuerdan entre sí.

Dificultad para tener acceso a los datos: Este ambiente no permite recuperar información requerida en forma conveniente o eficiente, dada su poca flexibilidad tanto en las estructuras utilizadas para el almacenamiento de la información como en los programas de aplicación específicos. Por lo tanto deben desarrollarse sistemas de recuperación de información de aplicación general.

Aislamiento de los datos: Dado que los datos están almacenados en varios archivos y éstos pueden tener diferentes formatos, es difícil escribir nuevos programas para extraer datos apropiados.

Usuarios múltiples: En un ambiente multiusuario, la interacción de las actualizaciones concurrentes puede resultar en información inconsistente, puesto que muchos programas de aplicación diferentes sin coordinación previa pueden tener acceso a los datos.

Problemas de seguridad: No es recomendable que todos los usuarios del sistema de base de datos puedan tener acceso a toda la información, puesto que los programas de aplicaciones se agregan al sistema en una forma precisa, es difícil implantar estas limitantes de seguridad.

Problemas de integridad: Los valores de los datos que se guardan en la base de datos deben de satisfacer ciertos tipos de limitantes de consistencia. El sistema debe obligar al cumplimiento de estas limitantes. Esto puede hacerse agregando el código apropiado a los distintos programas de aplicaciones. Sin embargo, cuando se añaden nuevas limitantes, es difícil cambiar los programas para asegurar su cumplimiento. El problema se complica aún más cuando las limitantes implican varios elementos de información de distintos archivos.

Estos y otros problemas han fomentado el desarrollo de los sistemas de manejo de Base de Datos. Al manejar gran cantidad de información los sistemas de base de datos involucran ciertos conceptos necesarios para su eficiencia.

La integridad en un sistema de base de datos implica considerar la unificación de varios archivos de datos independientes, es decir, se elimina la redundancia total o parcial entre los datos, cuando se requiere insertar un valor en la base de datos se deben satisfacer limitantes de consistencia.

Una consecuencia de la integridad en la base de datos es la compartibilidad de la misma, esta característica consiste en la capacidad de que varios usuarios puedan acceder al mismo subconjunto de la base de datos para ser utilizada con diferentes propósitos.

ABSTRACCIÓN DE LA INFORMACIÓN

Uno de los objetivos principales de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de la información. Es decir, el sistema oculta ciertos detalles relativos a la forma como los datos se almacenan y mantienen. Sin embargo, para que el sistema sea útil, la información debe recuperarse en forma eficiente. Existen tres niveles de abstracción de la información.

a) Nivel Físico: Este nivel comprende la forma real de almacenamiento de los datos. En este nivel se describen en detalle las estructuras de datos complejas del nivel más bajo.

b) Nivel Conceptual: En este segundo nivel se describen los datos reales almacenados y las relaciones que existen entre ellos. Este nivel contiene a toda la base de datos en términos de unas cuantas estructuras relativamente sencillas. El nivel conceptual lo utilizan los administradores de la base de datos, quienes deciden que información se guarda en la base de datos.

c) Nivel de Visión: Para este nivel, el sistema de base de datos puede mostrar diferentes vistas para cada usuario. Este nivel lo utilizan los administradores de las bases de datos, para restringir el acceso a determinados registros de una tabla de información, proporciona una tabla virtual sin crear una nueva.

MODELO DE DATOS

Los modelos de datos son un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes, los cuales se pueden dividir en tres grupos:

a) Modelos lógicos basados en objetos. Los modelos lógicos basados en objetos se utilizan para describir los datos en los niveles conceptual y de visión, se caracterizan por el hecho de que permiten una estructura bastante flexible y hacen posible especificar claramente las limitantes de los datos. Existen diversos modelos de este tipo, entre los que se encuentran:

- El modelo entidad-relación
- El modelo binario
- El modelo semántico de datos
- El modelo infológico

El representativo de la clase es el modelo entidad-relación, este modelo de datos entidad-relación se basa en una percepción de un mundo real, que consiste en un conjunto de objetos básicos llamado entidades y de las relaciones entre los objetos.

b) *Modelos lógicos basados en objetos y en registros.* Los modelos lógicos basados en objetos y en registros se utilizan para describir los datos en los niveles conceptuales y de visión, a diferencia de los modelos de datos basados en objetos, estos modelos sirven para especificar tanto la estructura lógica general de la base de datos como una descripción en un nivel más alto de la implantación, pero no permite especificar en forma clara las limitantes de los datos. Los modelos más comunes son:

- **Modelo Relacional.** Este modelo describe los datos y las relaciones por medio de una serie de tablas, cada una de las cuales tiene varias columnas con nombres únicos, como se observa en la figura 3.1.

nombre	calle	ciudad	número
Martín	Hidalgo	Tampico	900
Guadalupe	Juárez	Guadalajara	556
Guadalupe	Juárez	Guadalajara	647
Enrique	Chapultepec	Cd. de México	801
Enrique	Chapultepec	Cd. de México	647

número	saldo
900	55
556	10000
647	105366
801	10533

Fig. 3.1. Ejemplo de tablas usadas en el modelo relacional.

- **Modelo de Red.** Este modelo de datos se representa por medio de conjuntos de registros y las relaciones entre los datos se representan con ligas que pueden considerarse como apuntadores, los registros de la base de datos se organizan en forma de conjunto de gráficas arbitrarias, como se observa en la figura 3.2.

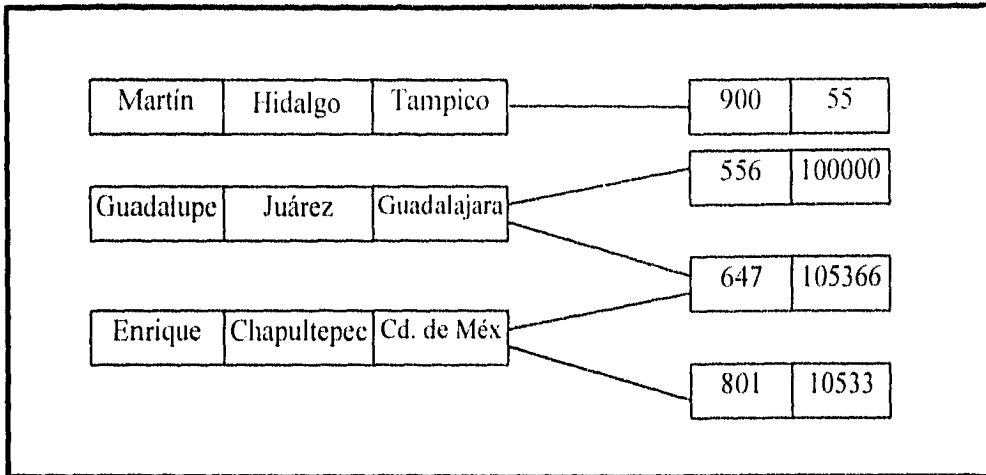


Fig. 3.2. Ejemplo de un esquema del modelo de red.

- **Modelo Jerárquico.** El modelo jerárquico es similar al modelo de red en cuanto a que los datos y las relaciones entre los datos se representan por medio de registros. Estos registros están organizados como conjuntos de árboles, en vez de gráficas arbitrarias, como se observa en la figura 3.3.

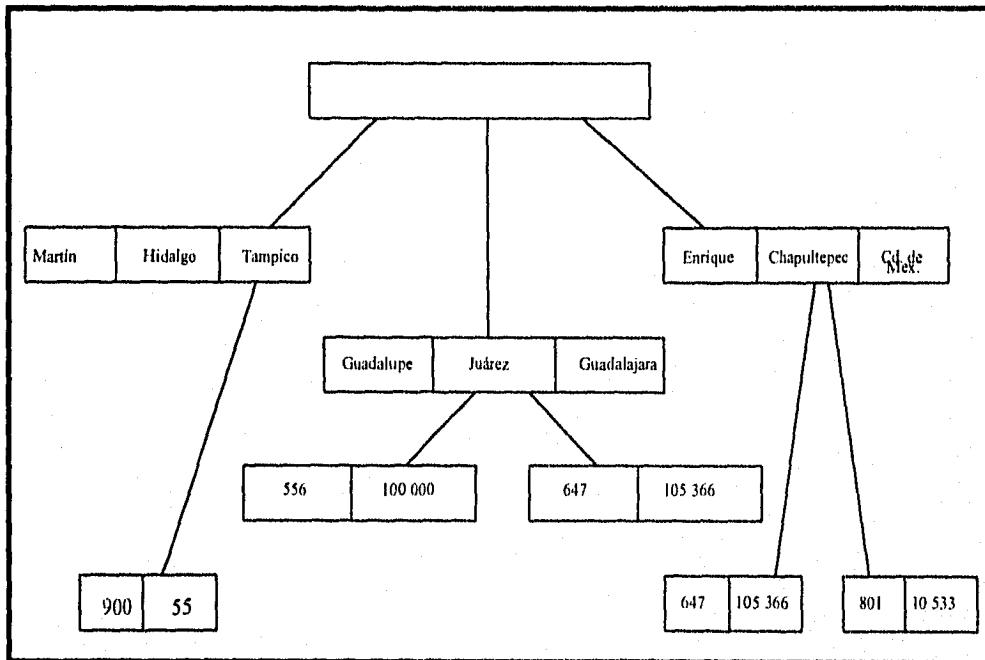


Fig. 3.3. Ejemplo de un esquema del modelo jerárquico.

c) **Modelos físicos de los datos.** Los modelos físicos de los datos sirven para describir los datos en el nivel más bajo. A diferencia de los modelos lógicos de los datos, son muy pocos los modelos físicos utilizados. Algunos de los más conocidos son:

- El modelo unificador
- La memoria de cuadros

Los modelos físicos de los datos capturan el aspecto de la implantación de los sistemas de base de datos.

INSTANCIAS, ESQUEMAS E INDEPENDENCIA DE LOS DATOS

Es importante mencionar que el conjunto de información almacenada en la base de datos se le llamará instancia, y a la estructura o diseño general de la base de datos se denominará esquema de base de datos. En la base de datos existen varios esquemas dependiendo del número de los niveles de abstracción de datos.

Otro concepto necesario es la independencia de datos, es decir, la posibilidad de modificar un tipo de esquema de la base de datos sin afectar el esquema del nivel inmediato superior. De esta forma la independencia de datos permite la capacidad de realizar cambios a la estructura de almacenamiento y de las estrategias de acceso sin afectar a las aplicaciones. Se distinguen dos tipos de independencias de datos y son:

Independencia Física: Si se modifica el esquema físico no se necesita rediseñar los programas de aplicación.

Independencia Lógica: Cuando se modifica el esquema conceptual no se necesita volver a escribir los programas de aplicación, este último concepto es difícil de lograr, puesto que gran parte de los programas de aplicación dependen de la estructura lógica de los datos.

DISEÑO DE SOFTWARE

Un buen diseño es la clave de una Ingeniería de Software efectiva. Un sistema de software bien diseñado es fácil de aplicar y mantener, además de ser comprensible y confiable. Los sistemas mal diseñados, aunque puedan funcionar, pueden ser caros de mantener, difíciles de probar y poco confiables. La etapa de diseño es, por tanto, la parte más importante del desarrollo del software.

No hay una manera definida de establecer un "buen diseño". Dependiendo de la aplicación y los requisitos del proyecto en particular, el buen diseño puede ser uno que permita producir una codificación muy eficiente; puede ser un diseño mínimo, donde la aplicación sea lo más compacta posible, o puede ser el diseño de más fácil mantenimiento. Este último implica minimizar los costos de los cambios del sistema, y eso significa que el diseño tiene que ser comprensible y que las modificaciones deben tener un efecto local.

Se considera el diseño como el proceso iterativo de tomar el modelo lógico del nuevo sistema junto con los objetivos del mismo y producir la especificación de un sistema físico que cumpla con esos objetivos.

El diseño estructurado es un conjunto de consideraciones generales y técnicas para diseño de programas, que facilitan codificación, depuración y modificación de éstos, a través, de un proceso que reduce la complejidad.

La reducción de la complejidad se obtiene mediante la separación de las funciones de un programa en módulos relativamente independientes.

Una de las ventajas más importantes del Diseño Estructurado es la posibilidad de reutilizar código.

MOTIVOS PARA REALIZAR EL DISEÑO ESTRUCTURADO

- Al hacer los programas más sencillos e independientes entre sí, se reduce el tiempo de desarrollo.
- Los beneficios aumentan si cada pieza realiza sólo una función.

CONSIDERACIONES GENERALES

- La medida primordial para evaluar entre diseños alternativos es la SIMPLICIDAD.
- La SIMPLICIDAD se puede conseguir dividiendo el sistema en pequeñas porciones.
- Estas porciones deberán poderse implementar y cambiar con un efecto mínimo sobre las demás porciones del sistema.
- Se deberá poder percibir fácilmente "que" y "como" hace su función cada porción.
- También deberá ser fácil cuantificar el impacto de un cambio de una porción.
- A esto que le hemos nombrado porción se le denomina formalmente MÓDULO.

Un MÓDULO es un conjunto de instrucciones que pueden ser llamadas a ejecución por medio de un nombre. El objetivo de la modularidad es permitir manejar las piezas de un programa de manera independiente. Los módulos serán más independientes si cada uno tiene su propio juego de variables.

METAS DEL DISEÑO ESTRUCTURADO

Las objetivos que se intentan lograr con un diseño estructurado es diseñar programas con:

- Estructuras independientes.
- Módulos que realizan una sola función.

Tomando lo anterior en cuenta, las características de los programas resultantes del diseño estructurado serán:

- Programas más simples.
- Módulos que se pueden codificar independientemente.
- Programas que se puede entender pieza por pieza.
- Facilidad en las pruebas.
- Los efectos colaterales de un cambio se reducen.
- Se reduce la cantidad de errores.

ACOPLAMIENTO Y COHESIVIDAD

Dos medidas para conseguir la independencia entre módulos son el Acoplamiento y la Cohesividad.

Para lograr una máxima independencia se deberá:

- Minimizar el acoplamiento.
- Maximizar la cohesividad.

ACOPLAMIENTO

- *El acoplamiento es una medida para determinar qué tan estrecha es una conexión entre dos módulos.*
- *Al evaluar alternativas para dividir un programa, resulta útil examinar los tipos de conexiones entre módulos.*
- *Una conexión es una referencia a una etiqueta o dirección definida en otro lado.*
- *Al minimizar las conexiones entre módulos, también se minimizan las trayectorias, a lo largo de las cuales se propagan los errores y los cambios.*
- *Entre menos y más simples sean las conexiones, más fácil será entender los módulos sin hacer referencia a otros.*
- *La propagación de un error o un cambio provoca un efecto de "cascada" que eleva el mantenimiento del sistema y propicia la aparición de nuevos errores.*
- *Un alto grado de acoplamiento complica al sistema ya que un módulo es más difícil de entender, cambiar o corregir si está altamente interrelacionado con otros.*
- *Factores que afectan el grado de acoplamiento.*
 1. *El tamaño de la conexión.*
 2. *El tipo de conexión.*
 3. *El tipo de comunicación: que es lo que se envía o se recibe.*

Si la medida en alguna de estas dimensiones es grande, el grado de acoplamiento será alto.

COHESIVIDAD

- *Un método alterno para reducir el acoplamiento entre módulos es maximizando la relación entre los elementos de un mismo módulo. A esto se le llama cohesividad.*
- *Un elemento es una instrucción, segmento o sub-función de un módulo.*
- *El objetivo es reducir el acoplamiento, elevando la ligadura entre los elementos de un módulo.*
- *La ligadura es la medida de cohesividad de un módulo.*
- *Escala de cohesividad, de menor a mayor:*

1. *Coincidental*
2. *Lógica*
3. *Temporal*
4. *Por comunicación*
5. *Secuencial*
6. *Funcional*

➤ *Se debe procurar que en cada módulo haya cohesividad funcional.*

TÉCNICAS DEL DISEÑO ESTRUCTURADO

El objetivo del Diseño Estructurado es construir programas como estructuras de una sola función, compilados por separado, con nombres apropiados que estén acoplados por datos a través del menor número de parámetros posibles.

La herramienta del diseño estructurado es el diagrama de estructura.

Las metodologías comúnmente usadas para el diseño son: los diagramas de flujo de datos (DFD), el diccionario de datos (DD), el modelo entidad relación (E-R) y la normalización.

Los diagramas de flujo de datos presentan una visión, lo más amplia posible, de las entradas al sistema, los procesos y las salidas.

Una vez terminados los diagramas de flujo de datos, éstos se utilizan para catalogar los procesos, el flujo, el almacenamiento, las estructuras y los elementos de un diccionario de datos, posteriormente se identifican las relaciones entre entidades y por último se efectúa la normalización usando todos los datos del sistema.

DISEÑO ORIENTADO AL FLUJO DE DATOS

La transición entre el flujo de la información (DFD) al diagrama de estructura es un proceso de cinco pasos:

1. *Se establece el tipo del flujo de la información.*
2. *Se indican los límites del flujo.*
3. *El DFD se convierte en la estructura del programa.*
4. *Se define la jerarquía de control mediante la factorización.*
5. *Se refina el diagrama de estructura resultante usando medidas y heurísticas de diseño.*

El tipo del flujo de la información es el conductor del método de conversión requerido en el paso 3.

➤ *Flujo de transformación*

- ⇒ Pueden ser identificados como entrada-proceso-salida.
- ⇒ De las entidades externas, los datos se transforman en una forma interna y se identifica como flujo de llegada.
- ⇒ En el núcleo del DFD ocurre una transición. Los datos de llegada ahora son transformados y comienzan a moverse en flujos de salida.
- ⇒ A este núcleo se le conoce con el nombre de centro de transformación o punto de máxima abstracción.

➤ *Flujo de transacción*

- ⇒ Este flujo de datos es identificado, cuando un flujo de datos sencillo, llamado transacción, desencadena otro flujo de datos a lo largo de uno de los muchos caminos.
- ⇒ El flujo de transacción se caracteriza por datos que se mueven a lo largo de un camino de llegada.
- ⇒ La transacción es evaluada, y basándose en su valor, el flujo se inicia por uno de los muchos caminos de acción.
- ⇒ El centro de donde emanan muchos caminos de acción se le denomina centro de transacción.

Nota: Es importante mencionar que en un DFD de un sistema mayor, pueden presentarse ambos tipos de flujos, tanto de transformación como de transacción.

Existen un gran número de metodologías que han sido desarrolladas para el diseño de sistemas, estas metodologías están integradas por una serie de herramientas y métodos que son similares en su concepto, significado o resultado, existiendo sólo pequeñas modalidades en su notación.

DIAGRAMA DE FLUJO DE DATOS (DFD)

Como se mencionó en el capítulo II un DFD se utiliza para describir un diseño de sistemas de alto nivel, además muestra como se transforman los datos al pasar de un componente del sistema a otro. La simbología que se utilizará para describir los procesos del sistema POHUER es la de Gane & Sarson, y sus componentes se muestra en la figura 3.4.

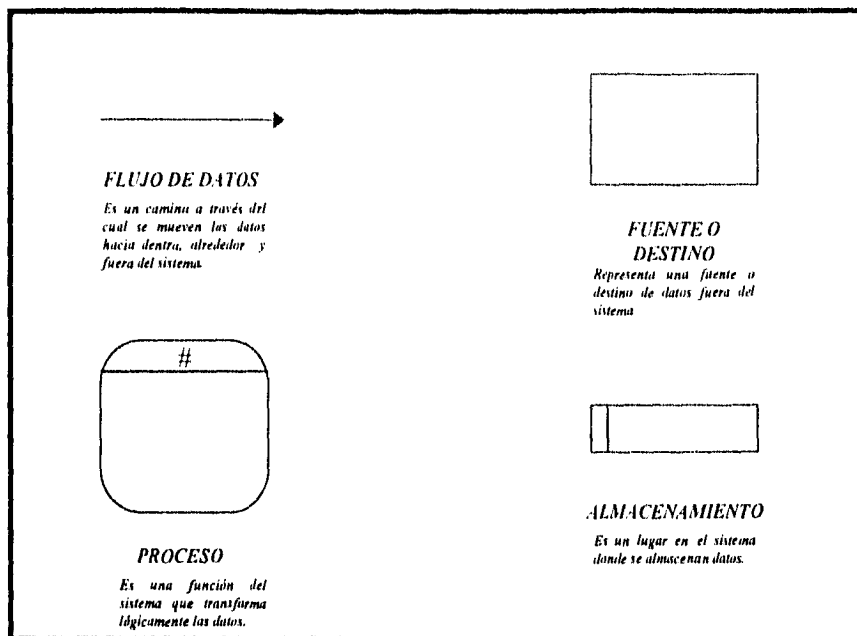


Fig. 3.4. Simbología, para realizar un DFD, de Gane & Sarson.

El flujo de datos debe partir de lo general a lo particular, haciendo uso de los cuatro símbolos descritos en la fig. 3.4. Algo para lo que servirá estructurar el DFD es para determinar los límites del sistema.

Para obtener mayor grado de detalle los diagramas sufren una descomposición manteniendo constantes las entradas y salidas a lo largo de los diagramas consecutivos. Las excepciones se ignoran de los primeros dos o tres niveles del DFD.

DICCIONARIO DE DATOS (DD)

Para el diseño de las estructuras de datos es necesario conocer las características de los elementos de información que componen cada flujo de datos de un DFD. La forma de representar el contenido de cada flujo de datos es utilizando el diccionario de datos.

El DD cuenta con una notación generalizada para definir los componentes de cada flujo. En otras palabras, el DD es una referencia de datos acerca de los datos que pueden ser compuestos (éstos se definen en términos de sus componentes) o elementales (éstos se definen en términos del significado de cada uno de sus valores). La figura 3.5, muestra la notación utilizada para el DD.

CONSTRUCCIÓN DE DATOS	NOTACIÓN	SIGNIFICADO
Secuencia	=	"Equivale" o "Está compuesto de"
Selección	+	"y" o "concatenación"
Repetición	[]	Selección de uno u otro de los datos
	{ }	Iteración o n repeticiones de datos
	()	Datos opcionales pertenecientes o no a la composición
	**	Comentarios o aclaraciones sobre el dato

Fig. 3.5. Notación para el Diccionario de Datos.

La notación convencional facilita la representación de elementos de información. Cada elemento puede ser un dato compuesto que requerirá ser refinado hasta tener datos elementales, o que cada elemento compuesto esté representado en forma clara sin que exista ambigüedad.

DIAGRAMA ENTIDAD RELACIÓN (E-R)

Como se describió en el capítulo II, el diagrama E-R (fig. 2.10.) se basa en la descripción del mundo real a través de la diferenciación de objetos básicos (entidades) y su interacción con otros objetos (relación). De esta forma la entidad es un ente que se distingue de los demás, por medio de una serie de características propias (atributos). Las entidades deben de ser identificadas en relación a la necesidad de almacenamiento de datos. Las relaciones que pueden tener dos entidades pueden ser:

- Uno a uno
- Uno a muchos
- Muchos a uno
- Muchos a muchos

El diagrama E-R representa un gran beneficio: Enfatiza las relaciones entre almacenes de datos en el DFD que de otra manera se verían sólo en la especificación del proceso. Es importante mencionar que no importa si se desarrolla primero el diagrama de flujo de datos o el diagrama entidad relación. Este orden puede estar sujeto a las preferencias del desarrollador de sistemas. Sin embargo también existe la posibilidad de que ambos modelos se desarrollen en paralelo.

La figura 3.6. muestra la representación gráfica de las diversas combinaciones de relaciones.

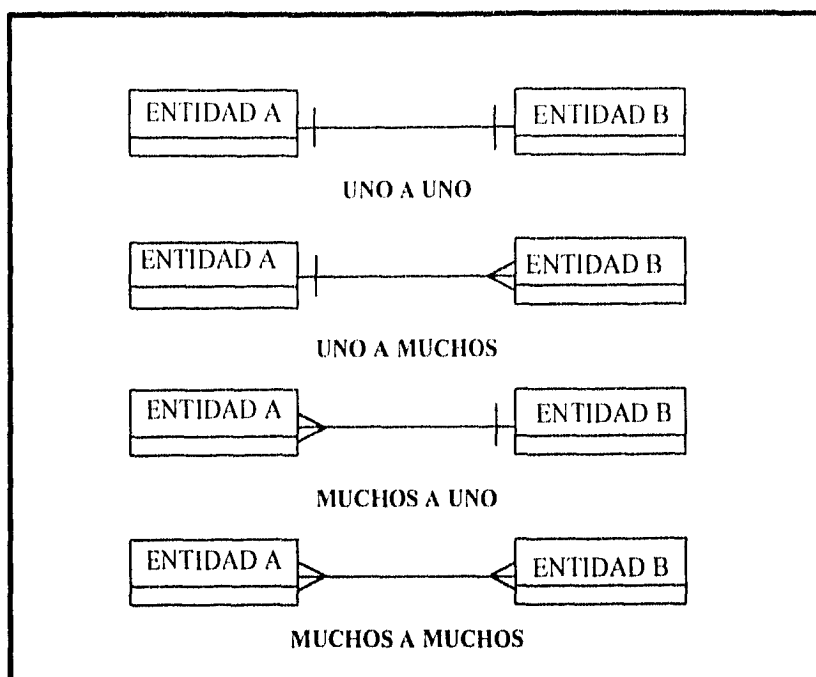


Fig. 3.6. Relaciones que pueden existir dentro de un diagrama E-R.

3.2 DIAGRAMA DE FLUJO DE DATOS (DFD) DEL SISTEMA POHUER

A continuación se realizarán los diagramas de flujo de datos correspondientes al diseño del sistema POHUER.

En la figura 3.7 se muestra el diagrama de nivel cero del sistema POHUER. El diagrama consta de cuatro procesos principales, los cuales son:

- Consulta y Reasignación
- Captura de Datos
- Mantenimiento
- Impresión de Reportes

Como se puede observar es indispensable que el usuario introduzca su clave de acceso al sistema. Primeramente el sistema verificará si la clave de acceso es correcta para permitir el acceso al sistema, en caso de que la clave sea correcta el usuario podrá escoger uno de los módulos del sistema POHUER. A continuación se mostraran los subprocesos internos dentro de cada proceso del sistema POHUER.

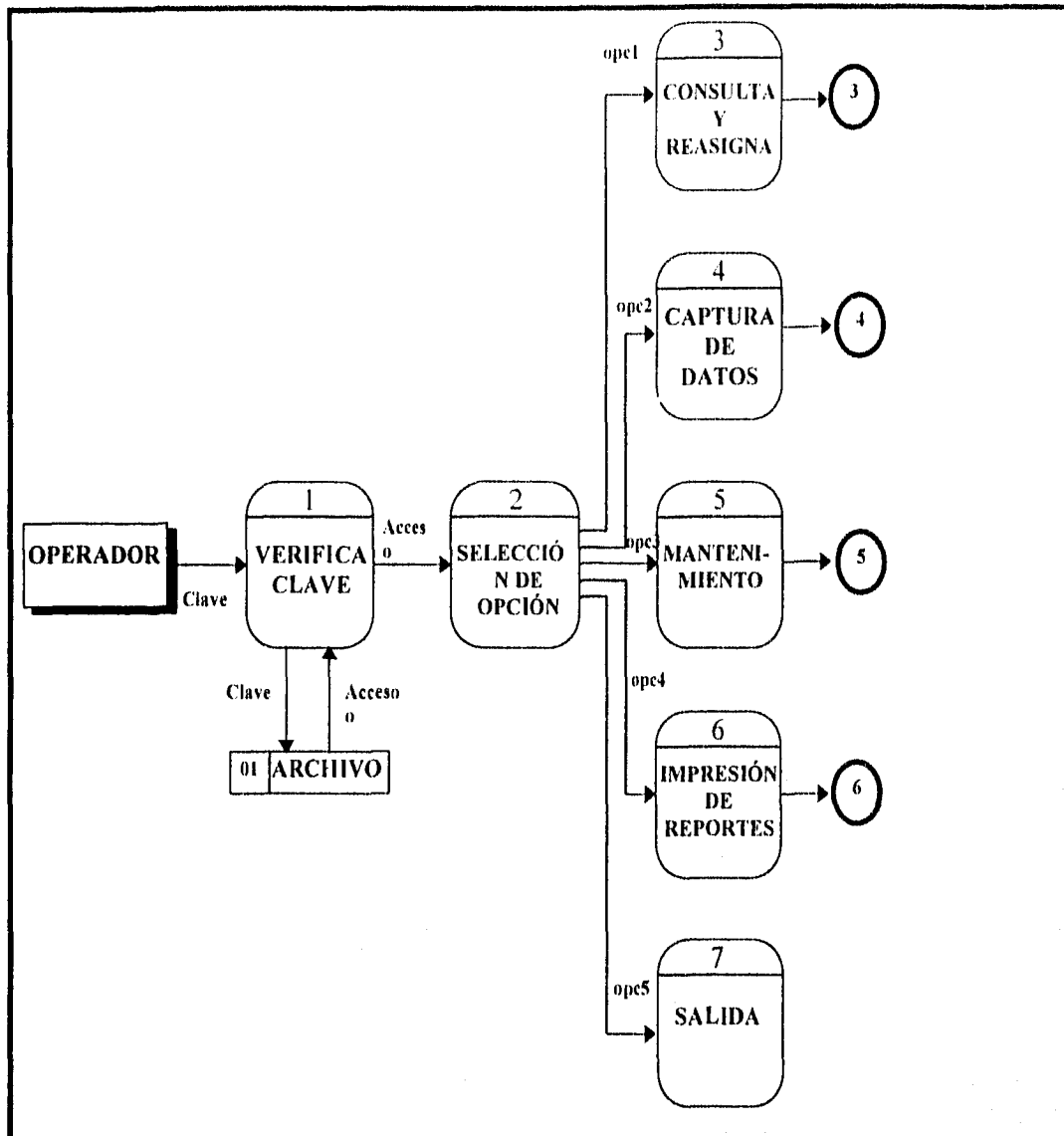


Fig. 3.7. Diagrama nivel cero "Sistema POHUER"

En la figura 3.8 se puede observar el diagrama de nivel cero del Módulo de Consulta y Reasigna. Este módulo cuenta con tres subprocesos. Éstos son: Consulta de Datos, Contactación, y Reasignación. En conjunto estos submódulos se encargarán, respectivamente, de recuperar datos de las pólizas y los agentes, guardar datos de llamadas hechas a los asegurados, y reasignar un nuevo agente a una póliza huérfana.

En la figura 3.9 se ilustra el diagrama de nivel cero para el Módulo de Captura de Datos. Este módulo se utilizará para capturar la información de las referencias de los asegurados que no se encuentran dentro de los sistemas que utiliza POHUER.

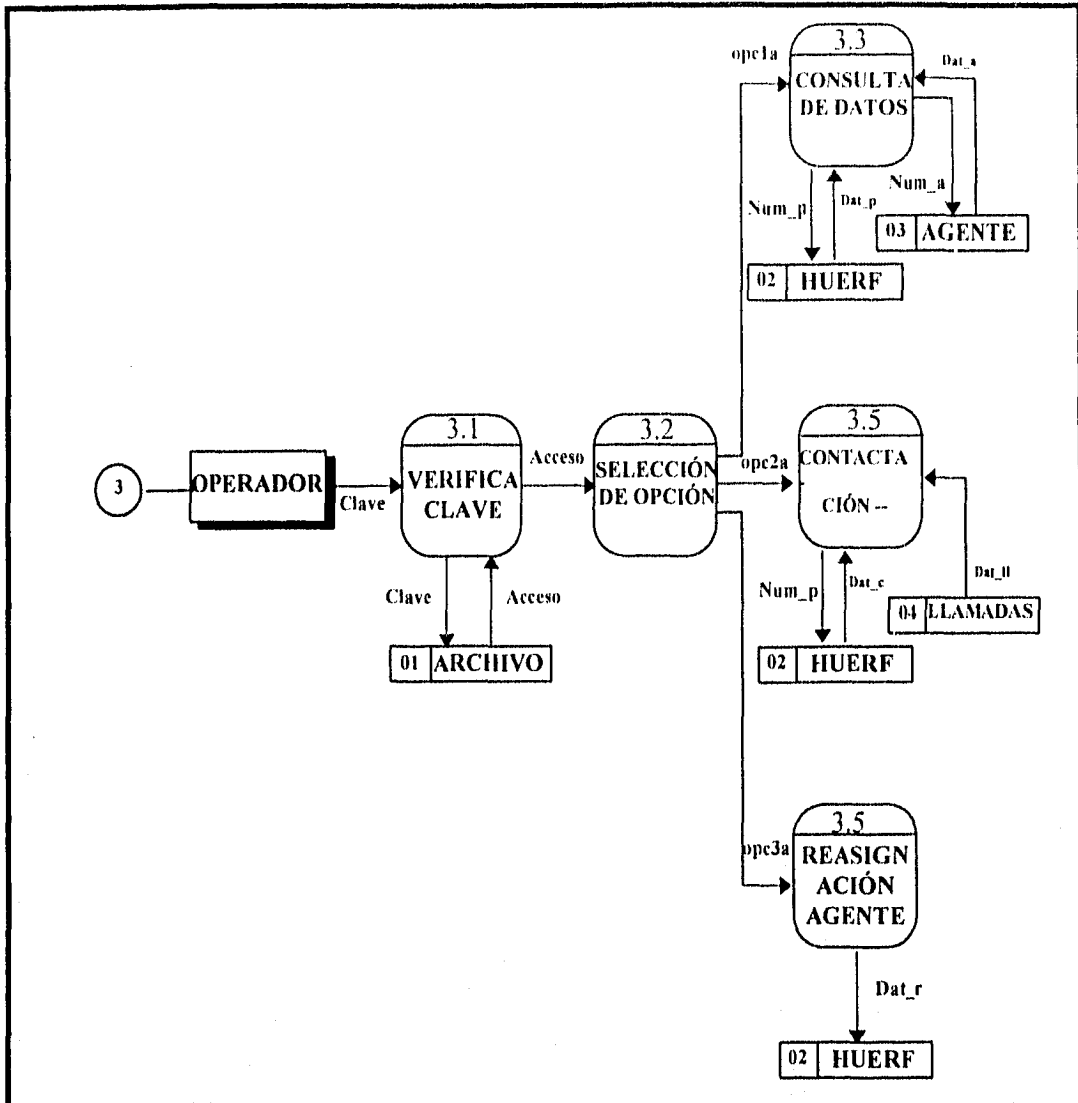


Fig. 3.8. Diagrama nivel cero "Consulta y Reasignación".

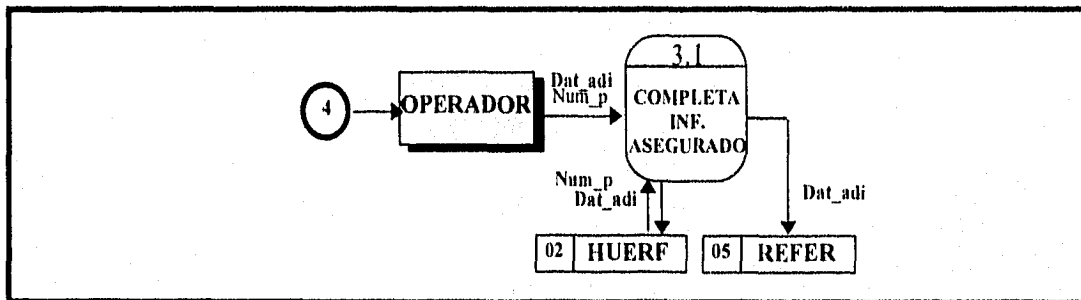


Fig. 3.9. Diagrama nivel cero "Datos Adicionales".

En las figura 3.10 y 3.11 se observa el diagrama de nivel cero para el Módulo de Impresión de Reportes. Este módulo cuenta con los siguientes diez submódulos: Crea Telegrama, Imprime Telegrama, Histórico de Telegrama, Consolidado, Estadístico, Megareporte, Estadístico del Universo Total, Agentes Dispositores, Agentes Cancelados, y Formas de Captura. Básicamente este módulo se utilizará para la impresión de reportes y generación de estadísticas.

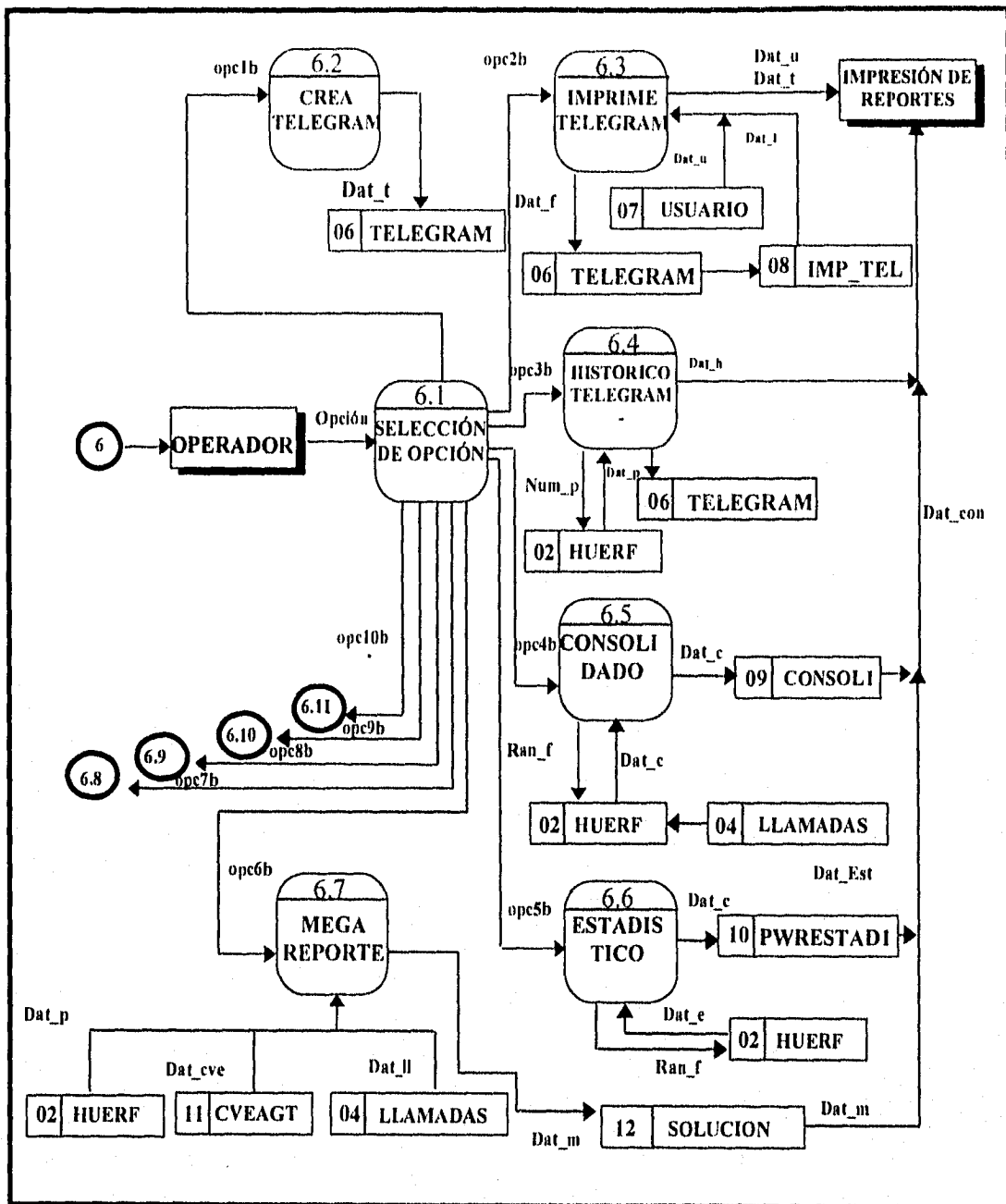


Fig. 3.10. Diagrama nivel cero "Impresión de Reportes".

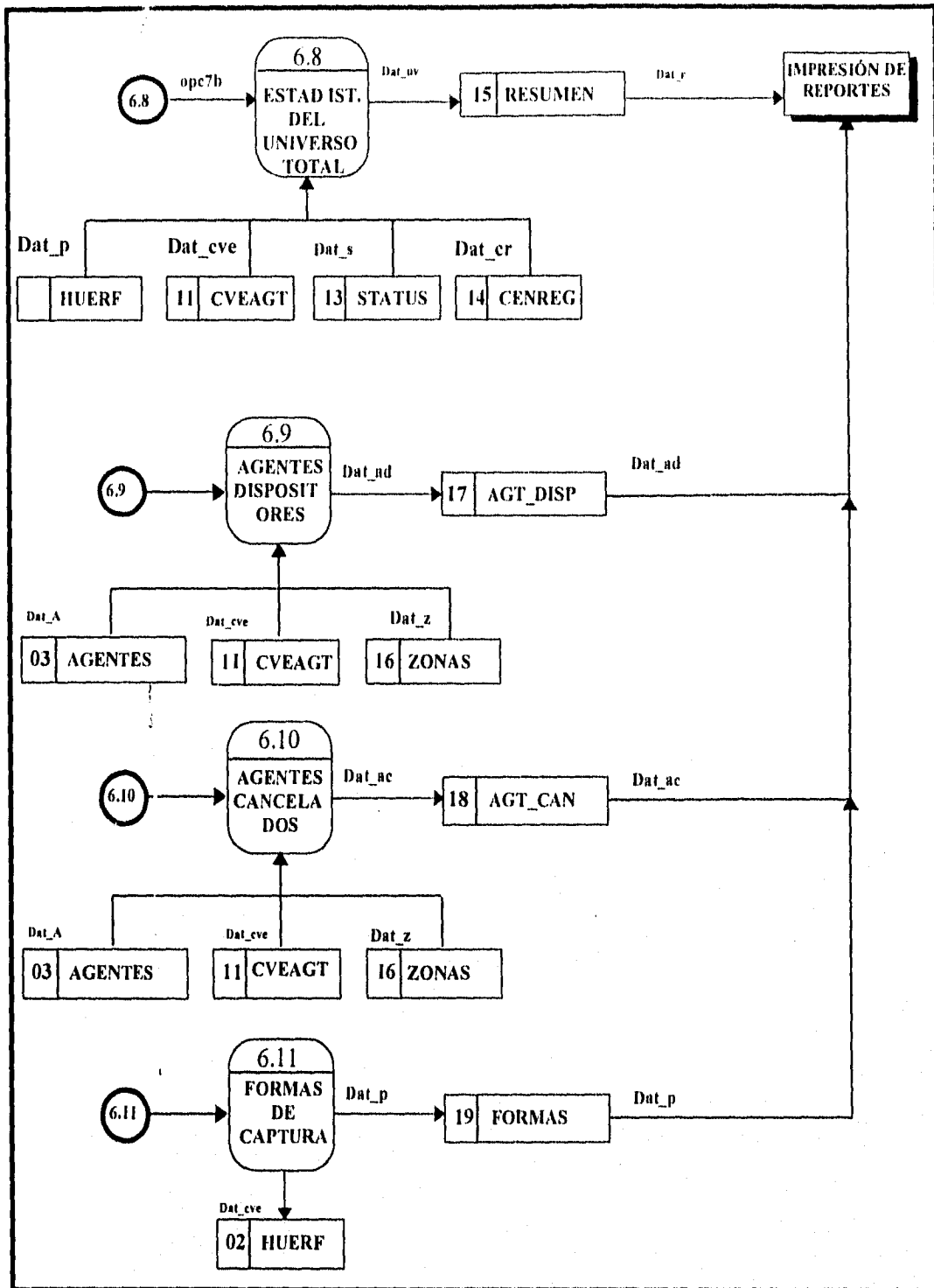


Fig. 3.11. Diagrama nivel cero "Impresión de Reportes" (continuación).

3.3 DICcionario DE DATOS

A continuación se presenta el Diccionario de Datos, segundo paso de la metodología de diseño de sistemas de Chris Gane & Trish Sarson, correspondiente a los flujos de información descritos en los diagramas de flujos de datos.

Clave= (Clave de acceso al sistema POHUER).

****** Cada operador contará con una clave única de acceso al sistema.

Dat_a= (Datos del Agente).

Nombre del agente + Status del agente + Fecha de conexión + Fecha de cancelación + Zona a la que pertenece.

Dat_ac= (Datos del Agente cancelado).

Nombre + Status + Clave + Centro Regional.

Dat_ad= (Datos de Agentes Dispositores).

Nombre + Status + Clave + Fecha de asignación + Zona + Fecha de status.

Dat_adi= (Datos adicionales).

******Datos adicionales se refiere a los datos que no se encuentran en la póliza, como son teléfono del asegurado, teléfono del contratante, referencias 1, 2 y 3.

Dat_c= (Datos del contratante).

Número de cliente + Nombre del asegurado + Domicilio Teléfono de casa + Teléfono de oficina + Nombre del contratante + Domicilio de contratante + Teléfono de casa del contratante + Teléfono de oficina del contratante + Suma Asegurada + Prima de la Póliza.

Dat_con= (Datos del Consolidado).

Clave de operador + Llamadas realizados + Tiempo de duración de la llamada + Tipologías.

Dat_cr= (Datos del los centros regionales).

Dat_cve= (Datos del agente).

Dat_e = (Datos del Estadístico).

Tipología + Descripción de la llamada + Número de Pólizas + % de participación de las pólizas + Prima + % de participación de la prima.

Dat_f= (Fecha de creación de telegrama).

Dat_h= (Datos del histórico de telegramas).

Clave de póliza + *Dat_f* + *Dat_t* + Número de telegramas enviados.

Dat_m= (Datos del megareporte).

** Para extraer datos específicos es decir datos del agente, de las pólizas y de las llamadas.

Dat_p= (Datos de la póliza).

Nombre del asegurado + Fecha de emisión de póliza + Fecha de vencimiento de póliza + Fecha límite de cobro + Suma asegurada + Prima según forma de pago + Monto del préstamo + Status de póliza + Forma de pago + Edad del asegurado al adquirir la póliza + Edad actual del asegurado + Beneficios adicionales.

Dat_r= (Datos de reasignación).

Número de zona asignada + Agente asignado + Fecha de asignación del agente
Nombre del nuevo agente.

Dat_s= (Datos del status).

Dat_t= (Datos del telegrama).

Datos del Telegrama + Fecha + Centro regional + Status del Agente + Prima.

Dat_ti= (Fecha de impresión de telegrama).

Dat_u= (Datos del usuario u operador que envía el telegrama).

Nombre del usuario + teléfono.

Dat_ul= (Datos de llamada).

Fecha de llamada + Hora de llamada + Duración de la llamada + Resultado de la llamada + Próxima llamada + Hora de la próxima llamada + Envío de telegrama + Observaciones.

Dat_v= (Datos del universo).

at_p + *Dat_cve* + *Dat_s* + *Dat_cr*.

Dat_z= (Datos de Gerencia de Zona).

Num_a= (Clave de agente).

** Campo clave para la extracción de información referente al agente.

Num_p= (Clave de la póliza).

Ran_f= (Rango de fecha).

Opc1= (Opción para acceder al proceso de Consulta y Reasignación).

*** Campo clave para la extracción de información referente a la póliza.*

Opc2= (Opción para acceder al proceso de Captura de Datos).

Opc3= (Opción para acceder al proceso de Mantenimiento).

Opc4= (Opción para acceder al proceso de Impresión de Reportes).

Opc5= (Opción de salida del sistema POHUER).

3.4 DIAGRAMA ENTIDAD-RELACIÓN DEL SISTEMA POHUER

Siguiendo la metodología mencionada en el inicio del capítulo, una vez que se han desarrollado los diagramas de flujo de datos, se continuará con la elaboración del diagrama Entidad-Relación para el sistema POHUER. La figura 3.12. muestra el diagrama Entidad-Relación para éste sistema.

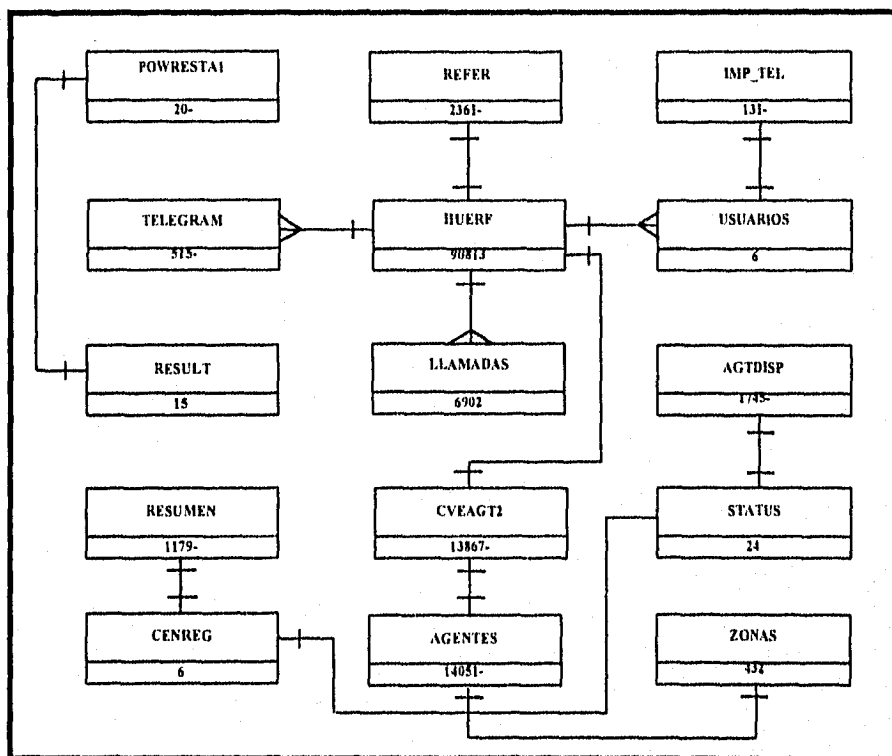


Fig. 3.12. Diagrama Entidad-Relación del sistema POHUER.

3.5 NORMALIZACIÓN

La normalización de datos está compuesta por una serie de reglas y técnicas que sirven para:

- Identificar las relaciones entre atributos.
- Combinar atributos para crear relaciones.
- Combinar relaciones para crear bases de datos.

Un atributo, también llamado campo, es un elemento de información. Una relación se puede definir como un grupo de atributos. Uno de los objetivos principales de la normalización es el de evitar anomalías que pueden ocurrir cuando los datos que se encuentran en dos atributos se pierden cuando se efectúa un borrado.

La normalización es útil como una herramienta analítica para el diseño de una base de datos. Los puntos de vista y requerimientos del usuario son considerados durante el proceso de diseño de la base de datos. Los atributos apropiados y sus relaciones son identificados y definidos adecuadamente.

Dentro de los beneficios que se pueden obtener con una normalización adecuada están:

- Desarrollo de estrategias para construir relaciones y selección de llaves.
- Reducción de problemas asociados con la inserción y eliminación de datos.
- Reducción de tiempo de modificación cuando se necesita cambiar la estructura de la información.
- Identificación de problemas potenciales que puedan requerir mayor análisis y documentación.

Una normalización correcta se traducirá en un mejor tiempo de respuesta de la base de datos. La normalización se realiza generalmente en tres pasos llamados formas normales que a continuación se mencionan.

Primera Forma Normal (1FN): Una relación "R" está en su 1FN si y sólo si todos sus atributos contienen valores atómicos. Esto es cada atributo no puede ser un vector. Incluye la eliminación de grupos repetidos y la identificación de la llave que define al criterio primario. Para poder realizar esto, la relación necesita desglosarse en dos o más relaciones.

(Segunda Forma Normal 2FN): Una relación "R" está en su 2FN si y sólo si está en la 1FN y cada atributo que no es llave depende funcionalmente de la llave primaria completa, es decir, se asegura que todos los atributos no-llave, sean completamente dependientes de la llave del criterio primario. Todas las dependencias normales se eliminan y se colocarán en otra relación.

Tercera Forma Normal (3FN): Una relación "R" está en su 3FN si y sólo si está en la 2FN y cada atributo que no sea llave depende funcionalmente, en forma directa, de la llave primaria (análisis de dependencia transitivas). Esto es, no deben existir dependencias funcionales entre atributos. En el tercer y último paso se elimina cualquier dependencia transitoria, entendiéndose como dependencia transitoria aquella en la cual sus atributos no-llave son dependientes de otros atributos no-llave.

Después de haber identificado las entidades y aplicado los pasos de la normalización se presentan las siguientes tablas ya normalizadas:

AGENTES: En esta entidad se encontrarán todos los agentes de GNP independientemente del status que tengan, conteniendo los campos que se muestran en tabla 3.1.

Nombre del Campo	Tipo	Tamaño	Descripción
RFC	ALP	13	Registro Federal de Causantes del agente
CLAVE_NACIONAL	NUM		Clave nacional del agente
CLAVE_PROV.INC	NUM		Clave provincial del agente
NOMBRE	ALP	50	Nombre del agente
FECHA_CONEXIÓN	DAT		Fecha de conexión del agente
STATUS	ALP	1	Status del agente
TIPO_AGENTE	ALP	2	Tipo de agente al que pertenece
ZONA	NUM		Zona a la que pertenece el agente

Tabla 3.1. AGENTES.

AGTDISP: En esta entidad se encontrarán los datos de los agentes dispositores. Se le dará una atención especial a este tipo de agentes debido a que las pólizas que tienen a cargo serán reasignadas lo más pronto posible. En la tabla 3.2. se pueden observar los campos, tipo y tamaño.

Nombre del Campo	Tipo	Tamaño	Descripción
NOMBRE	ALP	50	Nombre del agente
STATUS	ALP	2	Status del agente
CLAVE	NUM		Clave del agente
FECHA_ASIGNACIÓN	DAT		Fecha de asignación del agente
ZONA	NUM		Zona a la que pertenece el agente
FECHA STATUS	DAT		Fecha de cancelación del agente

Tabla 3.2. AGTDISP.

CENREG: Dentro de esta entidad se guardarán los datos de los centros regionales existentes en GNP, como se muestra en la tabla 3.3.

Nombre del Campo	Tipo	Tamaño	Descripción
CENTRO_REGIONAL	ALP	2	Clave del centro regional
DESCRIPCION	ALP	30	Descripción del centro regional

Tabla 3.3. CENREG.

CVEAGT2: En esta entidad se encontrarán la clave, RFC, status, y zona a la que pertenece el agente, en la tabla 3.4. se muestra el nombre, tipo, tamaño y descripción de los campos que se utilizarán en esta tabla.

Nombre del Campo	Tipo	Tamaño	Descripción
CLAVE	NUM		Clave del agente
RFC	ALP	13	RFC del agente
STATUS	ALP	2	Status que tiene el agente
ZONA	NUM		Zona a la que pertenece el agente

Tabla 3.4. CVEAGT2.

HUERF: En esta entidad se encontrarán todos los datos necesarios para conocer el estado actual de la póliza como son: datos del asegurado, del agente cancelado, del nuevo agente, datos del contratante, así como los datos de la forma en que se logró contactar al asegurado (telegrama o llamada telefónica) y el número de llamadas o telegramas que fueron necesarios para la contactación. En la tabla 3.5. se muestra el nombre, tipo, tamaño y descripción de cada campo a utilizar.

Nombre del Campo	Tipo	Tamaño	Descripción
POLIZA	ALP	10	Número de póliza
AGENTE	NUM		Clave del agente
NOMBRE	ALP	40	Nombre del asegurado
DOMICILIO	ALP	136	Domicilio del asegurado
TEL_CASA	ALP	20	Teléfono de la casa del asegurado
TEL_OF	ALP	40	Teléfono de la oficina del asegurado
C_NOMBRE	ALP	40	Nombre del contratante
C_DOM	ALP	136	Domicilio del contratante
C_TEL	ALP	20	Teléfono de la casa del contratante
C_TEL_OF	ALP	20	Teléfono de la oficina del contratante
P_F_EMI	ALP	8	Fecha de emisión de la póliza

P_COB_HASTA	ALP	8	Fecha de cobro
P_FORMA_PAGO	ALP	8	Forma de cobro de la póliza
P_AIO	SHO		Año en que va la póliza
P_PLAN	ALP	6	Tipo de plan de seguro de la póliza
P_IDPLAN	ALP	2	Clave del plan de seguro
P_CONS_REAL	NUM		Conservación real de la póliza
P_PRISFP	NUM		Cantidad acumulada en el sistema prisma
P_TIPO	SHO		Tipo de estado actual de la póliza
P_SUMA	NUM		Suma asegurada de la póliza
P_PRIMA	NUM		Prima de la póliza
P_STATUS	ALP	2	Status de la póliza
P_PRESTAMO	NUM		Préstamo de la póliza
P_EDAD	NUM		Edad real del asegurado
P_ACTUAL	NUM		Edad que se otorga al asegurado
P_FAD	NUM		Fondo disponible para el asegurado
P_BENEF	ALP	6	Beneficios adicionales de la póliza
N_ZONA	NUM		Número de nueva zona asignada
N_AGENTE	NUM		Clave del nuevo agente
N_N_AGENTE	ALP	40	Nombre del nuevo agente asignado
N_F_ASIG	DAT		Fecha de asignación del nuevo agente
A_OBS	MEM	45	Observaciones acerca del nuevo agente (puede no haber)
STATUS	ALP	2	Status del nuevo agente
ULT_RESULT	ALP	2	Resultados de las llamadas realizadas
TELEG	SHO		Enviar telegrama (si/no)
F_ENV_TELE	DAT		Fecha de envío de los telegramas
CEN_REG	ALP	2	Centro regional de la póliza
ZONA	NUM		Zona a la que pertenece la póliza
TRAMITE	SHO	1	Trámite en el que se encuentra la póliza
ESFUERZO	SHO	1	Tipo de esfuerzo que tiene la póliza
P_CONSERVA	NUM		Cantidad conservada de la póliza
ULT_USUAR	ALP	8	Último usuario que consultó la póliza
ULT_FECHA	DAT		Última Fecha de consulta de la póliza

Tabla 3.5. HUERF.

LLAMADAS: En esta entidad se encontrarán los datos relacionados con las llamadas realizadas a los asegurados cuando se les contacte para reportarles el status de su póliza. En la tabla 3.6. se hace referencia a la descripción de los campos.

Nombre del Campo	Tipo	Tamaño	Descripción
FECHA	DAT		Fecha de realización de llamada
HORA	NUM		Hora en que se efectúa la llamada
PÓLIZA	ALP	10	Número de póliza
DURACIÓN	NUM		Tiempo que duró la llamada realizada
RESULTADO	ALP	2	Resultado de la llamada
PROX_LLAMA	DAT		Hora en que se realizará la próxima llamada
HOR_PROX	NUM		Hora aprox. en que se realizará la siguiente llamada
CONTESTO	ALP	1	Contestaron la llamada (si / no)
OBS	MEM	25	Comentarios sobre la llamada (Si existen)
USUARIO	ALP	8	Clave del usuario

Tabla 3.6. LLAMADAS.

POWRESTAI: Esta entidad almacenará los datos necesarios para llenar el informe del estadístico general. En la tabla 3.7. se muestra la descripción de los campos que se utilizarán.

Nombre del Campo	Tipo	Tamaño	Descripción
TIPO	ALP	2	Clave de resultado de llamada
PÓLIZAS	N	0	Resultado de llamada realizada por usuario

Tabla 3.7. POWRESTAI.

RESULT: En esta entidad se registrará el resultado de las llamadas al contactar a los asegurados. Dentro de esta entidad, también se podrá evaluar a los operadores de POHUER debido a que, se generarán informes para describir el número de llamadas que realice cada operador, ver tabla. 3.8.

Nombre del Campo	Tipo	Tamaño	Descripción
CVE_RES	ALP	2	Clave de resultado de llamada
RESULT	ALP	25	Resultado de llamada realizada por usuario

Tabla 3.8. RESULT.

REFER: Esta entidad guardará los datos personales, de las referencias de un asegurado. En la tabla 3.9. se puede observar la descripción de cada campo.

Nombre del Campo	Tipo	Tamaño	Descripción
POLIZA	ALP	10	Número de la póliza
REFI	ALP	50	Nombre de la referencia 1

TEL_R1	ALP	20	Teléfono de la referencia 1
REF2	ALP	50	Nombre de la referencia 2
TEL_R2	ALP	20	Teléfono de la referencia 2
REF3	ALP	50	Nombre de la referencia 3
TEL_R3	ALP	20	Teléfono de la referencia 3

Tabla 3.9. REFER.

RESUMEN: En esta entidad se almacenará el monto de las primas que se genera cada año durante la vida de una póliza. En la tabla 3.10. se hace referencia a los campos utilizados.

Nombre del Campo	Tipo	Tamaño	Descripción
CEN_REG	ALP	2	Centro regional al que pertenece el agente
ZONA	NUM		Zona a la que pertenece el agente
STATUS	ALP	2	status del agente
PÓLIZA	ALP	10	Número de póliza
FEC_EMI	ALP	8	Fecha de emisión de la póliza
PRIMA	NUM		Prima de la póliza
DUR	NUM		Vigencia de la póliza
C1	NUM		Pólizas que tienen un año o menos
C2	NUM		Pólizas que tienen dos años
C3	NUM		Pólizas que tienen tres años
C4	NUM		Pólizas que tienen cuatro años
C5	NUM		Pólizas que tienen cinco años o más
P1	NUM		Cantidad acumulada de primas en el primer año
P2	NUM		Cantidad acumulada de primas en el segundo año
P3	NUM		Cantidad acumulada de primas en el tercer año
P4	NUM		Cantidad acumulada de primas en el cuarto año
P5	NUM		Cantidad acumulada de primas en el quinto año

Tabla 3.10. RESUMEN.

STATUS: En esta entidad se describirán los diferentes status de los agentes existentes. La tabla 3.11 describe los campos de la entidad.

Nombre del Campo	Tipo	Tamaño	Descripción
STATUS	ALP	2	Status del agente
DESCRIPCIÓN	ALP	50	Descripción del status

Tabla 3.11. STATUS.

TELEGRAM: Esta entidad contendrá los datos necesarios para llenar las formas de telegrama que se les enviarán a los asegurados cuando no tengan teléfono para ser contactados, como se puede observar en la tabla 3.12.

Nombre del Campo	Tipo	Tamaño	Descripción
POLIZA	ALP	10	Número de póliza
FECHA	DAT		Fecha de elaboración de telegrama
ASEGURADO	ALP	40	Nombre del asegurado
DIRECCIÓN	ALP	136	Dirección del asegurado
USUARIO	ALP	8	Nombre del usuario
FEC_IMPR	DAT		Fecha de impresión del telegrama

Tabla 3.12. TELEGRAM.

USUARIOS: En esta entidad se almacenarán las diferentes claves que cada usuario necesitará para acceder al sistema, así como su nombre y número telefónico. En la tabla 3.13. se pueden observar las claves para acceder al sistema POHUER.

Nombre del Campo	Tipo	Tamaño	Descripción
USUARIO	ALP	8	Clave del usuario
PASW	ALP	8	Clave de acceso de usuario
SIGNON	ALP	6	Segunda clave de acceso de usuario
PASW_SISI	ALP	6	Clave de acceso del usuario al sistema SISI
PASW_V4	ALP	8	Clave de acceso del usuario al sistema V4
VIDA	ALP	6	Clave de acceso del usuario al sistema VIDA
PASW_V2	ALP	6	Clave de acceso del usuario al sistema V2
TIPO_ENTRADA	SHO		Campo para diferenciar el tipo de acceso
TEL_US	ALP	20	Teléfono y extensión de trabajo del usuario
NOM_US	ALP	40	Nombre del usuario

Tablas 3.13. USUARIOS.

ZONAS: Esta entidad guardará la información de las diversas zonas existentes en cada centro regional. En la tabla 3.14. se observan los atributos que requiere la entidad.

Nombre del Campo	Tipo	Tamaño	Descripción
ZONA	NUM		Número de zona
OFICINA	NUM		Número de oficina
CENTRO_REGIONAL	ALP	2	Código del centro regional

Tabla 3.14. ZONAS.

3.6 CREACIÓN DE PROTOTIPOS

Durante la fase de diseño es muy útil la generación de prototipos pues de ello depende optimizar tiempo durante la etapa de desarrollo del sistema. Un prototipo es la simulación del funcionamiento del sistema antes de construirlo, es decir un simulacro del sistema. El objetivo de crear prototipos antes de entrar de lleno a la programación, es mostrar como se espera que trabaje el sistema, para que esto sirva como base para mejorarlo o hacer cambios antes de que se ocupe tiempo en la etapa de desarrollo. En las figuras 3.13. a 3.20. se presentan los prototipos desarrollados.

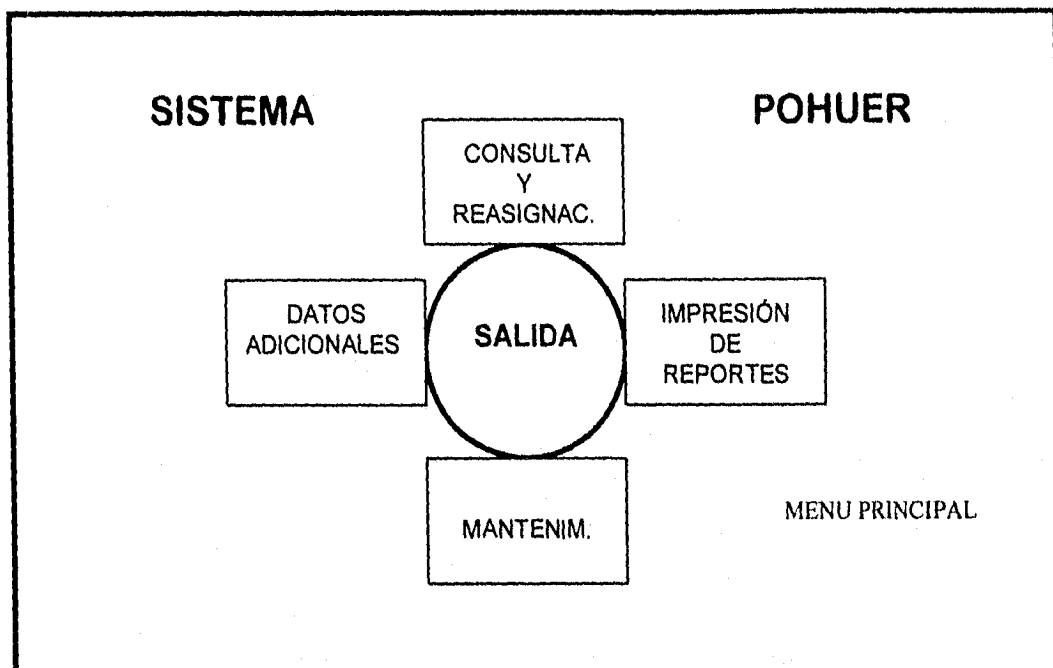


Fig. 3.13. Prototipo de la pantalla principal del sistema POHUER

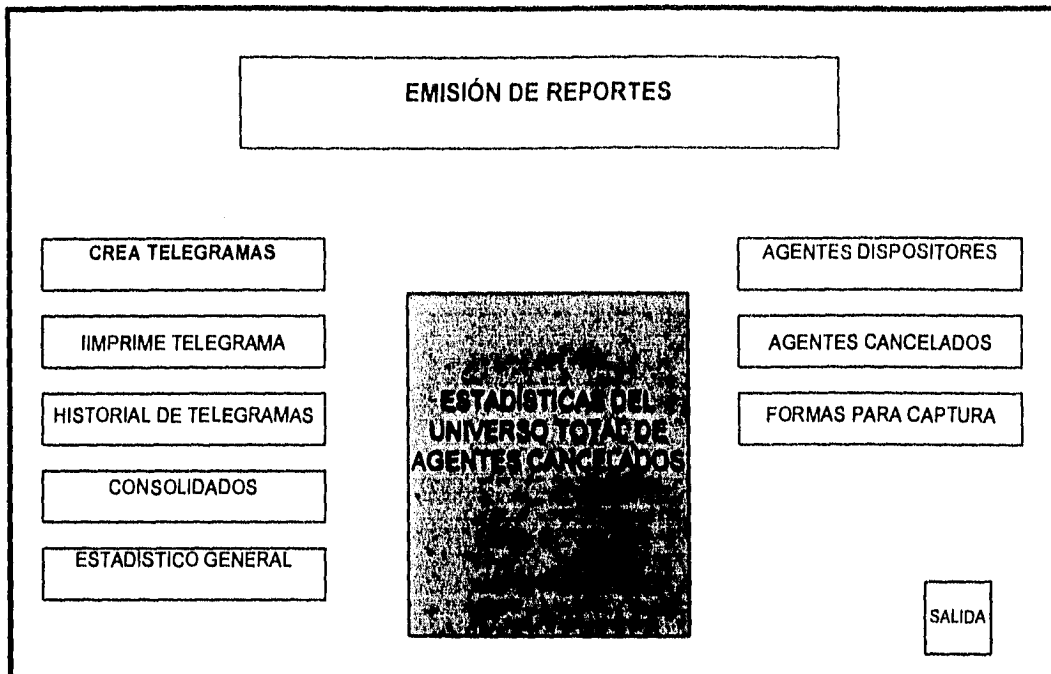


Fig. 3.14. Prototipo de la pantalla de Impresión de Reportes.

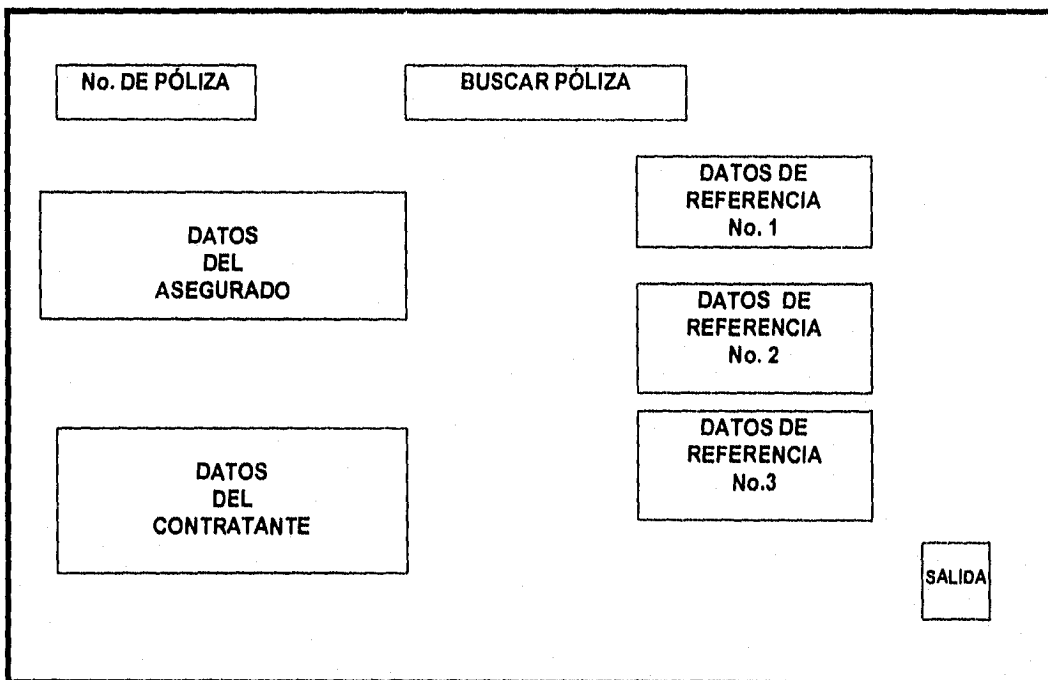


Fig. 3.15. Prototipo de la pantalla Captura de Datos Adicionales.

No. DE PÓLIZA	CONSULTAR PÓLIZA	AGENTES	SALIDA
DATOS DEL ASEGURADO NOMBRE: DOMICILIO:		TELÉFONOS DE ASEGURADO CASA: OFICINA:	
DATOS DEL CONTRATANTE NOMBRE: DOMICILIO:		TELÉFONOS DEL CONTRATANTE CASA: OFICINA:	
CLAVE DEL AGENTE			
REGRESA A LA PANTALLA DE DATOS DE RESULTADO	REGRESA A LA PANTALLA DE DATOS GENERALES DE PÓLIZA	REGRESA A LA PANTALLA DE DATOS DEL AGENTE	

Fig. 3.16. Prototipo de la pantalla principal de Consulta y Reasignación.

No. PÓLIZA	DATOS DEL ASEGURADO TELÉFONOS DE CASA Y DE OFICINA	
DATOS DE LA LLAMADA FECHA, HORA, DURACIÓN, RESULTADO DATOS DE LA PRÓXIMA LLAMADA OBSERVACIONES		
DATOS DEL AGENTE ASIGNADO ZONA ASIGNADA, NOMBRE DEL AGENTE, FECHA DE ASIGNACIÓN, FECHA DE FIRMA CLAVE DE AGENTE, OBSERVACIONES		
REGRESA A LA PANTALLA DE DATOS DEL AGENTE	REGRESA A LA PANTALLA DE DATOS RESULTADO	REGRESA A LA PANTALLA DEL MENU PRINCIPAL

Fig. 3.17. Prototipo de pantalla de Llamadas.

No. DE PÓLIZA		NOMBRE DEL ASEGURADO			
DATOS DE LA PÓLIZA FECHA DE EMISIÓN, FECHA DE VENCIMIENTO, SUMA ASEGURADA, PRIMA SEGÚN FORMA DE PAGO, STATUS DE LA PÓLIZA, FORMA DE PAGO.					
BENEFICIOS ADICIONALES: BIT IMA DIBA SET40 ISE DOP CADE					
REGRESA A LA PANTALLA DE DATOS DEL AGENTE		REGRESA A LA PANTALLA DE RESULTADO		REGRESA A LA PANTALLA DEL MENU PRINCIPAL	

Fig. 3.18. Prototipo de la pantalla de los Datos de la Póliza.

No. PÓLIZA		NOMBRE DEL ASEGURADO			
CLAVE DEL AGENTE:					
STATUS DEL AGENTE		FECHA DE CANCELACION DEL AGENTE			
FECHA DE CONEXIÓN DEL AGENTE		No. DE ZONA DEL AGENTE			
REGRESA A LA PANTALLA DE DATOS DE RESULTADO		REGRESO A LA PANTALLA DE DATOS GENERALES DE PÓLIZA		REGRESA A LA PANTALLA DEL MENU PRINCIPAL	

Fig. 3.19. Prototipo pantalla de los Datos del Agente.

Para el caso del submódulo de mantenimiento no existirán pantallas gráficas. La forma de cambiar o modificar la información en este submódulo será utilizando la tabla que se desea modificar. La figura 3.20. muestra el prototipo de lo que se observará cuando se llame al Módulo de Mantenimiento.

CONVENCIONES							
TABLA DE MANTENIMIENTO DE TIPOLOGIAS							
USUARIO	PASSWORD	PASSWORD SIS	PASSWORD V1	PASSWORD VIDA	PASSWORD V2	TEL USUARIO	NOMBRE DE USUARIO

TABLA DE MANTENIMIENTO DE USUARIOS

Fig. 3.20. Prototipo de la pantalla de Mantenimiento.

Hasta ahora se realizaron todos los pasos que conlleva un diseño para establecer las bases con las que se trabajarán en la fase de desarrollo del sistema POHUER. Con base a las tablas normalizadas se puede empezar a codificar el sistema y hacer las respectivas ligas entre ellas para la optimización de los informes requeridos. El diccionario de datos servirá de apoyo para definir las funciones de cada campo.

CAPÍTULO IV

DESARROLLO

El objetivo de este capítulo es presentar el sistema POHUER y la forma en que opera cada uno de sus módulos. Empezaremos por mostrar una descripción amplia y general del sistema POHUER. Se describirá cada uno de los módulos que conforman el sistema POHUER de manera individual. Se detallarán los aspectos más relevantes del sistema en cuanto a su operación. Finalmente, se mostrarán todos los reportes que genera el sistema.

4.1 CONCEPTOS BÁSICOS

El desarrollo, como se ha mencionado anteriormente, es la etapa del ciclo de vida de un sistema que prosigue a la etapa de diseño. La parte esencial de esta etapa la constituye la codificación del sistema.

CODIFICACIÓN

El paso de codificación traduce una representación del software, dada por un diseño detallado, a una realización en un lenguaje de programación. La interpretación equivocada de las especificaciones del diseño detallado pueden conducir a un código fuente muy complicado, que resulte difícil de probar y mantener.

Las características del lenguaje tienen un impacto directo sobre la calidad y la eficiencia de la traducción.

CARACTERÍSTICAS DE LOS LENGUAJES DE PROGRAMACIÓN

Se han estado escribiendo programas de computación desde que se desarrollaron las computadoras de propósito general hace aproximadamente 40 años. Es conveniente agrupar los distintos lenguajes de programación (existen cientos de lenguajes distintos que se utilizan en todo el mundo) en cuatro generaciones distintas:

- **Lenguajes de primera generación:** *Fueron los lenguajes máquina que se usaron en los años 50; los programadores que intentaban que la computadora hiciera algo útil codificaban sus instrucciones con unos y ceros binarios. Eran totalmente dependientes de la máquina.*
- **Lenguajes de segunda generación:** *Surgen a finales de los años 50 y principios de los 60. Son los sucesores del lenguaje de máquina, generalmente se conocen como lenguajes de ensamble o ensambladores. Son considerados como lenguajes de bajo nivel en el sentido de que el programador tiene que escribir una declaración por cada instrucción de máquina. Los lenguajes ensambladores son lenguajes de máquina con una correspondencia uno a uno entre las proposiciones del lenguaje de programación y las operaciones de la máquina. La programación en código ensamblador requiere que el programador traduzca el diseño de su programa a secuencias de acciones de la máquina. La programación en lenguaje ensamblador tiende a ser difícil y propensa a errores. Los lenguajes ensambladores son específicos de la familia de computadores en la que se aplican.*
- **Lenguajes de tercera generación:** *Incluyen BASIC, COBOL, FORTRAN, Pascal, C, Ada y muchos más. Son de alto nivel en el sentido de que una sola declaración usualmente representa cinco o diez declaraciones de lenguaje ensamblador (y a veces hasta cien); son de alto nivel en un sentido más importante, porque permiten al programador expresar pensamientos en una forma un tanto más compatible con el área del problema en el que está trabajando. Los lenguajes de tercera generación también se caracterizan como lenguajes guiados por procedimientos. Requieren que el programador piense con cuidado la secuencia de los cálculos o procedimientos necesarios para lograr alguna acción.*
- **Lenguajes de cuarta generación:** *Los lenguajes de cuarta generación, o 4GLs, son la moda actual y a menudo son considerados como el desarrollo más importante en el campo de software en los últimos 20 años.*

La mayor parte de los 4GLs tienen características de programación estructurada, además la mayoría de los detalles tediosos de programación se manejan automáticamente. Los detalles de programación pueden especificarse fácilmente con unas cuantas instrucciones. Los 4GLs han elevado aun más el nivel de abstracción que otros lenguajes, eliminan la necesidad de especificar detalles algorítmicos y han sido desarrollados para ser usados conjuntamente con aplicaciones de bases de datos.

CUESTIONES IMPORTANTES EN PROGRAMACIÓN

Sin tomar en cuenta el lenguaje de programación que se utilice, se debe de considerar para cualquier programa las siguientes cuestiones importantes:

- **Productividad:** *Probablemente la cuestión más importante de la programación actual sea la productividad: escribir más software, más rápidamente. Es por ello que se deben alentar los lenguajes y técnicas de programación que promueven la productividad; exceptuando casos raros, la productividad se considera más importante actualmente que la eficiencia.*
- **Eficiencia:** *Usualmente resulta importante minimizar la cantidad de tiempo de CPU requerido por el programa, también puede ser importante minimizar la utilización de memoria, al igual que la de otros recursos como el disco.*
- **Portabilidad:** *El usuario puede desear ejecutar el mismo sistema en distintos tipos de computadoras. Algunos lenguajes de programación son más portátiles que otros. No existe un lenguaje universalmente portátil; siempre hay forma de que el programador aproveche las características especiales de una computadora o sistema operativo específicos. Por ello, además del lenguaje de programación debemos preocuparnos por el estilo de programación, cuando la portabilidad es un factor importante.*
- **Mantenibilidad:** *Finalmente, se debe recordar que los sistemas viven durante mucho tiempo, por lo que el software debe mantenerse, es decir debe seguir cumpliendo con los requerimientos de los usuarios.*

PUNTOS CLAVE PARA LA PROGRAMACIÓN

- **La programación estructurada:** *Suponiendo que los programas se escriban en un lenguaje de tercera o cuarta generación, debe seguirse un enfoque de programación estructurada, en el que la lógica del programa (las decisiones y ciclos) se organiza en combinaciones anidadas.*

- **Módulos pequeños:** Es esencial que los programas se organicen en pequeños módulos para que la lógica de programación quepa en una sola página de listado de programa. Por ello, el programador puede tener que continuar la actividad de diseño, partiendo un módulo en submódulos de menor nivel, para que cada uno represente no más de 50 pasos de programación.
- **Sencillez de estilo:** Es muy importante crear programas que estén escritos de manera sencilla, para que cualquier otro programador pueda entenderlos y en determinado momento el darles mantenimiento resulte relativamente sencillo.

PROGRAMACIÓN ORIENTADA A OBJETOS

En la programación convencional, los datos asumen cualquier estructura y los procesos hacen de los datos todo lo que el programador desee. En el mundo orientado a objetos, las estructuras de datos se relacionan con los objetos y sólo pueden ser utilizadas mediante los métodos diseñados para ese tipo de objeto.

PARADOX

ObjectPAL es el lenguaje de programación integrado de PARADOX para Windows. Puede utilizarse para añadir a las aplicaciones de PARADOX funciones que no pueden añadirse de forma interactiva.

El lenguaje ObjectPAL cuenta con tres componentes:

- **Métodos:** Código anexo a un objeto que define su comportamiento.
- **Procedimientos:** Métodos encerrados entre los comandos PROC y ENDPROC.
- **Elementos básicos de lenguaje:** Los elementos de estructura básicos de ObjectPAL.

ObjectPAL se parece a los lenguajes tradicionales en que utiliza variables, ofrece estructuras de control como *if...then...else*, bucles *for* y *while*, realiza cálculos y proporciona un sistema para la creación de funciones denominados métodos y procedimientos.

La diferencia entre ObjectPAL y los lenguajes tradicionales es que se trata de un lenguaje basado en objetos. En los lenguajes tradicionales, la programación se propone como todo o nada. o bien se controla toda la aplicación de principio a fin o no se programa nada. Con ObjectPAL, sin embargo no es necesario proponerse una tarea tan desalentadora, ya que se centra en los objetos, y pueden programarse tantos o tan pocos como se deseen.

En PARADOX todo es un objeto, desde los botones y campos que se crean mediante las herramientas de la barra rápida, hasta las tablas y archivos de texto guardados en disco o los menús emergentes que se crean con código.

ENFOQUE POR PROCEDIMIENTOS

En la programación por procedimientos, son las funciones las que dictan la organización del programa.

ENFOQUE POR OBJETOS

En un lenguaje basado en objetos, se deja que el procesador de lenguaje decida el método que debe utilizarse según el tipo de objeto que llama al método: El código de programación subyacente está escrito para los objetos de forma específica.

4.2 PANORAMA GENERAL DEL SISTEMA POHUER

El objetivo principal del sistema POHUER es conjuntar la información referente a la Cartera Huérfana de GNP, en pantallas de fácil visualización para que el operador pueda tener una herramienta de consulta eficiente y rápida que le permita establecer el contacto con el asegurado y poder proporcionarle toda la información referente a su(s) póliza(s), hacer la reasignación del agente correspondiente y además llevar el control del número de llamadas realizadas por cada operador, así como los resultados obtenidos. También existe un módulo de mantenimiento que permite actualizar las tablas de tipologías y usuarios.

Como se mencionó en el capítulo anterior, sistema POHUER consta de cuatro secciones principales que conforman los módulos del sistema. Éstos son los siguientes:

- *Módulo de Consulta y Reasignación*
- *Módulo de Impresión de Reportes*
- *Módulo de Mantenimiento*
- *Módulo de Datos Adicionales*

El Módulo de Consultas y Reasignación opera por medio de tres pantallas. Estas tres pantallas realmente son una sola pero debido a la gran cantidad de datos que se tienen que insertar y visualizar, una pantalla es insuficiente para toda la información que se requiere. Podemos mencionar que este módulo es el más importante de los cuatro que existen, porque las tablas de datos que genera son la fuente de información para los otros módulos.

Como parte integral del sistema POHUER se contempla el Módulo de Impresión de Reportes. Este es un módulo generador de reportes específicos y generales, para que la Gerencia de Cartera Huérfana pueda analizar cada una de las pólizas, además de establecer las políticas y estrategias que le permitan la pronta recuperación de esta cartera, así como estadísticas generales y consolidados de operación. Una de las partes fundamentales de este módulo es la posibilidad de generar telegramas cuando se tiene la necesidad de contactar a los asegurados que no tienen teléfono.

El tercer módulo, *Módulo de Mantenimiento*, sirve para actualizar, dar de alta y modificar las claves de acceso de los usuarios, así como incrementar la tabla de tipologías.

Finalmente el *Módulo de Datos Adicionales*, se utiliza para capturar los datos de las referencias de un asegurado que no se encuentran dentro de los sistemas que alimentan a POHUER. Estos datos se tienen que buscar directamente en archivos físicos de los diversos centros regionales con que cuenta GNP.

A continuación se presenta la estructura del sistema POHUER dividida en módulos y submódulos:

1. CONSULTA Y REASIGNACIÓN

- ⇒ Principal
- ⇒ Generales Póliza
- ⇒ Datos del Agente
- ⇒ Resultados

2. IMPRESIÓN DE REPORTE

- ⇒ Crea Telegramas
- ⇒ Imprime Telegramas
- ⇒ Histórico de Telegramas
- ⇒ Consolidado
- ⇒ Estadístico General
- ⇒ Agentes Dispositores
- ⇒ Agentes Cancelados
- ⇒ Formas Para Captura
- ⇒ Estadísticas del Universo Total
- ⇒ Megareport

3. MANTENIMIENTO

- ⇒ Tabla de Usuarios
- ⇒ Tabla de Tipologías

4. DATOS ADICIONALES

- ⇒ Pantalla para Capturar la información de las referencias.

4.3 OPERACIÓN DEL SISTEMA

A continuación mencionaremos las generalidades que presenta el sistema POHUER para poder comprender claramente su funcionamiento. Se explicarán, además, las pantallas del sistema, así como el código que las genera, los aspectos más importantes de cada pantalla y las descripciones técnicas de los campos que se utilizan.

ENTRADA AL SISTEMA

Al momento de encender la terminal es necesario firmarse a la red, es decir introducir el nombre del usuario o LOGIN y posteriormente la clave de acceso o PASSWORD que tiene asignado. Una vez que son verificados, tanto el LOGIN como el PASSWORD, se inicia una sesión de WINDOWS 3.1 en donde se tiene que seleccionar la aplicación de PARADOX y ejecutarla. Cuando PARADOX esté debidamente cargado, será necesario abrir la ficha POHUMENU, como se muestra en la figura 4.1.

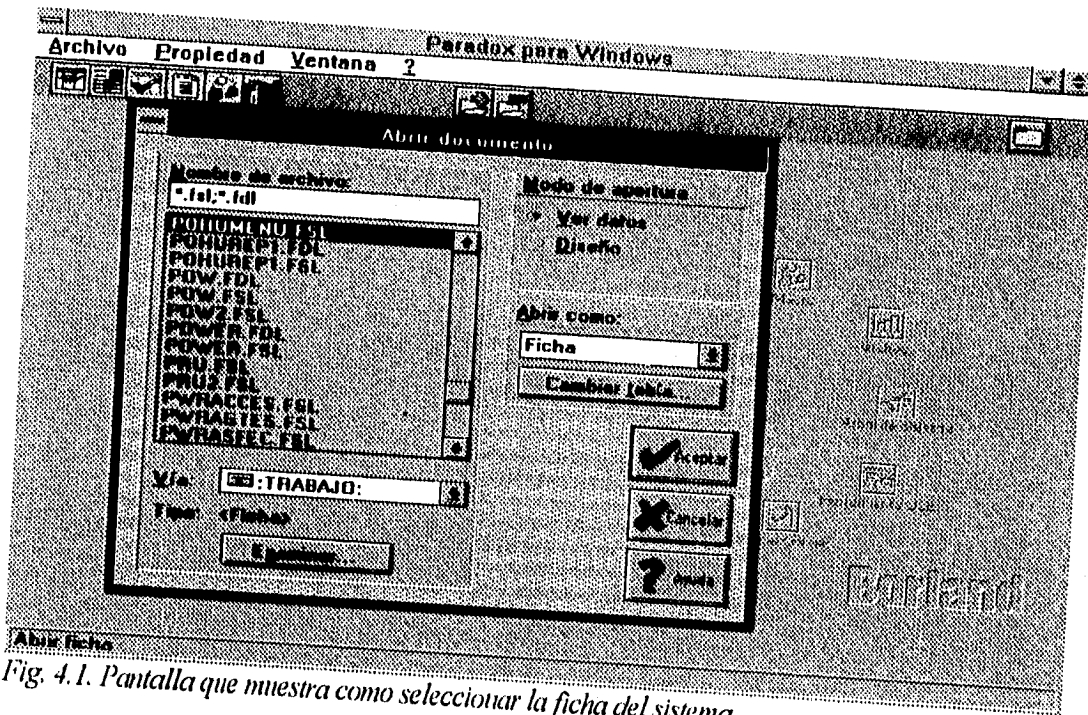


Fig. 4.1. Pantalla que muestra como seleccionar la ficha del sistema.

En la figura 4.2. se muestra la pantalla principal del sistema POHUER, en el momento en que se despliega un mensaje de ayuda. Es importante notar que el diseño del menú principal se identifica con el logotipo de la compañía, siendo cada uno de sus extremos un botón que nos permitirá entrar a cualquiera de los módulos que conforman el sistema. El botón de en medio (círculo) es la salida del sistema. Esta pantalla también nos permite hacer uso de las herramientas propias de PARADOX (Archivo, Edición, Ficha, Registro, Propiedad y Ventana) lo cual significa que podemos usar POHUER y al mismo tiempo editar, pegar, copiar o abrir alguna otra aplicación en caso de llegar a necesitarla.

El sistema cuenta con ayuda en línea que auxiliará al usuario en el momento que lo necesite, basta con que se posicione con el indicador del mouse en el botón en cuestión para que la ayuda se active.

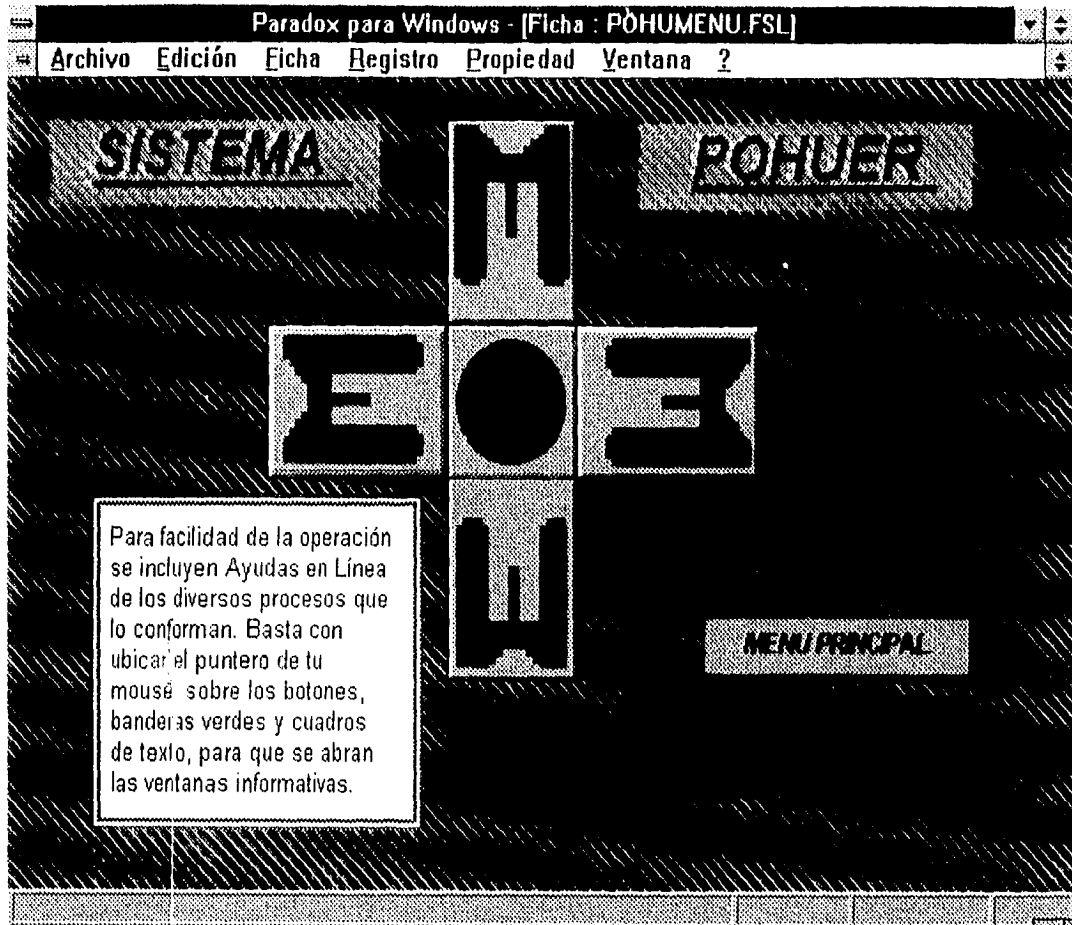


Fig. 4.2. Pantalla del menú principal con la ayuda en línea activada.

Si selecciona el botón superior de la cruz aparecerá un mensaje que indica que al presionar el botón se establecerá la conexión con los sistemas V2, V4 y COCOA, a través de la pantalla de Consulta y Reasignación.

Si se posiciona en el botón derecho se puede acceder al Módulo de Impresión de Reportes, cuando se posiciona en el botón inferior de la cruz se podrá entrar al Módulo de Mantenimiento. Finalmente en el botón izquierdo de la cruz se tiene el acceso al Módulo Datos Adicionales.

Prácticamente cualquier persona que tenga conocimientos mínimos de alguna aplicación en WINDOWS podrá hacer uso del sistema, considerando que las aplicaciones de este tipo, siempre tienen la misma forma de operación.

4.4 MÓDULO DE CONSULTA Y REASIGNACIÓN

Este módulo como se mencionó anteriormente es el más importante del sistema, debido a que en él se pueden hacer consultas de los datos de las pólizas y agentes cancelados con el fin de dar un tratamiento a las pólizas y reasignarles un nuevo agente.

Lo primero que realiza el sistema POHUER al abrir el módulo de consulta y reasignación es validar que sea un usuario autorizado la persona que intente abrir el sistema; es decir que esté previamente registrado en la tabla de usuarios. En caso de no estar registrado se desplegará un mensaje de error. En la figura 4.3. se presenta una pantalla donde se puede ver parte del código que realiza la validación de los usuarios.

```

method AbrirRumba()
var
  t1,t2      TCursor
  r,e,e1,e2  SmallInt
  Pass,dat   String
endVar

  Pass=""
  HI=0
  enumWindowNames("INF.DB")
  t1.open("INF.DB")
  if t1.locate("ClassName","PDOXWINDESKTOP") then
    t2.open("USUARIOS")
    if not t2.qlocate(GetNetUserName()) then
      beep()
      msgstop("Este usuario no esta registrado!",GetNetUserName())
    endif
    HI=t1."InstanceHandle"
    R=WD_RunProfile("U4.DSP",5)
    IF R<>0 THEN
      R.view("RunProfile")
    endif
    Conecta("")
    dat=space(100)
    e=size(dat)
  
```

Fig. 4.3. Código de validación de usuarios registrados.

Si es un usuario autorizado el programa permite establecer la emulación de terminal iniciando la llamada a los sistemas de V2 (VIDA), V4(SISI) y COCOA, utilizando los comandos de RUMBA. En la figura 4.4. se muestra parte del código en donde se ve claramente el proceso que se sigue para enviar las claves de acceso al host y las coordenadas donde se deben guardar. En la figura 4.5 se observan algunos de los comandos de RUMBA, que tienen que declararse para que PARADOX reconozca esas instrucciones y envíe datos al host.

```

Paradox para Windows [POWER:INTERFACE: AbrirRumba*]
Archivo Edición Lenguaje Depurar Propiedad Ventana ?
Envia(HI,"LOGON APPLID(IDMSDC)@E")
WD_Pause(HI,4)
Envia(HI,"SIGNON ")
Envia(HI,t2."SIGNON")
Envia(HI,"@E")
WD_Pause(HI,2)
Envia(HI,rtrim(t2."PASW_U4")+ "@E")
Envia(HI,"S SISI@E")
Envia(HI,rtrim(t2."PASW_SISI")+ "@E")
Envia(HI,"15")
Envia(HI,"@E")

R=WD_RunProfile("U2.DSP",5)
IF R<>0 THEN
  R.view("RunProfile")
endif
Conecta("A")
dat=space(100)
e=size(dat)
e2=0
while e2<20
  R=WD_CopyOIA(HI,dat,e)
  IF R<>0 THEN
    R.view("CopyOIA")
  quitloop

```

Fig. 4.4. Código que establece la conexión al Host.

```

Paradox para Windows [POWER:INTERFACE: Uses*]
Archivo Edición Lenguaje Depurar Propiedad Ventana ?
Uses EEHLAPI

WD_AdviseHostUpdate(hInstance CWORD, hWndMsg CWORD, Message CWORD) CWORD
WD_ConnectPS(hInstance CWORD, Name CPTR) CWORD
WD_Convert(hInstance CWORD, ConvertType CWORD, RowCol CLONG, Name CPTR) CWORD
WD_CopyFieldToString(hInstance CWORD, Position CWORD, Buffer CPTR, Length CWORD) CWORD
WD_CopyOIA(hInstance CWORD, Buffer CPTR, Length CWORD) CWORD
WD_CopyPS(hInstance CWORD, Buffer CPTR, Length CWORD) CWORD
WD_CopyPSToString(hInstance CWORD, Position CWORD, Buffer CPTR, Length CWORD) CWORD
WD_CopyStringToField(hInstance CWORD, Position CWORD, Buffer CPTR, Length CWORD) CWORD
WD_CopyStringToPS(hInstance CWORD, Position CWORD, Buffer CPTR, Length CWORD) CWORD
WD_DeletePS(hInstance CWORD, Name CPTR) CWORD
WD_DisconnectPS(hInstance CWORD) CWORD
WD_DisplayCursor(hInstance CWORD, Position CWORD, Name CPTR) CWORD
WD_DisplayPS(hInstance CWORD, Name CPTR) CWORD
WD_FindFieldLength(hInstance CWORD, Length CWORD, Position CWORD, FindData CWORD) CWORD
WD_FindFieldPosition(hInstance CWORD, Location CWORD, Position CWORD, FindData CWORD) CWORD
WD_GetKey(hInstance CWORD, GetKeyData CPTR) CWORD
WD_GetSessionHWND(hInstance CWORD) CWORD
WD_Pause(hInstance CWORD, Length CWORD) CWORD
WD_PostInterceptStatus(hInstance CWORD, Name CPTR) CWORD
WD_QueryCursorLocation(hInstance CWORD, Location CWORD) CWORD
WD_QueryFieldAttribute(hInstance CWORD, Attribute CWORD, Position CWORD) CWORD
WD_QueryHostUpdate(hInstance CWORD, Name CPTR) CWORD
WD_QuerySessionStatus(hInstance CWORD, SessionData CPTR) CWORD

```

Fig. 4.5. Comandos de Rumba, que se tienen que declarar en PARADOX.

➤ Pantalla Principal

La pantalla principal de este módulo, que se puede observar en la figura 4.6, se utiliza para la consulta de los datos de la póliza y el agente. Por medio de ésta es posible desplazarse a otras tres pantallas pulsando el botón correspondiente. Estas pantallas son: Generales de Póliza, Datos del agente y Resultados. Desde esta pantalla se puede regresar al menú principal.

Cuando se oprima el botón salir de esta pantalla, el sistema regresará a la pantalla del menú principal para poder elegir algún otro proceso o en su defecto salir del sistema.

Para consultar una póliza se pulsa el botón "Consultar Póliza". En donde se abre una caja de dialogo solicitando el número de póliza. Una vez que se inserta el número de póliza, se podrá recuperar la siguiente información:

- ⇒ ASEGURADO: Nombre del asegurado, domicilio y teléfono (casa y oficina).
- ⇒ CONTRATANTE: Nombre del contratante, domicilio y teléfono (casa y oficina).
- ⇒ AGENTE: Clave y nombre del agente.

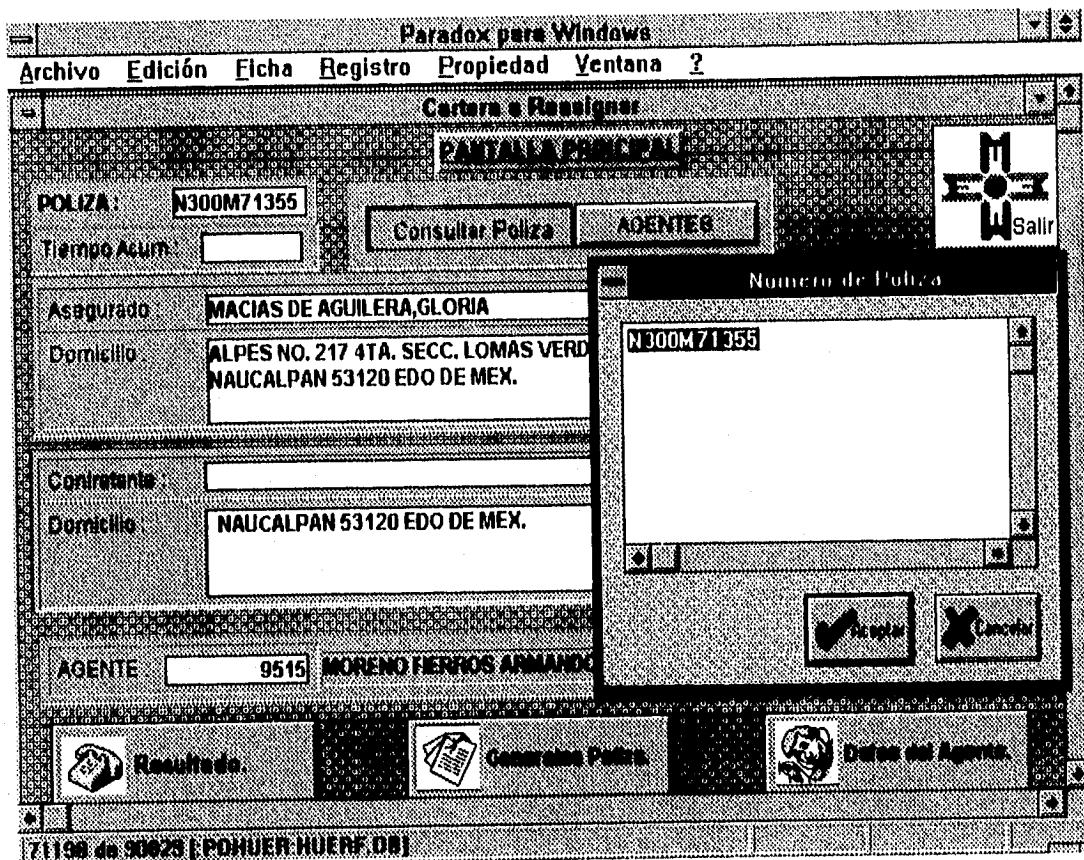


Fig. 4.6. Pantalla principal del Módulo de Consulta y Reasignación.

Descripción Técnica

La información que aparece dentro de esta pantalla es almacenada en la tabla HUERF. Los campos que se utilizan de esta tabla son los siguientes:

- ⇒ PÓLIZA: Número de póliza-ALP(10).
- ⇒ AGENTE: Clave del agente-NUM.
- ⇒ NOMBRE: Nombre del asegurado-ALP(40).
- ⇒ DOMICILIO: Domicilio del asegurado-ALP(136).
- ⇒ TEL_CASA: Teléfono de casa del asegurado-ALP(20).
- ⇒ TEL_OF: Teléfono de oficina del asegurado-ALP(40).
- ⇒ C_NOMBRE: Nombre del contratante-ALP(40).
- ⇒ C_DOM: Domicilio del contratante-ALP(136).
- ⇒ C_TEL: Teléfono de casa del contratante-ALP(20).
- ⇒ C_TEL_OF: Teléfono de oficina contratante-ALP(20).

► Generales de Póliza

En esta pantalla se pueden consultar los datos generales de la póliza, ver figura 4.7. Para llegar a esta pantalla sólo hay que presionar el botón de Generales Póliza desde la pantalla principal.

Paradox para Windows

Archivo Edición Ficha Registro Propiedad Ventana 2

PÓLIZA: N300M71355 Asegurado: MARIAS DE AGUILEA GONZA

EMISION: 08/31/87 Bono Asegurada: 270,000.00 GÍRNE Póliza: 43

VENCE: 08/07 Prima a/pago: 1,170.00 Forma de Pago: 12

COD. HABTA: 08/31/96 Monto Préstamo: 2,172.50 EDAD: Actual

Disponible en F.A.D.:

Beneficios Adicionales:

DIT INA DIBA ISETM ISE DOP CADE

71188 de 50828 (POHUEF HUERF.DB)

Fig. 4.7. Pantalla de los datos generales de la póliza.

La información que trae este submódulo es la siguiente:

- ⇒ PÓLIZA: Número de póliza que se está consultando.
- ⇒ ASEGURADO: Nombre del asegurado.
- ⇒ EMISIÓN: Fecha en la que el contratante adquiere una póliza.
- ⇒ VENCE: Fecha en la que expira la póliza.
- ⇒ COB. HASTA: Fecha de vigencia de la póliza por los pagos realizados.
- ⇒ SUMA ASEGURADA: Suma asegurada de la póliza.
- ⇒ PRIMA S/F PAGO: Cantidad que se paga según la forma de pago.
- ⇒ MONTO PRÉSTAMO: Cantidad que se le ha prestado al asegurado.
- ⇒ STATUS PÓLIZA: Situación en que se encuentra la póliza.
- ⇒ FORMA DE PAGO: Es un número que indica la forma en que se paga la póliza.
- ⇒ EDAD: Edad que tenía el asegurado al contratar la póliza.
- ⇒ ACTUAL: Edad con la que cuenta el asegurado actualmente.
- ⇒ DISPONIBLE EN F. A. D.: Fondo de Ahorro Disponible del asegurado.
- ⇒ BENEFICIOS ADICIONALES: Coberturas adicionales contratadas dentro de la póliza.

Los aspectos más importantes son los siguientes:

- ⇒ Desde esta pantalla se puede desplazar a la pantalla de Datos del agente, a la de Resultados o se puede regresar a la pantalla principal del módulo de Consulta y Reasignación, presionando el botón correspondiente.
- ⇒ El nombre del asegurado se carga desde la pantalla principal y se mantiene en esta pantalla sin necesidad de introducirlo por parte del usuario. De esta misma forma se carga el número de póliza.
- ⇒ En esta pantalla sólo se pueden consultar datos sin posibilidad de modificar alguno.
- ⇒ Para poder salir del Módulo de Consulta y Reasignación desde esta pantalla, será necesario regresar a la pantalla principal de este módulo y salir oprimiendo el botón de salida, ver figura 4.6.

Descripción Técnica

Los datos generales de una póliza se almacenan y actualizan en la tabla de HUERF. Los campos que utiliza son los siguientes:

- ⇒ P_F_EMI: Fecha de emisión de la póliza-ALP(8).
- ⇒ P_COB_HASTA: Fecha de cobro-ALP(8).
- ⇒ P_FORMA_PAGO: Forma de cobro de la póliza -ALP(8).
- ⇒ P_SUMA: Suma asegurada de la póliza-NUM.
- ⇒ P_PRIMA: Prima de la póliza-NUM.
- ⇒ P_STATUS: Status de la póliza-ALP(2).
- ⇒ P_EDAD: Edad real del asegurado-NUM.

- ⇒ P ACTUAL: Edad que se otorga al asegurado-NUM.
- ⇒ P BENEF: Beneficios adicionales de la póliza-ALP(6).
- ⇒ P FAD: Fondo de Ahorro disponible del asegurado-NUM.

➤ Datos del Agente

En esta pantalla se pueden consultar los datos del agente cancelado que representaba la póliza en cuestión, ver figura 4.8. Los datos que presenta el sistema POHUER dentro de esta pantalla son extraídos del sistema COCOA y se describen a continuación:

- ⇒ PÓLIZA: Número de póliza que se está consultando.
- ⇒ ASEGURADO: Nombre del asegurado.
- ⇒ AGENTE: Clave y nombre del agente.
- ⇒ STATUS: Está representado por una clave de dos dígitos que corresponde al motivo de cancelación del agente.
- ⇒ FECHA DE CONEXIÓN: Fecha de inicio de labores del agente.
- ⇒ FECHA DE CANCELACIÓN: Fecha en la que la clave del agente es cancelada.
- ⇒ ZONA: Número de la zona a la que está asignada el agente.

Parado para Windows

Archivo Edición Ficha Registro Propiedad Ventana ?

DATOS DEL AGENTE

POLIZA: **N300M71355** Asegurado: **MARIAS DE AGUILERA GEORJA**

AGENTE: **9515**

MORENO FERRAS ARMANDO

STATUS: **D4 BAJA PRODUCCION**

FECHA CONEXION: **31/Ago/87** FECHA CANCELA: **31/Dic/91**

ZONA: **039** **EMMA ANGELO DEL CABELLO**

3284998 0508/788/4098

Reservado Generar Ficha Principal

71158 de 50625 (POHUER-HUERF.DH)

Fig. 4.8. Pantalla que muestra los datos del agente.

Descripción Técnica

Para obtener la información requerida para llenar esta pantalla se utiliza la tabla HUERF y AGENTES de las cuales se emplean los siguientes campos:

- ⇒ PÓLIZA: Número de póliza-ALP(10).
- ⇒ NOMBRE: Nombre del asegurado-ALP(40).
- ⇒ AGENTE: Clave del agente-NUM.
- ⇒ STATUS: Status del agente-ALP(1).
- ⇒ FECHA_CONEXIÓN: Fecha de conexión del agente-DAT(1).
- ⇒ ZONA: Zona a la que pertenece el agente-NUM.

➤ Pantalla de Resultados

Esta pantalla es muy importante pues permite llevar el control de las llamadas realizadas a los asegurados y almacena los resultados obtenidos. Además guarda el nombre del último usuario que realizó la llamada, así como la fecha, hora y duración de la llamada, ver figura 4.9.

Paradox para Windows [Cartera a Reasignar]

Archivo Edición Ficha Registro Propiedad Ventana ?

RESULTADOS

POLIZA: N300M71355 Asegurado: MACIAS DE AGUILERA, GLORIA

Fecha Llamada: 03-08-06-55 Tel Casa: 03-08-06-55

FECHA	HORA	DURACION	RESULTADO	PROX LLAMA	HORA
10/Abr/96	13:07		05		
03/Mayo/96	11:17	4:21	06		

Esfuerzo: 3
 Telegrama

OBS: HABLO LA ASEGURADA Y SE LE COMENTO LA SITUACION DE SU POLIZA (EDO. 43) Y SOLICITO QUE SE LE VOLVIERA A LLAMAR PARA VER COMO SE PODIA RESOLVER LA

ZONA ABIO: AOTE ABIO: R Visita: 06 FECHA ABIO:

Nombre Agte: FECHA FIRMA:

OBS:

Datos del Agente Generar una Póliza Principal

2 de 2 [PONUER: LLAMADAS.DB] Campo

Fig. 4.9. Pantalla de Resultados.

La información que se maneja en esta pantalla es la siguiente:

- ⇒ *PÓLIZA: Número de póliza consultada.*
- ⇒ *ASEGURADO: Nombre del asegurado.*
- ⇒ *TEL_CASA: Teléfono de casa del asegurado.*
- ⇒ *TEL_OF: Teléfono de oficina del asegurado.*
- ⇒ *FECHA: Fecha en la que se hizo la llamada al asegurado.*
- ⇒ *HORA: Indica la hora en la que se hizo la llamada al asegurado.*
- ⇒ *DURACIÓN: Tiempo de duración de la llamada.*
- ⇒ *RESULTADO: Tipología que indica el resultado de la llamada.*
- ⇒ *PROX_LLAMA: Permite anotar la fecha en que se realizará la próxima llamada.*
- ⇒ *HORA: Permite anotar a que hora desea el asegurado se le llame nuevamente.*
- ⇒ *ESFUERZO: Indica a que número de esfuerzo pertenece esa póliza.*
- ⇒ *TELEGRAMA: Indica si se va enviar telegrama al asegurado.*
- ⇒ *OBS: En el se anotan los comentarios y peticiones que se consideren importantes..*
- ⇒ *ZONA ASIG: Número de la zona a que pertenece el agente reasignado.*
- ⇒ *AGTE ASIG: Clave del agente reasignado para atender la póliza.*
- ⇒ *R_VISITA: Indica el resultado obtenido al visitar al asegurado.*
- ⇒ *FECHA ASIG.: Fecha en que fue reasignado el nuevo agente.*
- ⇒ *NOMBRE AGTE: Nombre del nuevo agente.*
- ⇒ *FECHA FIRMA: Fecha en que se firma la autorización del agente*
- ⇒ *OBS: Observaciones sobre el nuevo agente.*

Los aspectos más importantes de esta pantalla son los siguientes:

- ⇒ *Con esta pantalla también podemos tener acceso a las pantallas de datos del asegurado, Datos Generales de póliza y regresar a la pantalla principal. Para salir del Módulo de Consulta y Reasignación es necesario seguir el mismo procedimiento de las otras pantallas, descrito anteriormente.*

Al presionar el botón de iniciar llamada aparecerá una caja de dialogo preguntando si contesto el asegurado y automáticamente se activa la fecha y la hora del inicio de la llamada, además el botón de iniciar llamada cambia por el de termina llamada. Si la respuesta es no, automáticamente se pone en el campo de resultado la tipología 00 que indica que no contesto. Si la llamada fue contestada, entonces se tiene que presionar el botón de "Termina Llamada" y se despliega la tabla de tipologías para que se pueda seleccionar el resultado que se obtuvo de esa llamada ver fig. 4.10. Una vez seleccionado el resultado obtenido el cursor se posiciona en el próximo campo que es el de próxima llamada y en el campo de duración automáticamente se pone el tiempo que duró la llamada.

Paradox para Windows

Archivo Edición Tabla Registro Propiedad Ventana ?

Cartera a Reasignar

POLIZA: Asegurado:

Terminar Llamada Tipología Tel Casa: Tel Ofi:

FECHA	HORA	DURACION	RESULTADO
05 Mayo/94	15.33	7.00	

OBS:

ZONA ASIG: AGTE ASIG:

Nombre Agte:

ORO:

Tabla RESULTADO

RESULTADO	RES	RESULTADO
1	01	Núm. equivocado
2	04	Se reporte con nosotros
3	06	Nuevo núm. comunicamos
4	08	Reportamos nuevamente
5	07	Ag. anterior de servicio
6	09	Agte. Rehabilitado
7	10	Visitar para reasignar ag
8	11	Visitar para realizar cob
9	12	Plan POR
10	13	Prestamos Elevados
11	14	L.U. incrementos
12	42	Asegurado no se interesa
13	50	Agente intocable

pagina 1 de 13

Fig. 4.10. Pantalla que muestra el inicio y término de una llamada.

Descripción Técnica

Para llenar los datos necesarios de esta pantalla es necesario utilizar la tablas de HUERF y LLAMADAS. Los campos que se utilizan de estas tablas son los siguientes:

- ⇒ PÓLIZA: Número de póliza-ALP(10).
- ⇒ NOMBRE: Nombre del asegurado-ALP(40).
- ⇒ TEL_CASA: Teléfono del asegurado-ALP(20).
- ⇒ TEL_OF: Teléfono de la oficina del asegurado-ALP(40).
- ⇒ FECHA: Fecha de realización de la llamada-DAT.
- ⇒ HORA: Hora en que se efectúa la llamada-NUM.
- ⇒ DURACIÓN: Tiempo de duración de la llamada-NUM.
- ⇒ RESULTADO: Resultado de la llamada-ALP(2).
- ⇒ PROX_LLAMA: Hora en que se realizará la próxima llamada-NUM.
- ⇒ CONTESTO: Contestaron la llamada (si/no)-ALP(1).
- ⇒ OBS: Comentarios sobre la llamada en caso de que existan-MEM(2).
- ⇒ N_ZONA: Número de nueva zona asignada-NUM.
- ⇒ N_AGENTE: Clave del nuevo agente-NUM.
- ⇒ N_F_ASIG: Fecha de asignación del nuevo agente-DAT.
- ⇒ N_N_AGENTE: Nombre del nuevo agente asignado-ALP(40).
- ⇒ A_OBS: Observaciones acerca del nuevo agente (puede no haber)-MEM(45).

Parte del código que genera la pantalla de Resultados se muestra en la figura 4.11.


```

method pushButton(var eventInfo Event)
var
  t,t1 Number
  s String
  tv TableView
endvar

IF Inicia=0 then
  FECHA.end()
  FECHA.InsertAfterRecord()
  FECHA=Today()
  ULT_USUAR=GetNetUserName()
  ULT_FECHA=Today()
  HORA=hour(DATETIME())+Number(minute(DateTime())/100)
  USUARIO=GetNetUserName()
  Tiempo=dateTime()
  s=msgQuestion("Resultado de llamada","¿Contesta el asegurado?")
  if s="No" then
    FECHA.end()
    Inicia=0
    DURACION=minute(DATETIME()-Tiempo)+number(second(DATETIME()-Tiempo))
    t=NUMBER(ACUM)+DURACION
    t1=t-int(t)
    if t1>.59 then
      t=t+1-.6

```

Fig. 4.11. Código para llevar el control y resultado de las llamadas realizadas.

4.5 MÓDULO DE IMPRESIÓN Y REPORTE

El módulo de Impresión de Reportes se utiliza para emitir reportes necesarios para la gerencia. La pantalla principal de impresión de reportes muestra las diversas opciones a las que se tendrá acceso con sólo presionar el botón de la opción deseada, ver figura 4.12.

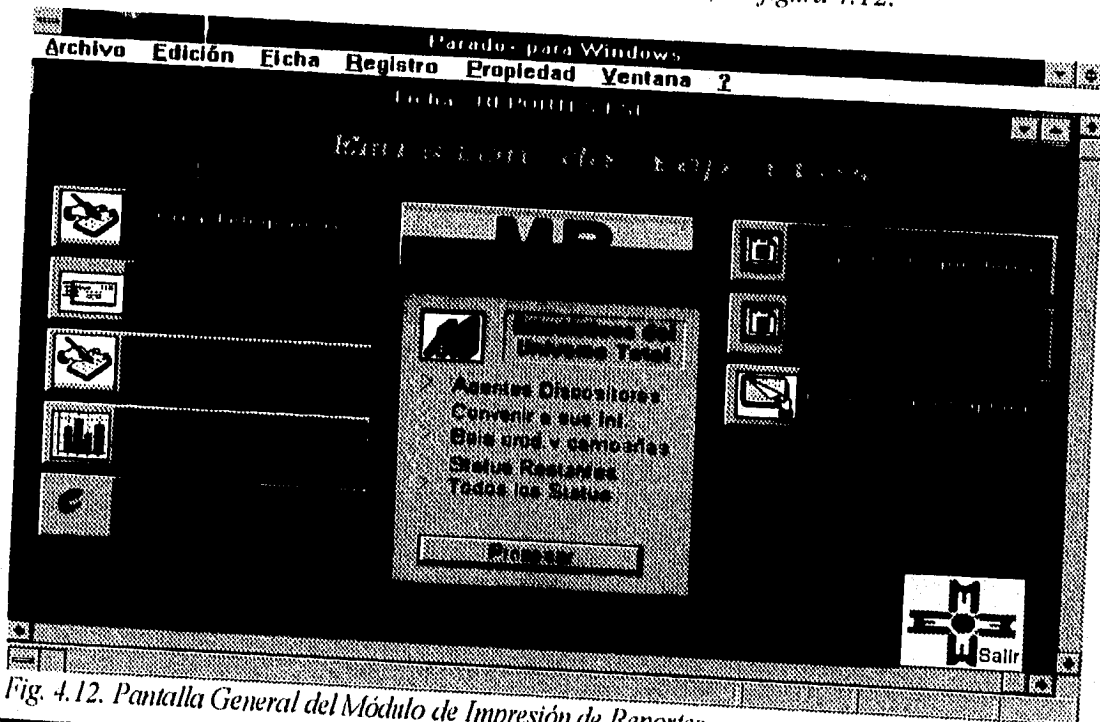


Fig. 4.12. Pantalla General del Módulo de Impresión de Reportes.

Las pantallas a las que se puede tener acceso por medio de la pantalla general de este módulo son las siguientes: *Crear Telegramas, Imprime Telegrama, Historial de Telegramas, Consolidados, Estadístico General, Estadísticas del Universo Total de agenes cancelados, Agentes dispositores, Agentes Cancelados y Formas para captura.* El botón que se puede observar en la parte inferior derecha de esta pantalla, se utiliza para salir al menú principal.

► Crea telegramas

Esta opción permite generar telegramas en forma masiva o individual. En la figura 4.13. se muestra como generar un sólo telegrama.

Los campos utilizados en esta pantalla se describen a continuación:

- ⇒ *FEC_GENER:* Fecha de generación de telegrama.
- ⇒ *REGIONAL:* Centro Regional a la que pertenece la póliza del asegurado.
- ⇒ *ZONA:* Zona a la que pertenece el asegurado.
- ⇒ *STATUS:* Status del Agente.
- ⇒ *NUM. PÓLIZA:* Número de póliza.
- ⇒ *AGENTE:* Clave del agente.
- ⇒ *RANGO DE PRIMA:* Rango de prima.



Fig. 4.13. Pantalla de Creación de Telegramas.

En las figuras 4.14. y 4.15. se encuentra el código que establece los parámetros requeridos para la creación de los telegramas. Esto se realiza a través de consultas realizadas a las tablas involucradas con estos parámetros. Se despliegan mensajes de error en caso de que el usuario introduzca datos no válidos. La solución derivada de la consulta se almacena en una tabla temporal en el directorio personal del usuario, esta información se necesita para llenar el formato de los telegramas y también se acumula en otra tabla para llevar un histórico de los telegramas enviados.

Aspectos importantes de la creación de telegramas.

- ⇒ Automáticamente se escribe la fecha actual, pero puede modificarse de acuerdo a las necesidades del operador..
- ⇒ No es necesario llenar todos los parámetros de la pantalla.
- ⇒ En caso de que se deje en blanco alguno de los parámetros de la pantalla, por default el sistema traerá todos los datos que existen de ese parámetro.
- ⇒ Para el caso de los parámetros regional, zona, status, mín. de póliza y agente, el sistema verifica que los datos insertados sean del tipo correcto permitido para cada parámetro.

```

method pushButton(var eventInfo Event)

var
  s,s1,s2,s3,s4  string
  t1,t2,t3      tCursor
  n              LongInt
  d              date
  r              report
endVar

d=(T_FECHA)
message("Procesando...")
s1="Query\n\nHUERF.DB|POLIZA"
s2="| check "
if T_Poliza<>blank() then
  s2=s2+upper(T_Poliza)
endif
if T_REG<>blank() then
  s1=s1+"| CEN_REG"
  s2=s2+"| "+T_REG
endif
if T_ZONA<>blank() then
  s1=s1+"| N_ZONA"

```

Fig. 4.14. Código que genera los telegramas.

```

s=s1+"|n"+s2+"|n\nEndQuery"
IF not executeQBEStrng(s) then
  beep()
  msgStop("Error","Datos Incongruentes")
else
  message("Creando Telegramas...")
  t1.open("PERSONAL:SOLUCION.DB")
  t2.open("HUERF.DB")
  t3.open("TELEGRAM.DB")
  t3.edit()
  n=0
  n1=0
  scan t1:
    n=n+1
    if not t2.qlocate(t1."POLIZA") then
      msgstop("Error","en sistema y Poliza "+t1."POLIZA")
    endif
    t3.insertAfterRecord()
    t3."POLIZA"=t2."POLIZA"
    t3."FECHA"=d
    t3."ASEGURADO"=t2."NOMBRE"
    t3."DIRECCION"=t2."DOMICILIO"
    t2.edit()

```

Figura 4.15. Continuación del código que genera los telegramas.

Descripción Técnica

La información que se introduce en la forma de captura de creación de telegramas se direcciona a la tabla TELEGRAM. Una vez generado el o los telegramas, la información se guarda en los siguientes campos:

- ⇒ PÓLIZA: Núm. de póliza - ALP(10).
- ⇒ FECHA: Fecha de Generación - DAT.
- ⇒ ASEGURADO: Nombre del asegurado.- ALP(40).
- ⇒ DIRECCIÓN: Dirección del asegurado -ALP(136).
- ⇒ USUARIO: Nombre del Usuario - ALP(8).

Al iniciar la ejecución de esta forma el cursor se posiciona automáticamente en el parámetro de regional. A partir de aquí se puede mover el cursor a cualquier otro parámetro o a cualquier otro lugar como ya se explicó. Una vez que se generan los telegramas aparece una pantalla más pequeña indicando el número de telegramas que se han generado.

► Imprime Telegrama

Esta parte del módulo fue creado para imprimir los telegramas que con anterioridad se hayan creado, es importante notar que la impresión de un telegrama puede o no ser el mismo día en que se genera. En el caso de que se genere un telegrama y no se imprima inmediatamente, el sistema guardará la información del o los telegramas generados para posteriormente imprimirlos.

Cuando se requiera de este submódulo será necesario oprimir el botón de "Imprime Telegrama", como se puede apreciar en la figura 4.16. Una vez que se presiona este botón aparecerá una pantalla pequeña esperando que se introduzca la fecha de generación de los telegramas que se quieren imprimir. Posteriormente pedirá la fecha de impresión, es decir la fecha que se desee aparezca en el formato del telegrama. Esta fecha podrá ser elegida libremente por el usuario, con el fin de poder imprimir telegramas con anticipación y enviarlos oportunamente.



Fig. 4.16. Esta pantalla muestra la activación de la pantalla de Imprime Telegrama.

En la figura 4.17 se muestra el código que hace la conversión del formato de la fecha que aparece en la pantalla de imprime telegrama. En este código se puede observar la llamada a un método denominado `pushButton`, que por medio de un arreglo realiza la conversión del formato de fecha numérico al formato de fecha que se utiliza tradicionalmente en los telegramas, es decir, la fecha completa.

```

method pushButton(var eventInfo Event)
VAR
  r Report
  d,d1 date
  t Table
  t1,t2 Tcursor
  mes array[12] string
ENDVAR

mes[1]="Enero"
mes[2]="Febrero"
mes[3]="Marzo"
mes[4]="Abril"
mes[5]="Mayo"
mes[6]="Junio"
mes[7]="Julio"
mes[8]="Agosto"
mes[9]="Septiembre"
mes[10]="Octubre"
mes[11]="Noviembre"
mes[12]="Diciembre"

d=Today()

```

Fig. 4.17. Código que hace la conversión del formato de la fecha.

En la figura 4.18. se observa el formato de un telegrama antes de ser enviado a impresión. Cuando son generados muchos telegramas, el sistema distribuye equitativamente los telegramas a cada uno de los usuarios, para equilibrar la carga de trabajo y no saturar la línea telefónica de un sólo usuario.

```

TELEGRAMA
CLASE DE SERVICIO ORDINARIO MEXICO D.F.
25 de Abril de 1995.
DESTINATARIO
ASEGURADO: STENY BALLINAS GUTIERREZ, ROGER
DIRECCION: FAISAN NO 1-C FUENTES DE SATELITE
MENSAJE
URGENTE. COMUNICARSE GRUPO NACIONAL PROVINCIAL.
REFERENTE SEGURO DE VIDA. 2273999 EXT. 2521 .
REMITENTE
ALEJANDRO HASTIDA

```

Fig. 4.18. Formato del telegrama.

Descripción Técnica

Cuando se le introduce la fecha de generación el sistema busca en la tabla de TELEGRAM los telegramas que se hayan generado en esta fecha.

⇒ FECHA: Fecha de generación de telegrama-DAT.

➤ Histórico de Telegramas

Esta opción presenta un histórico de los telegramas que se han enviado a un asegurado, muestra las fechas en que fue generado el telegrama, la fecha en que fue impreso o enviado, el nombre del asegurado y el nombre del usuario que lo generó. En la figura 4.19. se muestra el formato de la pantalla de Histórico de Telegramas.

Descripción Técnica

Para obtener esta información es necesario acceder a las tablas de HUERF y TELEGRAM. Los campos que se emplean de estas tablas son los siguientes:

⇒ PÓLIZA: Número de póliza-ALP(10).

⇒ FECHA: Fecha de generación-DAT.

⇒ ASEGURADO: Nombre del asegurado-ALP(40).

⇒ USUARIO: Nombre del usuario-ALP(8).

⇒ FEC_IMPR: Fecha de impresión del telegrama.-DAT

➤ Consolidado

Esta selección se utiliza para generar reportes estadísticos con respecto a los usuarios y a las llamadas realizadas para contactar a los asegurados.

Para generar el Consolidado es necesario presionar el botón correspondiente a esta opción y posteriormente se abrirá una pantalla, ver fig. 4.20. Esta pantalla se maneja por rangos de fecha y también se clasifica por esfuerzo realizado. Sin embargo, el Consolidado puede generar estadísticas de un solo día si fuera necesario. Además hace uso de tablas de paso, con el fin de traer siempre información actualizada y no almacenar datos que sólo sirven en un periodo determinado. Estas tablas proveen de información a 3 reportes: Pwropdib, Pwropdic, Propdid, que son los que se generan al presionar el botón que se encuentra a la derecha de la banderilla que tiene el mensaje de "se imprime". Aquí también se maneja ayuda en línea, que nos indica el formato que debe llevar la fecha.

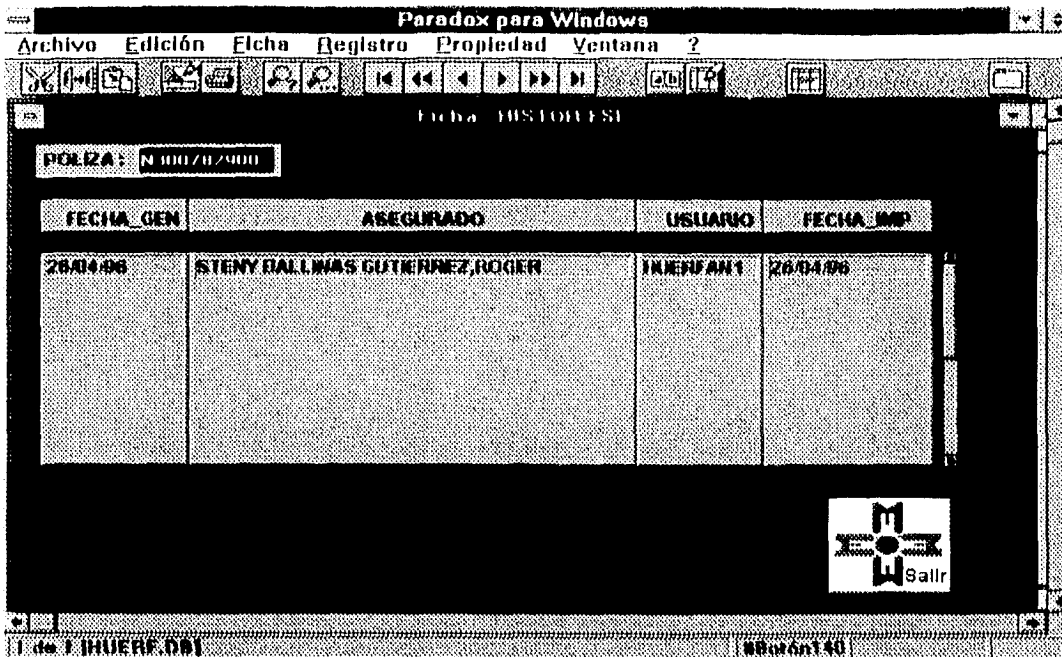


Fig. 4.19. Pantalla del Subproceso de Histórico de Telegramas.

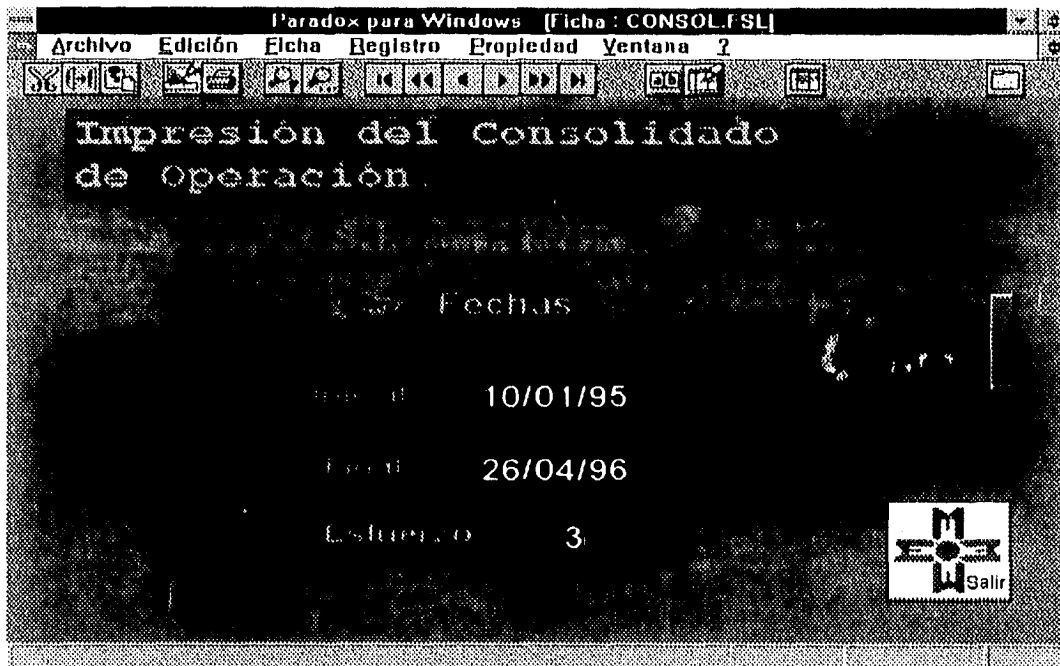


Fig. 4.20. Pantalla del Consolidado de Operación.

En el reporte llamado *pwpropdid* se ve a detalle el número de llamadas realizadas por operador, el tiempo ocupado en esas llamadas, los resultados obtenidos en cada una de esas llamadas desglosadas por tipología, ver fig. 4.21.

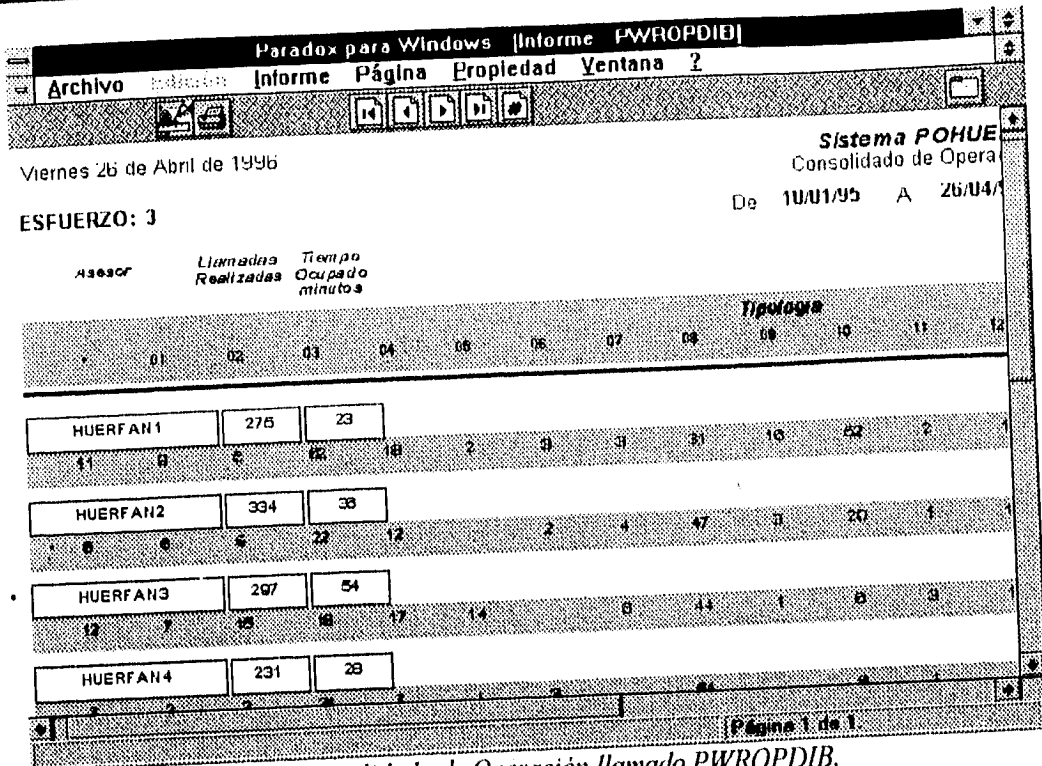


Fig. 4.21. Reporte del Consolidado de Operación llamado PWRDPDIB.

En el reporte llamado pwpropdic se muestran los porcentajes de rendimiento de cada usuario y el porcentaje de cada una de las tipologías por usuario, ver figura 4.22.

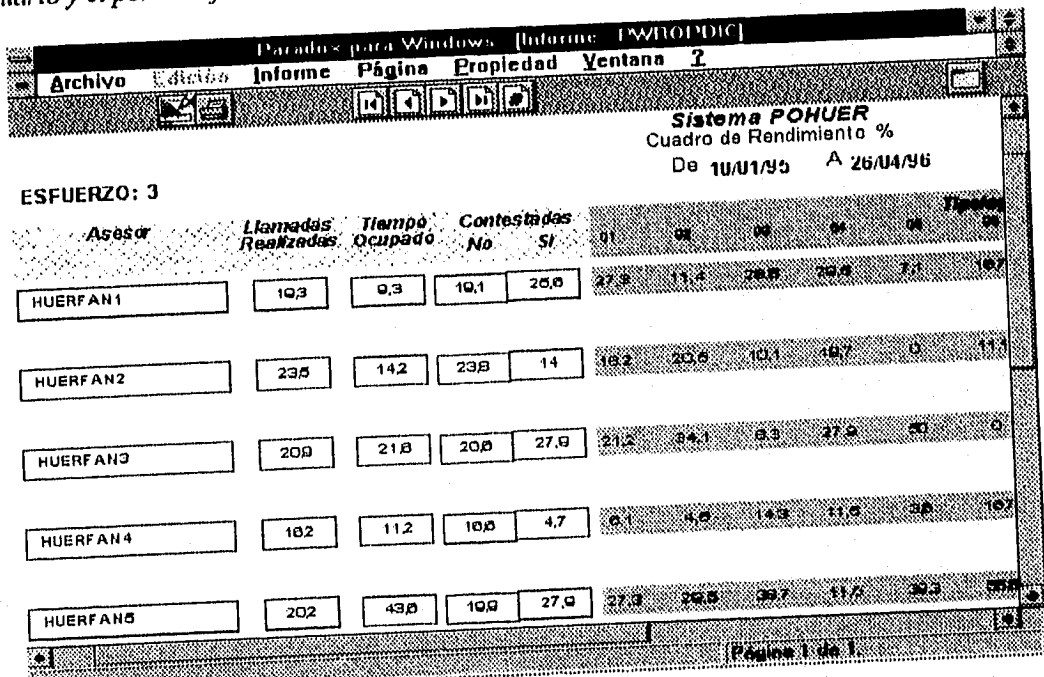


Fig. 4.22. Reporte del Cuadro de Rendimiento Porcentual.

En el último reporte llamado *pwpropdib* se muestra una gráfica de barras donde se ve el número de llamadas contestadas y no contestadas por cada operador, también el volumen de llamadas realizadas contra el tiempo que se tardan en ellas. Indicando también el número de esfuerzo al que pertenece, ver figura 4.23.

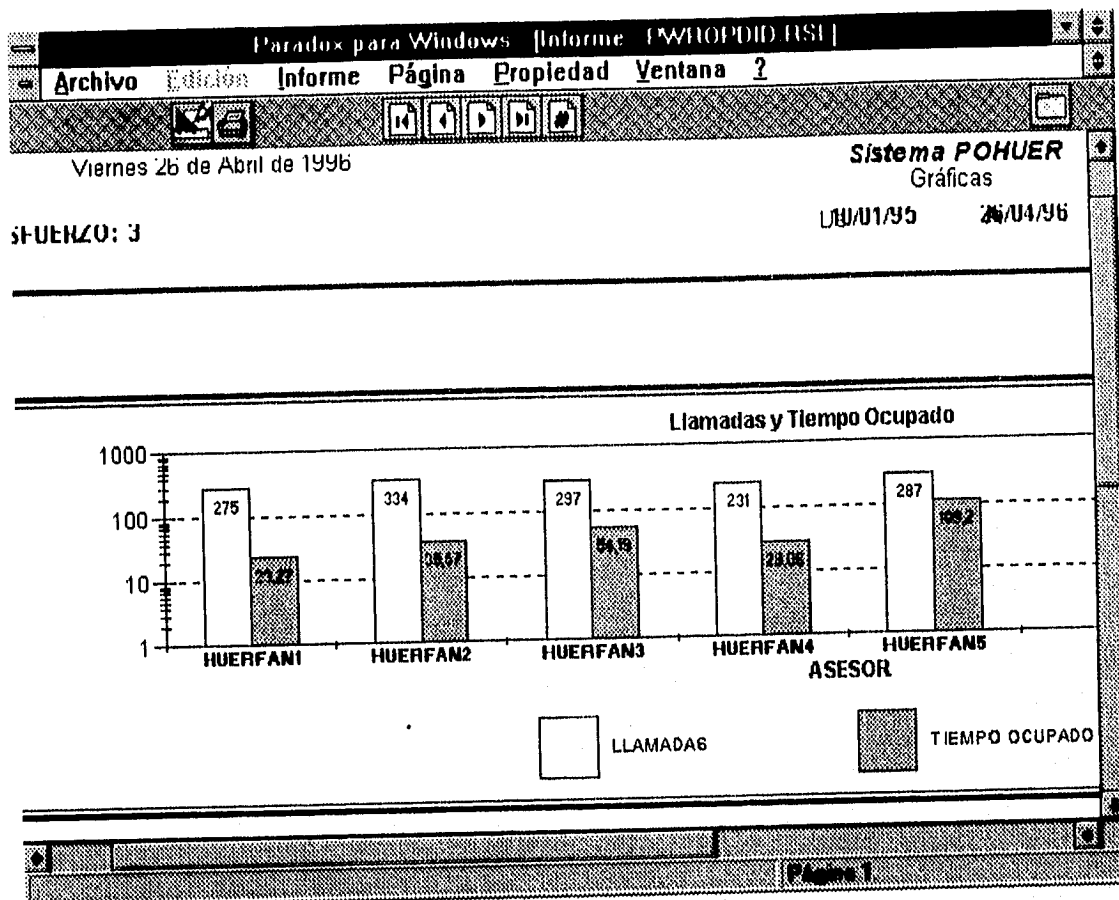


Fig. 4.23. Reporte de las Gráficas de Llamadas y Tiempo Ocupado.

Para poder realizar el conteo y generar los reportes correspondientes es necesario que se accese a las tablas de *HUERF* y *LLAMADAS*. Los campos que se utilizan de estas tablas son:

- ⇒ *USUARIO*: Nombre del usuario-ALP(8).
- ⇒ *DURACIÓN*: Tiempo que duró la llamada realizada-*NUM*.
- ⇒ *RESULTADO*: Resultado de la llamada.
- ⇒ *CONTESTO*: Contestaron la llamada (si/no).

Es importante mencionar que este módulo otorga información principalmente para saber el desempeño general de cada usuario.

➤ Estadístico General

Al igual que el Consolidado de Operación, el Estadístico General solicita un rango de fechas, así como un número de esfuerzo para poder generar la estadística correspondiente. La diferencia de este reporte con los que genera el submódulo de Consolidado, es que se manejan las llamadas considerando únicamente el tipo de topología que se le asignó. El formato del reporte no se incluye debido a que es muy similar a los reportes de Consolidado General y el Cuadro de Rendimiento Porcentual.

En la figura 4.24, se muestra parte del código que se utiliza para el Estadístico General. Este código abre la tabla pwrestad y la vacía por si tiene alguna información, después copia la estructura al directorio personal para que ésta provea de información al reporte que se genera.

Se realizan consultas a la tabla HUERF para obtener la información de la póliza, el último resultado, la prima, la última fecha y número de esfuerzo realizado.

```

method pushButton(var eventInfo Event)
UAR
  T
  T1,T2,T3      Table
  N,N1         Tcursor
  Fecha1,FechaF Number
  R           Date
  Q           Report
  S           Query
  S           String
ENDUAR

Fecha1=Fecha_Inicial
FechaF=Fecha_Final

T.Attach("PWRESTAD")
T.Empty()
T.Copy("":PERSONAL:PWRESTA1.DB")
T.Unattach()

s=ESFUERZO
q=Query
  huerf.db  |POLIZA  |ULT_RESULT  |ULT_FECHA  |P_PRIMA  |ESFUERZO
            |Check   |CHECK      |CHECK      |CHECK     |CHECK ~S
  
```

Fig. 4.24. Código del Estadístico General.

➤ Estadísticas del Universo Total

Dentro de esta sección se generan estadísticas sobre los agentes con los siguientes status:

- ⇒ AGENTES DISPOSITORES
- ⇒ CONVENIR A SUS INTERESES
- ⇒ BAJA PRODUCCIÓN Y CAMPAÑAS
- ⇒ STATUS RESTANTES

⇒ TODOS LOS STATUS

Agente dispositor: Este tipo de status pertenece a los agentes que por algún motivo u otro disponen de la prima.

Convenir a sus intereses: Este tipo de status pertenece a los agentes que por alguna razón en particular no desean seguir atendiendo una póliza.

Baja producción y campañas: Este status se otorga a los agentes que han demostrado a través de un periodo de tiempo una venta insuficiente de seguros y pérdida de clientela.

Status restantes: Se generan estadísticas de los agentes con los status diferentes a los que se mencionaron anteriormente.

Todos los status: Esta opción permite generar estadísticas tomando en cuenta todos los status que puedan existir.

Para generar algún reporte de este tipo basta con seleccionar que tipo de datos se necesitan y oprimir el botón de "Procesar". En este momento aparecerá una pequeña ventana que pedirá que se le introduzca un rango de primas, ver figura 4.25. Después de oprimir el botón de "Aceptar" se generará un reporte específico sobre la información requerida. Este reporte se muestra en la figura 4.26.

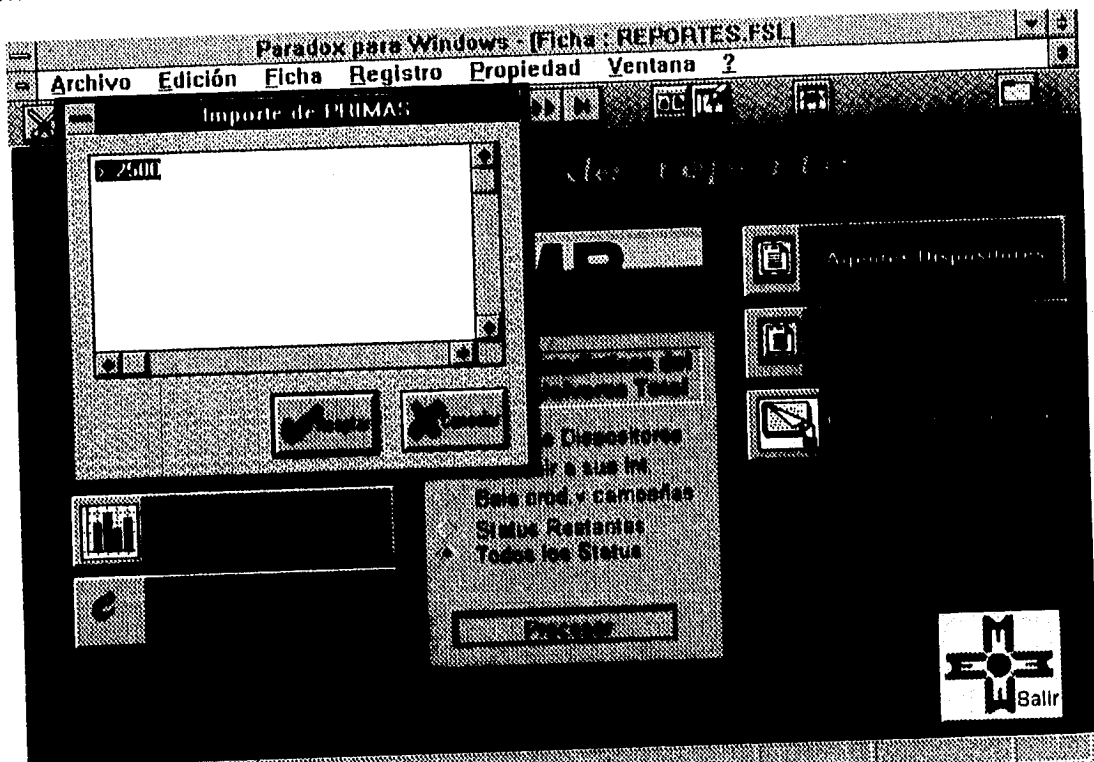


Fig. 4.25. Pantalla de captura de parámetros para el Estadístico del Universo Total.

Una vez que se ha impreso el reporte, el sistema regresa nuevamente a la pantalla principal de Impresión de Reportes.

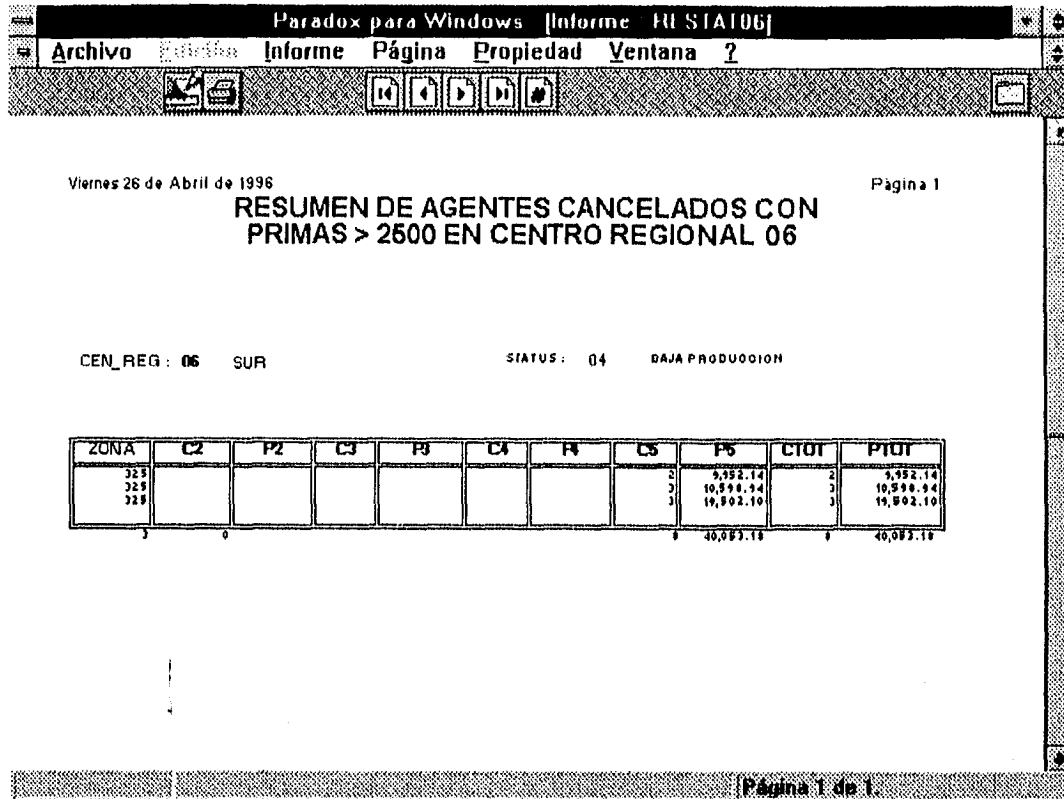


Fig. 4.26. Informe que se obtiene del Estadístico del Universo Total.

Descripción Técnica

En esta sección se hará uso de las tablas HUERF, CVAGT, STATUS y CENREG.

➤ AGENTES DISPOSITORES, AGENTES CANCELADOS Y FORMAS PARA CAPTURA.

En el caso de estos submódulos, su objetivo principal es crear un reporte de agentes dispositores o agentes cancelados y formas para captura tomando en cuenta algunos parámetros. Los parámetros son:

- ⇒ Importe de primas
- ⇒ Centro Regional
- ⇒ Zona

En la figura 4.27. se muestra el informe de los Agentes Dispositores cumpliendo las restricciones que se le fueron dadas.

Paradox para Windows [Informe AGTDISP RSL]

Archivo Informe Página Propiedad Ventana ?

AGENTES DISPOSITORES CENTRO REGIONAL

STATUS: 05 CONVENIR A LA CIA.

ZONA: 16

HOMBRE	CVE_HAC	CVE_PROV	FECHA CONEXION	FECHA CANCELACION
ABAQI BASIL JEAN STEPHAN	8882	8848	13/04/84	14/10/88
JIMENEZ ARENAS AIDA	8437	5990	20/02/84	4/12/87
LIRA Y MONCADA RICARDO	8388	5740	3/02/84	8/12/88
OLAETA ABURTO ESTHER	2784	480	23/02/79	22/04/83

4 REGISTROS POR ZONA

ZONA: 18

HOMBRE	CVE_HAC	CVE_PROV	FECHA CONEXION	FECHA CANCELACION
AQUILAR MENDOZA ANTONIO	7241	3803	8/04/83	30/07/87
REGALADO LOBO ALFONSO	8823	8823	4/05/83	23/10/85
WARNHOLTZ UNOHVARY VON R ELIZABE	3294	3294	1/08/75	24/02/89

3 REGISTROS POR ZONA

ZONA: 326

Página 1

Fig. 4.27. Informe de agentes dispositores.

La forma de insertar los parámetros es idéntica a la forma en que se introduce el rango de primas para el caso de los reportes de Estadísticas del Universo Total. Se abren ventanas que van solicitando la información del centro regional, la zona y rango de primas, para completar la información necesaria y generar los reportes requeridos. En el momento en que se llena un parámetro y se oprime el botón de aceptar desaparece esta pantalla y aparece la siguiente, hasta llenar los parámetros solicitados.

Descripción Técnica

Para generar estos reportes es necesario abrir las tablas AGENTES, CVEAGT, y ZONAS. Para el caso del submódulo de Formas de Captura es necesario acceder a las tablas de HUERF y FORMAS. Estos reportes ya no se muestran porque son muy similares a los ya mencionados.

➤ Megareport

El propósito de este reporteador es evitar que el usuario se vuelva dependiente del área de sistemas. Puede realizar los listados que quiera en el momento que desee, con las restricciones que solicite. El Megareport muestra los campos de todas las tablas involucradas.

En la figura 4.28. se muestra parte del código usado en el Megareporte. Se puede observar que lo que más se maneja son las consultas entre varias tablas ligadas a través de algunos campos clave, como son el número de la póliza, la clave del agente, la zona, y el centro regional.

En esta pantalla podemos ver que primero se definen las variables que se van a usar, después se utiliza un ciclo anidado para reconocer los nombres de cada una de las casillas utilizadas y establecer los casos en que son solicitados estos datos. Las herramientas más utilizadas en el lenguaje de programación de PARADOX (ObjecPAL) son: el icono de ejecutar la acción, representado por un rayo; el icono de impresión representado por una impresora y el icono de verificación de sintaxis del método, representado por una hoja palomeada, además de las tradicionales para cortar, copiar y pegar.

```

method mouseClick(var eventInfo MouseEvent)
var
    des,des1,x          UIObject
    s,s1,s2,s3,s4,y,u  String
    r,c,m              SmallInt
endvar

if eventInfo.isPreFilter()
then
    ; este código se ejecuta para cada objeto de la ficha
    eventInfo.getObjectHit(des)
    if des.class="Button" then
        s=Des.name
        if s.size()<=5 then
            r=smallInt(s.substr(3,s.size()-3))
            c=smallInt(s.substr(s.size(),1))
            m=0
            message(string(r)+" "+string(c))
            switch
                case s.substr(1,2)="ba":
                    m=1
                case s.substr(1,2)="bp":
                    m=2

```

Fig. 4.28. Código que se utiliza para el Megareporte.

Este reporte fue diseñado para generar datos sin una estructura definida, es decir que genera la información que en algún momento se pueda requerir y que no cuenta con un módulo específico para obtenerlos. La pantalla de captura se muestra inmediatamente después de que se oprime el botón con las letras MR dentro de la pantalla principal de Impresión de Reportes. En la figura 4.29. se presenta la pantalla del submódulo de Megareporte.

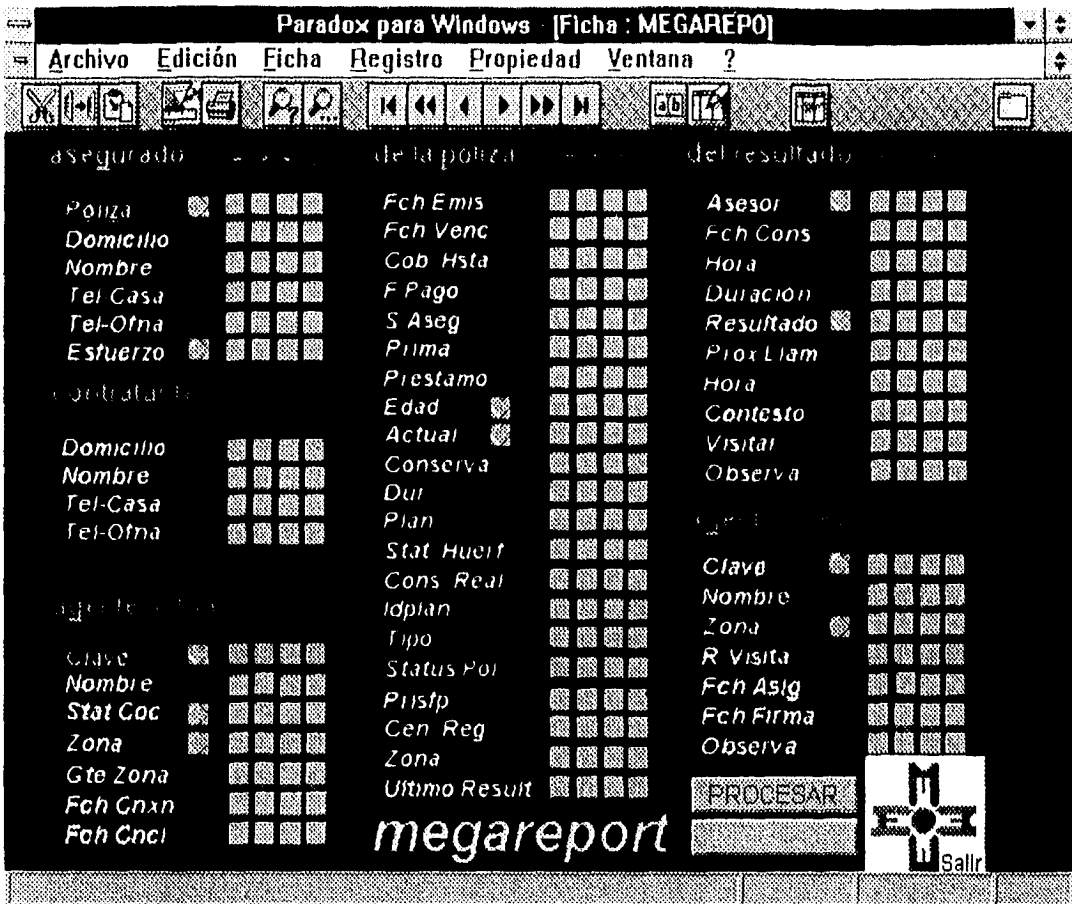


Fig. 4.29. Pantalla del Megareport.

Es importante mencionar que se tienen que seleccionar los campos que se soliciten. Si una casilla se deja en blanco el sistema no traerá esa información. Por ejemplo, si queremos saber que póliza atendió el usuario huerfem1 en una semana, tendremos que seleccionar el campo de póliza dentro de la casilla que aparece con el signo igual y no darle un número de póliza específico, así nos traerá todas. Después seleccionamos el campo "asesor" e introducimos la clave huerfem1 y únicamente nos traerá las pólizas que él atendió. Para darle los rangos de fecha se selecciona la casilla que tenga el signo "/" (entre que fechas) y así con cada uno de los campos que necesitemos. Después se presiona el botón "PROCESAR" y aparecerá un reporte con los campos solicitados. En caso de que se quiera empezar otro reporte bastará con oprimir el botón que dice "INICIALIZAR" y en ese instante se reiniciarán todas las variables usadas y el Megareport estará disponible para realizar otra consulta.

Descripción Técnica

Debido a que este reporte puede generar información de cualquier tipo es importante mencionar que puede llegar a utilizar todas las tablas del sistema y todos los campos de éstas.

4.6 MÓDULO DE MANTENIMIENTO.

Este módulo realmente no contiene alguna forma para captura o lectura de información. Los cambios que se quieran realizar en este módulo se modifican directamente en las tablas que están interrelacionadas con este módulo. Como ya se ha explicado anteriormente, el Módulo de Mantenimiento se utiliza para modificar los datos referentes a los usuarios y las tipologías de las llamadas.

Antes de poder realizar algún cambio, el sistema checa la clave de la persona que intenta entrar en este módulo para verificar si tiene acceso. De esta manera se mantiene cierta seguridad sobre este módulo, ver figura 4.30. Una vez que se ha confirmado que la clave del usuario es correcta, el sistema permitirá el acceso a la pantalla de mantenimiento, ver figura 4.31.

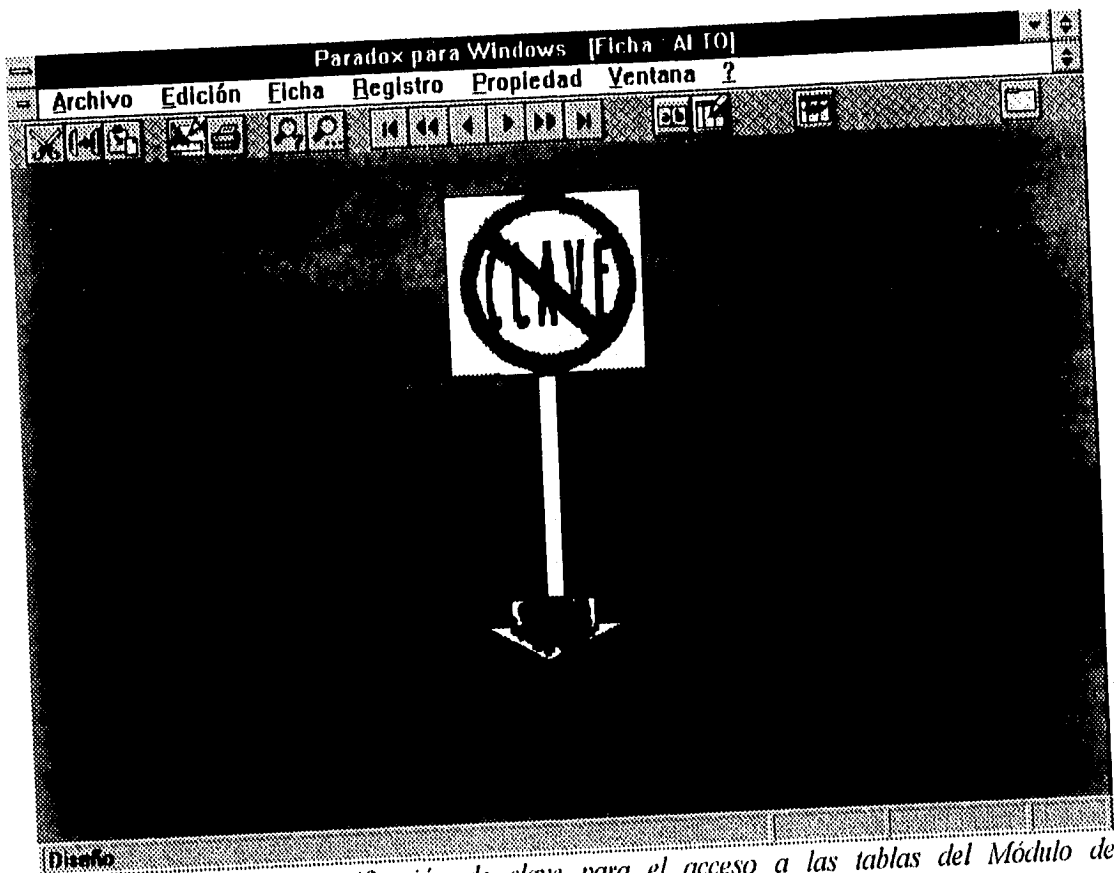


Fig. 4.30. Pantalla de verificación de clave para el acceso a las tablas del Módulo de Mantenimiento.

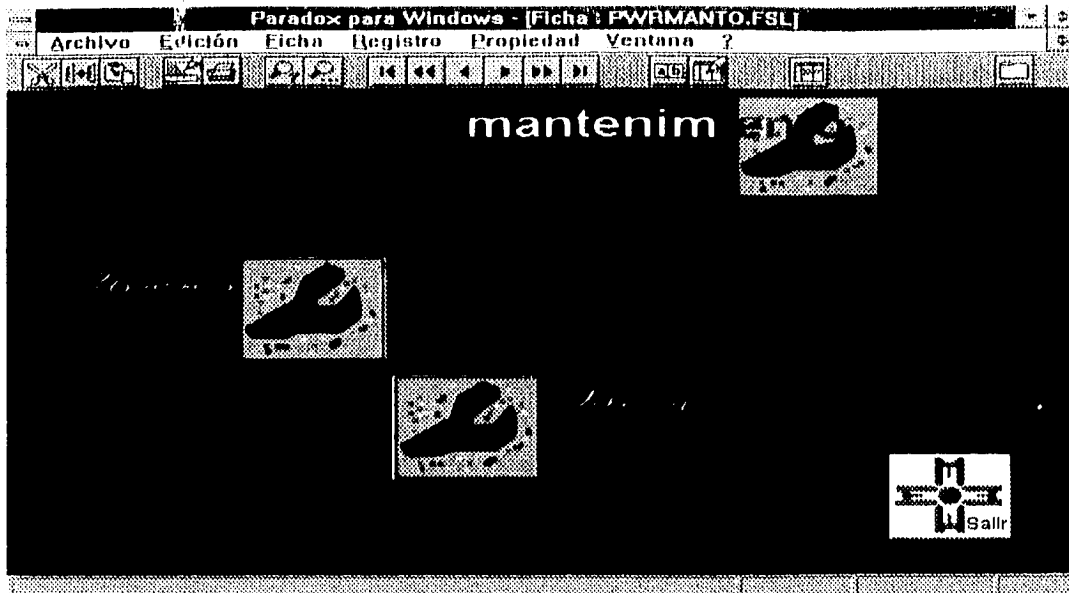


Fig. 4.31. Pantalla principal del Módulo de Mantenimiento.

El Módulo de Mantenimiento muestra las dos opciones que existen para modificar los nombres de los usuarios o las tipologías. Se puede observar que existe ayuda en línea que nos indica que hace cada una de las opciones con sólo seleccionar con el puntero del mouse.

En la figura 4.32, se muestra la pantalla que nos permite modificar las diferentes tipologías que existen. Todos los cambios que se hagan aquí, automáticamente se aplican a los otros módulos donde son usadas estas tablas.



Fig. 4.32. Pantalla donde se pueden realizar las modificaciones a las tipologías.

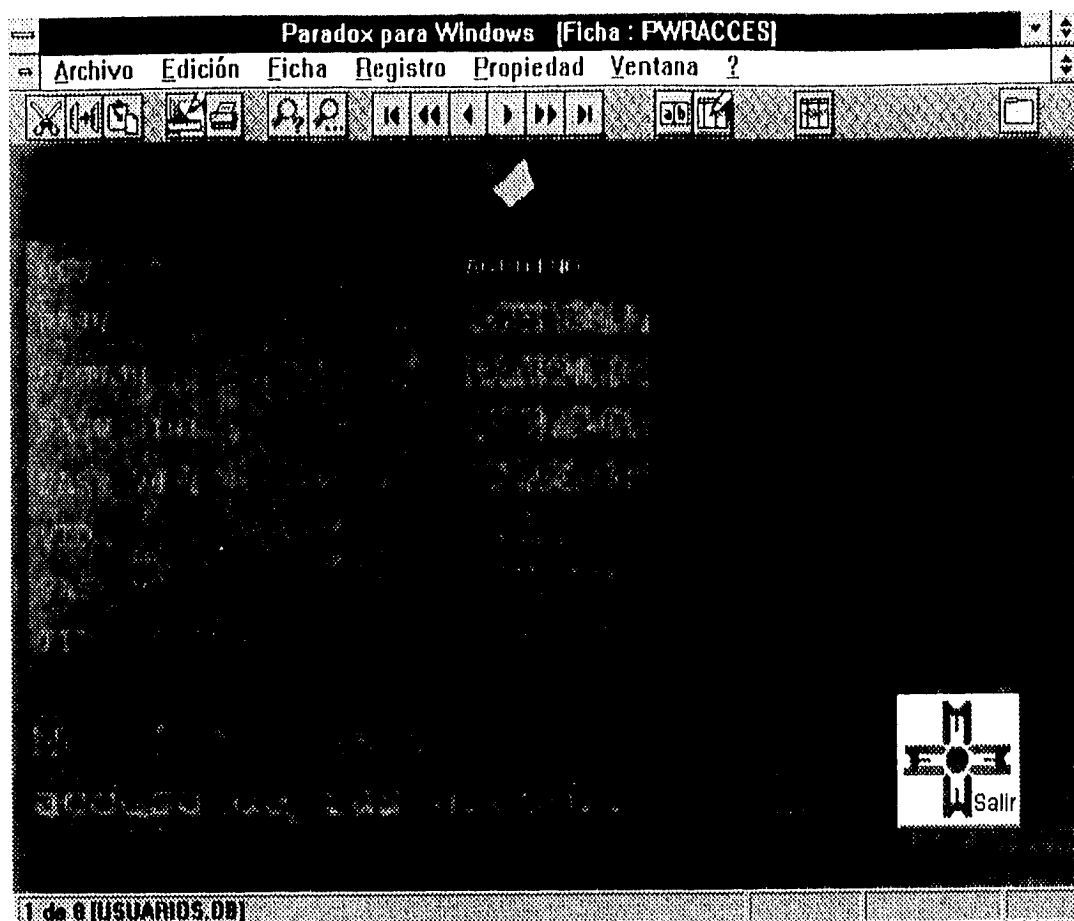


Fig. 4.33. Pantalla que muestra la tabla de usuarios .

En la figura 4.33. se muestra una pantalla donde se pueden modificar las claves de acceso a los sistemas. También se puede incrementar el número de usuarios. Todo dependerá de las necesidades de la gerencia.

Descripción Técnica

Debido a que las modificaciones se hacen directamente en las tablas, es necesario utilizar la tabla de USUARIOS y la tabla de TIPOLOGÍAS.

4.7 MÓDULO DE DATOS ADICIONALES

Este módulo fue diseñado para capturar los nombres, teléfonos y direcciones de las referencias de los asegurados. Esta información no se encuentra en ninguno de los sistemas que utiliza POHUEER para su funcionamiento. Por esta razón se implementó este módulo, ver figura 4.34.

Fig. 4.34. Pantalla que muestra el módulo de captura de Datos Adicionales.

Los datos de las referencias son necesarios para cuando no es posible contactar al asegurado y el único contacto indirecto que se puede tener es a través de alguna de sus referencias

Los datos que se capturan en esta forma son los siguientes:

- PÓLIZA: Número de póliza.
- NOMBRE: Nombre del asegurado.
- DOMICILIO: Domicilio del asegurado.
- TEL_CASA: teléfono de la casa del asegurado.
- C_NOMBRE: Nombre del contratante.
- C_DOM: Domicilio del contratante.
- REF1: Nombre de la referencia 1.
- TEL_R1: Teléfono de la referencia 1.
- REF2: Nombre de la referencia 2.

- *TEL_R2: Teléfono de la referencia 2.*
- *REF3: Nombre de la referencia 3.*
- *TEL_R3: Teléfono de la referencia 3.*

Para obtener los datos requeridos en esta pantalla, esta sección cuenta con un botón adicional "Buscar Póliza", que se activa después de introducir el número de la póliza que se quiere consultar.

Descripción Técnica

Las tablas que utiliza para la generación de esta forma son HUERF y REFER. Los campos que se requieren son:

- ⇒ *PÓLIZA: Número de póliza-ALP(10).*
- ⇒ *NOMBRE: Nombre del asegurado-ALP(40).*
- ⇒ *DOMICILIO: Domicilio del asegurado-ALP(136).*
- ⇒ *TEL_CASA: Teléfono de la casa del asegurado-ALP(20).*
- ⇒ *C_NOMBRE: Nombre del contratante-ALP(40).*
- ⇒ *C_DOM: Domicilio del contratante-ALP(136).*
- ⇒ *REF1: Nombre de la referencia 1-ALP(10).*
- ⇒ *TEL_R1: Teléfono de la referencia 1-ALP(50).*
- ⇒ *REF2: Nombre de la referencia 2-ALP(10).*
- ⇒ *TEL_R2: Teléfono de la referencia 2-ALP(50).*
- ⇒ *REF3: Nombre de la referencia 3-ALP(10).*
- ⇒ *TEL_R3: Teléfono de la referencia 3.-ALP(50).*

En el presente capítulo se hizo una descripción detallada de la operación del sistema POHUER, se especificaron los campos que requiere cada módulo y se mostró parte del código utilizado para la generación de pantallas.

La etapa de desarrollo se basó fielmente en la etapa de diseño. Como se pudo observar, existieron algunas modificaciones con respecto a los prototipos creados durante el diseño, sin embargo se utilizaron como base de las pantallas que finalmente se implementaron.

El haber desarrollado el sistema en PARADOX, fue relativamente sencillo, debido a que ofrece herramientas de reportes, fichas y tablas que se crean fácil y rápidamente. Además se logró aprovechar las tablas generadas por el sistema COCOA.

Como ya se explicó, PARADOX es un lenguaje de cuarta generación en el cual se programa a base de objetos, en donde cada objeto es independiente.

Los módulos que se describieron, están constituidos por un conjunto de objetos interrelacionados.

En el Módulo de Reportes, y en específico en el Megareport, la programación fue más compleja, debido a que se tuvo que establecer la liga entre varias tablas, además de que este reporte exigía establecer varias opciones como son: "=", ">", "<", y "/".

Otro punto importante es el hecho de que se desarrolló el sistema POHUER de manera que fuera amigable al usuario, facilitándole ayuda en línea. Sin embargo, para poder manejar completamente el sistema será necesario de una capacitación básica que se describirá en el siguiente capítulo.

CAPÍTULO V

PRUEBAS E IMPLANTACIÓN

La etapa siguiente al desarrollo de un sistema es la de pruebas. En el presente capítulo se explica que es una prueba, cual es su proceso y los tipos de pruebas más utilizados, además de recalcar la importancia que tiene dentro del ciclo de vida de un sistema. Posteriormente se describen las pruebas a las que se sometió el sistema POHUER, antes de proseguir con la etapa de implantación. Dentro de la etapa de implantación se mencionan los procesos que se llevan a cabo para la instalación del sistema POHUER, así como las actividades que se realizan paralelamente dentro de esta etapa, entrega de manuales y capacitación. Por último se describen los tipos de mantenimiento que se pueden dar al sistema y en que consisten.

5.1 CONCEPTOS BÁSICOS

PRUEBAS

La prueba de los programas es la parte del proceso de confirmación que suele realizarse durante la aplicación y también, en forma distinta, cuando ésta ha terminado. La prueba consiste en ejercitar el programa utilizando datos similares a los datos reales que habrán de ser ejecutados por el programa, observar los resultados y deducir la existencia de errores o insuficiencias del programa a partir de anomalías de ese resultado.

En ocasiones, se piensa que la prueba y la depuración de programas son una misma cosa. Aunque están muy relacionadas, en realidad son procesos distintos. La prueba es el proceso de establecer la existencia de errores en el programa. Depuración es el proceso de localizar dónde se produjeron esos errores y corregir el código.

Es muy importante comprender que la prueba nunca demuestra que un programa es correcto. Siempre es posible que existan errores aún después de la prueba más completa. La prueba de programas sólo puede demostrar la presencia de errores en un programa, no su ausencia, por lo tanto, se considera prueba acertada aquella que establece la presencia de uno o más errores en el software objeto de prueba.

Las pruebas de sistemas son sumamente importantes dentro del ciclo de vida de un sistema, debido a que de éstas depende que se puedan corregir los posibles errores antes de la implantación del sistema y que se evite incurrir en costos de mantenimiento, cuando lleguen a surgir errores no descubiertos, en la vida útil del sistema.

EL PROCESO DE PRUEBA

El proceso de prueba, al igual que el de programación, debe avanzar en etapas, siendo cada una de ellas la continuación lógica de la etapa anterior.

En el proceso de prueba se pueden identificar cinco etapas:

- **Prueba de funciones:** *La prueba de funciones o de unidades es el nivel básico en donde se prueban las funciones que componen un módulo para garantizar que operan de manera correcta.*
- **Prueba de módulos:** *Un módulo se compone de varias funciones que pueden cooperar entre sí. Después de haber probado cada función individual, es necesario probar la cooperación de estas funciones cuando componen un módulo. Debe ser posible probar un módulo como una entidad aislada, sin la presencia de otros módulos del sistema.*
- **Prueba de subsistemas:** *Esta prueba es el siguiente paso del proceso en el cual los módulos se agrupan para formar subsistemas. Puesto que los módulos cooperan y se comunican, la prueba de subsistemas se debe centrar en la prueba de **interfaces** de aquéllos, dando por supuesto que los módulos son correctos.*
- **Prueba del sistema:** *La prueba del sistema (a veces llamada prueba de integración) se lleva a cabo cuando se integran los subsistemas para conformar el sistema completo. En esta etapa, el proceso de prueba tiene que ver con el hallazgo de errores en el diseño y la codificación.*

Prueba de aceptación: Hasta esta etapa, todas las pruebas se realizan mediante el empleo de datos generados por la organización encargada de construir el sistema. La prueba de aceptación del sistema se efectúa con datos reales: la información con la que el sistema deberá operar. El proceso de la prueba de aceptación a menudo descubre errores en la definición de requisitos del sistema.

TIPOS DE PRUEBA

Existen distintas estrategias de prueba, las dos más comunes se conocen como prueba ascendente y prueba descendente. El enfoque ascendente empieza por probar módulos individuales y separadamente; esto a menudo se conoce como prueba de unidades, prueba de módulos, o prueba de programas. Luego, los módulos individuales se combinan para formar unidades cada vez más grandes que se probarán en masa; esto se conoce como prueba de subsistemas. Finalmente, todos los componentes del sistema se combinan para probarse; esto se conoce como prueba del sistema, y suele estar seguido de las pruebas de aceptación, donde se permite al usuario usar sus propios casos de prueba para verificar que el sistema esté trabajando de manera correcta.

El enfoque de prueba descendente empieza con un esqueleto del sistema, es decir, la estrategia de prueba supone que se han desarrollado los módulos ejecutivos de alto nivel del sistema, pero que los de bajo nivel existen sólo como módulos vacíos (módulos que no procesan nada, sino que simplemente terminan luego de ser llamados). Dado que muchas de las funciones detalladas del sistema no se han implantado, las pruebas iniciales están muy limitadas; el propósito es simplemente comenzar a ejercitar las **interfaces** entre los subsistemas principales.

IMPLANTACIÓN

Después de haber desarrollado el sistema y haberle efectuado las pruebas pertinentes en forma exhaustiva, continúa la fase de implantación. En esta fase una vez que se ha verificado que el sistema está libre de errores, se procede a la instalación del mismo dentro del ambiente real en donde operará desde ese momento en adelante. Junto con la instalación se realizan otras actividades paralelas, de las cuales las más importantes son: la creación de archivos para el sistema, entrega de manual de usuario y capacitación del personal.

5.2 PRUEBAS DEL SISTEMA POHUER

Para tener la certeza de que el sistema POHUER opera de acuerdo a las especificaciones planteadas con anterioridad y detectar posibles errores o anomalías en los procesos que realiza, se aplicaron diferentes tipos de pruebas. El realizar estas pruebas tiene como objetivo principal liberar un sistema de alta calidad, que garantice el funcionamiento óptimo de todos los procesos que en él se realicen, además de brindar un alto grado de confiabilidad y un porcentaje nulo de error.

La estrategia que se utilizó para probar el sistema POHUER fue la prueba ascendente, por lo que se procedió primeramente a verificar el funcionamiento correcto de los módulos independientes entre sí de cada subsistema (Prueba de unidad), para localizar los errores posibles; posteriormente se probó la integración de cada módulo en el subsistema (Prueba de integración), para ver la compatibilidad de los módulos. El siguiente paso fue probar todos los componentes que integran el sistema (Prueba de sistema), para verificar la interacción de los subsistemas.

En la figura 5.1. se observan algunas de las principales operaciones de prueba que se aplican durante el desarrollo de un sistema.

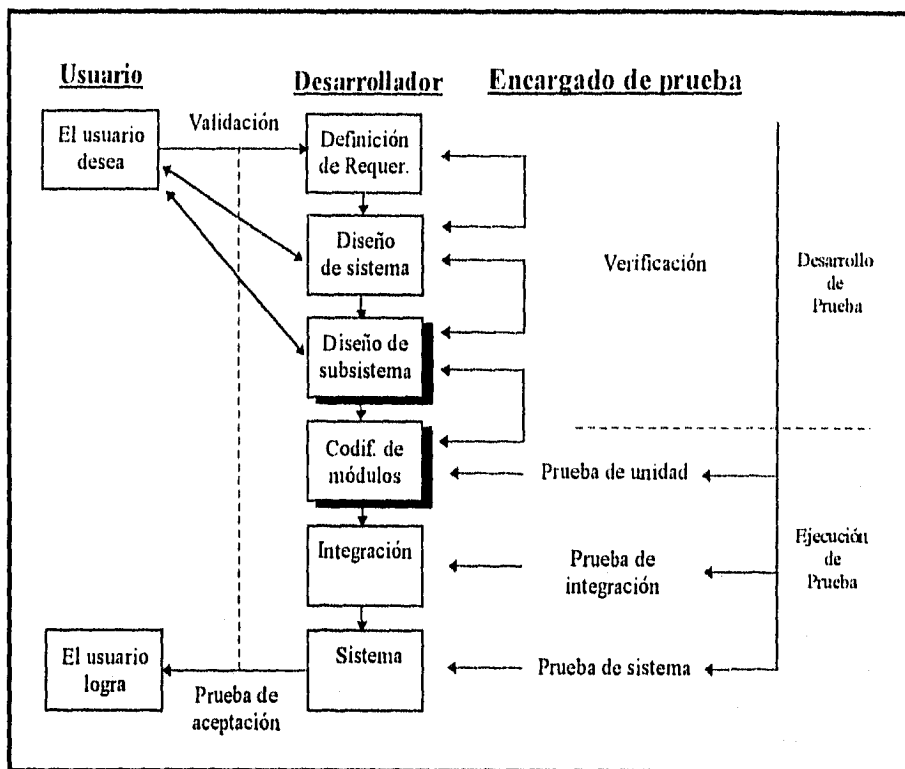


Fig. 5.1. Algunas de las principales operaciones de prueba durante el desarrollo.

Posteriormente se procedió a la instalación del sistema, para lo cual fue necesario preparar el equipo como se describe más adelante.

El proceso de pruebas es de gran utilidad, porque permite conocer las condiciones que pueden ocurrir y que no se tienen contempladas dentro del sistema, por lo que permite puntualizar las omisiones que lleguen a existir.

Una vez habiendo aplicado las pruebas pertinentes al sistema POHUER y después de depurar el sistema al máximo, se inició la etapa de implantación.

5.3 IMPLANTACIÓN DEL SISTEMA POHUER

INSTALACIÓN

La Gerencia de Cartera Huérfana con anterioridad dispuso del equipo en el cuál se instalaría el sistema POHUER, según los requerimientos que desde el inicio del desarrollo del sistema se habían contemplado. Dicho equipo está conformado básicamente por 5 computadoras PC's con emulación de terminal, que están conectadas en red.

Antes de instalar el sistema fue necesario que se instalara el software RUMBA, que permite establecer la comunicación entre aplicaciones del host y la PC, en la unidad "C" de cada máquina. Posteriormente se configuraron los archivos *autoexec.bat*, *config.sys*, *shell.cfg* y *net.cfg*, con el fin de que permitieran operar adecuadamente a RUMBA y al sistema POHUER. El siguiente paso fue realizar propiamente la instalación del sistema POHUER, en un subdirectorio del servidor de la red, con el fin de que los usuarios u operadores de la Gerencia de Cartera Huérfana pudieran tener acceso al sistema. Cabe mencionar que el sistema estuvo sometido durante algún tiempo a observación y estuvo sujeto a modificaciones, de acuerdo a su desempeño y aceptación por parte del usuario.

Una vez puesto en marcha el sistema se procedió a la capacitación de los usuarios y se les hizo entrega del manual del sistema POHUER.

La etapa precedente a la instalación consistió básicamente en probar la aceptación del sistema por parte del usuario. Durante esta etapa de pruebas los usuarios desempeñaron un papel muy importante, debido a que ellos colaboraron en el desarrollo de datos de prueba para la verificación del sistema. Los usuarios no generaron transacciones normales únicamente, sino que también generaron datos con errores. Lo que permitió verificar todas las combinaciones de formatos y valores que fueron posibles. Por describir un ejemplo, se trató de violar las reglas de validación al introducir datos erróneos en las formas de captura, es decir escribir caracteres alfanuméricos cuando se deben introducir caracteres numéricos o viceversa. Estas pruebas garantizaron que el sistema está apto para detectar errores y señalarlos oportunamente al usuario.

Se le aplicaron al sistema otras pruebas, como la prueba de carga máxima, activando varias máquinas al mismo tiempo y operando el sistema con los procesos más complejos; para observar como actúa el sistema ante esta situación.

De la misma manera se aplicó una prueba de tiempo de ejecución, con el fin de determinar el tiempo de máquina que el sistema necesita para procesar los datos de cada transacción.

Una vez realizada la etapa de confirmación del sistema, habiendo dado por hecho que el sistema POHUER cumple con los requisitos planteados desde el diseño y satisfaciendo las necesidades de la Gerencia de Cartera Huérfana, se procedió a la liberación del sistema. Es importante mencionar que para lograr lo anterior se tuvieron que realizar ciertas modificaciones al sistema.

MANUAL DE USUARIO

El manual de usuario es de suma importancia, ya que con éste el usuario puede explotar el sistema al máximo. Además de ser un apoyo para resolver cualquier problema que surja por el desconocimiento de algún procedimiento.

La información contenida en el manual de usuario debe proporcionar una visión clara y precisa del sistema. Debe ser una guía que lleve de la mano al usuario a través de los diferentes procesos que realiza el sistema.

El manual de usuario, además de contemplar una descripción detallada sobre lo que puede realizar el sistema, debe incluir una explicación de cómo instalarlo y adecuarlo a configuraciones específicas de hardware.

Para la realización del manual de usuario se utilizó la documentación que se generó durante las fases del ciclo de vida del sistema, pero no se incluye, para no incrementar el volumen de información de esta tesis.

Después de la elaboración de este manual, se entregó a los usuarios y se prepararon los cursos de capacitación.

CAPACITACIÓN

Como se expuso anteriormente, el sistema POHUER corre bajo WINDOWS, está desarrollado en PARADOX y cuenta con ayuda en línea, lo que hace que sea amigable al usuario. Cualquier persona que tenga conocimientos mínimos de alguna aplicación en WINDOWS, podrá hacer uso del sistema POHUER, debido a que las aplicaciones de este tipo siempre tienen el mismo formato de operación.

Fue necesario familiarizar a los usuarios con PARADOX mediante un curso con duración de 20 hrs. y capacitarlos en los diferentes procesos que se requieren para operar el sistema. Dichos procesos son: Procesos de consulta, captura, revisión, validación, impresión, elaboración de telegramas, reportes y estadísticas.

Surgieron muchas preguntas durante la etapa de capacitación, dando oportunidad al usuario de discutir sus dudas y proponer sugerencias libremente.

Es importante mencionar que con el fin de dar una asesoría constante que sirva de apoyo al usuario en el manejo del sistema POHUER, se optó por proporcionarles formas en las que deben escribir los errores más frecuentes que hayan tenido durante el día, para resolverles sus dudas y tener un control de los errores que más se repiten, para buscar una solución adecuada.

MANTENIMIENTO DEL SISTEMA POHUER

Como se ha mencionado con anterioridad, el Mantenimiento es el proceso de modificar un programa cuando ya se ha entregado y está en uso. Esas modificaciones pueden implicar cambios sencillos para corregir errores de codificación, cambios mayores para corregir errores de diseño o reestructuras drásticas para corregir errores de especificación o introducir nuevos requisitos. Los tipos de mantenimiento pueden clasificarse en cuatro categorías: *Mantenimiento correctivo* (mantenimiento de reparación), *mantenimiento adaptativo* (para hacer frente a un nuevo sistema operativo, etc.), *mantenimiento de mejora o perfectivo* (mantenimiento para optimizar el desempeño, etcétera), y *mantenimiento preventivo* (para facilitar el mantenimiento mismo).

El mantenimiento correctivo se da debido a que no necesariamente las pruebas realizadas al sistema permitirán identificar los errores o anomalías. Una vez instalado el sistema POHUER, se detectaron ciertas anomalías del sistema que se pudieron corregir satisfactoriamente. Como ejemplo se puede mencionar que los reportes estadísticos se generaban en el servidor de la red y cuando más de un usuario requería del mismo reporte, modificaba los datos de otro usuario, por lo que se decidió que los reportes estadísticos se generaran dentro del disco "C" de cada terminal.

El mantenimiento adaptativo se da debido al cambio rápido inherente de la informática. Diariamente se anuncian nuevos cambios en cuanto al uso del hardware, software e infraestructura de comunicaciones, así como la aparición de nuevos sistemas operativos, software de red, y frecuentemente la mejora o modificación de los equipos periféricos. Concientes de las innovaciones de la tecnología se buscará estar a la vanguardia en cuanto a los cambios que ésta ofrezca, sobre todo a los que permitan que el sistema POHUER pueda mejorar su desempeño.

El mantenimiento de mejora o perfectivo se realizará durante la operación del sistema, mediante recomendaciones de los usuarios, sobre necesidades de modificación de los módulos del sistema y sobre mejoras generales del sistema. Se intentará que el sistema sea del agrado del usuario con el fin de que éste aproveche el sistema al máximo.

El mantenimiento preventivo se realizará con el fin de facilitar el mantenimiento mismo. Para evitar que surjan dificultades al operar el sistema POHUER, regularmente se verificará que exista espacio suficiente en el servidor que alberga las bases de datos. Así mismo, se verificará también el espacio existente en el disco "C" de cada máquina de los usuarios, para que puedan generarse de manera correcta los reportes estadísticos.

Los costos para llevar a cabo el mantenimiento pueden variar de acuerdo a la forma en que fueron realizadas, las etapas anteriores del ciclo de vida de un sistema, y otros factores técnicos (interdependencia de los módulos, lenguaje de programación, estilo de programación, prueba del programa, etcétera) y no técnicos (estabilidad del personal, dependencia del programa de su ambiente externo, estabilidad del hardware, etcétera).

La metodología utilizada para el desarrollo del sistema POIHUER, ha permitido que el mantenimiento realizado al sistema se haya logrado con relativa facilidad y nos permite afirmar que desde este momento se cuenta con los elementos necesarios para darle cualquiera de los mantenimientos anteriormente descritos.

En resumen, se puede decir que las pruebas constituyen una de las etapas más importantes del ciclo de vida de un sistema, porque depende de éstas, la confiabilidad y calidad del sistema.

Las actividades que se realizaron para la implantación del sistema básicamente se pueden resumir en instalación, entrega de manuales y capacitación del usuario. Estas actividades se realizaron con éxito, porque el sistema está operando en forma correcta y los usuarios se sienten plenamente identificados con éste.

Es importante mencionar que la facilidad para dar mantenimiento a un sistema es proporcional a la calidad de la realización de las etapas anteriores, además de que el tiempo y el esfuerzo que se invierta en la etapa de pruebas evitará que surjan errores o deficiencias en el sistema, cuando se encuentre en operación, minimizando los costos de mantenimiento.

RESULTADOS Y CONCLUSIONES

Con base en el análisis realizado a la problemática de la Cartera Huérfana de GNP, una vez identificados los procedimientos que debían de realizarse, tomando en cuenta los recursos y estándares de la Cía y aunado a nuestra experiencia en el desarrollo de sistemas, se obtuvo como resultado un sistema que permite el control y la reasignación de agentes a la Cartera Huérfana. Dicho sistema ha sido puesto en operación en Plaza GNP, dentro del área asignada a la Gerencia de Cartera Huérfana.

El sistema como se esperaba, ha tenido un alto grado de aceptación por parte del usuario, debido al ambiente gráfico en que está desarrollado, que resulta ser amigable y de fácil aprendizaje, además de proporcionar ayuda en línea que lo guía en cada uno de los procesos que realiza.

El sistema es una herramienta eficaz de consulta, además de que permite la emisión automática de reportes, estadísticas y listados, que auxilian a la Gerencia de Cartera Huérfana en la determinación de políticas que coadyuvan a la toma acertada de decisiones.

Gracias al uso de este sistema, la Gerencia de Cartera Huérfana tiene contemplado, recuperar un porcentaje importante de las pólizas que han quedado sin la representación de un agente, que se verá reflejado en la obtención de una suma monetaria considerable.

Consideramos que dentro de las mejoras a futuro del sistema POHUER se encuentra la integración de la Cartera Huérfana del Ramo de Daños. El conjuntar dentro de un mismo sistema las pólizas del Ramo de Vida y Daños nos permitirá agrupar las pólizas de un solo cliente dentro un grupo de pantallas interactivas.

También pensamos que podría ser sustancial la instalación en cada una de las PC's de los operadores, de un módem que permita el marcaje automático, mediante ratón o una función del teclado y envío de documentos relacionados con la operación, vía fax.

La evolución vertiginosa de la tecnología, origina un cambio continuo en herramientas de software. Por lo que más adelante se podría utilizar un manejador de bases de datos como SQL for Windows de Gupta para agilizar los procesos.

En forma personal, podemos concluir que el haber desarrollado el sistema mediante el ciclo de vida descrito, fue de gran utilidad para nuestro desarrollo profesional.

BIBLIOGRAFÍA

MARK L. GILLENSON, *INTRODUCCIÓN A LAS BASES DE DATOS*, McGRAW HILL, MÉXICO 1987.

JOHN WILLWEY & SONS, *LOCAL AREA NETWORKS (LAN)*, WILEY, CANADA 1987.

JUAN MANUEL MÁRQUEZ VITE, *SISTEMAS DE INFORMACIÓN POR COMPUTADORA*, TRILLAS, MÉXICO 1987.

ROGER S. PRESSMAN, *INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO (2a. EDICIÓN)*, McGRAW HILL, MÉXICO 1994.

KENDALL Y KENDALL, *ANÁLISIS DISEÑO DE SISTEMAS*, PRINTICE HALL, MÉXICO 1991.

IAN SOMMERVILLE, *INGENIERÍA DE SOFTWARE*, ADDISON-WESLEY IBEROAMERICANA, MÉXICO 1988.

JHON P. VAN GIGCH, *TEORÍA GENERAL DE SISTEMAS (2a. EDICIÓN)*, TRILLAS, MÉXICO 1987.

RICHARD E. FAIRLEY, *INGENIERÍA DE SOFTWARE*, McGRAW HILL, MÉXICO 1988.

NORRIS - RIGBY, *INGENIERÍA DE SOFTWARE EXPLICADA*, MEGABYTE, MÉXICO 1994.

LUDWIN VON BERTALAN, *TEORÍA GENERAL DE SISTEMAS (1a. EDICIÓN)*, FONDO DE CULTURA ECONÓMICA, COLECCIÓN CIENCIA Y TECNOLOGÍA, MÉXICO 1976.

WILLIAM A. BOCCHINO, *SISTEMAS DE INFORMACIÓN PARA LA ADMINISTRACIÓN (2a. EDICIÓN)*, TRILLAS, MÉXICO 1991.

HENRY C. LUCAS JR., *CONCEPTOS DE LOS SISTEMAS DE INFORMACIÓN PARA LA ADMINISTRACIÓN (1a. EDICIÓN)*, McGRAW HILL, MÉXICO 1983.

A. ZIYA AKTAS, *STRUCTURED ANALYSIS AND DESING INFORMATION SYSTEMS*, PRENTICE HALL, MÉXICO 1987.

SILVER SCHÄTZ, FUNDAMENTOS DE LAS BASES DE DATOS, MCGRAW HILL, MÉXICO 1987.

EDWARD YOURDON, ANÁLISIS ESTRUCTURADO MODERNO, PRENTICE HALL, MÉXICO 1989.

JAMES MARTIN, JAMES J. ODELL, ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS, PRENTICE HALL, MÉXICO 1992.

ÍNDICE DE MATERIAS

A

- Acoplamiento, 63, 64
- Agente, 17
 - cancelado, 2, 43, 46
 - datos generales, 41
 - dispositor, 43, 46
 - identificación de, 52
 - tabla de status, 52
- Almacenamiento de datos, 56
- Análisis, 20
 - planeación de las actividades, 22
 - propuesta de solución, 21
 - requerimientos y alcances, 22
 - situación actual, 21
- Archivos, 56
- Aseguradora(s), 2
- Asegurado, 16
 - contacto con, 53
- Asegurador, 16

B

- Base de datos, 56
 - alternativas de selección, 48
 - centralizada, 50
 - comparativo, 49

C

- Capacitación de POHUER, 25, 133
- Captura de datos adicionales, 42
- Cartera Huérfana, 2, 19, 26, 31
 - Gerencia de, 19, 54
 - Problemática, 39, 40
- Centros regionales de GNP, 12
- Chris Gane, 2, 55
- Ciclo de Vida, 19, 20
- COCOA, 38, 96
- Codificación, 89
- Cohesividad, 63, 64
- Consolidado de Operación, 43, 44

Consulta y reasignación, 40
 Control de llamadas, 41
 resultados, 41
 Datos
 adicionales, 42
 principales, 42
 dBASE, 48, 49
 Desarrollo, 2, 24
 Diagrama de Estructura de datos, 23, 55

D

Datos generales de agente, 41
 Datos principales, 42
 DED, (ver diagrama de Entidad-Relación)
 Desarrollo, 24
 Diagrama de flujo de datos, 23, 55, 65, 66
 de POHUER, 69-73
 Diagrama Entidad Relación, 24, 55, 65, 68
 de POHUER, 76
 Diccionario de datos, 65, 67
 de POHUER, 74-76, 88
 notación de, 68
 Diseño, 2, 23, 55
 de software, 61
 estructurado, 62, 63
 orientado al flujo de datos, 65
 técnicas de, 65
 División accidentes y enfermedades, 13
 División automóviles, 13
 División daños, 14
 División pensiones, 13
 División vida, 13

E

Eficiencia, 91
 Enfoque por procedimientos, 93
 Enfoque por objetos, 93
 Esquema
 de datos, 61
 centralizados, 50
 ventajas, 50
 desventajas, 50
 Estadísticas del Universo de Datos, 43, 46

F

Flujo de transformación, 66
 Flujo de transacción, 66

G

Gerencia de la Cartera Huérfana, 27, 28
 GNP, 2, 19
 actividades de, 12
 antecedentes de, 7-10
 centros regionales, 12
 emblema de, 14
 estructura de servicio, 12
 fuerza productora, 11
 gerencia de zona, 53
 instalaciones de cómputo, 11, 14
 oficinas centrales, 15
 organigrama general, 15
 plaza, 14, 15
 ramos de seguros de, 12-14
 recursos, 11
 Grupo Bal, 8

H

Histórico telegramas, 43, 44

I

IDMSDC, 29, 38
 Implantación, 2, 25, 130
 de POHUER, 132
 Impresión
 reportes, 43
 Independencia de datos, 61
 Información
 abstracción, 58
 INFORMIX, 48, 49
 Ingeniería de Software, 2, 20
 Ingeniería, 20
 Instalación de POHUER, 132
 Instancia, 61

L

- Lenguajes de programación, 90
 - primera generación, 90
 - segunda generación, 90
 - tercera generación, 90
 - 4GL, 90, 91

M

- Manual de usuario, 25, 133
- Mantenibilidad, 91
- Mantenimiento, 2, 3, 26, 42
 - adaptativo, 26
 - correctivo, 26
 - perfeccionamiento, 26
 - preventivo, 26
 - usuarios, 43

Megareporte (macroreporte), 43, 45

Metodología, 55, 66

Modelos

- binario, 58
- de abstracción, 58
- de datos, 58
- del desarrollo de software, 20
- entidad relación, 58, 65
- físicos, 61
- infológico, 58
- lógicos, 59
 - relacional, 59
 - red, 59
 - jerárquico, 60
- semántico de datos, 58
- V o cascada, 20

Módulo, 62, 63

- de consulta y reasignación, 93, 97
 - datos de agente, 102
 - generales de póliza, 100
 - pantalla de resultados, 103
 - pantalla principal, 99
- de datos adicionales, 93, 94, 124
- de impresión de reportes, 93, 106
 - agentes cancelados, 118
 - agentes dispositores, 118
 - consolidado, 112

- crea telegramas, 107
- de consolidado de op, 114
- de llamadas y tiempo, 115
- de rendimiento, 114
- estadísticas inv. total, 116
- estadístico general, 116
- formas para captura, 118
- histórico de telegramas, 112
- imprime telegrama, 109
- megareport, 119-121
 - de mantenimiento, 93, 94, 122
 - pantalla principal, 123
 - tipología, 123
 - usuarios, 124
- diagrama de, 47
- pequeños, 92

N

Normalización, 77

de POHUER, 78-83

O

- ORACLE, 48, 49
- ObjectPAL, 92
- Operación de sistema, 93

P

- Pantallas de Vida, 31-38
- PARADOX, 48, 49, 95
- Planeación, 21
- POHUER, 2, 55, 40, 46, 47
 - análisis del sistema, 51
 - capacitación de, 133
 - diagrama de flujo de datos, 69-73
 - diagrama entidad relación, 76
 - diccionario de datos, 74-76, 88
 - entrada al sistema, 95
 - implantación de, 132, 133
 - instalación de, 132
 - manual de usuario de, 133
 - mantenimiento de, 134, 135
 - normalización, 78-83

panorama general, 93
plan de trabajo, 51
planeación de actividades, 50
prototipos, 84-88
pruebas, 130, 131
requerimientos de, 48

Póliza (s), 2, 17
datos generales de, 40, 41
detección de, 52
huérfana, 17
reasignación 53

Portabilidad, 91
Programación estructurada, 91
Programación orientada a objetos, 92
Productividad, 91

Prima, 17
Proceso de reasignación, 51-54

PROGRESS, 48, 49

Prototipos, 2, 84
de POHUER, 84

Pruebas, 2, 25, 128-130
de POHUER, 130, 131
tipos de, 130

R

Ramo
de Daños 2, 12-14
de seguros, 12
de Vida 2, 30, 13

Riesgo, 16
especulativo, 16
puro, 16

RUMBA, 48, 97

S

SAETA, 38, 39

Seguro(s), 2, 16
clasificación de, 17
historia de, 4-7
principios fundamentales, 16-17

Sencillez de estilo, 92

Simplicidad, 62

Simiestro, 16

SISI, 38, 53

Software estructurado, 19

SO-1, 28, 29, 30, 40

SO-2, 28, 29, 42

Subdirección de información de agentes y
control de comisiones, 27

Status, 52

SYBASE, 48, 49

T

Telegrama, 44

Thrish Sarson, 2, 55

Tipología, 42, 43

V

Vida, 29, 30

V2, 30, 52, 96, 97

V4, 96, 97

U

Usuario, 43

W

Woodrow William V, 7

APÉNDICE A

GLOSARIO

Agente Cancelado: Es aquel agente que deja de prestar sus servicios en la aseguradora al tener status de cancelación.

Agente Dispositor: Es el agente cancelado que actúa de mala fe o con dolo, ya que sigue cobrando las primas a los asegurados, pero ya sin brindarles el servicio.

Agente: Fuerza productora de una compañía aseguradora, del cual depende la venta de seguros.

Ajustador: Personal encargado de acudir al lugar en donde ocurrió el siniestro, evalúa las circunstancias y decide si cumple con las coberturas y condiciones del seguro correspondiente.

Asegurado: Persona física o moral que ha contratado un seguro (en el caso del seguro de bienes materiales, el asegurado no necesariamente corresponde al propietario de los bienes o al beneficiario del seguro).

Aseguradora: Empresa que asume o toma riesgos ajenos a cambio del pago de la prima correspondiente.

Beneficiario: Persona que asigna el asegurado en la póliza como titular de los derechos de indemnización establecidos

Cartera de Clientes: Son todas las pólizas que tienen un agente que las represente ante la compañía.

Cartera Huérfana: Son los pólizas que se han quedado sin ningún agente que las represente ante la compañía de seguros

Centro Regional de la Póliza: Es el lugar geográfico al cual pertenece la póliza.

Cobertura: Compromiso aceptado por la aseguradora en virtud del cual se hace cargo hasta el límite estipulado, de las consecuencias económicas derivadas de la realización del riesgo amparado.

Contratante: Es la parte que hace un convenio con la compañía de seguros.

Edad Actual del Asegurado: Es la edad que se le asigna al asegurado por parte de la compañía de seguros.

Edad Real del Asegurado: Es la edad cronológica del asegurado.

Esfuerzo: Es una clasificación arbitraria que se le da a las pólizas en cuanto al monto de su prima, tipo de seguro contratado, o cualquier otro rubro que se desee.

Item: Es un objeto o ente aislado.

GNP: Grupo Nacional Provincial.

Póliza: Documento que instrumenta el contrato del seguro en el que se reflejan las normas generales, particulares, y especiales que regulan las relaciones contractuales convenidas entre asegurado y la aseguradora.

Préstamo de la Póliza: De las aportaciones que ha logrado reunir el asegurado en su prima, la compañía de seguros le puede otorgar un préstamo.

Prima: Cantidad que cobra la compañía aseguradora al asegurado por el riesgo que asume.

Ramo de Daños: Protección financiera para bienes inmuebles, empresas y responsabilidad civil.

Ramo de Vida: Brinda toda la protección a las personas en caso de muerte, accidentes o enfermedades.

Reasignar una Póliza: Otorgar un nuevo agente a una póliza huérfana con el previo consentimiento del asegurado.

Resultado de la Llamada: También llamada Tipología, es el tipo de respuesta que se encontró al tratar de comunicarse telefónicamente con el asegurado.

Riesgo: Una eventualidad que tiene cierta probabilidad de ocurrir y generalmente conlleva una pérdida material o económica.

Riesgo Especulativo: Son aquéllos cuyas consecuencias pueden ser favorables o adversas.

Riesgo Puro: Este tipo de riesgo acarrea siempre consecuencias adversas.

Seguro: Contrato por el que una persona o sociedad (aseguradora) asume un riesgo que debe recaer sobre otra persona (asegurado), a cambio del pago de una cantidad de dinero (prima).

Siniestro: Daño, destrucción o pérdida que sufren las personas o propiedades por distintas causas, y que hacen entrar en acción la garantía de la compañía aseguradora.

Sistema POHUER: Sistema creado para detectar agentes cancelados, contactar con los asegurados y reasignarles un nuevo agente.

Sistemas Satélites: Sistemas auxiliares de cómputo que han surgido conforme a las necesidades de la empresa y se van integrando al sistema principal.

Status de la Póliza: Verifica si la póliza está vigente o no.

Status del Agente: Es una clasificación que se le otorga al agente de acuerdo a su desempeño.

Suma asegurada: Valor que fija el asegurado al bien que quiere proteger y que servirá de límite máximo de responsabilidad para el asegurador.

Subdirección de Información de Agentes y Control de Comisiones: Departamento en donde se realiza el control de los agentes y al cual pertenece la Gerencia de Cartera Huérfana.

Tipología: Es el tipo de resultado que arroja una llamada telefónica a un asegurado para informarle que su agente ha sido cancelado.

Valuación: Es el precio que se le otorga a un bien inmueble.

APÉNDICE B
CÓDIGO DE POHUER

Objeto : #Ficha1

Nombre del Metodo : Var

Source : Var

```

    ts          TCursor
    Inicia      Number
endVar

```

Objeto : #Ficha1

Nombre del Metodo : open

Source : rmethod open(var eventInfo Event)

```

    if eventInfo.isPreFilter()
    then
        ; este código se ejecuta para cada objeto de la ficha
    else
        ; este código se ejecuta sólo para la ficha en sí
        ts.open("STAT_POL.DB")
        Inicia=0
    endif
endmethod

```

Objeto : #Ficha1

Nombre del Metodo : close

Source : method close(var eventInfo Event)

```

    If eventInfo.isPreFilter()
    then
        ; este código se ejecuta para cada objeto de la ficha
    else
        ; este código se ejecuta sólo para la ficha en sí
        ts.close()
    endif
endmethod

```

Objeto : #Ficha1

Nombre del Metodo : action

Source : method action(var eventInfo ActionEvent)

```

    var
        idval      smallInt
    endVar

    if eventInfo.isPreFilter()
    then
        ; este código se ejecuta para cada objeto de la ficha
    else
        ; este código se ejecuta sólo para la ficha en sí

        idval=eventInfo.id()
        if idval=DataUnlockRecord or idVal=DataPostRecord then
            if Inicia=1 then
                beep()
                msgStop("Error","Hay una llamada en espera")
                eventInfo.setErrorCode(UserError)
            endif
        endif
    endif

endmethod

```

Objeto : #Página27.ACEPTAR4 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)

```

        MoveToPage(2)
    endmethod

```

Objeto : #Página27.EDITAR **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)

```

        MoveToPage(3)
    endmethod

```

Objeto : #Página27.ACEPTAR3 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)

```

        MoveToPage(1)
    endmethod

```

Objeto : #Página27.#Cuadro54.P_STATUS **Nombre del Metodo :** changeValue

Source : method changeValue(var eventInfo ValueEvent)

```

    do default
    if ts.locate("ESTADO",P_STATUS) then
        STATUS_NOM=ts."NOMBRE"
    else
        STATUS_NOM=""
    endif
endmethod

```

Objeto : #Página58.#Botón255 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)

```

VAR
FICHA1 FORM
TC,T1 TCURSOR
ENDVAR

IF RESULTADO=11 OR RESULTADO=41 THEN
    IF POLIZA<>BLANK()THEN
        TC.OPEN("TEMPORA1.DB")
        TC.EDIT()
        TC.TEMPORAL=POLIZA
        TC.ENDEDIT()
        TC.CLOSE()
        ficha1.open("recapt.fsl")
    ENDIF
ELSE

```

```

MSGINFO("ERROR","PARA TRASPASAR A COBRANZA TIENE TENER RESULTADO 11 O 41")
ENDIF
endmethod

```

Objeto : #Página58.#Botón86 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)

```

var
s      string
endvar
s="00"
s.view("Nueva Tipología para la póliza")
if s="00" or s="03" or s="08" or s="09" or s="12" or s="13" or s="14" or s="15" or s="40" or s="41" or s="42" or s="50"
then
        FECHA.end()
        FECHA.insertAfterRecord()
        FECHA=TODAY()
        HORA=hour(DATETIME()+Number(minute(DateTime()))/100
        USUARIO=GetNetUserName()
        ACUM=0
        RESULTADO=s
        ULT_RESULT=s
        ULT_USUAR=GetNetUserName()
        ULT_FECHA=today()
        Asegurado2.moveto()
else
        msgstop("Error", "Esta tipología no esta permitida")
endif
endmethod

```

Objeto : #Página58.Registro **Nombre del Metodo :** Var

Source : Var

```

Tiempo DateTime
endVar

```

Objeto : #Página58.Registro **Nombre del Metodo :** open

Source : method open(var eventInfo Event)

```

        Tregistro="Inicia Llamada"
        Tregistro.font.color=blue
        Inicia=0
endmethod

```

Objeto : #Página58.Registro **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)

```

var
t,t1    Number
s      String
tv     TableView
endvar

If Inicia=0 then
        FECHA.end()
        FECHA.insertAfterRecord()
        FECHA=TODAY()
        ULT_USUAR=GetNetUserName()

```

```

ULT_FECHA=today()
HORA=hour(DATETIME()+Number(minute(DateTime()))/100
USUARIO=GetNetUserName()
Tiempo=dateTime()
s=msgQuestion("Resultado de llamada","¿Contesto el Asegurado?")
if s="No" then
    FECHA.end()
    Inicia=0
    DURACION=minute(DATETIME()-Tiempo)+number(second(DATETIME()-Tiempo))/100
    t=NUMBER(ACUM)+DURACION
    t1=t-int(t)
    if t1>.59 then
        t=t+1-.6
    endif
    ACUM=t
    RESULTADO="02"
    ULT_RESULT="02"
    Asegurado2.moveto()
else
    Tregistro="Termina Llamada"
    Tregistro.font.color=Red
    inicia=1
endif
else
    Inicia=0
    FECHA.end()
    DURACION=minute(DATETIME()-Tiempo)+number(second(DATETIME()-Tiempo))/100
    t=NUMBER(ACUM)+DURACION
    t1=t-int(t)
    if t1>.59 then
        t=t+1-.6
    endif
    ACUM=t
    tv.open("RESULTV.DB")
    tv.wait()
    RESULTADO=tv."CVE_RES"
    ULT_RESULT=tv."CVE_RES"
    tv.close()
    maximize()
    Tregistro="Inicia Llamada"
    Tregistro.font.color=blue
    PROX_LLAMA.moveto()
endif
endmethod

```

Objeto : #Página58.#Cuadro257.LLAMADA **Nombre del Metodo :** depart
S.#Registro244.HOR_PROX

Source : method depart(var eventInfo MoveEvent)

Registro.moveto()

endmethod

Objeto : #Página58.#Cuadro257.TELEG **Nombre del Metodo :** changeValue

Source : method changeValue(var eventInfo ValueEvent)

var

t TCursor

endVar

```

DoDefault
if self.value=1 then

    t.open("TELEGRAM.DB")
    t.edit()
    t.insertAfterRecord()
    t."POLIZA"=POLIZA
    t."FECHA"=today()
    t."ASEGURADO"=ASEGURADO2
    t."DIRECCION"=DOMICILIO
    t."USUARIO"=USUARIO
    t.endedit()
    t.close()

endif

endmethod

```

Objeto : #Página58.ACCEPTAR2 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)
 MoveToPage(2)
 endmethod

Objeto : #Página58.ACCEPTAR5 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)
 MoveToPage(4)
 endmethod

Objeto : #Página58.ACCEPTAR1 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)
 MoveToPage(1)
 endmethod

Objeto : PAG2.ACCEPTAR5 **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)
 MoveToPage(4)
 endmethod

Objeto : PAG2.EDITAR **Nombre del Metodo :** pushButton

Source : method pushButton(var eventInfo Event)

MoveToPage(3)

endmethod

Objeto : PAG2.ACEPTAR1

Nombre del Metodo : pushButton

Source : method pushButton(var eventInfo Event)

MoveToPage(1)

endmethod

Objeto : PAG1

Nombre del Metodo : arrive

Source : method arrive(var eventInfo MoveEvent)

maximize()

endmethod

Objeto : PAG1.#Botón140

Nombre del Metodo : pushButton

Source : method pushButton(var eventInfo Event)

close()

endmethod

Objeto : PAG1.ACEPTAR5

Nombre del Metodo : pushButton

Source : method pushButton(var eventInfo Event)

MoveToPage(4)

endmethod

Objeto : PAG1.ACEPTAR2

Nombre del Metodo : pushButton

Source : method pushButton(var eventInfo Event)

MoveToPage(2)

endmethod

Objeto : PAG1.EDITAR

Nombre del Metodo : pushButton

Source : method pushButton(var eventInfo Event)

MoveToPage(3)

endmethod

Objeto : PAG1.INTERFACE

Nombre del Metodo : Uses

Source :
 : Llamadas a DLL EHLLAPI para interface con RUMBA.
 :
 : Uses EEHLLAPI

WD_AdviseHostUpdate(hInstance CWORD, hWndMsg CWORD, Message CWORD) CWORD
 WD_ConnectPS(hInstance CWORD, Name CPTR) CWORD
 WD_Convert(hInstance CWORD, ConvertType CWORD, RowCol CLONG, Name CPTR) CWORD
 WD_CopyFieldToString(hInstance CWORD, Position CWORD, Buffer CPTR, Length CWORD) CWORD
 WD_CopyOIA(hInstance CWORD, Buffer CPTR, Length CWORD) CWORD
 WD_CopyPS(hInstance CWORD, Buffer CPTR, Length CWORD) CWORD
 WD_CopyPSToString(hInstance CWORD, Position CWORD, Buffer CPTR, Length CWORD) CWORD
 WD_CopyStringToField(hInstance CWORD, Position CWORD, Buffer CPTR) CWORD
 WD_CopyStringToPS(hInstance CWORD, Position CWORD, Buffer CPTR, Length CWORD) CWORD
 WD_DeletePS(hInstance CWORD, Name CPTR) CWORD
 WD_DisconnectPS(hInstance CWORD) CWORD
 WD_DisplayCursor(hInstance CWORD, Position CWORD, Name CPTR) CWORD
 WD_DisplayPS(hInstance CWORD, Name CPTR) CWORD
 WD_FindFieldLength(hInstance CWORD, Length CWORD, Position CWORD, FindData CPTR) CWORD
 WD_FindFieldPosition(hInstance CWORD, Location CWORD, Position CWORD, FindData CPTR) CWORD
 WD_GetKey(hInstance CWORD, GetKeyData CPTR) CWORD
 WD_GetSessionHWND(hInstance CWORD) CWORD
 WD_Pause(hInstance CWORD, Length CWORD) CWORD
 WD_PostInterceptStatus(hInstance CWORD, Name CPTR) CWORD
 WD_QueryCursorLocation(hInstance CWORD, Location CWORD) CWORD
 WD_QueryFieldAttribute(hInstance CWORD, Attribute CWORD, Position CWORD) CWORD
 WD_QueryHostUpdate(hInstance CWORD, Name CPTR) CWORD
 WD_QuerySessionStatus(hInstance CWORD, SessionData CPTR) CWORD
 WD_QuerySessions(hInstance CWORD, Count CWORD, Data CPTR) CWORD
 WD_ReceiveFile(hInstance CWORD, TransferData CPTR) CWORD
 WD_Release(hInstance CWORD) CWORD
 WD_Reserve(hInstance CWORD) CWORD
 WD_ResetSystem(hInstance CWORD) CWORD
 WD_RunProfile(Profile CPTR, ShowWindow CWORD) CWORD
 WD_SearchField(hInstance CWORD, Location CWORD, Position CWORD, Buffer CPTR) CWORD
 WD_SearchPS(hInstance CWORD, Location CWORD, Position CWORD, Buffer CPTR) CWORD
 WD_SendFile(hInstance CWORD, TransferData CPTR) CWORD
 WD_SendKey(hInstance CWORD, Buffer CPTR) CWORD
 WD_SetCursor(hInstance CWORD, Position CWORD) CWORD
 WD_SetSessionParameters(hInstance CWORD, SSPParameters CWORD) CWORD
 WD_ShowSession(hInstance CWORD, ShowWindow CWORD) CWORD
 WD_StartHostNotification(hInstance CWORD, NotifyData CPTR) CWORD
 WD_StartKeystrokeIntercept(hInstance CWORD, InterceptData CPTR) CWORD
 WD_StopHostNotification(hInstance CWORD, Name CPTR) CWORD
 WD_StopKeystrokeIntercept(hInstance CWORD, Name CPTR) CWORD
 WD_Wait(hInstance CWORD) CWORD

endUses

Objeto : PAG1.INTERFACE Nombre del Metodo : Var

Source : Var
 HI Smallint
 endVar

Objeto : PAG1.INTERFACE Nombre del Metodo : open

Source : method open(var eventInfo Event)
 edit()
 AbrirRumba()

```
self.setTimer(300000)
```

```
endmethod
```

Objeto : PAG1.INTERFACE

Nombre del Metodo : close

Source : method close(var eventInfo Event)
 CerrarRumba()
 endmethod

Objeto : PAG1.INTERFACE

Nombre del Metodo : timer

Source : method timer(var eventInfo TimerEvent)

```
conecta("B")
Envia(HI,"@9")
conecta("A")
```

```
endmethod
```

Objeto : PAG1.INTERFACE

Nombre del Metodo : proc

Source : proc Conecta(pant String) SmallInt

```
var
    R    SmallInt
endVar
R=WD_ConnectPS(HI,pant)
IF R<>0 THEN
    R.view("Connect")
endif
return R
endProc
```

Proc Espera(hInstance SmallInt) SmallInt

```
var
    Resp,C SmallInt
endVar
c=0
Resp=-1
while Resp<>0
    Resp=WD_Wait(hInstance)
    if Resp=0 or c>3 then
        quitloop
    endif
    WD_Pause(hInstance,2)
    c=c+1
    if c>3 then
        msgstop("Error","Tiempo de Espera Excedido")
    endif
endwhile
return Resp
endProc
```

Proc Envia(hInstance SmallInt,Cadena String) SmallInt

```
var
    Resp SmallInt
endVar
Resp=WD_SendKey(HInstance,Cadena)
IF Resp<>0 THEN
    Resp.view("Send Key")
endif
```



```

        endif
        Espera(hInstance)
        Return Resp
    endProc

proc Manda(HI Smallint, dat String, len Smallint) Smallint
var
    a      String
endVar

    a=dat+space(len)
    Envia(HI,substr(a,1,len))
    Espera(HI)

    return 0
endproc

proc MandaFecha(HI Smallint, fecha Date) Smallint
var
    a,a1   String
endVar
    a="00"+string(year(fecha))
    a1=substr(a,size(a)-1,2)
    a="00"+string(month(fecha))
    a1=a1+substr(a,size(a)-1,2)
    a="00"+string(day(fecha))
    a1=a1+substr(a,size(a)-1,2)
    Envia(HI,a1)
    Espera(HI)

    return 0
endproc

Proc SoloNum(a string) string
var
    i smallint
    a1 string
endVar

    a1=""
    for i from 1 to size(a)
        if a.substr(i,1)>="0" and a.substr(i,1)<="9" then
            a1=a1+a.substr(i,1)
        endif
    endfor
    return a1
endproc

```

Objeto : PAG1.INTERFACE

Nombre del Metodo : AbrirRumba

Source : method AbrirRumba()

```

var
    t1,t2          TCursor
    r,e,e1,e2 SmallInt
    Pass,dat      String
endVar

    Pass=""
    HI=0
    enumWindowNames("INF.DB")
    t1.open("INF.DB")

```

```

if t1.locate("ClassName","PDOXWINDESKTOP") then
  t2.open("USUARIOS")
  if not t2.qlocate(GetNetUserName()) then
    beep()
    msgstop("Este usuario no esta registrado!",GetNetUserName())
  endif
  HJ=t1."InstanceHandle"
  R=WD_RunProfile("V4.DSP",5)
  IF R<>0 THEN
    R.view("RunProfile")
  endif
  Conecta("B")
  dat=space(103)
  e=size(dat)
  e2=0
  while e2<20
    R=WD_CopyOIA(HI,dat,e)
    IF R<>0 THEN
      R.view("CopyOIA")
      quitloop
    endif
    e1=ansiCode(substr(dat,82,1))
    if e1=32 then
      quitloop
    endif
    WD_Pause(HI,2)
    e2=e2+1
  endwhile
  Envia(HI,"LOGON APPLID(IDMSDC)@E")
  WD_Pause(HI,4)
  Envia(HI,"SIGNON ")
  Envia(HI,t2."SIGNON")
  Envia(HI,"@E")
  WD_Pause(HI,2)
  Envia(HI,rtrim(t2."PASW_V4")+ "@E")
  Envia(HI,"S SISl@E")
  Envia(HI,rtrim(t2."PASW_SISl")+ "@E")
  Envia(HI,"15")
  Envia(HI,"@E")

  R=WD_RunProfile("V2.DSP",5)
  IF R<>0 THEN
    R.view("RunProfile")
  endif
  Conecta("A")
  dat=space(103)
  e=size(dat)
  e2=0
  while e2<20
    R=WD_CopyOIA(HI,dat,e)
    IF R<>0 THEN
      R.view("CopyOIA")
      quitloop
    endif
    e1=ansiCode(substr(dat,82,1))
    if e1=32 then
      quitloop
    endif
    WD_Pause(HI,2)
    e2=e2+1
  endwhile
  Envia(HI,"LOGON APPLID(VIDA) LOGMODE(AUTO3270)@E")

```

```

        WD_Pause(HI,4)
        Envia(HI,"CSSN@E")
        WD_Pause(HI,2)
        Envia(HI,t2."SIGNON")
        Envia(HI,"@T")
        Envia(HI,t2."PASW_V2"+"@E")
        BbringToTop()
    else
        msgstop("ERROR","No encontramos PDOXWINDESKTOP")
    endif
    t1.close()

endmethod

```

Objeto : PAG1.INTERFACE

Nombre del Metodo : CerrarRumba

Source : method CerrarRumba()

```

var
    r      SmallInt
endVar

R=WD_ConnectPS(HI,"B")
Envia(HI,"@1")
Espera(HI)
Envia(HI,"@1")
Espera(HI)
Envia(HI,"BYE@E")
Espera(HI)

R=WD_DeletePS(HI,"B")
IF R<>0 THEN
    R.view("DeletePS")
endif

R=WD_ConnectPS(HI,"A")
Envia(HI,"CSSF LOGOFF@E")
Espera(HI)
R=WD_DeletePS(HI,"A")
IF R<>0 THEN
    R.view("DeletePS")
endif
HI=0

endmethod

```

Objeto : PAG1.INTERFACE.#Botón138

Nombre del Metodo : pushButton

Source : method pushButton(var eventInfo Event)

```

var
    T      TableView
    TC     TCursor
endvar

TC.attach(POLIZA)
T.open("HUERF",WinStyleDefault,10,400,1440*6.5,1440*3.5)
T.MoveToRecord(TC)
T.wait()
TC.attach(T)
POLIZA.locate("POLIZA",TC."POLIZA")
T.close()
maximize()

```

endmethod

Objeto : PAG1.INTERFACE.#Bolón101 Nombre del Metodo : pushButton

Source : method pushButton(var eventInfo Event)

```

var
    dat,dat1,datp,p   String
    e,e1,e2,r,f,f2    SmallInt
    n                  Number
endVar

dat=POLIZA
dat.view("Numero de Poliza")
f=1
if dat.size()<> 10 then
    beep()
    msgStop(datp,"El No. de Poliza debe tener 10 caracteres")
    f=0
else
    dat=upper(dat)
    datp=dat
    if not POLIZA.locate("POLIZA",dat) then
        POLIZA.insertAfterRecord()
        POLIZA=dat
    endif
    Envia(HI,"IMG ")
    Envia(HI,POLIZA.substr(3,8))
    Envia(HI,",CO=")
    Envia(HI,POLIZA.substr(1,2))
    Envia(HI,"S1 @E")
    Espera(HI)
    p="I"
    dat=space(5)
    e1=size(dat)
    e=1*80+2
    R=WD_CopyPSToString(HI,e,dat,e1)
    if dat="IN104" then
        Envia(HI,"TIMG ")
        Envia(HI,POLIZA.substr(3,8))
        Envia(HI,",CO=")
        Envia(HI,POLIZA.substr(1,2))
        Envia(HI,"S1 @E")
        Espera(HI)
        dat=space(5)
        e1=size(dat)
        e=1*80+2
        R=WD_CopyPSToString(HI,e,dat,e1)
        if dat="IN104" then
            beep()
            MsgInfo(datp,"No existe este numero de POLIZA")
            f=0
        else
            p="T"
        endif
    endif
endif

if f=1 then
    dat=space(33)
    e1=size(dat)

```

```

e=5*80+3
R=WD_CopyPSToString(HI,e,dat,e1)
NOMBRE=dat
if rtrim(DOMICILIO)=" " then
e=6*80+3
R=WD_CopyPSToString(HI,e,dat,e1)
dat1=rtrim(dat)
e=7*80+3
R=WD_CopyPSToString(HI,e,dat,e1)
dat1=dat1+" "+rtrim(dat)
e=8*80+3
R=WD_CopyPSToString(HI,e,dat,e1)
dat1=dat1+" "+rtrim(dat)
DOMICILIO=dat1
endif

e=5*80+43
R=WD_CopyPSToString(HI,e,dat,e1)
C_NOMBRE=dat
if rtrim(C_DOM)=" " then
e=6*80+43
R=WD_CopyPSToString(HI,e,dat,e1)
dat1=rtrim(dat)
e=7*80+43
R=WD_CopyPSToString(HI,e,dat,e1)
dat1=dat1+" "+rtrim(dat)
e=8*80+43
R=WD_CopyPSToString(HI,e,dat,e1)
dat1=dat1+" "+rtrim(dat)
C_DOM=dat1
endif

Envia(HI,"@E")

dat=space(5)
R=WD_CopyPSToString(HI,2*80+32,dat,5)
if dat="LINEA" then
; Linea Universal
f2=1
else
f2=0
endif
dat=space(2)
e=12*80+34
e1=size(dat)
R=WD_CopyPSToString(HI,e,dat,e1)
P_STATUS=dat

e=10*80+34
R=WD_CopyPSToString(HI,e,dat,e1)
P_FORMA_PAGO=dat

dat=space(11)
e=16*80+24
e1=size(dat)
R=WD_CopyPSToString(HI,e,dat,e1)
dat=Ltrim(dat)
dat=Rtrim(dat)
dat=substr(dat,2,dat.size()-1)
while dat.search(",")>0
e=search(dat,",")

```

```

        dat=substr(dat,1,e-1)+substr(dat,e+1,dat.size()-e)
    endwhile
    P_PRIMA=dat

    dat=space(8)
    e=7*80+28
    e1=size(dat)
    R=WD_CopyPSToString(HI,e,dat,e1)
    P_COB_HASTA=dat

    dat=space(11)
    e=22*80+47
    e1=size(dat)
    R=WD_CopyPSToString(HI,e,dat,e1)
    dat=Ltrim(dat)
    dat=Rtrim(dat)
    if dat.size(>0) then
        dat=substr(dat,2,dat.size()-1)
        while dat.search(",")>0
            e=search(dat,",")
            dat=substr(dat,1,e-1)+substr(dat,e+1,dat.size()-e)
        endwhile
    endif
    P_PRESTAMO=dat

    Envia(HI,p+"IMG ")
    Envia(HI,POLIZA.substr(3,8))
    Envia(HI,"CO=")
    Envia(HI,POLIZA.substr(1,2))
    Envia(HI,"S8Z @E")
    Espera(HI)

    e=2*80+11
    dat=space(5)
    e1=size(dat)
    R=WD_CopyPSToString(HI,e,dat,e1)
    AGENTE=Number(dat)

    Envia(HI,p+"IMG ")
    Envia(HI,POLIZA.substr(3,8))
    Envia(HI,"CO=")
    Envia(HI,POLIZA.substr(1,2))
    Envia(HI,"S6 @E")
    Espera(HI)

    n=0
    dat1="6"
    while dat1="6"
        e2=0
        while e2<4
            e=12*80+31+e2*15
            R=WD_CopyPSToString(HI,e,dat1,1)
            if dat1="0" then
                dat=space(2)
                e=1*80+30+e2*15
                R=WD_CopyPSToString(HI,e,dat,2)
                if f2=0 or dat="04" then
                    dat=space(8)
                    e=5*80+24+e2*15
                    e1=size(dat)
                    R=WD_CopyPSToString(HI,e,dat,e1)
                    P_F_EMI=dat
                endif
            endif
            e2=e2+1
        endwhile
        dat1=R
    endwhile

```

```

        dat=space(15)
        e=10*80+17+e2*15
        e1=size(dat)
        R=WD_CopyPSToString(HI,e,dat,e1)
        dat=Ltrim(dat)
        dat=Rtrim(dat)
        dat=substr(dat,2,dat.size()-1)
        while dat.search(",")>0
            e=search(dat,",")
            dat=substr(dat,1,e-1)+substr(dat,e+1,dat.size()-e)
        endWhile
        n=n+number(dat)

        dat=space(2)
        e=4*80+30+e2*15
        R=WD_CopyPSToString(HI,e,dat,2)
        EDAD=dat
        EDADA=number(dat)+year(today()-1900-number(substr(P_F_EMl,7,2))

        dat=space(5)
        e=19*80+27+e2*15
        e1=size(dat)
        R=WD_CopyPSToString(HI,e,dat,e1)
        P_F_VENC=dat
    endif
    endif
    e2=e2+1
endWhile
R=WD_CopyPSToString(HI,23,dat1,1)
Envia(HI,"@E")
endWhile
P_SUMA=n

Envia(HI,p+"IMG ")
Envia(HI,POLIZA.substr(3,8))
Envia(HI,"CO=")
Envia(HI,POLIZA.substr(1,2))
Envia(HI,"S7 @E")
Espera(HI)

dat=space(2)
e1=dat.size()
e=5*80+2
R=WD_CopyPSToString(HI,e,dat,e1)
B_BIT=false
B_IMA=false
B_DIBA=false
B_ISET40=false
B_ISE=false
B_DOP=false
B_CADE=false
while dat<>" "
    switch
        case dat="01": B_BIT=true
        case dat="03": B_IMA=true
        case dat="04": B_DIBA=true
        case dat="20": B_ISET40=true
        case dat="21": B_ISE=true
        case dat="25": B_DOP=true
        case dat="30": B_CADE=true
    endSwitch

```

```
e=e+80
R=WD_CopyPSToString(HI,e,dat,e1)
endWhile

Conecta("B")
Espera(HI)
Envia(HI,POLIZA.substr(3,8))
Envia(HI,"@E")
dat="*** CONSULTA EFECTUADA ***"
dat1=dat
e=23*80+1
e1=size(dat)
R=WD_CopyPSToString(HI,e,dat,e1)
if dat=dat1 then
    dat=space(11)
    e=16*80+18
    e1=size(dat)
    R=WD_CopyPSToString(HI,e,dat,e1)
    dat=Ltrim(dat)
    dat=Rtrim(dat)
    if dat.size()>0 then
        while dat.search(",")>0
            e=search(dat,",")
            dat=substr(dat,1,e-1)+substr(dat,e+1,dat.size()-e)
        endwhile
    endif
    Envia(HI,"@9")
else
    dat=""
endif
FAD=dat
Conecta("A")
endif

endmethod
```