

9
25



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**"SISTEMA DE INFORMACION GERENCIAL
PARA CUENTAS ESPECIALES"**

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N

CYNTHIA CRISTINA ALVARADO MENDEZ

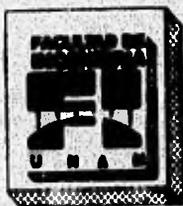
F. FRANCISCO EDUARDO FIERRO TORRES

AARON DAVID GARCIA NUREZ

JORGE ANTONIO GUERRERO ESPINAL

ANGEL MACIEL MARQUEZ

ASESOR: ING. ALBERTO TEMPLOS CARBAJAL



MEXICO, D. F.

MAYO 1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A la Facultad de Ingeniería de la Universidad Nacional Autónoma de México por habernos permitido formar parte de ella y a sus profesores por transmitirnos sus conocimientos.

Al Ing. Alberto Templos Carbajal por dirigir la elaboración del presente trabajo.

A la Dirección de Cuentas Especiales de Grupo Nacional Provincial por permitir el uso de su infraestructura e información.

A nuestros compañeros de la Facultad, que fueron un gran apoyo para cada uno de nosotros.

Mi más sincero agradecimiento a todas las personas que me apoyaron y comprendieron para lograr llegar a la culminación de este periodo tan importante de mi vida:

A Héctor Alvarado, mi padre, por proporcionarme todos los recursos necesarios para la obtención de una educación universitaria.

A mi madre por haberme acompañado y apoyado en los momentos difíciles, por los sacrificios que tuvo que hacer para que yo llegara a este momento, y por ser siempre la mejor amiga que he tenido, pero sobre todo por que es la mejor mamá del mundo.

A mi hermana que siempre ha sido mi mejor confidente, mi apoyo en las noches de desvelo y por ser una gran amiga, además de que se que siempre podré contar con ella.

A mis tías Mary, Lydia, Irma y Ana por ser las mejores segundas mamás que cualquiera pudiera desear o tener.

A mi prima Mónica por ser la mejor hermana del corazón y por estar conmigo siempre que la he necesitado.

A mis primos Rafa, Luis, Ricardo y Max por apoyarme y quererme siempre, sin importar lo que pase.

A mis tíos Maximiliano y Rafael quienes forman parte de mi vida, por su cariño desinteresado y apoyo incondicional.

A Mela, Anita e Iflgo (q.e.d) que ya no están entre nosotros, pero que son la guía en diferentes ámbitos de mi vida y mis aspiraciones futuras, GRACIAS.

A todos mis amigos y amigas por su incomparable apoyo durante todos estos años de estudio, gracias por su amistad.

Pero sobre todas las cosas, Gracias a DIOS por permitirme llegar a este momento y por contar con la una gran familia.

CYNTHIA CRISTINA ALVARADO MÉNDEZ

Este pequeño esfuerzo significa el término de una de las etapas más importantes de mi vida, mi formación profesional, por lo cual quiero agradecer a las siguientes personas:

A Ma. Teresa Torres:

Por su esfuerzo, su amor y sus sacrificios; por su constante optimismo y alegría; por ayudarme e impulsarme siempre para lograr mis objetivos. Gracias mamá.

A Victor Fierro:

Por su incondicional apoyo en todos los sentidos, su cariño, su firme decisión de forjar mi futuro y sus consejos siempre acertados. Gracias papá.

A Carmen, Vicky, Armando y Victor:

Por ser un ejemplo para mí y permitirme serlo, por el cariño que siempre me han brindado y por el gran significado que le han dado a la palabra hermano.

A Gaby Sierra:

Por ser el impulso final para lograr esta meta y por su contagioso optimismo.

A los pequeños Scarlet, Pamela y Alfonso:

Por su sonrisa que alienta a seguir con cualquier proyecto en la vida.

A Martín y Alfonso:

Mis cuñados, por su apoyo moral y material.

A mi tío Luis Torres y al buen amigo Beto (q.e.d.):

Dos personas que ya no están con nosotros pero que siempre creyeron en mí.

F. EDUARDO FIERRO TORRES

Esta tesis se la dedico con mucho cariño a todas las personas que me apoyaron desde niño y actualmente, a la gente que supo poner algo bueno en mí, a sus palabras, a su tiempo y su cariño, en especial se la dedico a mi padre a quien debo todo lo que soy, gracias.

A mi Abuelita Margarita (q.e.d) , una mujer adelantada a su tiempo , por guiar mis pasos en momentos difíciles haciendo de mí una persona responsable y trabajadora.

A mis hermanas Olivia y Corina por su cariño y apoyo, por ser parte de mí .

A mis tíos Esperanza , Luz Clara, Georgina, Antonio, Teresa, Fidel, Teresa C. , Ernesto por enseñarme, guiarme y por que siempre he contado con su apoyo.

A mis primos Ernesto, Mauricio, Fernando, Luis Miguel, Corina, Margarita, Luz María, por brindame su apoyo y su cariño, preocupados por la conclusión de mi carrera profesional.

A mis amigos Raúl, Adrián y Víctor Hugo, con quienes he compartido éxitos y fracasos, momentos llenos de amistad, he conocido lugares y me han ayudado a conocerme mejor, gracias a su valioso apoyo ha sido posible continuar el camino que me he trazado.

A mis compañeros de Generación con quienes comparti el ideal y el entusiasmo de concluir una carrera universitaria, a mis profesores de la facultad por sus enseñanzas y transmitirme el orgullo de ser ingeniero .

A la gente a quien le debo su tiempo, cariño y enseñanzas y a quienes no tengo palabras para agradecerles su apoyo su confianza y sobre todo el creer en mí. Gracias Mary, Norma, Alejandra, Cuitlahuac.

En especial a Grupo Nacional Provincial y a todos mis compañeros de trabajo, por su profesionalismo, por dejarme aprender de ellos y ayudarme a ser mejor cada día.

AARÓN DAVID GARCÍA NÚÑEZ.

A la memoria de mi madre que ha sido la persona más maravillosa que yo he conocido.

A mi padre y a Lily por ser mi familia y permitirme vivir con ellos momentos de alegría y tristeza.

A Paty por ser parte importante de mi vida y concederme la oportunidad de aprender a convivir con una compañera.

A mis amigos Ernesto Flores, Gabriel Chávez, Ana María López, Juan Castro, Juan Carlos Castellán, Ezequiel Quiñonez, Arturo García y algunos otros no menos importantes con los cuales me divertí (y me divierto) enormemente.

A mis compañeros de Tesis por que sin el esfuerzo de todos este trabajo no hubiera sido posible.

JORGE ANTONIO GUERRERO ESPINAL.

La conclusión de mi carrera profesional significa una de las metas más importantes en mi vida. Este objetivo no podría llevarse a cabo sin la colaboración y apoyo de las siguientes personas.

Dedico la culminación de esta tesis principalmente a mi madre (Gloria E. Márquez Escalera), por darme la existencia, por su constante y desinteresado apoyo, su esfuerzo y constancia durante toda mi carrera, quien me ha hecho sentir que soy capaz de alcanzar cualquier objetivo fijado y de vencer cualquier obstáculo encontrado en mi vida.

A mi abuelita (Teresa Escalera G.), hermana (Ana G. Maciel), primos, tíos y sobrinos maternos quienes me han inculcado un sentido de responsabilidad y superación tanto personal como profesional.

A todos los compañeros de mi trabajo actual, que con ayuda de sus conocimientos y larga experiencia en el área de sistemas informáticos han contribuido constantemente en mi formación profesional.

A todos mis amigos y compañeros universitarios con los que he compartido mis éxitos y fracasos, quienes compartieron desvelos y cansancios en la culminación de mi carrera.

ANGEL MACIEL MARQUEZ.

SISTEMA DE INFORMACION GERENCIAL PARA CUENTAS ESPECIALES ÍNDICE

I. Introducción.

I.1 Antecedentes de "Grupo Nacional Provincial".	1
I.2 Organización de "Grupo Nacional Provincial".	2

II. Identificación de Necesidades.

II.1 Importancia de la Información a nivel Dirección.	5
II.2 Descripción de la situación actual.	6
II.3 Diseño y Elaboración de Entrevistas.	7
II.4 Determinación de la factibilidad del proyecto.	12
II.5 Propuestas de Mejoramiento.	13

III. Análisis y selección de Metodología.

III.1 Yourdon.	15
III.2 Pressman.	17
III.3 Merise.	18
III.4 Elección de la Metodología.	21

IV. Análisis Estructurado.

IV.1 Modelo ambiental del sistema.	23
IV.2 Diagramas de Flujo de datos.	25
IV.3 Elaboración del Diccionario de Datos.	36

V. Diseño Estructurado.

V.1 Elaboración del Diagrama entidad-relación.	38
V.2 Diseño de la base de datos.	39
IV.3 Establecimiento de niveles de seguridad.	46

V.4 Elaboración de la carta de estructura de procesos.	52
V.5 Diseño de la interfaz con el usuario.	56
VI. Aspectos de la codificación del sistema.	
VI.1 Descripción de Paradox 4.5 como herramienta de desarrollo.	59
VI.2 Implantación de estándares de codificación.	62
VI.3 Pruebas modulares.	64
VI.4 Programación del sistema.	75
VII. Implementación del sistema.	
VII.1 Descripción de Hardware.	80
VII.2 Pruebas de desempeño.	81
VII.3 Implantación.	82
VII.4 Capacitación del usuario.	84
VIII. Conclusiones.	86
Bibliografía.	88
Apéndice A Ejemplos de reportes del Sistema.	A1
Apéndice B Términos utilizados	B1
Apéndice C Manual Técnico	C1
Apéndice D Manual del Usuario	D1

OBJETIVO.

Desarrollar un sistema de consulta dirigido a ejecutivos y funcionarios, que proporcione información referente a los indicadores económicos y estadísticos de los clientes pertenecientes a la Dirección de Cuentas Especiales de Grupo Nacional Provincial, con el propósito de facilitar la toma de decisiones.

I. Introduction.

I. INTRODUCCIÓN

1.1 Antecedentes de Grupo Nacional Provincial.

La historia de las compañías de seguros en México nos remonta a los inicios de Grupo Nacional Provincial, S.A. la cual atravesó varias etapas de retos, incertidumbres económicas, movimientos armados, que acontecían en el país, saliendo siempre adelante gracias a la capacidad creadora, a la imaginación y esfuerzo por crear una conciencia nacional sobre la seguridad y sus beneficios de compartir el riesgo. Los retos anteriores llevaron a la fusión de las dos empresas de seguros líderes en el país "La Nacional" y "La Provincial", consolidándose como una empresa de liderazgo y desarrollo.

En los primeros años del siglo XX se cristalizó un ambicioso proyecto del inglés William B. Woodrow y de varios hombres de negocios, estableciendo la primera compañía de seguros sobre la vida con sede en México, creando así "La Nacional". En medio de tiempos de cambios y tensiones sociales como la huelga de Cananea y Río Blanco, el régimen del porfiriato amenazado con el grito revolucionario del plan de San Luis Potosí, la entrada triunfal de Francisco I. Madero el 17 de julio a la ciudad de México y la lucha armada paralizando las actividades económicas de la nación, transcurren los primeros años de "La Nacional". En 1916 contaba con 14 empleados y se habían emitido 384 pólizas ese año. La empresa continuó desarrollándose y es así como el 27 de Diciembre de 1932 se construye el primer rascacielos de México, sede de "La Nacional".

El 20 de mayo de 1936 Arthur Woodrow y un grupo de hombres de negocios obtienen la autorización de la Secretaría de Hacienda para construir una nueva empresa de seguros "La Provincial" formándose así las dos empresas de seguros con más desarrollo en el país. A finales de los 60 comenzó a gestarse la operación conjunta de Nacional Provincial, dejando La Nacional de operar el ramo de Daños y Provincial el de Vida para buscar una productiva especialización. Pero no es hasta el año de 1972 cuando se integra como grupo en uno de los conglomerados empresariales de mayor prestigio en México.

Grupo Nacional Provincial forma parte de Grupo Bal, que es la denominación que se utiliza para describir al Grupo Bailleres, el cual se constituye como uno de los grupos empresariales de mayor renombre nacional, ya que agrupa a empresas vanguardistas en diversos sectores especializados de la actividad económica y social de nuestro país; entre ellas destacan:

Sector Industrial

Grupo Peñoles.

Sector Financiero

Grupo Nacional Provincial, Seguros.
Crédito Afianzador, S.A. Compañía Mexicana de Garantías.
Casa de Bolsa Cremi.
Arrendadora Cremi.
Cambinter.
Finco.

Sector Comercial

El Palacio de Hierro, S.A. De C.V.

Sector Educativo

Instituto Tecnológico Autónomo de México (ITAM).

1.2 Organización de Grupo Nacional Provincial.

Actualmente la estructura de Grupo Nacional Provincial esta conformada como se describe a continuación:

Por un presidente del consejo de administración de las empresas que conforman Grupo Bal, por miembros del consejo de administración de las empresas de Grupo Bal, el director general de Grupo Nacional Provincial, los directores ejecutivos, los directores de área, los directores de centros regionales y los subdirectores de las direcciones de área.

Entre las direcciones ejecutivas se encuentra la dirección de Sector Financiero, Cuentas Especiales y Negocios Internacionales. La Dirección de Cuentas Especiales se desprende de esta dirección ejecutiva. Se describirá con mayor detalle la Dirección de Cuentas Especiales dado que será esta área quien operará el sistema informático.

La Dirección de Cuentas Especiales perteneciente al área de Automóviles se forma por cuatro subdirecciones: Cuentas Especiales I, Cuentas Especiales II, Cuentas Especiales III y Soporte a la Operación.

La Subdirección de Cuentas Especiales I esta constituida por dos gerencias: la Gerencia de Operación de cuya cabeza dependen un analista y cuatro asistentes, y la Gerencia de Servicio.

La Subdirección de Cuentas Especiales II se forma de dos gerencias: la Gerencia de Operación, de la cual dependen un analista, cuatro asistentes y un auxiliar administrativo, y una Gerencia de Servicio.

La Subdirección de Cuentas Especiales III esta formada por dos Gerencias: la Gerencia de Operación con cuatro asistentes y la Gerencia de Servicio.

La Subdirección de Soporte a la Operación aglutina a la Gerencia de Estadística, la Gerencia de Estudios Especiales y la Jefatura de Riesgos Mayores. La Gerencia de Estadística está integrada por tres analistas de cuentas especiales, uno de autos, tres asistentes y un auxiliar administrativo, la misma estructura corresponde a la Gerencia de Estudios Especiales, en tanto que la Jefatura de Riesgos Mayores cuenta con un analista de autos y dos auxiliares.

Es en la Dirección de Cuentas Especiales donde a través de las Gerencias de Operación se lleva el movimiento operativo para las cuentas más importantes de la compañía en el ramo de automóviles, y a través de las Gerencias de Servicio se contacta con posibles clientes para lograr un aumento de los mismos. La Subdirección de Soporte a la Operación se encarga de proporcionar toda la información referente a los movimientos realizados mensualmente para cada cuenta, así como proporcionar cualquier información necesaria a la diferentes subdirecciones.

La Dirección de Cuentas Especiales se encarga también del pago a asegurados y genera los indicadores estadísticos y económicos de los clientes, siendo de vital importancia en este punto contar con información actualizada. La información que genera esta Subdirección es canalizada a las diferentes direcciones ejecutivas que así lo soliciten.

Dada la situación plasmada en el párrafo anterior se vuelve indispensable el conocimiento oportuno de la información. El poder contar con cifras actuales y de diferentes periodos de tiempo, nos lleva a la necesidad de contar con un sistema de consulta que permita analizar y actualizar la información relacionada con los cortes efectuados de cada cuenta, sus principales características de identificación de cuenta, tipo de negociación, fórmulas de cálculo de dividendos, siniestralidad acumulada anual, así como los diferentes reportes que son generados a nivel dirección y para otras áreas.

Las figuras 1.1 y 1.2 describen el organigrama general de Grupo Nacional Provincial y el organigrama de la Dirección ejecutiva de Cuentas Especiales, Sector Financiero y Negocios Internacionales respectivamente.

Aprovechando los recursos técnicos de equipo e instalaciones del área, se desarrolló un sistema gerencial de consulta para Cuentas Especiales, al cual se le llamó SIGCE (Sistema de Información Gerencial Para Cuentas Especiales), éste permite un conocimiento oportuno de la información a nivel dirección, subdirección y gerencia, con lo cual se puede llevar a cabo una mejor toma de decisiones de manera ágil y oportuna. Asimismo el SIGCE permite a la Dirección proporcionar puntualmente la información que los clientes requieran.

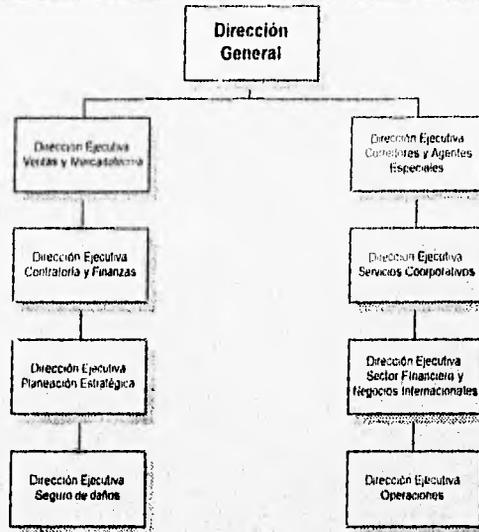


Figura 1.1 Organigrama de la Dirección General de Grupo Nacional Provincial.

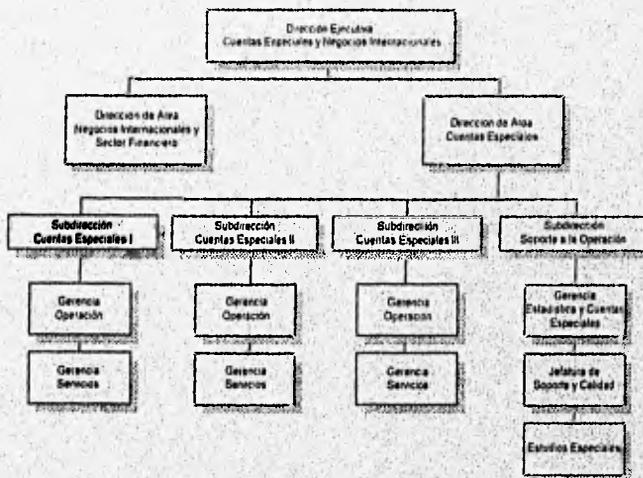


Figura 1.2 Organigrama de la Dirección Ejecutiva Sector Financiero, Cuentas Especiales y Negocios Internacionales.

II. Identificación de Necesidades.

II. IDENTIFICACIÓN DE NECESIDADES.

II.1. *Importancia de la Información a nivel dirección.*

En una empresa como Grupo Nacional Provincial el tiempo puede representar tanto el atender a un asegurado con calidad y servicio profesional, logrando su satisfacción, como el presentar cifras a nivel dirección ejecutiva para tomar una decisión referente al futuro de una cuenta nueva o la permanencia de otra y por lo tanto evaluar el nivel de ingresos de la compañía.

En Cuentas Especiales se consideran a los clientes más importantes de la compañía en el ramo de automóviles, asegurándose flotillas de los mismos. Son cuentas muy grandes en cuanto al volumen de información ya que se manejan de forma histórica, algunas tendrán cierta vigencia dependiendo de la negociación realizada. Esta información es generada mes a mes, procesada y revisada para poder presentársela al cliente y poder indicarle la siniestralidad de la cuenta a un determinado corte, es decir, obtener la relación de lo que nos a pagado de prima (monto del seguro) y lo que se ha cubierto (pagos de siniestros ocurridos) por compartir el riesgo en algo que se denomina porcentaje de siniestralidad.

A nivel dirección el contar con todos estos indicadores puede significar una buena negociación entre el agente del cliente que lleva la cuenta y Grupo Nacional Provincial a través de Cuentas Especiales. El conocimiento de estos indicadores puede considerarse como base para tener argumentos confiables para la toma de decisiones, o para el informe de cierre. A nivel Subdirección el tener información confiable representa una consulta directa sobre la siniestralidad de cada cuenta a través de un periodo de tiempo para conocer el comportamiento de ésta, así también conocer los cálculos de dividendos efectuados, datos relacionados con la vigencia, la forma de extracción, próxima emisión de estadísticas y poder contar con los reportes solicitados al momento. A nivel ejecutivo la información oportuna permite el control sobre la correcta calendarización y emisión de estadísticas, permite también llevar un control sobre el estatus de elaboración de la cuenta y asegurar que la forma en que se identifica la cuenta sea la correcta y se encuentre actualizada.

En todos los factores presentados anteriormente se encuentra presente el tiempo de respuesta que es importante tanto para usuarios internos de la compañía, como para los clientes externos.

II.2. Descripción de la situación actual.

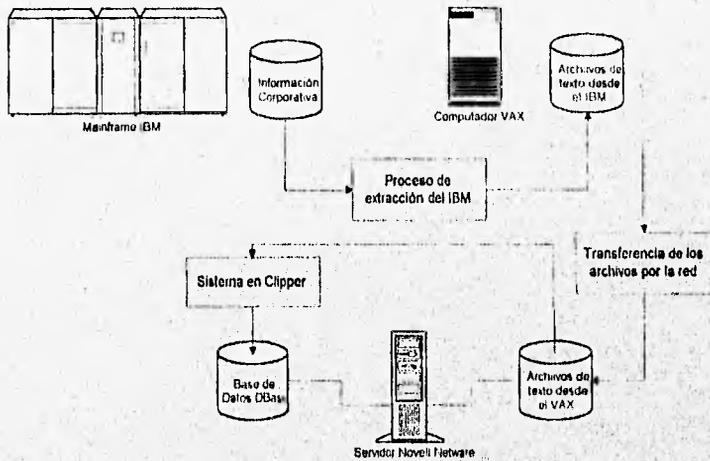
Actualmente el área de Cuentas Especiales cuenta con la Gerencia de Estadística que se encarga de generar toda la información acumulada a un mes, denominada fecha de corte. La planificación para producir esta información inicia con la calendarización de cuentas de las subdirecciones de cuentas especiales.

El proceso comienza creando archivos que son un reflejo de la situación actual de la base de datos principal (la cual se encuentra bajo una mainframe IBM 4381) sobre siniestros en un determinado período. Estos archivos son explotados para obtener la información de las cuentas en calendario. Esta información se obtiene a través de aplicaciones desarrolladas en lenguaje Cobol produciendo archivos texto, los cuales son transferidos por medio de una VAX 11760 hasta el servidor Novell donde se almacena para su procesamiento.

Dichos archivos son utilizados para alimentar sistemas ya existentes elaborados en Clipper y otros que son realizados manualmente a través de paquetes como Dbase y Quattro Pro, obteniendo un reporte de siniestros asociados a pólizas pertenecientes a una determinada cuenta. Este reporte es formado en un juego de impresión para revisión posterior de las Gerencias de Operación. La Gerencia de Operación correspondiente revisa que los pagos efectuados a tales siniestros estén cargados correctamente a las coberturas correspondientes, y verifica la reserva pendiente de los siniestros en caso de cerrar o abrir el estatus de una reclamación.

Al término de la revisión se devuelven los listados y se realizan las modificaciones en los sistemas tanto a nivel micro como en el sistema mainframe. Se realizan modificaciones y se genera nuevamente un juego de impresión para una nueva revisión por parte de la Gerencia de Operación. Si no existen modificaciones se forman juegos de impresión final de los cuales uno es para el cliente y otro para la Gerencia de Estadística. En caso contrario se repite el ciclo.

Una vez entregadas todas la estadísticas se actualizan cuadros o reportes con las nuevas cifras obtenidas que son parte de los informes dirigidos a la dirección, estos son generados una vez que se concluye el proceso de revisión y modificación para obtener concentrados por cuenta al periodo de elaboración de la estadística. El ciclo de elaboración se representa gráficamente en la figura II.1.



II.3. Diseño y elaboración de entrevistas.

Las entrevistas con los usuarios finales del sistema es una fase de vital importancia para el análisis y recolección de información del sistema. Las razones por las cuales se requiere la elaboración de entrevistas son las siguientes:

- La necesidad de reunir información sobre el comportamiento y explotación del método con el cual se opera y administra actualmente la información, además de conocer los requerimientos de cada uno de los usuarios para poder contemplarlos en el nuevo sistema.
- La necesidad de reunir información acerca del sistema actual con el fin de realizar cálculos de costo-beneficio.

Algunas de las reglas que se toman en cuenta para obtener una entrevista exitosa con el usuario son las siguientes:

- Desarrollo de un plan global de las entrevistas. Comprende la selección de usuarios finales del sistema en base a los organigramas de la empresa, así como la secuencia y combinación de las mismas.
- Planeación de las entrevistas para una administración efectiva del tiempo.

- Identificar la información que le interesa a cada uno de los usuarios y las necesidades de cada uno de los mismos.

En las páginas siguientes se muestran las entrevistas realizadas a diversos funcionarios de Grupo Nacional Provincial. Cada una de las entrevistas tiene un objetivo e indica puesto y nombre de los usuarios finales entre otros datos.

Objetivo.- Identificar las necesidades de información a nivel Subdirección.

Nombre del entrevistado : Lic. Mario Aguillón

Puesto del entrevistado : Subdirector de Cuentas Especiales III

Fecha de aplicación : 21 de Diciembre de 1995

1.- ¿Qué cuentas especiales maneja su Subdirección?

En cuentas especiales se administran varias cuentas , cada una se encuentra dentro de un grupo específico , como puede ser cuentas de pago a declaración en el caso de las cuentas englobadas dentro de Tráslado y plan piso , cuentas multianuales , anuales como puede ser Volkswagen Credit , Volkswagen Leasing , Grupo Ica o carreteras.

Cuentas Especiales III , atiende a cuentas como Grupo Ica, Volkswagen Credit, Volkswagen Leasing, Volkswagen de México , los tramos de carreteras , autopistas del sol , Autopistas Grupo Ica, Protexa, Capufe (Caminos y Puentes Federales) , básicamente .

2.- ¿ Qué tipo de negociación tienen las cuentas que opera su Subdirección?

El tipo de negociación depende de varios factores , Si la cuenta es nueva se hace un análisis sobre la conveniencia de la administración , en caso de carreteras se fijan montos de acuerdo reportes de siniestralidad anterior mejorando o conservando las cuotas al cliente por compartir el riesgo. En el caso de Grandes negocios se realiza en cuanto si es una siniestralidad acumulada o por periodo , si el calculo de dividendos se realizara anual o Semestral. Para las cuentas de pago a declaración su manejo es diferente , el cliente nos paga lo que nosotros demostramos que hemos pagado por atender la cuenta .

3.- ¿ Qué información requiere de cada cuenta y por qué es importante?

De cada cuenta me interesa saber su siniestralidad a un corte en particular , es decir contar con la estadística de la cuenta , si se le otorga dividendo o no , pues es dinero que sale de la compañía. Me interesa saber como están programadas la generación de las mismas. Todo esto es muy importante debido a la importancia en si de la cuenta y del cliente , brindándole una información oportuna.

4.- ¿ Recibe oportunamente la información que necesita?

En ocasiones existen retrasos , se pierde y hay que volver ha solicitar dicha información. El cliente puede hacer una petición especial como solicitar un calculo de dividendos anticipado, a un determinado corte y se necesita saber cuando esta calendario la emisión de la estadística.

5.- ¿ Con qué frecuencia requiere la información?

Mucho depende de la cuenta , hay cuentas que contar con la información bimestral , o cada vez que lo solicite el cliente ,pero hay cuentas que se necesitan mes a mes la información, porque constantemente se actualizan las emisiones de primas.

6.- ¿Requiere comparativos de cortes de información?

Sí, son solicitados por la dirección ejecutiva , para planear o estimar presupuestos anuales, para elaborar proyecciones sobre el comportamiento de las cuentas en un año, poder corregir desviaciones de la siniestralidad de la cuentas.

7.- ¿Si contará con un sistema que satisfaga sus necesidades de información cómo le gustaría que fuera?

Que me otorgara el manejo oportuno de la información de la cuentas de la subdirección, que me permitiera obtener las cifras más recientes , históricas y por periodo , que me permitiera actualizar o modificar cifras para ver diferentes comportamientos en los cálculos. Un sistema funcional .

Objetivo.- Conocer los requerimientos y manejo de la información a nivel de ejecutivo de operación.

Nombre del entrevistado : Lic. Ramón Flores .

Puesto del Entrevistado : Gerente de Operación Cuentas Especiales I.

1.- ¿Podría indicar cuáles son sus principales funciones?

Básicamente mis funciones están orientadas al manejo operativo de las cuentas de la subdirección , es decir en esta gerencia se atiende al asegurado a través del agente encargado de la cuenta. Se controlan los pagos de los siniestros , emitiéndose cheques de pagos de siniestros , se atienden siniestros donde es necesario investigar si el pago procedo ,si se contempla en la cobertura de la póliza se , es atender todo lo que involucra la parte operativa de una cuenta revisan las estadísticas de las diferentes cuentas. Asi como un control de las pérdidas totales tanto de daños y robo.

2.- ¿Qué informaciones históricas requiere?

Las carátulas de las cuentas , los dividendos otorgados y en un momento dado el soporte de las estadísticas.

3.- ¿Cuáles son sus clientes especiales?

Todos los clientes pertenecientes a la Dirección de Cuentas Especiales y en particular los clientes de la Subdirección de Cuentas Especiales I , a los cuales se les da un trato preferencial por la importancia de la cuenta y estar englobada en Cuentas Especiales .

4.- ¿Cuáles son las características más importantes de sus clientes?

Son Cuentas muy grandes de flotillas de autos , donde el número de siniestros llega ha ser considerable , siendo el compromiso ante el cliente de atender estas cuentas y reportarle mensualmente al cliente la siniestralidad de la cuenta, para ver su comportamiento.

5.- ¿Qué necesidades de información ha detectado por parte de sus clientes?

Información del comportamiento histórico , comparativos de cuentas a un determinado periodo de tiempo, actualización de información ya reportada , Reportes de salvamentos vendidos y reportados en la estadística.

6.- ¿ Si contará con un sistema que satisfaga sus necesidades de información como le gustaría que fuera?

Un sistema funcional que proporcione información histórica , a un determinado periodo , permanente acerca de las cuentas , su calendario de emisión , los que me proporcionara la emisión de primas por cliente a un periodo , contar con información oportuna .

II.4. Determinación de la factibilidad del proyecto.

Actualmente la Dirección de Cuentas Especiales tiene la infraestructura necesaria para el manejo operativo y administrativo de las cuentas. Posee equipo Olivetti 486 con 8 Mb de memoria RAM, conectado a una Red Novell con ambiente Gráfico Windows 3.1 desde donde se accesan diferentes paquetes como Paradox, Quattro Pro y Wordperfect, realizándose también la conexión a otros equipos como la IBM 4381 y la VAX 11760. El equipo Olivetti emula terminales para la realización de la captura complementaria, las consultas, modificación y realización de pagos, por otro lado se ejecutan los procesos para generar los archivos texto de las diferentes cuentas.

Los requerimientos del sistema son :

- Desarrollar un sistema de consulta enfocado al uso exclusivo de funcionarios y ejecutivos de la Dirección de Cuentas Especiales, el cual permita realizar consultas dinámicas de consolidados de información, en periodos cortos de tiempo, así como poder generar los reportes técnicos y de resultados a nivel dirección.
- Un sistema que funcione en red, permitiendo la consulta simultánea de varios usuarios y el uso de periféricos.
- Acceso directo a través de una clave y una contraseña asignando niveles de seguridad por usuario.
- El sistema debe permitir la consulta de los diferentes reportes, y su impresión, el mantenimiento de los catálogos de cuentas y las cifras de cada corte por cuenta.
- Los reportes que debe generar el sistema son los siguientes:
 1. Resultado técnico.
 2. Resumen de estadística.
 3. Reporte de producción y cobranza
 4. Reporte de emisión de primas por periodo
 5. Datos contables
 6. Catálogo de cuentas por funcionarios
 7. Calendarios de emisión de cuentas por mes
 8. Calendario de estatus de emisión
 9. Catálogo general de cuentas.
- El sistema debe desarrollarse en Paradox para Windows 4.5 basándose en los estándares internos de la compañía, ya que se cuenta con la licencia corporativa de Borland.

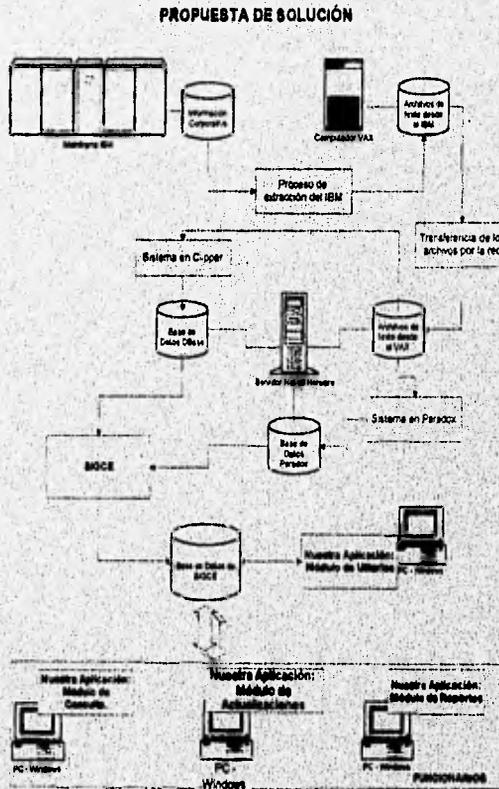
El equipo y la Infraestructura con que cuenta Grupo Nacional Provincial permite convertir en una realidad el Sistema de Información Gerencial de Cuentas Especiales. La Dirección y Subdirección dan su apoyo al proyecto puesto que están convencidas de que

es necesario contar con un sistema de información que facilite la toma de decisiones en estos niveles.

II.5. Propuesta de Mejoramiento.

El proceso actual de elaboración de estadísticas se modernizará con el desarrollo de nuevos sistemas en Paradox, lo cual mejorará la operación de las cuentas, acortará el calendario de emisión, automatizando el mayor número de cuentas, lo que permitirá dedicar más tiempo al análisis y no al desarrollo, evitando errores y permitiendo la detección oportuna de errores. Actualmente durante la generación de reportes de concentrados primero se elabora una estadística y posteriormente se vacían los datos, en este sentido el sistema agilizará las consultas complementándose con los sistemas existentes.

Este sistema cubrirá los requerimientos de los usuarios y permitirá un mejor manejo y control de la información generada. En la figura II.2 se plasma la propuesta de solución.



III. Análisis y selección de Metodología.

III. ANÁLISIS Y SELECCIÓN DE METODOLOGÍA.

Uno de los elementos más importantes en las empresas de desarrollo de software son los individuos de alta especialización, no obstante, el contar con excelente programadores no asegura el éxito en la consecución de los objetivos planteados, debido a que se puede presentar falta de conjunción, resultado de que cada individuo utiliza una personal manera de desarrollar, todo ello lleva a la imposibilidad de trabajar en un perfecto equipo.

Debido a lo anterior es necesario seguir pasos estandarizados, es decir, una metodología de desarrollo de software, que establezca los pasos a tomar con sus procedimientos y técnicas para producir el software. De tal manera que se establezcan sistemas cooperativos para atacar de manera integral, ordenada y eficiente las múltiples facetas de un determinado problema.

Lo anterior es más comprensible si se analiza la definición de PARNAS para la Ingeniería de Software "Multi-Person Construction of Multi-Version Software"¹, es decir, la Ingeniería de software es la construcción de sistemas de información, que son tan grandes, que se necesitan varios grupos de programadores e Ingenieros para elaborarlos, y de los cuales se generan varias versiones en el transcurso del tiempo, y a los que se les proporcionará mantenimiento durante un tiempo relativamente largo. De la definición de PARNAS observamos que se requiere trabajar en equipo y comprender que lo que unos realizan será el complemento de los otros, que a su vez se combinarán con otros y así se formará un gran sistema de fácil mantenimiento y de gran calidad.

El desarrollo de un producto de software de alta calidad se realiza con más efectividad, si se usan técnicas que aporten productos intermedios, como serían los documentos de especificación y de diseño. En el proceso de desarrollo debe existir una validación para cada etapa, evitando al máximo el error humano e incrementando la calidad final.

Recientemente, ha empezado a cambiar el enfoque que se le da al desarrollo de sistemas. Cada vez son más las organizaciones grandes y pequeñas que están adoptando un ciclo de vida uniforme y único para sus proyectos. Lo anterior se conoce como la metodología de desarrollo de sistemas. La metodología de desarrollo de sistemas ofrece un procedimiento común para producir un sistema computacional que pueda orientar a cualquier miembro de la organización.

Existen diversas metodologías de desarrollo de software, por lo que es necesario analizarlas y evaluarlas, para seleccionar la que mejor se adapte a las necesidades planteadas. Normalmente se selecciona una metodología estándar con las variaciones y modificaciones que sean necesarias. En este trabajo se analizarán los conjuntos metodológicos de Yourdon, Pressman y Merise, los cuales presentan características comunes en los aspectos generales, aunque difieren en la profundidad con que aborden las diversas etapas del desarrollo de sistemas de información.

¹ Parnas Software Design.

III.1 Yourdon.

Yourdon llama a su metodología de desarrollo de sistemas ciclo de vida estructurado del proyecto, y propone los siguientes pasos para llevar a cabo satisfactoriamente cualquier producto de software:

- La encuesta.

Este punto se conoce también como el estudio de factibilidad o como estudio inicial del negocio. Comúnmente comienza cuando el usuario solicita que una o más partes de los procesos que realiza se automatizen. Entre los objetivos de la encuesta se tienen los siguientes:

1. Identificar a los usuarios responsables y crear un entorno inicial del sistema.
2. Establecer metas y objetivos para el nuevo sistema.
3. Determinar si es factible automatizar el sistema y de ser así, sugerir escenarios aceptables.
4. Preparar el esquema que se usará para guiar el resto del proyecto.

- Análisis Estructurado.

El propósito principal de la actividad de análisis es transformar las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente del usuario con diagramas de flujo de datos, diagramas entidad-relación, etc.

Adicionalmente al modelo del sistema que describe los requerimientos del usuario, generalmente se prepara un conjunto de presupuestos y cálculos de costos y beneficios más precisos y detallados al final de la actividad de análisis.

- Diseño estructurado.

La actividad de diseño se dedica a la creación de una jerarquía apropiada de módulos de programas y de interfaces entre ellos para implementar la especificación creada durante el análisis. Además la actividad de diseño se ocupa de la transformación de modelos de datos entidad-relación en un diseño de base de datos.

- Codificación.

Durante esta actividad se desarrolla la programación y la integración de módulos. En este punto se consideran herramientas tales como la programación estructurada. La implantación se propone que sea descendente, es decir, se elaboren los esqueletos de

los módulos principales, y posteriormente se realice la programación de los submódulos que componen cada módulo principal.

- **Elaboración de pruebas de aceptación.**

La especificación estructurada debe contener toda la información necesaria para definir un sistema que sea aceptable desde el punto de vista del usuario. Por eso, una vez generada la especificación puede realizarse la actividad de producir un conjunto de casos de prueba de aceptación a partir de la especificación estructurada.

- **Garantía de calidad**

Se requiere de alguna actividad que verifique que el sistema tenga un nivel apropiado de calidad, es decir, que cumpla no sólo con los requisitos del usuario si no que adicionalmente cuente con buena documentación, tiempo de respuesta aceptable, codificación entendible, etc. Para que los puntos anteriores sean posibles de alcanzar, es importante llevar a cabo actividades de garantía de calidad en cada una de las fases de la metodología puesto que con ello se asegura en gran medida la calidad final del producto.

- **Elaboración de manuales de usuario.**

Una de las actividades importantes a realizar es la generación de una descripción formal de las partes del sistema que se harán en forma manual, lo mismo que la descripción de cómo interactuarán los usuarios con la parte automatizada del nuevo sistema.

- **Conversión de base de datos.**

Esta actividad requiere como entrada la base de datos actual del usuario, al igual que la especificación del diseño. El diseño de la base de datos se establece a partir del diagrama entidad relación. Durante la implementación física de la base de datos se pueden utilizar las ventajas otorgadas por cada manejador DBMS, por ejemplo creación de roles de usuarios, establecimiento de restricciones, indicaciones de cómo utilizar el espacio en disco, etc.

- **Instalación.**

La instalación representa el último paso de la metodología de Yourdon. Las entradas de la instalación son el manual del usuario, la base de datos convertida y el sistema aceptado. La puesta en marcha del nuevo sistema puede acompañarse de ejecuciones en paralelo del sistema anterior (en caso de que exista) en tanto se comprueba que funciona adecuadamente en un medio real.

III.2 Pressman.

Pressman sugiere las siguientes etapas para el desarrollo adecuado de software de sistemas:

- Definición.
 1. Análisis del Sistema.
 2. Planificación del Proyecto de Software.
 3. Análisis de los requerimientos.
- Desarrollo.
 1. Diseño del software.
 2. Codificación.
 3. Prueba del software
- Mantenimiento.
 1. Corrección.
 2. Adaptación.
 3. Aumento.
 4. Prevención.

Para la etapa de definición, Pressman menciona que es necesario desarrollar un enunciado definitivo del problema por resolver. También es necesario incluir una descripción de la situación actual, restricciones del problema y las metas que se lograrán. Además de justificar que es necesaria una estrategia computacional en la solución del problema. Se deben identificar las funciones por realizar, las restricciones, el equipo electrónico con que se cuenta, el personal, etc. Es necesario determinar los objetivos y requisitos en el nivel del sistema para el proceso de desarrollo y los productos finales y por último establecer criterios de alto nivel para la aceptación del sistema. Según Pressman se analizan los requerimientos para especificar las necesidades funcionales y de desempeño del sistema, así como para la detección de excepciones. Pressman proporciona la siguiente lista de especificaciones de los requisitos para la producción de software:

1. Resumen y panorama del producto.
2. Ambientes de desarrollo, operación y mantenimiento.
3. Interfaces externas y flujo de datos.
4. Requisitos de operación.
5. Manejo de excepciones.
6. Subconjuntos iniciales y prioridades instrumentales.
7. Modificaciones y mejoras previsible.
8. Criterios de aceptación.
9. Guías y sugerencias de diseño.
10. Índice cruzado.
11. Glosario de términos.

El diseño del software requiere de concebir, planear y especificar las características del producto final, las cuales incluyen la definición de las pantallas de presentación, de los formatos de reportes, la definición de las entradas y salida de datos, entre otras cosas. Es necesario incluir la concepción, planeación y especificación de la estructura interna y de los detalles del proceso del producto, así como las detalles de los algoritmos, las estructuras de datos y planes de pruebas.

En resumen, el proceso de diseño comprende el desarrollo de una visión conceptual del sistema, el establecimiento de una estructura, la identificación de las cadenas de datos y su almacenamiento, la descomposición de funciones de alto nivel en subfunciones, el establecimiento de las relaciones e interconexiones entre componentes, el desarrollo de la representación de datos en forma concreta y la especificación de los detalles de los algoritmos.

La fase de codificación tiene por objetivo principal escribir el código fuente y la documentación interna de modo que sea fácil verificar y corregir posibles errores, o dar mantenimiento, se debe contar con un conjunto de requisitos de programación bien definidos y una descripción del diseño detallado.

Las pruebas del software sirven para determinar el apego a las especificaciones, incluyéndose las especificaciones de requisitos, la documentación del diseño, diversos principios generales de estilo, estándares del lenguaje de programación, estándares del proyecto, estándares organizacionales y expectativas del usuario.

Los errores ocurren cuando una parte del producto es incompleta, los tres errores más comunes son de requisito que se provocan por una propuesta incorrecta de las necesidades del usuario, por falta de una especificación completa de los requisitos funcionales y de desempeño, entre otras cosas. Los errores de diseño, los cuales se producen por fallas al traducir los requisitos en estructuras de solución correctas y completas, por inconsistencias, etc. y por último los errores de codificación, que son los cometidos al traducir las especificaciones del diseño en código fuente.

Las actividades de mantenimiento del software implican realizar mejoras en los productos de software, adaptar productos a nuevos ambientes y corregir problemas. El mantenimiento consume en la mayoría de los casos el 50 % del presupuesto del ciclo de vida de un producto. La mejora de los productos puede dar como resultado proporcionar nuevas capacidades funcionales.

III.3 Merise.

La metodología Merise surge en Francia a partir del año 1977, como un intento de definir una metodología a emplear en la Administración Pública para el desarrollo y diseño de sistemas informáticos.

Los principios generales en que se apoya Merise son:

- Desglose del desarrollo por etapas.

- Definición de los documentos estándar de cada una.
- Uso del modelo Entidad-Relación y sus principios para la representación de datos.
- Uso de las redes de Petri para la representación de procesos y tratamientos.
- Definición de grupos de trabajo y reparto de las responsabilidades y funciones a lo largo del desarrollo.
- Especificación del reparto de tareas y tratamientos entre los usuarios y el ordenador.
- Definición de los flujos de información entre las unidades del sistema.

La concepción del sistema para su análisis y construcción consiste en trabajar sobre dos elementos: datos y tratamientos sobre los mismos. Los primeros son la información que se maneja en el entorno del sistema y las relaciones dentro de ella. Los tratamientos reflejan los procesos que se aplican a la información y su secuencia en el tiempo para obtener los resultados deseados.

Estos dos elementos del sistema son tratados en tres niveles: conceptual, lógico y físico.

La metodología utiliza técnicas para modelar ambos elementos a todos los niveles. Algunas técnicas utilizadas son las siguientes:

- Utilización de grupos de trabajo para el desarrollo del proyecto (comité de directores, comité de usuarios y grupo de desarrolladores).
- Representación gráfica de diagramas de flujo.
- Modelado de datos (conceptual, lógico y físico).
- Modelado de tratamientos (conceptual, organizativo y operacional).

La metodología propone que la concepción de los datos y los tratamientos puede ser llevada a cabo de manera paralela y con algunas interfaces entre sí; sin embargo, es muy importante tomar en cuenta el orden de los niveles de concepción del sistema: En primer lugar realizar la conceptualización de los dos elementos, después la concepción lógica y al final obtener un resultado físico.

Merise sugiere una secuencia de etapas, fases y pasos a seguir en el desarrollo de un proyecto informático como metodología para desarrollo de proyectos y distingue cuatro etapas a lo largo de un proyecto:

- Estudio Preliminar.

Fase 1: Recolección de Datos:

Recolección Inicial.
Estudio de la Situación Actual.
Síntesis y Crítica de la Situación actual.

Fase 2: Concepción de la nueva solución.
Objetivos a alcanzar.
Descripción de la solución.

Fase 3: Evaluación y plan de desarrollo.
Evaluación de la nueva solución.
Plan de Desarrollo.

- Estudio Detallado.

Fase 1: Concepción general.

Fase 2: Concepción detallada de las fases.
Realización de las especificaciones detalladas de procesos.

Fase 3: Plan de desarrollo.

- Realización.

Fase 1: Estudio Técnico.

Fase 2: Producción.

- Puesta en marcha.

Fase 1: Preparación de los recursos físicos y humanos.

Fase 3: Recepción y lanzamiento del sistema.

El modelo Entidad-Relación es una representación gráfica con el fin de mostrar, en primer lugar mediante técnicas de diagramación del mundo real las entidades o agrupaciones de datos con significado dentro del sistema, y las relaciones entre ellas.

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

- Qué sucesos desencadenan la ejecución de procesos.
- Bajo qué condiciones de esos sucesos desencadenantes se ejecuta el proceso.
- En qué consiste el proceso, es decir que operaciones hay que realizar.

- Quién ejecuta el proceso.
- Resultados obtenidos o salidas del proceso.

Para la estructuración de las acciones a realizar se define "Proceso" como toda actividad de la empresa con sus entradas y salidas establecidas y "Operación o Procedimiento" como una serie de acciones ejecutables sin interrupción y que se ejecutan bajo las mismas condiciones en cuanto a factores desencadenantes.

III.4 Elección de Metodología.

Las tres metodologías expuestas en los apartados anteriores mencionan facetas similares para el desarrollo de sistemas, sintetizando podemos distinguir las fases generales como sigue :

- Análisis.
- Diseño y Codificación.
- Pruebas e Instalación.

Sin embargo, cada metodología resalta ciertos aspectos, por ejemplo, Pressman da gran importancia a la fase de mantenimiento y corrección del software, en tanto que Yourdon plantea que si en cada fase de la metodología se aplican reglas de calidad, las cuales consisten en una revisión detallada de cada documento generado en cada fase y la congruencia de los mismos; el sistema calificará de acuerdo a los requerimientos iniciales, y por lo tanto la fase de mantenimiento y corrección estará limitada.

Merise menciona la necesidad de tener grupos organizados de trabajo para desarrollar proyectos y de las tres metodologías analizadas es el único autor que detalla este paso como parte esencial de la misma, en tanto que los otros dos autores no ponen énfasis en este aspecto, aunque implícitamente dejan ver que el desarrollo de sistemas se lleva a cabo en grupos de trabajo, sin embargo no exponen reglas claras de cómo formar y administrar éstos.

Un aspecto importante de la implantación de sistemas lo constituyen la capacitación de los usuarios y la entrega de manuales de usuario. Yourdon marca como parte de su metodología la elaboración de manuales de usuario, no obstante, en la fase de instalación no habla explícitamente de la capacitación a los usuarios finales, aunque se entiende que si se quiere implantar un sistema que tenga ciertos parámetros de calidad se hará necesaria la capacitación de los usuarios. En tanto, Merise sí menciona en su metodología la capacitación del usuario, sin detallar los aspectos del manual de usuario, sin embargo, este documento podría ser cubierto en la fase de capacitación.

En cuanto a las herramientas de análisis y diseño Merise se enfoca en las redes de Peiri para conceptualizar los procesos del sistema y utiliza el diagrama Entidad-Relación para

mostrar los datos. Queda confuso el saber qué herramientas maneja Merise para el diseño, así también no profundiza en aspectos y técnicas de codificación.

Pressman enfoca el análisis y diseño de sistemas en un concepto llamado jerarquía de datos, manejado inicialmente por Jackson y Warnier. Los datos son definidos dentro de los procesos y esto vuelve difícil su manejo.

Los diagramas de flujos de datos son la herramienta con la cual Yourdon nos muestra los procesos y flujos entre entidades pertenecientes al sistema. Al igual que Merise maneja el diagrama Entidad-Relación para modelar los datos del sistema, y agrega un concepto muy importante el diccionario de datos. Bajo la metodología de Yourdon las herramientas de análisis y diseño están íntimamente ligadas y deben estar "balanceadas", es decir, todo flujo de datos debe estar especificado en el diccionario de datos, todo almacén de datos debe estar plasmado en el diagramas Entidad-Relación, etc.

Consideramos que la metodología de Yourdon ofrece las siguientes ventajas sobre las otras dos metodologías:

- Menos compleja y por lo tanto más clara.
- Manejo de herramientas de análisis y diseño fáciles de entender por parte del usuario.
- Permite pasar fácilmente a la codificación.
- Documenta coherentemente la aplicación y hace fácil el mantenimiento.
- Garantiza la calidad en cada etapa del desarrollo.

Por las ventajas anteriormente expuestas, se concluye que Yourdon ofrece el mejor conjunto metodológico de desarrollo de sistemas de información y que de acuerdo a nuestras necesidades (sistema con un enfoque administrativo) nos proporciona las mejores herramientas para llevar a cabo exitosamente el sistema que queremos generar. Lo anterior no excluye por supuesto, que tomemos ideas de los otros dos autores, así pues, el resto del documento estará basado en la metodología de Yourdon sin olvidar aspectos que consideremos adecuados de las otras dos.

IV. Análisis Estructurado.

IV. Análisis Estructurado.

IV.1. Modelo ambiental del sistema.

Para realizar el análisis del sistema el primer paso a seguir es la definición de todas las interfaces entre el sistema y el resto del entorno, es decir, el medio ambiente. Yourdon llama "Modelo Ambiental" al modelo por medio del cual se deben determinar los aspectos que se encuentran en el interior del sistema y los aspectos que se encuentran en el exterior; los elementos que no forman parte de éste deben ser excluidos del análisis. Es importante tomar en cuenta la información que está en el exterior del sistema y la información que se produce como salida para determinados acontecimientos. La identificación de dichos acontecimientos para los cuales el sistema debe responder, es un aspecto crítico del modelo ambiental; sin embargo se debe notar que el número de acontecimientos a considerar no es infinito, ya que solo algunos de ellos que acontecen en el ambiente exterior del sistema requieren de una respuesta automática del mismo.

Es muy importante el correcto dimensionamiento de la cobertura del sistema en su ambiente, ya que se puede llegar a extremos como los de cubrir requerimientos muy específicos sin tomar en cuenta los aspectos más amplios, o tratar de realizar un sistema demasiado amplio para los recursos y tiempo que se tengan.

En general, la importancia del modelo ambiental es la identificación de los alcances del sistema, interactuando con un medio ambiente (otros sistemas de mayor envergadura), mediante determinadas interfaces; estableciendo una frontera arbitraria en la cual el analista puede tener una gran influencia, pero que se define mediante acuerdos con el usuario o políticas determinadas.

El modelo del ambiente consta de tres componentes :

1. **Declaración de propósitos.** Es una declaración textual breve y concisa del propósito del sistema, dirigida al nivel administrativo superior, la administración de los usuarios, y otros que no están directamente involucrados con el desarrollo del sistema. Puede constar de una, dos o varias frases, sin embargo jamás debe llegar a más de un párrafo, ya que la intención no es proporcionar una descripción completa y detallada del sistema, sino simplemente describir el propósito y en algunos casos resumir los beneficios tangibles y cuantificables que se logren.
2. **Diagrama de contexto.** Es un caso especial del diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema. El diagrama de contexto enfatiza varias características importantes del sistema :
 - Las personas, organizaciones y sistemas con los que se comunica los cuales son conocidos como terminadoras.
 - Los datos que el sistema recibe del exterior y deben ser procesados.
 - Los datos que el sistema produce y que se envían al exterior.

- Los almacenes de datos que el sistema comparte con los terminadores.
- La frontera del sistema.

3. Lista de acontecimientos. Es una lista narrativa de los "estímulos" que ocurren en el exterior y a los cuales el sistema debe responder, estos acontecimientos pueden ser de flujo de datos, de tiempo o de control.

Los conceptos anteriores son aplicados a continuación al Sistema de Información Gerencial de Cuentas Especiales objeto de esta tesis.

Declaración de propósitos.

"El propósito del Sistema de Información Gerencial de Cuentas Especiales (SIGCE) es almacenar y actualizar información consolidada, necesaria para la consulta de indicadores económicos y estadísticos de los clientes pertenecientes a la Dirección de Cuentas Especiales, lo cual permitirá una mejor toma de decisiones a nivel directivo".

Diagrama de Contexto.

La figura IV.1 muestra el diagrama de contexto del SIGCE, sobresalen los siguientes terminadores la Gerencia de Operación, la Gerencia de Estadística, la Gerencia de Servicios y la Subdirección de Cuentas Especiales.

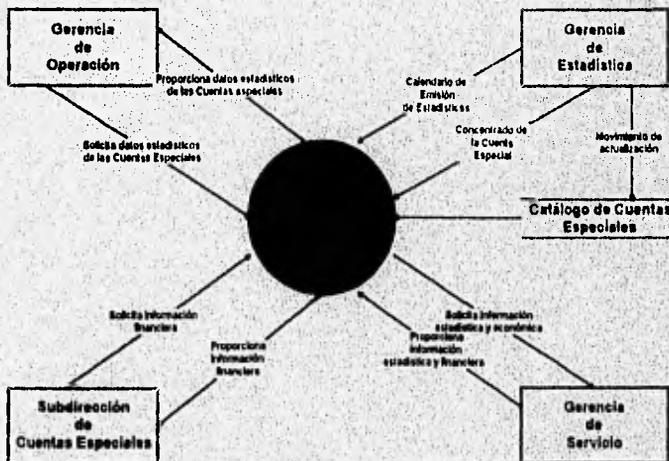


Figura IV.1 Diagrama de Contexto para el sistema SIGCE

Lista de acontecimientos.

1. La Gerencia de Estadística ingresa carátulas de cada cuenta especial a la fecha de corte.
2. La Gerencia de Estadística ingresa el calendario de emisión de estadísticas.
3. La Gerencia de Estadística actualiza el reporte de estatus de elaboración de cuentas especiales.
4. La Gerencia de Estadística actualiza el catálogo de cuentas especiales.
5. La Gerencia de Estadística actualiza el reporte de producción y cobranza de primas de cuentas especiales por período.
6. La Gerencia de Estadística solicita la estimación de presupuesto anual.
7. La Gerencia de Estadística ingresa datos para cálculo de dividendos.
8. La Gerencia de Operación solicita las carátulas por cuenta especial a una fecha de corte.
9. La Gerencia de Operación consulta el calendario de emisión de estadísticas.
10. La Gerencia de Operación consulta el estatus de elaboración de cuentas especiales.
11. La Gerencia de Operación consulta el catálogo de cuentas especiales.
12. La Gerencia de Servicios consulta carátulas de cada cuenta especial a una fecha de corte.
13. La Gerencia de Servicios consulta el reporte de resultado técnico.
14. La Gerencia de Servicios consulta el catálogo de cuentas especiales.
15. La Subdirección de Cuentas Especiales solicita el reporte de cálculo de dividendos.
16. La Subdirección de Cuentas Especiales solicita el reporte de producción y cobranza.
17. La Subdirección de Cuentas Especiales solicita el reporte de emisión de primas.

IV.2. Diagramas de Flujo de datos.

Un Diagrama de Flujo de Datos (DFD) es una herramienta gráfica para el análisis estructurado de sistemas, que permite visualizar un sistema como una red de procesos con intercambio de datos entre sí. Estos diagramas pueden permitir el modelado no tan solo de sistemas de información, sino de incluso organizaciones enteras, utilizándolos como herramientas de planeación estratégica. Sin embargo, puede darse el caso que no sea necesario el uso de esta herramienta, como en sistemas en los que son más

importantes las relaciones entre datos o el comportamiento de estos en el tiempo, sin llegar a tener un número de procesos significativo.

Un DFD debe tener las siguientes características para su funcionalidad:

- Se debe dibujar con una notación sencilla y clara, ya que es la manera común de plasmar los objetivos del usuario y del analista.
- Debe ser reducido a dibujarse en una sola hoja, de tal modo que sea fácil tener una visión global para entenderlo mejor.
- El diseño del diagrama debe ser ordenado y estandarizado, para que sea fácil de efectuar cualquier cambio necesario. El autor de la metodología propone siempre utilizar un paquete de computadora para llevar a cabo este diseño en la forma que se plantea, por lo que en nuestro caso, los diagramas serán diseñados en el paquete "Visio for MS Windows 2.0" de Shapeware Corp.

Los componentes de un DFD son los siguientes:

1. Proceso, o Burbuja o Función o transformación. Es una parte del sistema que transforma entradas en salidas, se representa gráficamente como un círculo, un rectángulo, o un rectángulo con esquinas redondeadas (en nuestro caso se utilizará la primera opción para seguir los lineamientos de Yourdon). Un proceso se describe con una sola palabra, frase u oración sencilla que describa lo que hace el proceso con los datos. Sin embargo, se presentan casos en los que el nombre del proceso refleja quién o qué es lo que está efectuando el proceso, lo cual depende de la naturaleza del sistema.

2. Flujo. El flujo de datos es la parte del sistema que representa el movimiento de bloques o paquetes de información de una parte del sistema a otra, es decir, los datos en movimiento. Se representa mediante una flecha que entra o sale de un proceso y es la parte que alimenta o retira información de los mismos. El flujo consta de un nombre, que representa el significado del paquete que se mueve a lo largo del mismo; y de una dirección, representada por la cabeza de la flecha y que significa si el flujo es hacia o desde el proceso.

Un flujo puede conducir un solo un tipo de paquete de información, pero un mismo paquete puede tener diferentes significados de acuerdo al flujo por el que esté pasando.

Un flujo puede ser también convergente o divergente, ya que puede ser formado de, o puede formar varios flujos con paquetes más sencillos, desde varios procesos o hacia varios procesos diferentes respectivamente. Sin embargo, el DFD no tiene la tarea de tomar en cuenta los tiempos en que estos flujos se efectúan, para lo cual se requiere definir el procedimiento interno del sistema.

3. Almacén. Se utiliza para modelar una colección de paquetes de datos en reposo y se denota por dos líneas paralelas arriba y abajo de su identificador. Un almacén significa un archivo o base de datos física, un medio de almacenamiento magnético, óptico, un

archivo en papel, un directorio, o cualquier lugar en el que la información permanezca estática desde que entra al almacén hasta que sale de él mediante flujos.

Algo importante en el análisis es el propósito del almacén, el cual es importante identificar, ya que puede ser no necesario. Algunos motivos para tener un almacén dentro de un sistema pueden ser los siguientes:

- Como archivo intermedio para ejecutar procesos en tiempos diferentes.
- Como respaldo en caso de aborto de un proceso.
- Para prueba y corrección de errores.
- Como anticipación a necesidades futuras.

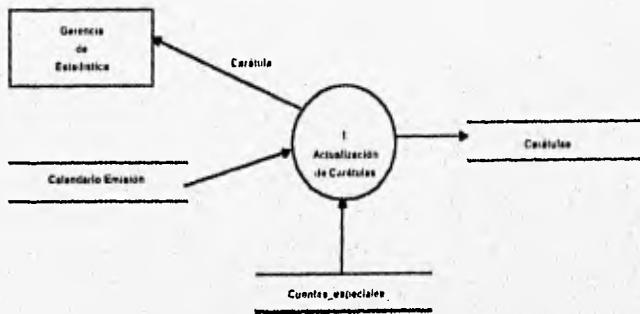
Un almacén es pasivo, es decir, los datos que contiene no viajarán a menos que un proceso lo solicite y cuando un paquete de información se mueve a través de un flujo, el almacén no cambia. Por otro lado, cuando el flujo de datos entra al almacén, se lleva a cabo una inserción, un borrado o una actualización de datos.

4. Terminador. Representan entidades externas al ambiente del sistema con las cuales se comunica. Gráficamente se representan como un rectángulo con un identificador. Normalmente representa una persona, un grupo, un departamento o una organización externa. Puede ser también incluso otro sistema que suministra o pide información al que se está modelando.

Como consecuencia de que el terminador es totalmente ajeno al sistema, el analista no puede cambiar los contenidos, la organización ni los procedimientos de un terminador; solamente puede modelar un sistema que se adapte a los requerimientos y a las salidas que puede proveer.

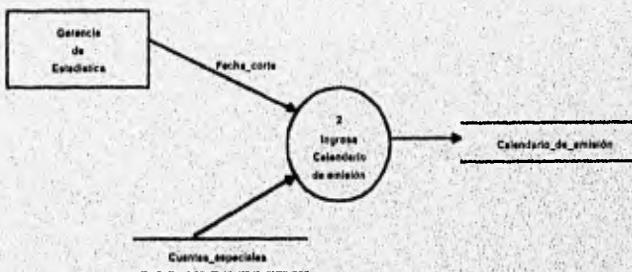
Es importante señalar que en ningún caso se deben representar en el DFD las relaciones entre terminadores, ya que estas no forman parte del sistema.

A continuación se presentan los Diagramas de Flujo de Datos para el Sistema de Información Gerencial para Cuentas Especiales, los cuales sirven como base para la posterior construcción del Diccionario de Datos.



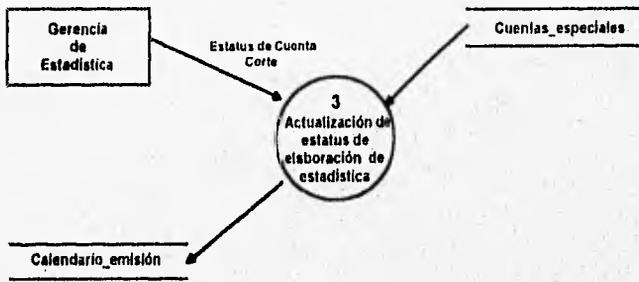
Acontecimiento 1. La Gerencia de Estadística ingresa carátulas de cada cuenta especial a la fecha de corte

Figura IV. 2.



Acontecimiento 2. La Gerencia de Estadística ingresa al calendario de emisión de estadística.

Figura IV. 3.



Acontecimiento 3. La Gerencia de Estadística Ingresa estatus de elaboración.

Figura IV. 4.



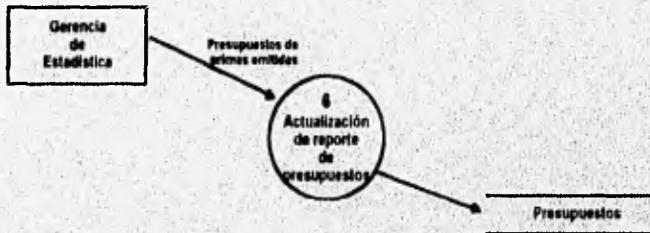
Acontecimiento 4. La Gerencia de Estadística actualiza el catálogo de Cuentas Especiales.

Figura IV. 5.



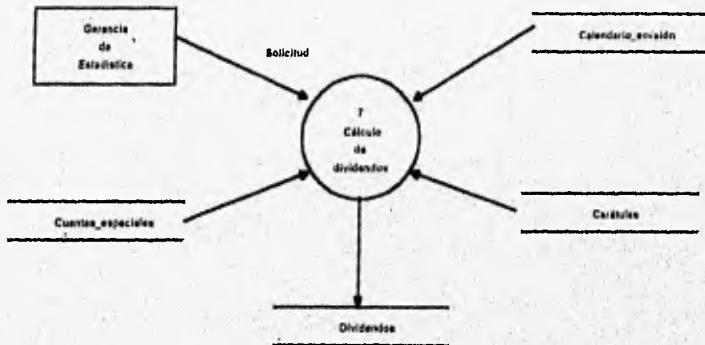
Acontecimiento 5. La Gerencia de Estadística actualiza reporte de primas por periodo.

Figura IV. 6.



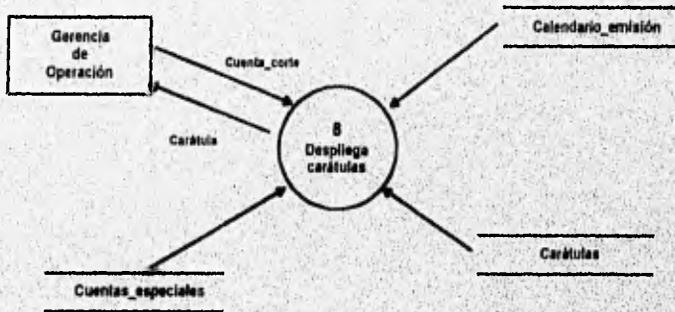
Acontecimiento 6. La Gerencia de Estadística actualiza reporte de presupuesto.

Figura IV. 7.



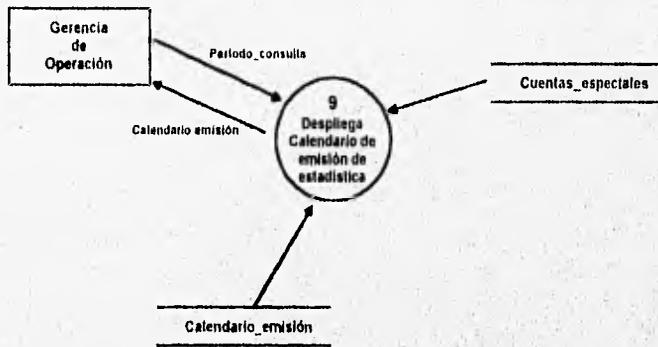
Acontecimiento 7. La Gerencia de Estadística Ingressa datos para cálculo de dividendos.

Figura IV. 8.



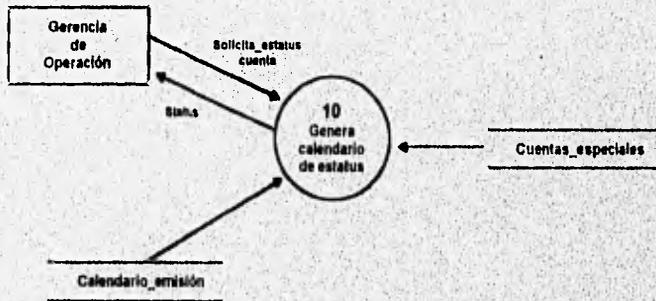
Acontecimiento 8. La Gerencia de Operación solicita las carátulas por cuenta especial a una fecha de corte.

Figura IV. 9.



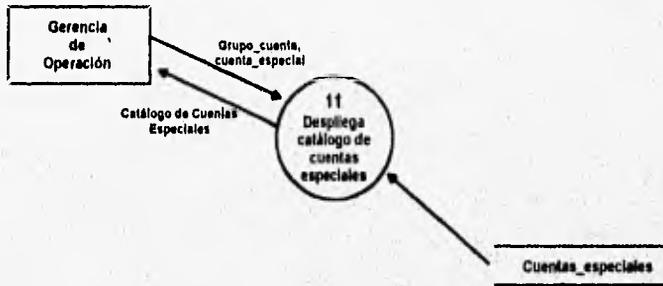
Acontecimiento 9. La Gerencia de Operación consulta el calendario de emisión de estadísticas.

Figura IV. 10.



Acontecimiento 10. La Gerencia de Operación consulta estatus de elaboración de cuentas especiales.

Figura IV. 11.



Acontecimiento 11. La Gerencia de Operación consulta el catálogo de cuentas especiales.

Figura IV. 12.



Acontecimiento 12. La Gerencia de Servicio consulta carátula de cada cuenta a una fecha de corte.

Figura IV. 13.



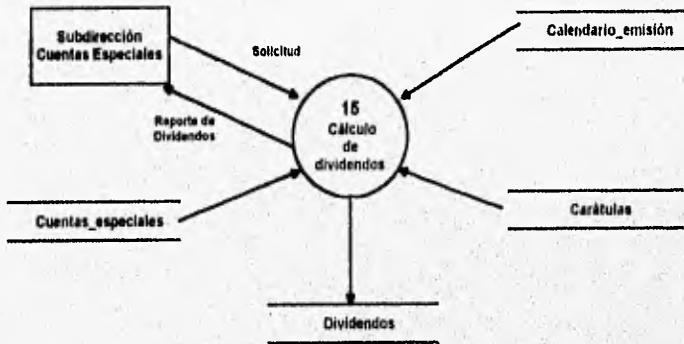
Acontecimiento 13. La Gerencia de Servicios consulta reporte de resultado técnico.

Figura IV. 14.



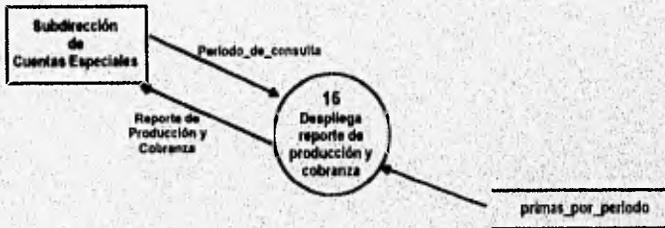
Acontecimiento 14. La Gerencia de Servicios consulta el catálogo de cuentas especiales.

Figura IV. 15.



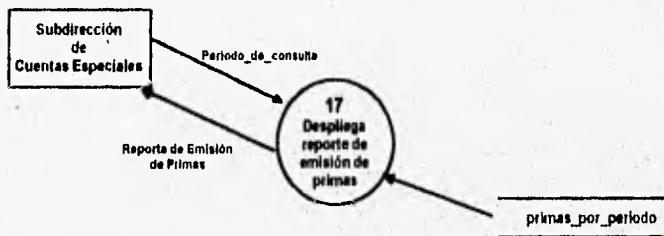
Acontecimiento 16. La Subdirecciones de Cuentas Especiales consultan reporte de cálculos de dividendos.

Figura IV. 16.



Acontecimiento 16. La Subdirección de Cuentas Especiales solicita el reporte de producción y cobranza.

Figura IV. 17.



Acontecimiento 17. La Subdirección de Cuentas Especiales solicita el reporte de emisión de primas.

Figura IV. 18.

IV.3. Elaboración del Diccionario de Datos.

Concepto y Antecedentes:

El diccionario de datos es un listado organizado de todos los datos pertinentes del sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista de sistemas tengan un entendimiento claro de todos los conceptos, entradas del sistema, salidas, entidades y cálculos que desarrolle internamente el sistema. Sin los diccionarios de datos, el modelo de los requerimientos del usuario no puede considerarse completo.

El diccionario de datos es desarrollado por el analista durante el desarrollo del modelo del sistema, pero el usuario debe ser capaz de leerlo y entenderlo.

El diccionario de datos comprende los siguientes puntos:

- Describe el significado de los flujos y entidades que se muestran en el diagrama de flujo de datos.
- Describe la descomposición de atributos del sistema en niveles inferiores, esto es por ejemplo: el atributo del nombre del cliente puede considerarse integrado por tres subatributos: Apellido paterno, materno y nombre.
- Describe las características y tipos de datos de cada uno de los atributos de las entidades incluidas en el sistema.

- Describe a detalle las relaciones entre las entidades del sistema.

Algunos manejadores de bases de datos de cuarta generación, por ejemplo, Oracle, Informix, etc.; ya cuentan con subsistemas desarrollados especialmente a la documentación y creación del diccionario de datos. Si no se cuenta con un gestor de bases de datos como los anteriores, es recomendable desarrollar un sistema interno para esta función, ya que en la mayoría de los casos, el diccionario de datos suele ser realmente voluminoso.

V. Diano
E. ...

V. Diseño estructurado.

V.1 Elaboración del Diagrama Entidad-Relación.

Un diagrama Entidad-Relación (DER) es una descripción gráfica de la distribución de la información en un sistema. Esta descripción es referente solo a los datos del sistema y no modela ninguna función del mismo como lo hace el Diagrama de Flujo de Datos.

El diagrama Entidad-Relación es importante debido a que las estructuras de datos y las relaciones de éstos en un sistema pueden ser tan complejas que se requiera un modelado de los datos de forma que sea posible facilitar su entendimiento por parte del analista y los que mantienen el sistema.

Tener un modelo de datos en forma gráfica, significa la posibilidad de actualizar, corregir, aumentar y complementar los datos de la organización con los datos de otros sistemas internos o inclusive externos. Estas tareas referentes al mantenimiento de datos de la base de datos implica a veces tener un grupo de personas dedicado a ello llamadas Administradores de la Base de Datos (DBA). Tales administradores deben tener un conocimiento generalizado de todos los datos que pueden ser factibles de almacenar, de qué forma almacenarlos, con qué relaciones, con qué privilegios para qué usuarios e incluso la manera de organizar los datos físicamente en el hardware.

Para los analistas, el DER es de mucha utilidad para conceptualizar los almacenes de datos en los diagramas de flujos de datos, ya que en la mayoría de los casos (como debería ser siempre), tales almacenes son comunes a todos los sistemas, por lo que cualquier afectación a la base de datos, debe tener en cuenta la interacción con otros sistemas que afectan o consultan los mismos almacenes. En el caso del SIGCE, los almacenes de datos son generados desde otros sistemas, en tanto que la consulta a los mismos no es compartida con ningún otro sistema, sin embargo, el diagrama Entidad-Relación es necesario para que a partir de él se realice un diseño de la base de datos con la información necesaria, así como con la seguridad e integridad requeridas.

Un DER tiene los siguientes componentes :

1. Entidades.
2. Relaciones.
3. Asociaciones de tipos de objetos.
4. Indicadores de supertipo/subtipo.

A continuación se presenta el diagrama Entidad-Relación de SIGCE, indicándose el tipo de relación establecida por las entidades que lo forman, la cual puede ser uno a uno, uno a muchos o muchos a muchos.

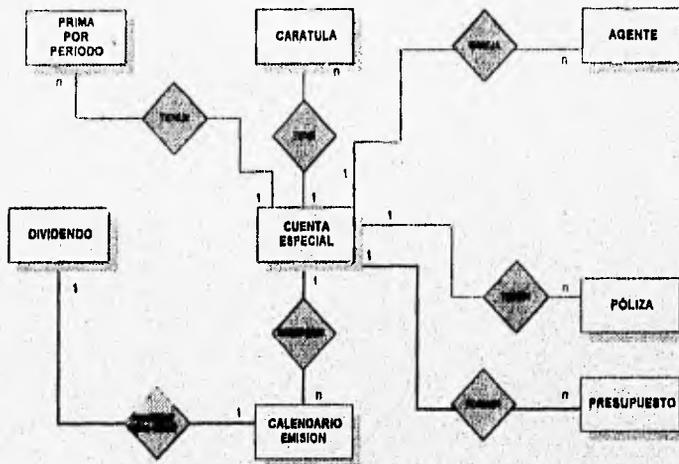


Figura V.1 Diagrama Entidad-Relación de SIOCE

V.2 Diseño de la Base de Datos.

Un sistema de base de datos es un esquema basado en computadoras cuyo propósito general es registrar y mantener información, esta información esta relacionada con elementos significativos de una empresa. Para Grupo Nacional Provincial estos elementos pueden ser cuentas especiales, agentes, presupuestos, etc.

Las ventajas de utilizar un sistema de base de datos son las siguientes :

- Se disminuye o elimina la redundancia de información.
- Se evita la inconsistencia de datos.
- Los datos pueden ser compartidos por distintos usuarios.
- A través del manejador de base de datos se pueden hacer cumplir reglas del negocio.
- Se pueden aplicar restricciones de seguridad.

Conceptos generales

El modelo relacional únicamente soporta la definición de dos tipos de objetos : Entidades y Relaciones. A continuación se describen los elementos de un sistema relacional :

- **Entidad.** Tabla de dos dimensiones con las siguientes propiedades :
Cada columna contiene valores relativos al mismo atributo y cada valor de una columna debe ser simple. Cada columna tiene un nombre distinto .

Cada renglón es distinto. Esto es, un renglón no puede duplicarse para un grupo de columnas seleccionadas como llaves. La secuencia de los renglones no es importante.

Todos los valores no llave deben ser totalmente dependientes de toda llave.

Cada atributo no llave debe depender sólo de la llave de la relación, no de ningún otro no llave.

- **Relación.** Es la correspondencia entre dos entidades.
- **Tupla.** Conjunto de entradas que componen un renglón de la tabla. Es equivalente a una instancia de una entidad.
- **Grado de una tupla.** Número de atributos que tiene una tupla.
- **Cardinalidad.** Número de tuplas de una relación.
- **Domnio.** Conjunto de todos los valores posibles para un atributo.
- **Llave primaria.** Es el atributo para el cual no más de una tupla puede tener el mismo valor.
- **Llave candidata.** Atributo o conjunto de atributos que podrían servir como llave primaria.
- **Llave secundaria.** Todas aquellas llaves candidatas que no se eligieron como llave primaria.
- **Llave foránea.** Atributo de una entidad que es llave primaria de otra entidad.

Normalización de bases de datos.

La normalización de base de datos nos permite evitar guardar información de manera redundante en tanto sea posible. La teoría de la normalización se basa en el concepto de formas normales las cuales fueron definidas por E. Codd. Se establece que entre mayor es el número de forma normal que se aplica, menor será la redundancia de datos. A continuación se presentan las formas normales :

- **Primera forma normal.** Una entidad se encuentra en primera forma normal si los valores que contienen cada uno de sus campos sólo contienen valores atómicos, es decir valores cuyo contenido no es divisible.
- **Segunda forma normal.** Una entidad se encuentra en segunda forma normal si y sólo si está en primera forma normal y cada campo que no pertenece a la llave primaria es totalmente dependiente de ésta.

- **Tercera forma normal.** Una entidad se encuentra en tercera forma normal si se encuentra en segunda forma normal y además para todo instante cada registro que la forma, se compone de un conjunto de atributos que representan la llave primaria, junto con un conjunto de campos mutuamente independientes que contienen características propias de la entidad.
- **Forma de Boyce-Codd.** Esta forma soluciona satisfactoriamente los problemas que surgen cuando una entidad posee varias llaves candidatas que son compuestas que poseen un atributo común. Cuando una entidad solo tiene una llave candidata se encuentra en esta forma.

Tomemos como ejemplo la entidad cuenta especial, inicialmente contamos con el dato de nombre y dirección del cliente al que pertenece la cuenta especial¹, es pertinente recordar que sólo existe una cuenta especial por cliente. La entidad cuenta especial quedaría como sigue :

nombre = FORD DE OCCIDENTE
dirección = LÓPEZ MATEOS 34 GUADALAJARA JALISCO.

Aplicando la primera forma normal tenemos que el atributo dirección se puede descomponer en calle, número, ciudad y estado obteniéndose que la relación cuenta especial queda de la siguiente manera :

nombre = FORD DE OCCIDENTE
calle = LÓPEZ MATEOS
número = 34
ciudad = GUADALAJARA
estado = JALISCO

Tenemos ahora la entidad con atributos atómicos, donde una posible llave candidata sería (nombre), sin embargo, podría haber otra cuenta especial cuyo nombre del cliente fuera "FORD DE OCCIDENTE" (por ejemplo en Colima), con lo cual la llave candidata no sería primaria, entonces se agrega un atributo que es clave_cuenta con el cual se puede identificar (como lo indica la segunda forma normal) de manera única a una cuenta especial. Así la entidad queda como :

clave_cuenta = FOR938
nombre = FORD DE OCCIDENTE
calle = LÓPEZ MATEOS
número = 34
ciudad = GUADALAJARA
estado = JALISCO

La tercera forma normal indica que no debe existir dependencia entre los atributos que no son parte de la llave. En este caso observamos que estado depende de

¹ SIGCE no requiere guardar la dirección del cliente asociado a la cuenta especial ésta se agrega sólo para ejemplificar.

ciudad (suponemos que los nombres de las ciudades no se repiten en los estados), así entonces para eliminar esta dependencia funcional quitamos el atributo estado de la entidad cuenta especial y creamos una entidad llamada ciudad_estado² donde indicamos la relación de ciudades con estados, asumiendo que no hay estados con el mismo nombre. Entonces la entidad cuenta especial quedaría de la siguiente forma :

clave_cuenta = FOR938
nombre = FORD DE OCCIDENTE
calle = LÓPEZ MATEOS
número = 34
ciudad = GUADALAJARA

En tanto que ciudad_estado sería :

ciudad = GUADALAJARA
estado = JALISCO

La entidad cuenta especial no tiene más de una llave candidata, por lo tanto cumple con la forma de Boyce-Codd.

Paso del diagrama Entidad-Relación al diseño de base de datos.

A partir del diagrama entidad relación podemos derivar el diseño de la base de datos, para ello tendremos en cuenta estas reglas :

- Por cada entidad del diagrama Entidad-Relación se creará una tabla en la base de datos.
- Para cada relación que sea del tipo muchos a muchos generar una tabla cuyos campos se formen a partir de los atributos llave de cada relación.
- Marcar como llaves foráneas a aquellos atributos que pertenezcan a entidades hijo y que deben de estar incluidas en los atributos que forman la llave primaria de la entidad padre. Esto se percibe en una relación de 1 a muchos, donde la entidad que está del lado marcado con '1' es la relación padre y la entidad que está del lado marcado como 'n' es la entidad hijo. Validar las llaves foráneas de la entidad hijo a partir de la llave primaria de la relación padre.

Aplicando hasta la tercera forma normal propuesta por Codd³ y en base a las reglas para pasar del diagrama Entidad-Relación al modelo de base de datos tenemos que el esquema de datos del SIGCE se compondrá de la siguiente forma (se consideran los tipos de datos manejados por PARADOX 4.5) :

² Esta entidad no será elaborada en SIGCE, se agrega sólo para ejemplificar.

³ Hay una excepción en el campo descripción de la entidad agente, donde en un solo campo se guardan tres datos: es decir, apellido paterno, apellido materno y nombre de un agente. Se decidió utilizar un solo campo debido a que el agente puede ser una persona única o una compañía, en el caso de una persona única, después del análisis no se consideró importante desglosar el nombre del agente en apellido paterno, materno y nombre.

Entidad : Agente.
Descripción : Intermediario entre el cliente y la aseguradora. Puede manejar varias cuentas especiales.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_agente	*	Identificación del agente.	short number	no nulo.
Descripcion		Nombre o razón social del agente.	alpha(50)	no nulo.
Clave_oficina		Oficina a la que pertenece el agente.	short number	Se valida con el catálogo de oficinas

Entidad : Calendario_emisión.
Definición : Concentrado de fechas de corte por cuenta especial, en las cuales se generará la carátula de estadísticas de siniestralidad.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_cuenta	*	Clave de una cuenta especial.	alpha(3)	Con respecto al catálogo de cuentas especiales.
Fecha_corte	*	Es la fecha en la que se realizará la carátula de estadísticas de siniestralidad.	date	no nulo.
Status_transferencia		Bandera que nos permite conocer si la transferencia de archivos fue realizada a la fecha de corte.	alpha(1)	S o N.
status_revision		Bandera que nos permite conocer si la revisión de la actualización de la cuenta se ha realizado correctamente.	alpha(1)	S o N.
status_impresión_revisión		Bandera que nos permite conocer si la impresión de la revisión se realizó.	alpha(1)	S o N.
status_entrega		Bandera que nos determina si se ha entregado al cliente la última actualización de la estadística.	alpha(1)	S o N.
status_correccion		Bandera que denota si la corrección se ha realizado.	alpha(1)	S o N.
status_impresion_correccion		Bandera que denota si la impresión de la última corrección se ha realizado.	alpha(1)	S o N.

Entidad : Carátula

Definición : Concentrado de información estadística sobre cuentas especiales

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_cuenta	*	Clave de una cuenta especial.	alpha(3)	Con respecto a la entidad calendario emisión.
Fecha_corte	*	fecha de corte.	date	con respecto a la entidad calendario emisión.
Prima_pagada		Monto pagado a la fecha de corte por el servicio de seguro.	currency	>=0. no nulo.
Prima_devengada		Monto consumido de la prima pagada a la fecha de corte.	currency	>=0. no nulo.
Importa_por_siniestro		Monto por la aseguradora debido a siniestros a la fecha de corte.	currency	>=0. no nulo.
Gastos		Monto de gastos generados por siniestros a la fecha de corte.	currency	>=0. no nulo.
Reserva_pendiente		Monto calculado por la aseguradora para cubrir pagos de siniestros.	currency	>=0. no nulo.
Ingreso		Utilidad para la aseguradora a la fecha de corte.	currency	>=0. no nulo.
Tasa_ajuste		Tasa aplicable a gastos indirectos.	number	>=0. <= 100. no nulo.
Tasa_udi		Tasa de uso de instalación.	number	
fecha_elaboración		Fecha del día.	date	no nulo.

Entidad : Cuenta especial.

Descripción : Catálogo de cuentas especiales donde se guardan las características de cada cuenta.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_cuenta	*	Clave de una cuenta especial.	alpha(3)	no nulo.
Nombre		Nombre del cliente al cual pertenece la cuenta especial.	alpha(30)	no nulo.
Tasa_udi		Tasa de uso de instalación.	number	
tasa_gie		Tasa de gastos indirectos.	number	
grupo		Grupo al que pertenece la cuenta.	short number	Validado con el catálogo grupo cuenta.
Fecha_inicio		Fecha de inicio de la vigencia de la cuenta.	date	no nulo.
Fecha_fin		Fecha de finalización de la vigencia de la cuenta.	date	

Entidad: Cuenta_agente.

Descripción: Tabla que relaciona a las cuentas especiales con sus agentes.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_cuenta	*	Clave de una cuenta especial.	alpha(3)	Con respecto al catálogo de cuentas especiales
clave_agente	*	Clave del agente.	short number	Con respecto a la entidad agente.

Entidad: Grupo_cuenta.

Definición: Grupo al que puede pertenecer una cuenta especial.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_grupo	*	Clave del grupo	short number	No nulo.
Descripcion		Descripción del grupo.	alpha(30)	No nulo.

Entidad: Oficina.

Descripción: Dependencia encargada de la emisión de pólizas.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_oficina	*	Clave que identifica una oficina.	short number	no nulo.
Descripcion		Descripción de la oficina.	alpha(30)	no nulo.

Entidad: Póliza.

Descripción: Contrato individual que se establece entre un cliente y la aseguradora. Un conjunto de pólizas de clientes importantes forman una cuenta especial.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_cuenta	*	Clave de una cuenta especial a la cual pertenece la póliza.	alpha(3)	Se valida de acuerdo al catálogo de cuentas especiales.
poliza_inicial	*	Número de póliza inicial.	short number	No nulo. Mayor a 0.
poliza_final	*	Número de póliza final.	short number	No nulo. Mayor a póliza inicial.
fecha_inicial		Fecha de inicio de vigencia del conjunto de pólizas.	date	no nulo.
fecha_final		Fecha de finalización de la vigencia del conjunto de pólizas.	date	

Entidad: Presupuesto
Descripción: Total de primas emitidas durante un año.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_cuenta	*	Clave de una cuenta especial.	alpha(3)	Con respecto al catálogo de cuentas especiales.
Año	*	Año que ampara el monto total de primas.	date	no nulo.
total_prima		Monto total de primas emitidas en el año en curso.	currency	No nulo. Mayor o igual a cero.

Entidad: Prima_perodo
Definición: Producción y cobranza de cuentas especiales.

Atributo	Llave (s/n)	Descripción	Tipo de dato	Validaciones
clave_cuenta	*	Clave de una cuenta especial.	alpha(3)	Con respecto al catálogo de cuentas especiales.
Periodo	*	Fecha a la cual se obtiene el acumulado.	date	No nulo.
prima_emitida		Monto de los recibos de emisión de las primas.	currency	>=0. no nulo.
prima_pagada		Monto cubierto de los recibos de emisión.	currency	>=0. no nulo.
prima_devengada		Monto de la prima utilizada en el período.	currency	>=0. no nulo.

V.3 Establecimiento de niveles de seguridad.

La seguridad de la información dentro de una aplicación es un elemento esencial del proceso de diseño de la misma. El éxito del diseño y el buen funcionamiento del sistema dependen en gran medida de una estructura de seguridad planeada acorde a las necesidades de todos los tipos de usuarios y de la naturaleza misma de la información que manejará dicho sistema.

Todo sistema de información debe contar con medidas de seguridad, pero sobre todo los sistemas "compartidos" o multiusuario, que pueden llegar a permitir el acceso a cualquier usuario de la red local o incluso de una red amplia.

Estas situaciones imponen la necesidad de diseñar medidas con el fin de asegurarse de que únicamente las personas autorizadas tengan acceso a determinados reportes, tablas, formas o consultas. El diseño y evaluación de estos puntos de seguridad, deben hacerse durante el análisis de un sistema, para posteriormente implementarlo en la construcción del mismo.

La seguridad en un sistema puede ser tan compleja como importante sea la información que manejará y los recursos que se puedan asignar para implementarla. En el caso del

SIGCE, por ser un sistema de información consolidada para nivel gerencial, la seguridad de la información se implementará de manera que solo las personas autorizadas a hacer consultas, cambios y reportes lo puedan hacer. Esto es de vital importancia para el funcionamiento del sistema, ya que perdería sus objetivos si se alimentara con basura, y podría ser muy peligroso para la organización de Grupo Nacional Provincial que cualquier persona pudiera obtener alguno de estos reportes o realizar consultas de dicha información.

Dentro de los alcances del SIGCE no está la seguridad de las redes de computadoras de Grupo Nacional Provincial, ni la de los sistemas operativos de las computadoras que integran estas redes, pero para el establecimiento de niveles de seguridad se tomarán en cuenta las facilidades que ofrece Paradox 4.5 para Windows instalado en un ambiente de red Novell Netware 3.12.

Para establecer niveles de seguridad en el Sistema de Información Gerencial para Cuentas Especiales de Grupo Nacional Provincial, se dispone de las facilidades que tiene la seguridad de la red Novell Netware para protección de archivos y subdirectorios mediante nombres de usuario y palabras clave para el acceso de los usuarios, además de la opción de protección con palabras clave para acceso a tablas en Paradox.

Para compartir información con Paradox para Windows en una red, es posible efectuar la instalación en una de tres configuraciones:

Configuración Local:

Cada estación en la red tiene una instalación completa de Windows y Paradox en su disco duro.

Configuración Combinada:

Cada estación en la red tiene instalado Windows en su disco duro, pero todas utilizan una sola instalación en red de Paradox, con las licencias suficientes para el número de usuarios concurrentes. Entonces cada estación corre Windows localmente, pero accesa el programa Paradox y el área de datos en forma compartida.

Configuración de Servidor:

Las estaciones en red pueden o no tener disco duro local, ya que la instalación de Windows y de Paradox son compartidas en el disco duro del servidor, así que cada estación corre Windows y Paradox desde el servidor de archivos y accesa el área de datos en forma compartida.

De las tres configuraciones anteriores, la primera es la más rápida pero la más cara, ya que los programas no son compartidos, pero los archivos ocupan espacio en los discos de las estaciones; y por el contrario, la última es la más barata pero la más lenta, puesto que los archivos de programas ocupan un solo espacio, pero deben ser compartidos.

Para la operación del SIGCE se utiliza la configuración de servidor, que presenta la ventaja de tener Windows, los programas de Paradox y la información del sistema

concentrados en el servidor, con lo cual se ahorra tiempos de instalaciones y espacio en discos duros, además de reducir la posibilidad de diversas versiones sin control.

La estrategia de seguridad que se seguirá para el SIGCE es la siguiente:

La instalación de los programas de Paradox en la red deberá ser en el siguiente directorio del servidor de archivos Novell:

VOL_NAME:\PDOXWIN

Dicho directorio tendrá atributos de sólo lectura para todos los usuarios del sistema.

El área de datos compartidos para Paradox será en el siguiente directorio de Novell:

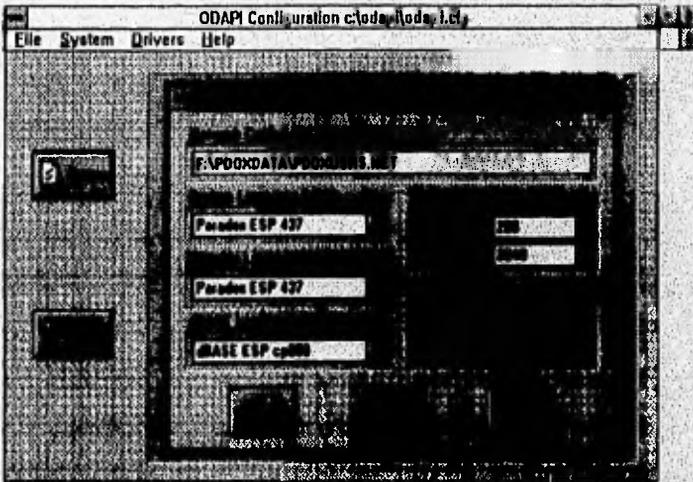
VOL_NAME:\SIGCE

En este directorio se encontrará el archivo de control PDOXUSRS.NET, el cual es indispensable para que los usuarios compartan información. Sus atributos serán de lectura, escritura y creación de archivos para todos los usuarios del sistema.

El área de intercambio con otras aplicaciones (ODAPI), estará en el directorio:

VOL_NAME:\ODAPI

con atributos de solo lectura para todos los usuarios del sistema. Este espacio se debe definir mediante la Utilidad de Configuración ODAPI, como se muestra en seguida:



El directorio privado de cada usuario ("Private Directory") será establecido en cada estación como el directorio hogar de cada usuario en Novell:

C:\PRIVATE en el disco duro de cada estación.

Los archivos propios del SIGCE estarán organizados bajo un solo directorio:

VOL_NAME\SIGCE

Se identificaron los siguientes grupos de usuarios de acuerdo a su utilización del sistema:

Grupo de funcionarios:

Formado por directores y subdirectores de la Dirección de Cuentas Especiales que harán uso de formas de consulta de información consolidada para tener elementos para la toma de decisiones. Este grupo no podrá modificar o eliminar la información contenida en la base de datos.

Grupo de Ejecutivos:

Formado por los ejecutivos que harán uso directo de los productos del sistema (consultas y reportes en pantalla o impresos de la información consolidada) para tener elementos para la toma de decisiones. Este grupo no podrá modificar o eliminar la información contenida en la base de datos.

Grupo de Estadística:

Formado por la Gerencia de Estadística, en el cual habrán personas encargadas de alimentar información al sistema, que no tendrán acceso a los productos de consulta y reportes. Este grupo solo podrá agregar información al sistema o modificarla en caso de errores de captura.

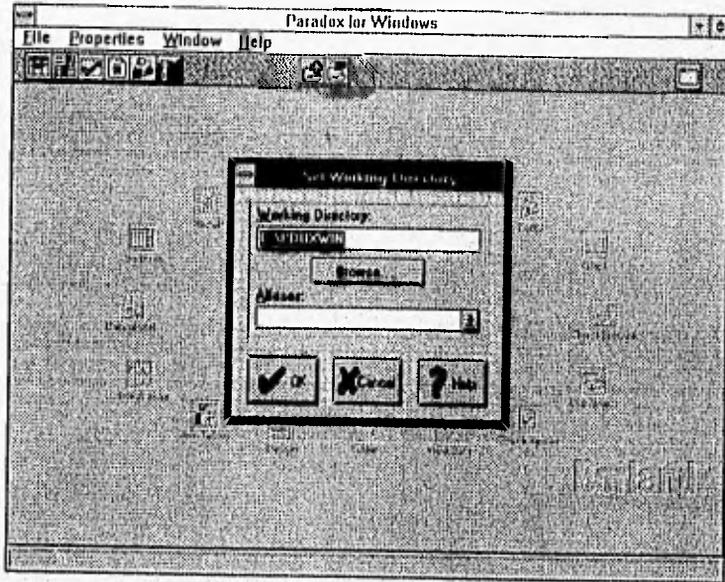
Grupo de Administración:

Formado por solo dos personas de la Gerencia de Estadística para la administración de la base de datos, la seguridad del sistema, los respaldos y la estructura del sistema. Este grupo tendrá la capacidad de agregar, modificar, borrar y reestructurar cualquier elemento de la base de datos y además será el medio de comunicación con los administradores del servidor Novell en el que se instalará el sistema, para cualquier reestructura o respaldo de archivos, administración de usuarios y sus privilegios.

El directorio de trabajo de todos los usuarios del sistema será:

VOL_NAME\SIGCE

Para especificar el directorio de trabajo de cada usuario se utiliza la opción Working Directory de la opción File del menú principal como se muestra en la siguiente figura:



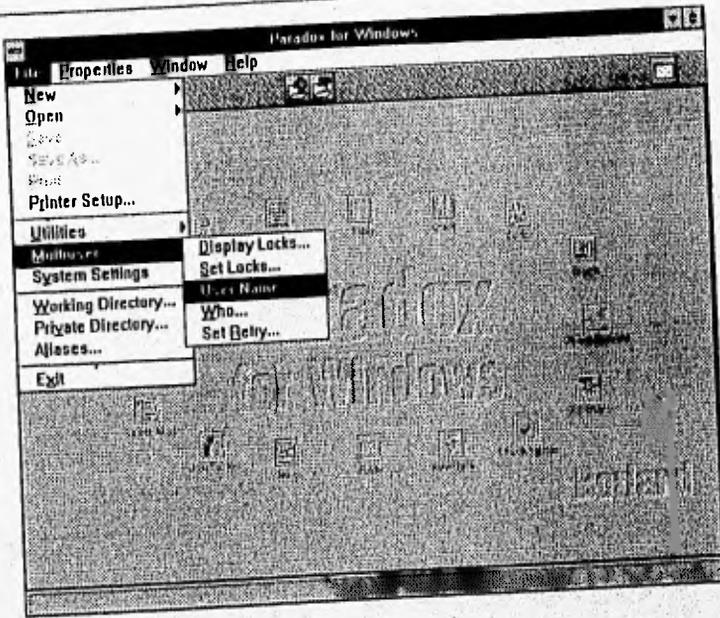
Los privilegios de cada grupo sobre cada directorio se muestran en la siguiente matriz:

GRUPO	ACTUALIZACIÓN	CONSULTA	REPORTES	UTILERIAS
FUNCIONARIOS				
EJECUTIVOS				
ESTADÍSTICA				
ADMINISTRACIÓN				

Para implementar estos grupos y privilegios en el servidor Novell Netware se requiere tener un usuario con privilegio de supervisor, lo cual no está a cargo del área de Cuentas Especiales, ya que esta es una área usuaria de la red. Por lo tanto, el requerimiento de grupos, usuarios, privilegios, instalación de software y estructura de directorios compartidos se hará a los administradores de la red Novell; realizando pruebas y haciendo correcciones posteriormente.

Borland Paradox para Windows en su versión 4.5, tiene las siguientes facilidades para manejo de información en forma compartida o multiusuario:

En la opción File del menú principal, se encuentra el submenú Multiuser, dentro del cual existen las siguientes opciones:



Display Locks: Despliega los bloqueos que en el momento se están llevando a cabo por diferentes usuarios sobre tablas o registros determinados.

Set Locks: Efectúa bloqueos sobre registros o tablas determinadas, registrados como bloqueos del usuario actual.

User Name: Despliega el nombre del usuario actual (el mismo nombre de usuario que en Novell).

Who: Despliega los nombres de los usuarios en Novell que se encuentran utilizando la instalación compartida de Paradox en el servidor de archivos.

Set Retry: Para establecer el tiempo en segundos que una consulta debe esperar al encontrar un bloqueo por parte de otro usuario.

Todas estas opciones son útiles para la compartición de la base de datos en un ambiente multusuario.

Otra opción útil para aumentar la seguridad es el establecimiento de palabras clave para acceder a las diferentes tablas del sistema. Esto se puede llevar a cabo mediante la opción Utilities/Passwords del menú File.

V.4 Elaboración de la carta de estructura de procesos.

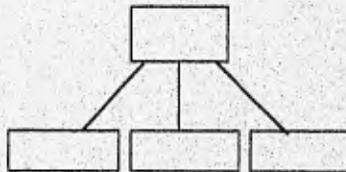
Dentro de una tarea individual, la computadora opera de una manera no sincronizada; solo se puede llevar a cabo una actividad a la vez. El modelo más común de organización de la actividad en una sola unidad sincronizada es el diagrama de estructura. La carta de estructura muestra la partición del sistema en módulos y la relación jerárquica entre estos. Además muestra los flujos de datos y control entre los módulos.

Elementos de una carta de estructura.

- Un rectángulo con un nombre inscrito para indicar un módulo. El nombre indica la función del mismo.



- Líneas que indican la liga entre módulos (llamadas a módulos).



- Flechas que indican el flujo de datos y de control respectivamente.

Un módulo representado como:



significa que es un módulo predefinido:

- El nombre del módulo debe resumir los nombres de sus subordinados inmediatos o resumir su función y las funciones de sus subordinados inmediatos.

Atributos básicos de los módulos.

Un módulo tiene cuatro atributos básicos:

Entrada: Los datos que le pasa quien lo invoca.

Salida: Los datos que regresa a quien lo invoca.

Función: Lo que hace a sus datos de entrada para producir sus datos de salida.

Mecánica: Como realiza su función, es decir, su lógica.

Datos internos: Su propio espacio de trabajo, es decir, las variables locales.

Las figura V.2 presenta la carta de estructura principal del SIGCE. A partir de la figura V.3 y hasta la figura V.6 se muestran las cartas de estructura de cada módulo de SIGCE. En base a estas cartas, se desarrollarán los módulos de programación.

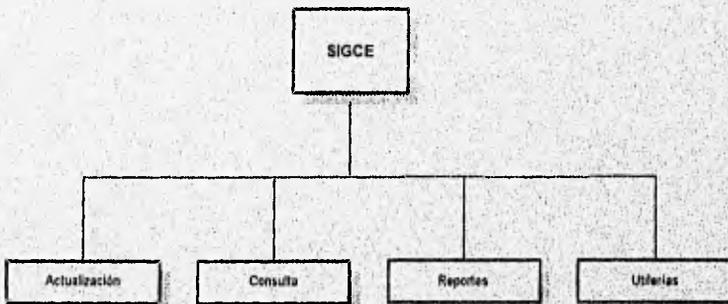


Figura V.2. Carta de estructura de SIGCE. Módulos principales.

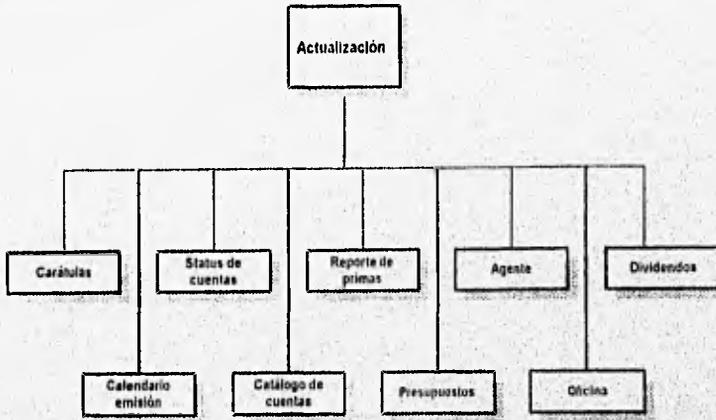


Figura V.3. Carta de estructura de SIGCE. Módulo de actualización.

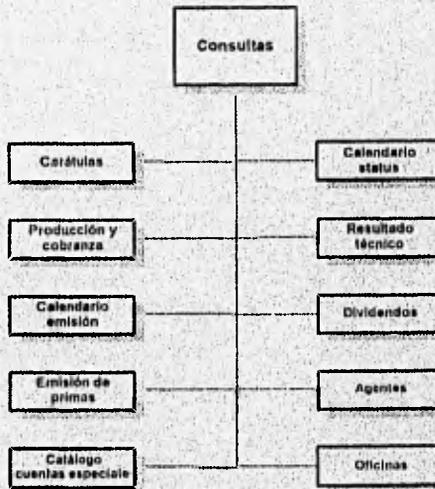


Figura V.4. Carta de estructura de SIGCE. Módulo de consulta.

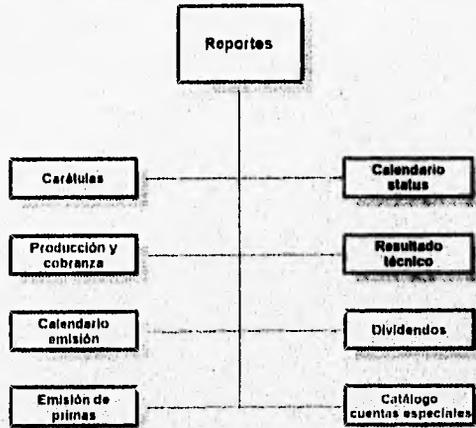


Figura V.5. Carta de estructura de SIGCE. Módulo de reportes.

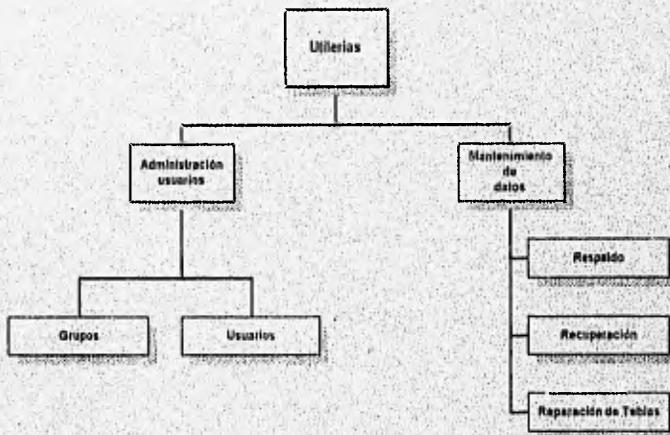


Figura V.6. Utilerías de SIGCE

Para el momento de programar cada módulo debemos tomar en cuenta los siguientes dos conceptos:

Cohesión. Grado en el cual los componentes de un módulo son necesarios y suficientes para llevar a cabo una sola función bien definida. Esto significa no fragmentar los procesos esenciales en módulos y no juntar en un mismo módulo procesos no relacionados.

Acoplamiento. Grado en el cual los módulos se interconectan o se relacionan entre ellos. Entre más fuerte sea el acoplamiento entre módulos en un sistema, más difícil es implantarlo y mantenerlo. Un módulo dado no debe modificar la lógica interna o los datos de algún otro módulo.

V.5 Diseño de la Interfaz con el usuario.

El primer paso para desarrollar una aplicación para Windows es planear que elementos gráficos contendrá la forma que el usuario va a operar, es decir, realizar el diseño de las pantallas respetando la filosofía de Windows. Esta actividad se lleva a cabo contestando una serie de preguntas como ¿Qué menús se desean? ¿Cómo ve la ventana en la que se ejecuta la aplicación? ¿Cuántas ventanas debe haber?, ¿Debe poder modificar el usuario el tamaño de las ventanas? ¿Dónde se colocarán los botones de ordenes, los botones dónde se hará clic para activar aplicaciones?, es decir los objetos que contendrá la aplicación.

Programar una aplicación en Paradox es como diseñar un pantalla, literalmente se dibuja la interfaz del usuario, es como si se utilizara un programa de dibujo. Además cuando se ha terminado de dibujar la interfaz, los botones de ordenes, cajas de texto y otros controles colocados en la ventana, reconocerán automáticamente acciones de usuario, tales como el movimiento del ratón y los clics de los botones.

Después de diseñar la interfaz de usuario se procede a programar. En los lenguajes de programación convencionales el flujo del programa se ejecutan secuencialmente (de arriba a abajo), es decir, la ejecución comienza en la primera línea y se desplaza con el flujo del programa a las distintas partes según se necesite. Un programa en Paradox funciona de un modo totalmente diferente. El núcleo de un programa en Paradox es un conjunto de diferentes partes de código que son activadas por eventos, y que solo responden a los sucesos que se les ha indicado que reconozcan. La mayor parte del código de programación en Paradox indica al programa el modo de responder a determinados sucesos, como el clic del ratón los cuales se denominan procedimientos de suceso.

A continuación se resumen los pasos a seguir para diseñar una aplicación en Paradox:

1. Personalizar la ventana que utiliza el usuario.
2. Decidir los sucesos que deben reconocer los controles en la ventana.
3. Escribir los procedimientos de sucesos para éstos.

4. Escribir cualquier función que necesiten los procedimientos de suceso para realizar su labor.

Esto es lo que sucede cuando se ejecuta la aplicación:

1. Paradox supervisa cada uno de los objetos que constituyen la forma, captando que suceso aplica a cada uno, por ejemplo, movimientos de ratón, clics, pulsaciones de teclas y demás.
2. Cuando Paradox reconoce un suceso, examina la aplicación para comprobar si se ha escrito un procedimiento para ese suceso.
3. Si se ha escrito un procedimiento de suceso, Paradox ejecuta el código de procedimiento y vuelve al paso número 1.
4. Si no se ha escrito un procedimiento de suceso, Paradox espera al siguiente suceso y vuelve al paso 1.

Este proceso se repite continuamente hasta que se finaliza la aplicación. Normalmente un suceso debe ocurrir para que Paradox haga algo.

Aunque se puede pasar por al alto la filosofía básica de Windows, debería haber una buena razón para hacerlo, ya que si se hace los usuarios del programa, en su mayoría se sentirán muy molestos. Windows está orientado a los gráficos, lo que significa que proporciona una Interfaz Gráfica de Usuario (GUI). Aunque el hardware de gráficos y los modos de video son muy diversos, muchas de las diferencias son manejadas por Windows, esto significa que, en la mayor parte, un programa no se ha de preocupar por el tipo de Hardware gráfico o modo de video que se use.

Aparte de las ventanas estándar, Windows proporciona ventanas de propósito especial, las más comunes de ellas son los menús y los cuadros de diálogo. Un menú es una ventana que contiene una lista en la que el usuario hace una selección. Un cuadro de diálogo es un ventana especial que permite una interacción más compleja con la aplicación que la permitida por el menú y a la cual el usuario debe responder necesariamente. Windows permite (aunque no lo requiere) el uso de iconos e imágenes gráficas de mapas de bits. Un icono es un pequeño símbolo que representa alguna función o programa que se puede activar llevando el ratón sobre el icono y pulsando dos veces sobre él.

Todas las ventanas tienen un borde que define los límites de las ventanas y se usa para mover o cambiar el tamaño de la ventana. En la parte superior de la ventana hay diversos elementos. En la parte superior izquierda se encuentra el icono del menú del sistema (o cuadro, como se denomina normalmente). Pulsando sobre ese cuadro se muestra el menú estándar del sistema. A la derecha del cuadro del menú del sistema se encuentra la barra de título de la ventana. En el extremo derecho están los cuadros de maximización y minimización. En el área cliente (la parte interior) es la parte de la ventana donde se desarrollarán las actividades del programa. Muchas ventanas también tienen barras de desplazamiento horizontal y vertical que se usan para desplazar el texto a través de la ventana.

A continuación se muestran dos pantallas apegadas a la filosofía de Windows realizadas en Paradox como parte del interfaz gráfico de la aplicación de usuario.

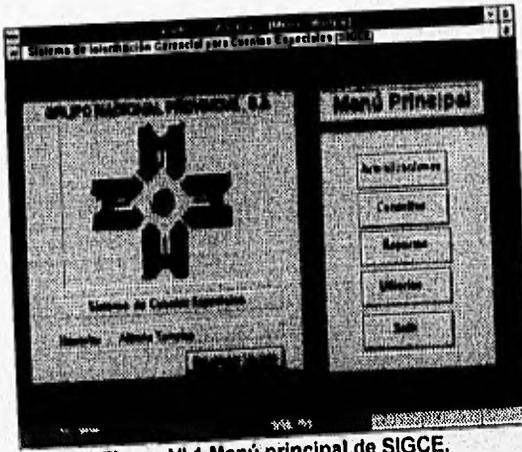


Figura VI.1 Menú principal de SIGCE.

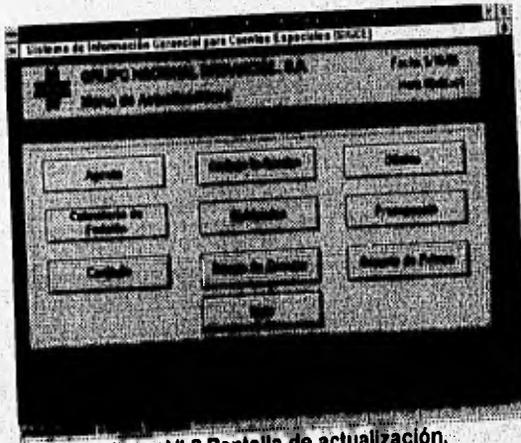


Figura VI.2 Pantalla de actualización.

VI. Aspectos de la codificación del sistema.

VI. Aspectos de la Codificación del sistema.

VI.1 Descripción de Paradox 4.5 como herramienta de desarrollo.

Paradox es un gestor de información que posee su propia base de datos para brindar al usuario posibilidades de desarrollo en aplicaciones de software (DBMS). Existen varias versiones sobre este producto, la utilizada en este proyecto es la versión 4.5 que opera en ambiente Windows.

Algunos usos de Paradox 4.5 para Windows.

En general, algunos de los usos y ventajas más frecuentes de este sistema manejador de bases de datos son los siguientes:

- Programación orientada a objetos que facilita el desarrollo de aplicaciones.
- Representación gráfica de datos.
- Manejo de información en forma de tablas relacionales.
- Cuenta con librerías utilizables con otras aplicaciones Windows.
- Operable en equipos independientes y en ambiente de red.

Requerimientos para operar Paradox.

Paradox 4.5 para Windows puede ser ejecutado en cualquier microcomputadora personal que cuente con las siguientes características:

- Un equipo IBM o computadora compatible con éste que posea un microprocesador Intel 80386 o superior (80486 o Pentium) con 4 megabytes de memoria RAM como mínimo.
- Para utilizar Paradox es requerido tener instalada la versión 3.1 de Windows manufacturado por la corporación Microsoft. La versión 3.0 de Windows no soporta Paradox 4.5.
- Contar con disco duro, que por lo general se encuentra instalado internamente en el equipo. Paradox necesita al menos 20 megabytes de espacio libre en disco para operar correctamente, sin embargo, es recomendable contar con espacio en disco adicional, y éste esta en función del nivel de información que se procese en la base de datos.
- Se requiere un monitor EGA (enhanced graphics adapter) para operar Windows, ya que este no opera con monitores CGA (color graphics adapter).

Nombre y tipos de datos en Paradox para Windows.

El nombre de archivos en Paradox (mnemónico de una tabla) puede contener hasta 8 caracteres incluyendo caracteres alfabético, dígitos del 0 al 9. El nombre de una tabla debe iniciar con un carácter alfabético.

Paradox para Windows acepta y manipula diez diferentes tipos de datos: alfabéticos, numéricos, porcentuales, de tipo fecha, numéricos cortos, de texto libre (memo), de texto libre formateado (formatted memo), gráficos, Ole y binarios.

- **Alfanuméricos (A).** Dato que puede ser cualquier combinación de caracteres válidos. El mínimo de un campo alfabético es 1 carácter y el máximo de 255 caracteres.
- **Numéricos (N).** Dato que incluye números con o sin decimales, con signo o sin él. Se pueden introducir números hasta con 15 dígitos incluyendo decimales. Las cantidades con más de 15 dígitos significativos son redondeados o visualizados en notación científica.
- **Porcentuales (\$).** Es similar al tipo de dato numérico, sin embargo, Paradox para Windows visualiza un signo \$ a la izquierda del número, adiciona comas para separar dígitos en tercias y redondea el número a 2 decimales. Los números negativos son visualizados entre paréntesis.
- **Fecha (D).** Dato que incluye fechas del 1 de enero al 31 de diciembre, y es visualizado en diferentes formatos (mm/dd/aa, dd/mm/aa, dd.mm.aa, etc.). El sistema posee un calendario interno para validar automáticamente fechas inhábiles.
- **Numérico corto (S).** Dato usado por usuarios que necesitan aprovechar espacio en disco. Este tipo de dato puede contener solo números enteros en el rango de -32,767 a 32,767.
- **Texto libre (M).** Dato utilizado para almacenar grandes cantidades de texto con fines de documentación cuyo tamaño no está limitado. Un campo memo es almacenado en un archivo con extensión MB que es abierto automáticamente por Paradox cuando se desea acceder al contenido del campo.
- **Texto libre formateado (F).** Es similar al tipo de dato anterior, sin embargo, tiene mayor flexibilidad ya que el texto puede ser formateado. Este tipo de dato no puede ser soportado como un campo en una tabla.
- **Gráficos (G).** Dato que contiene imágenes gráficas como fotografías. Este tipo de dato en una tabla puede requerir grandes cantidades de espacio en disco.
- **OLE (Object Linking and Embedding - O).** Contiene objetos provenientes de otras aplicaciones Windows que pueden ser utilizados como atributos en una tabla de Paradox.

- **Binarios (B).** Campo que almacena información binaria (unos y ceros). Cuando utilizamos campos de tipo binario, debemos explotarlos a través del ObjectPAL, el lenguaje de programación de Paradox para Windows.

Objetos en Paradox para Windows.

Paradox 4.5 cuenta con un lenguaje de programación orientado a objetos llamado ObjectPAL. Es muy importante entender los conceptos involucrados en esta nueva plataforma de programación, los más importantes son:

- **Clase:** Una clase es una estructura para la creación de objetos, definiendo las operaciones y variables asociadas a ese objeto.
- **Objeto:** Es un elemento de una clase. Cada objeto posee uno o más parámetros llamados variables modelo. Un objeto puede recibir mensajes y ejecutar ciertas operaciones (métodos). Las variables modelo sólo pueden recuperarse o modificar su contenido mediante el envío de mensajes al objeto. Un objeto se crea mediante la llamada a una función especial que depende de la clase de objeto a crear.
- **Mensaje:** Es la forma por la cual se solicita a un objeto la ejecución de una determinada tarea. El envío de un mensaje a un objeto devuelve un valor. Para enviar un mensaje hay que especificar el nombre del objeto. El selector debe tener el mismo nombre del método a ser ejecutado.
- **Método:** Es la operación que se ejecutará en el objeto en respuesta a un mensaje enviado al mismo.
- **Selector:** Es el elemento que accionará el método a ser ejecutado.
- **Variables modelo.** Son los atributos o datos que forman un objeto. Cada objeto tiene su propio conjunto de variables modelo, con sus valores iniciales. Las variables modelo que tienen su contenido accesible se llaman VME (variables modelo exportables)
- **Función de creación:** Es la función que crea el objeto de una determinada clase.

El término objeto en Paradox para Windows se define como diferentes cosas. Los siguientes son ejemplos de algunos objetos manejados en Paradox.

- Una tabla de datos.
- Un campo o atributo pertenecientes a una tabla.
- Un registro en una tabla.
- Un reporte.
- Una forma.
- Un script. (Documento electrónico que almacena una serie de comandos de Paradox).

Un objeto puede ser temporal o permanente. Paradox para Windows genera diferentes tipos de objetos temporales durante la sesión de un usuario. El uso de objetos nos permitirá ahorrar código y elaborar aplicaciones en poco tiempo.

VI.2 Implantación de Estándares de codificación.

Sin tomar en cuenta el lenguaje de programación que se use, hay cuestiones comunes que todo programador debe tomar en cuenta, por ejemplo las siguientes :

- **Productividad.** La productividad se resume en escribir más software más rápidamente. Por lo anterior se deben alentar los lenguajes y técnicas de programación que promuevan la productividad.
- **Eficiencia.** La eficiencia es sumamente importante para aplicaciones que deben tener una respuesta inmediata a peticiones de usuario. Sistemas que procesan grandes volúmenes de datos se preocupan por minimizar la cantidad de tiempo de CPU requerido por el programas, también puede ser importante minimizar la utilización de memoria así como el disco.
- **Facilidad de corrección.** Se prefieren lenguajes estructurados tipo Pascal puesto que facilitan el seguimiento de la lógica del programa. El uso de lenguajes orientados a objetos puede ser permitir fáciles correcciones al software en tanto que mantengan la utilización de las estructuras de control de la programación estructurada.
- **Portabilidad.** La portabilidad busca que un mismo código pueda ejecutarse en distintas plataformas. El lenguaje C es el más portable actualmente, el resto tiene poca portabilidad.
- **Elaborar 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. Un módulo cuya programación abarque demasiado debe dividirse en submódulos.
- **Sencillez en la programación.** Se deben escribir programas sencillos que el programador promedio pueda entender, por ejemplo se deben evitar el uso de reglas booleanas demasiado complejas.

Estándares para la definición de objetos en Paradox.

Como mencionamos anteriormente Paradox maneja diversos tipos de objetos tales como tablas, índices, formas, reportes, etc. Para cada uno de estos tipos de objetos definiremos la manera de nombrarlos, con la finalidad de poderlos controlar fácilmente durante la programación.

- **Tabla.** Los nombres de las tablas en Paradox 4.5 quedan limitados a ocho caracteres puesto que cada una de éstas es almacenada en un archivo individual. Tomaremos la regla de tomar los ocho primeros caracteres del nombre de nuestra entidad para determinar el nombre de la tabla. Por ejemplo la entidad "agente" será representada

con una tabla llamada de la misma forma, es decir, "agente" , en tanto que a la entidad "cuenta_agente" le corresponderá una tabla llamada "cuenta_a".

- **Campo.** El tamaño del nombre de un campo en Paradox está restringido a veinticinco caracteres. Se tomarán los primeros veinticinco de los nombres de atributos definidos en cada entidad.
- **Forma.** Los nombres de las formas se encuentran limitados al igual que el de las tablas a ocho caracteres. Si la forma involucra el manejo de una sola tabla seguiremos las mismas reglas que se mencionaron para definir el nombre de una tabla, si intervienen dos o más tablas tomaremos tantos caracteres iniciales del nombre de cada tabla como sea posible, considerando el caso especial de las tablas llamadas "cuenta_es" y "cuenta_ag" las cuales deberán ser diferenciadas.
- **Reporte.** Para los nombres de reportes aplican exactamente las mismas reglas que para los nombres de formas.
- **Índice secundario.**¹ El nombre de los índices secundarios está limitado a veinticinco caracteres; a partir de esta consideración formaremos este nombre con base al nombre de la tabla agregando un guión y la primera letra del nombre del campo ó en caso de que el nombre del campo tenga un guión bajo (por ejemplo clave_oficina), consideraremos la primera letra del nombre del campo junto con la primera letra del nombre del campo que está después del guión bajo. Por ejemplo si definimos un índice secundario sobre el campo descripción de la tabla agente será "agente_d" y si lo hacemos sobre clave_oficina será "agente_co".
- **Referencia de integridad.** Cuando definimos una relación padre-hijo en Paradox se nos solicita un nombre para ella, cuyo tamaño está restringido a 30 caracteres. Las reglas para definir estos nombres son las mismas que aplican para los nombres de índices secundarios.

Estándares de programación en Paradox 4.5

Como se mencionó Paradox cuenta con un lenguaje de programación orientado a objetos llamado ObjectPAL mediante el cual especificamos las acciones que ocurrirán cuando ocurra un evento asociado a un objeto dado.

El código fuente de ObjectPAL se compone de una sección de variables y el resto son sentencias sencillas o manejadas por medio de estructuras de control como if..then..else..endif, switch..endswitch, etc. Es conveniente acotar que Paradox no distingue entre mayúsculas y minúsculas.

La sentencia sencilla en Paradox tiene la siguiente forma :

variable_tipo_objeto.acción(parámetros)(parámetros)

¹ Es conveniente aclarar que los índices principales o primarios Paradox los crea automáticamente al definir un campo como clave de la tabla, por lo tanto no es posible controlar su nombre.

Las siguientes reglas serán consideradas cuando codifiquemos en Paradox :

- Las palabras reservadas serán escritas con mayúsculas.
- Las variables serán escritas en minúsculas.
- En una sentencia la variable_tipo_objeto y los parámetros serán codificados en minúsculas en tanto que la acción será especificada en mayúsculas. Por ejemplo, el siguiente código abre la forma llamada "agente" :

```
agente.OPEN("agente.fsl")
```

En base a las reglas para crear nombres de objetos y a los estándares de codificación anotados arriba se elaborará el código fuente del SIGCE.

VI.3 Pruebas modulares.

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

En esta sección haremos referencia a la pruebas modulares, es decir, a aquellas pruebas que tienen por propósito conocer el rendimiento y eficacia de cada una de las partes que forman la aplicación global. En el siguiente capítulo presentaremos las pruebas de rendimiento las cuales involucran a la aplicación de manera completa.

Existen ciertas reglas que aplican a las pruebas de código :

- La prueba es un proceso de ejecución de un programa con el objetivo de descubrir algún error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

A medida que se van recopilando y evaluando los resultados de la prueba, comienza a vislumbrarse una medida cualitativa de la calidad y fiabilidad del software. Si se encuentran con regularidad serios errores que requieren modificaciones en el diseño, la calidad del software queda en entredicho, siendo necesarias pruebas posteriores. Si, por otro lado, el funcionamiento parece ser adecuado y los errores que se encuentran son fácilmente corregibles, se puede concluir que el software es aceptable.

Un producto de software puede ser probado en alguna de las siguientes dos formas : 1) conociendo la función específica para la que fue diseñado se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa sin importar su forma interna; 2) conociendo el funcionamiento interno del producto y dándole seguimiento paso a paso para asegurar que cumplirá con los requisitos de la

especificación. El primer enfoque es conocido como *prueba de caja negra* y el segundo como prueba de *caja blanca*, estos conceptos se definen a continuación.

VI.3.1 Pruebas de caja negra.

La prueba de la caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del producto, esto es los casos de prueba pretenden demostrar que las funciones del software son adecuadas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. La prueba de caja negra examina aspectos del modelo funcional del sistema sin tener en cuenta la estructura interna del software.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de validación.
- Errores en formato de datos.
- Errores en accesos a bases de datos.
- Errores de inicialización o terminación.

VI.3.2 Prueba de caja blanca.

La prueba de caja blanca se basa en un minucioso examen de los procedimientos internos. Se comprueban los caminos lógicos del código proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado coincide con el esperado.

Mediante la prueba de caja blanca se obtienen casos de prueba que garanticen que se ejecuten los siguientes elementos del programa :

- Cuando menos una vez todos los caminos independientes de cada módulo.
- Todas las decisiones lógicas en sus dos posibilidades verdadero y falso.
- Se ejecuten todos los ciclos hasta llegar a sus límites.
- Se manejen todas las estructuras de datos del módulo.

Las razones para considerar las pruebas de caja blanca además de las pruebas de caja negra (descritas anteriormente) se destacan a continuación :

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa. Los errores tienden a reproducirse en las funciones poco usadas.
- Posibilidad de encontrar errores tipográficos. Aunque un programa haya compilado exitosamente en ocasiones se asignan variables de manera errónea, debido a haber equivocado el nombre de las mismas en ciertas asignaciones.

Pruebas de caja negra para SIGCE

Para ejemplificar el uso de la técnica de caja negra tomaremos de SIGCE tres ejemplos². Del menú de actualizaciones consideraremos el módulo de "actualización de cuentas especiales", del menú de consultas probaremos el módulo de "consulta de agentes" y del menú de reportes se analizará el módulo de "reporte de carátulas".

Módulo de "actualización de cuentas especiales."

Objeto	Descripción de la prueba	Observaciones
clave	No puede ser modificada a menos que sea un nuevo registro.	Cumple.
clave	En un alta no puede ser nula.	Cumple.
nombre	No pueden ser modificados a menos que sea un nuevo registro.	Cumple. El mensaje que indica que no se puede modificar es incorrecto. Se procedió a corregirlo.
nombre	En un alta no puede ser nulo.	Cumple.
grupo	En un alta no puede ser nulo.	Cumple.
grupo	Debe validarse contra la tabla grupo_cu.	Cumple.
grupo	Debe aceptar solo números.	Cumple.
oficina	En un alta no puede ser nulo.	Cumple.
oficina	Debe validarse contra la tabla oficina.	Cumple.
oficina	Debe aceptar solo números.	Cumple.
agente	En un alta no puede ser nulo.	Cumple.
agente	Debe validarse contra la tabla agente.	Cumple.
agente	Debe aceptar solo números.	Cumple.
tasa udi	Puede ser nula.	Cumple.
tasa udi	Debe aceptar solo números.	Cumple. No despliega letrero que indique "carácter inválido." Corregido.
tasa reserva	Puede ser nula.	Cumple.
tasa reserva	Debe aceptar solo números.	Cumple. No despliega letrero que indique "carácter inválido." Corregido.
tasa gia	Puede ser nula.	Cumple.

² Teniendo en cuenta que se aplicarán pruebas de caja negra y caja blanca a cada módulo de S.I.G.C.E.

tasa_gia	Debe aceptar solo números.	Cumple. No despliega letrero que indique "carácter inválido." Corregido.
póliza_inicial	En un alta no puede ser nula.	Cumple.
póliza_inicial	Debe aceptar solo números.	Cumple.
póliza_inicial	No pueden ser modificada a menos que sea un nuevo registro.	NO cumple. Error corregido.
póliza_final	Puede ser nula.	Cumple.
fecha_inicio	Se debe capturar en un formato dd/mm/aa.	Cumple.
fecha_inicio	No puede ser nula.	Cumple.
fecha_fin	Se debe capturar en un formato dd/mm/aa.	Cumple.
fecha_fin	Puede ser nula.	Cumple.
fecha_fin	Si no es nula debe ser mayor o igual a la fecha inicio.	Cumple.
botón alta	Debe insertar un nuevo registro y colocar en los campos clave, nombre, grupo y oficina del nuevo registro los valores del registro anterior. Al terminar de capturar el campo fecha_fin debe aparecer un diálogo de confirmación. Si ya existe el registro (la llave es clave+póliza inicial) debe aparecer un mensaje de error y se cancela el registro.	Cumple.
botón baja	Borra el registro actual. Aparece un diálogo de confirmación.	Cumple.
botón cambio.	Muestra el diálogo de búsqueda de Paradox y ubica un registro para modificarlo. Después de teclear <enter> en el campo fecha_fin debe aparecer un diálogo de confirmación. Doble clic para ver el siguiente registro que cumple con la condición de búsqueda.	Cumple.
botón cancelar	Cancela el alta o el cambio hecho al registro.	Cumple.
botón salir	Regresa al menú de actualizaciones.	Cumple.
forma	La forma puede cerrarse solo con el botón de salir.	Cumple.

Módulo de "consulta de agentes".

Objeto clave	Descripción de la prueba	Observaciones
	No puede ser modificado.	Cumple.

descripción	No puede ser modificado	Cumple.
oficina	No puede ser modificado.	Cumple.
botón primer registro	Coloca el apuntador de la tabla al inicio.	Cumple.
botón registro anterior	Retrocede el apuntador de la tabla un registro.	Cumple.
botón registro siguiente	Avanza el apuntador de la tabla un registro.	Cumple.
botón último registro	Coloca el apuntador de la tabla al final.	Cumple.
botón buscar	Muestra el diálogo de búsqueda de Paradox y se ubica en algún registro de acuerdo a la condición establecida.	Cumple.
botón salir	Regresa al menú de consultas.	Cumple.
forma	La forma puede cerrarse solo con el botón de salir.	Cumple.

Módulo de "reporte de carátulas".

Objeto	Descripción de la prueba	Observaciones
clave	Debe validarse contra la tabla carátula.	Cumple. No se despliega mensaje de error cuando la clave que se teclea no está en la tabla carátula. Corregido.
fecha de corte	Validar formato dd/mm/aa.	No cumple. Formato corregido.
fecha de corte	Debe validarse contra la tabla carátula.	Cumple. No se despliega mensaje de error cuando la clave que se teclea no está en la tabla carátula. Corregido.
fecha de corte	Validar que se introduzcan solo números.	Cumple. Falta mensaje de error. Corregido.
botón aceptar	Debe validar que ninguno de los dos parámetros sea nulo.	Cumple.
botón aceptar	Debe validar que los parámetros existan en la tabla carátula.	No cumple. Función implementada.
forma	La forma puede cerrarse solo con el botón de salir.	Cumple.

Pruebas de caja blanca para SIGCE

Las pruebas de caja blanca como mencionamos anteriormente se hacen directamente sobre el código, probando todos los caminos por los cuales la ejecución se lleva a cabo, para ello nos apoyaremos en el depurador de Paradox. Debido a la extensión del código generado por el depurador a manera de ejemplo nos concentraremos en el módulo de "actualización de cuentas" uno de los más extensos del sistema.

Módulo de "actualización de cuentas".

*** Apertura de la forma.

```
#FormData!:Entering method open.
#FormData!:open:2=> IF eventInfo.isPreFilter()
#FormData!:open:8=> salir = FALSE
#FormData!:Leaving method open.
:Entering method open.
:open:2=> self.ACTION(databeginedit)
#FormData!:Entering method status.
#FormData!:status:4=> SWITCH
#FormData!:status:23=> eventInfo.SETSTATUSVALUE("Modo de edición.")
#FormData!:Leaving method status.
:Entering method arrive.
:arrive:4=> MAXIMIZE()
:arrive:5=> HIDESPEEDBAR()
:Leaving method arrive.
```

*** Apuntar con el mouse el botón de alta.

```
:Entering method mouseEnter.
:mouseEnter:2=> MESSAGE ("Nuevo registro.")
:Leaving method mouseEnter.
:Entering method mouseExit.
:mouseExit:2=> MESSAGE (" ")
:Leaving method mouseExit.
```

*** Oprimiendo botón de alta.

```
:Entering method pushButton.
:pushButton:10=> IF DMGET("cuenta","clave_cuenta",v_clave) AND
DMGET("cuenta","nombre",v_descr)
:pushButton:13=> IF self.RECORDSTATUS("Modified") THEN
:pushButton:16=> self.ACTION(fieldfirst)
:pushButton:17=> self.ACTION(datainsertrecord)
#FormData!:Entering method status.
#FormData!:status:4=> SWITCH
#FormData!:status:10=> eventInfo.SETSTATUSVALUE("Nuevo registro.")
#FormData!:Leaving method status.
:pushButton:18=> IF NOT v_clave.ISBLANK() THEN
DMPUT("cuenta","clave_cuenta",v_clave)
:pushButton:19=>
:pushButton:21=> IF NOT v_descr.ISBLANK() THEN
DMPUT("cuenta","nombre",v_descr)
:pushButton:22=>
:pushButton:24=> IF NOT v_grupo.ISBLANK() THEN
DMPUT("cuenta","clave_grupo",v_grupo)
:pushButton:25=>
:pushButton:27=> IF NOT v_ofici.ISBLANK() THEN
:pushButton:28=> DMPUT("cuenta","clave_oficina",v_ofici)
:Leaving method pushButton.
```

```
Clave_cuenta:Entering method canDepart.
Clave_cuenta::canDepart:2=> IF self.ISBLANK() THEN
Clave_cuenta:Leaving method canDepart.
Clave_cuenta:Entering method changeValue.
Clave_cuenta::changeValue:2=> IF NOT self.RECORDSTATUS("New") THEN
Clave_cuenta:Leaving method changeValue.
Nombre:Entering method canDepart.
Nombre::canDepart:2=> IF self.ISBLANK() THEN
Nombre:Leaving method canDepart.
Nombre:Entering method changeValue.
Nombre::changeValue:2=> IF NOT self.RECORDSTATUS("New") THEN
Nombre:Leaving method changeValue.
Clave_grupo:Entering method canDepart.
Clave_grupo::canDepart:2=> IF self.ISBLANK() THEN
Clave_grupo:Leaving method canDepart.
#FormData!:Entering method error.
#FormData!::error:3=> SWITCH
#FormData!::error:12=>      MSGINFO("Aviso","Valor inválido. Ctrl+Barra
Espaciadora para ver ayuda.")
#FormData!:Entering method status.
#FormData!::status:4=> SWITCH
#FormData!::status:6=>      eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra
Espaciadora para ver ayuda")
#FormData!:Leaving method status.
Clave_grupo:Entering method canDepart.
Clave_grupo::canDepart:2=> IF self.ISBLANK() THEN
Clave_grupo::canDepart:3=>      MSGINFO("Aviso","El grupo no puede ser
nulo.")
&&& ERROR detectado falta indicar eventinfo.SETERRORCODE(cannotdepart)

#FormData!:Entering method status.
#FormData!::status:4=> SWITCH
#FormData!::status:6=>      eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra
Espaciadora para ver ayuda")
#FormData!:Leaving method status.
Clave_grupo::canDepart:4=>      eventinfo.SETERRORCODE(cannotdepart)
Clave_grupo:Leaving method canDepart.
Clave_grupo:Entering method canDepart.
Clave_grupo::canDepart:2=> IF self.ISBLANK() THEN
Clave_grupo:Leaving method canDepart.
#FormData!:Entering method error.
#FormData!::error:3=> SWITCH
#FormData!::error:18=>      MSGINFO("Aviso","Carácter inválido.")
#FormData!:Entering method status.
#FormData!::status:4=> SWITCH
#FormData!::status:6=>      eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra
Espaciadora para ver ayuda")
#FormData!:Leaving method status.

Clave_grupo:Entering method canDepart.
Clave_grupo::canDepart:2=> IF self.ISBLANK() THEN
Clave_grupo:Leaving method canDepart.
Clave_oficina:Entering method canDepart.
Clave_oficina::canDepart:2=> IF self.ISBLANK() THEN
Clave_oficina::canDepart:3=>      MSGINFO("Aviso","La oficina no puede ser
nula.")
#FormData!:Entering method status.
#FormData!::status:4=> SWITCH
```

```
#FormDatal::status:6=> eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra  
Espaciadora para ver ayuda")  
#FormDatal:Leaving method status.  
Clave_oficina::canDepart:4=> eventinfo.SETERRORCODE(cannotdepart)  
Clave_oficina:Leaving method canDepart.  
Clave_oficina:Entering method canDepart.  
Clave_oficina::canDepart:2=> IF self.ISBLANK() THEN  
Clave_oficina:Leaving method canDepart.  
#FormDatal:Entering method error.  
#FormDatal::error:3=> SWITCH  
#FormDatal::error:18=> MSGINFO("Aviso", "Carácter inválido.")  
#FormDatal:Entering method status.  
#FormDatal::status:4=> SWITCH  
#FormDatal::status:6=> eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra  
Espaciadora para ver ayuda")  
#FormDatal:Leaving method status.  
Clave_oficina:Entering method canDepart.  
Clave_oficina::canDepart:2=> IF self.ISBLANK() THEN  
Clave_oficina:Leaving method canDepart.  
Clave_agente:Entering method canDepart.  
Clave_agente::canDepart:2=> IF self.ISBLANK() THEN  
Clave_agente::canDepart:3=> MSGINFO("Aviso", "El agente no puede ser  
nulo.")  
#FormDatal:Entering method status.  
#FormDatal::status:4=> SWITCH  
#FormDatal::status:6=> eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra  
Espaciadora para ver ayuda")  
#FormDatal:Leaving method status.  
Clave_agente::canDepart:4=> eventinfo.SETERRORCODE(cannotdepart)  
Clave_agente:Leaving method canDepart.  
Clave_agente:Entering method canDepart.  
Clave_agente::canDepart:2=> IF self.ISBLANK() THEN  
Clave_agente:Leaving method canDepart.  
#FormDatal:Entering method error.  
#FormDatal::error:3=> SWITCH  
#FormDatal::error:18=> MSGINFO("Aviso", "Carácter inválido.")  
#FormDatal:Entering method status.  
#FormDatal::status:4=> SWITCH  
#FormDatal::status:6=> eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra  
Espaciadora para ver ayuda")  
#FormDatal:Leaving method status.  
Clave_agente:Entering method canDepart.  
Clave_agente::canDepart:2=> IF self.ISBLANK() THEN  
Clave_agente:Leaving method canDepart.  
Poliza_ini:Entering method canDepart.  
Poliza_ini::canDepart:2=> IF self.ISBLANK() THEN  
Poliza_ini::canDepart:3=> MSGINFO("Aviso", "La póliza inicial no puede  
ser nula.")  
Poliza_ini::canDepart:4=> eventinfo.SETERRORCODE(cannotdepart)  
Poliza_ini:Leaving method canDepart.  
Poliza_ini:Entering method canDepart.  
Poliza_ini::canDepart:2=> IF self.ISBLANK() THEN  
Poliza_ini:Leaving method canDepart.  
Poliza_ini:Entering method changeValue.  
Poliza_ini::changeValue:2=> IF NOT self.RECORDSTATUS("New") THEN  
Poliza_ini:Leaving method changeValue.  
Poliza_fin:Entering method depart.  
Poliza_fin::depart:2=> IF self.VALUE <= poliza_ini.VALUE AND NOT  
poliza_fin.ISBLANK() THEN
```

```
Poliza_fin::depart:3=> MSGINFO("Aviso","La poliza final no puede ser
menor o igual que la poliza inicial")
Poliza_fin::depart:4=> eventinfo.SETERRORCODE(1)
Poliza_fin::depart:2=> IF self.VALUE <= poliza_ini.VALUE AND NOT
poliza_fin.ISBLANK() THEN
Poliza_fin:Leaving method depart.
Fecha_ini:Entering method canDepart.
Fecha_ini::canDepart:2=> IF self.ISBLANK() THEN
Fecha_ini:Leaving method canDepart.
Fecha_fin:Entering method depart.
Fecha_fin::depart:15=> IF fecha_ini.VALUE >= fecha_fin.VALUE AND NOT
fecha_fin.ISBLANK() THEN
Fecha_fin::depart:16=> MSGINFO("Aviso","La Fecha inicial no puede ser
menor o igual que la fecha final")
Fecha_fin::depart:17=> eventinfo.SETERRORCODE(1)
Fecha_fin::depart:15=> IF fecha_ini.VALUE >= fecha_fin.VALUE AND NOT
fecha_fin.ISBLANK() THEN
Fecha_fin::depart:20=> IF self.RECORDSTATUS("New") THEN
Fecha_fin::depart:21=> IF MSGQUESTION("Confirmación",";Acepta la
alta?")="Yes" THEN
Fecha_fin:Leaving method depart.
```

*** No se acepta el alta y el registro puede ser modificado antes de confirmar el alta.

```
Clave_cuenta:Entering method canDepart.
Clave_cuenta::canDepart:2=> IF self.ISBLANK() THEN
Clave_cuenta:Leaving method canDepart.
Nombre:Entering method canDepart.
Nombre::canDepart:2=> IF self.ISBLANK() THEN
Nombre:Leaving method canDepart.
Clave_grupo:Entering method canDepart.
Clave_grupo::canDepart:2=> IF self.ISBLANK() THEN
Clave_grupo:Leaving method canDepart.
Clave_oficina:Entering method canDepart.
Clave_oficina::canDepart:2=> IF self.ISBLANK() THEN
Clave_oficina:Leaving method canDepart.
Clave_agente:Entering method canDepart.
Clave_agente::canDepart:2=> IF self.ISBLANK() THEN
Clave_agente:Leaving method canDepart.
Poliza_ini:Entering method canDepart.
Poliza_ini::canDepart:2=> IF self.ISBLANK() THEN
Poliza_ini:Leaving method canDepart.
Poliza_fin:Entering method canDepart.
Poliza_fin:Leaving method canDepart.
Poliza_fin:Entering method depart.
Poliza_fin::depart:2=> IF self.VALUE <= poliza_ini.VALUE AND NOT
poliza_fin.ISBLANK() THEN
Poliza_fin:Leaving method depart.
Fecha_ini:Entering method canDepart.
Fecha_ini::canDepart:2=> IF self.ISBLANK() THEN
Fecha_ini:Leaving method canDepart.
Fecha_fin:Entering method canDepart.
Fecha_fin:Leaving method canDepart.
Fecha_fin:Entering method depart.
Fecha_fin::depart:15=> IF fecha_ini.VALUE >= fecha_fin.VALUE AND NOT
fecha_fin.ISBLANK() THEN
Fecha_fin::depart:20=> IF self.RECORDSTATUS("New") THEN
```

```

Fecha_fin::depart:21=> IF MSGQUESTION("Confirmación","¿Acepta la
alta?")="Yes" THEN

*** Se confirma el alta.

Fecha_fin::depart:22=> self.ACTION(DataPostRecord)
#FormData1:Entering method status.
#FormData1::status:4=> SWITCH
#FormData1::status:26=> eventinfo.SETSTATUSVALUE("Movimiento aceptado.")
#FormData1:Leaving method status.
Fecha_fin::depart:23=> IF NOT self.UNLOCKRECORD() THEN
Fecha_fin::depart:27=> DMGET("cuenta","clave_cuenta",v_clave)
Fecha_fin::depart:28=> DMGET("cuenta","nombre",v_nombre)
Fecha_fin::depart:29=> v_grupo.ATTACH(clave_grupo)
Fecha_fin::depart:30=> DMGET("cuenta","fecha_ini",v_fecha)
Fecha_fin::depart:31=> v_grupo=NUMBER(v_grupo)
Fecha_fin::depart:32=> v_tu.ATTACH(tasa_udi)
Fecha_fin::depart:33=> v_tr.ATTACH(tasa_reserva_div)
Fecha_fin::depart:34=> v_tc.ATTACH(tasa_costo_adquisicion)
Fecha_fin::depart:35=> v_tg.ATTACH(tasa_gia)
Fecha_fin::depart:36=> qq =
Fecha_fin::depart:49=> IF NOT executeqbe(qq) THEN
Fecha_fin::depart:52=> MESSAGE ("Movimiento aceptado en cuenta
especial.")
Fecha_fin::depart:57=> self.ACTION(dataUNLOCKRECORD)

*** Apuntando con el mouse el botón de cambio.

:Entering method mouseEnter.
:mouseEnter:2=> MESSAGE ("Ubica un registro para modificarlo. Doble
click derecho para ver siguiente.")
:Leaving method mouseEnter.

*** Oprimiendo el botón de cambio.

:Entering method pushButton.
:pushButton:2=> IF self.RECORDSTATUS("New") OR
self.RECORDSTATUS("Modified") THEN
:pushButton:5=> IF NOT self.ACTION(datasearch) THEN
#FormData1:Leaving method status.
Clave_cuenta:Entering method canDepart.
Clave_cuenta::canDepart:2=> IF self.ISBLANK() THEN
Clave_cuenta:Leaving method canDepart.
Clave_cuenta:Entering method canDepart.
Clave_cuenta::canDepart:2=> IF self.ISBLANK() THEN
Clave_cuenta:Leaving method canDepart.
Nombre:Entering method canDepart.
Nombre::canDepart:2=> IF self.ISBLANK() THEN
Nombre:Leaving method canDepart.
Clave_grupo:Entering method canDepart.
Clave_grupo::canDepart:2=> IF self.ISBLANK() THEN
Clave_grupo:Leaving method canDepart.
#FormData1:Entering method status.
#FormData1::status:4=> SWITCH
#FormData1::status:6=> eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra
Espaciadora para ver ayuda")
#FormData1:Leaving method status.
Clave_oficina:Entering method canDepart.
Clave_oficina::canDepart:2=> IF self.ISBLANK() THEN

```

```
Clave_oficina:Leaving method canDepart.
#FormData!:Entering method status.
#FormData!::status:4=> SWITCH
#FormData!::status:6=> eventinfo.SETSTATUSVALUE("Presione Ctrl+Barra
Espaciadora para ver ayuda")
#FormData!:Leaving method status.
Clave_agente:Entering method canDepart.
Clave_agente::canDepart:2=> IF self.ISBLANK() THEN
Clave_agente:Leaving method canDepart.
Poliza_ini:Entering method canDepart.
Poliza_ini::canDepart:2=> IF self.ISBLANK() THEN
Poliza_ini:Leaving method canDepart.
Poliza_fin:Entering method canDepart.
Poliza_fin:Leaving method canDepart.
Poliza_fin:Entering method depart.
Poliza_fin::depart:2=> IF self.VALUE <= poliza_ini.VALUE AND NOT
poliza_fin.ISBLANK() THEN
Poliza_fin:Leaving method depart.
#FormData!:Entering method status.
Fecha_ini:Entering method canDepart.
Fecha_ini::canDepart:2=> IF self.ISBLANK() THEN
Fecha_ini:Leaving method canDepart.
Fecha_fin:Entering method canDepart.
Fecha_fin:Leaving method canDepart.
Fecha_fin:Entering method depart.
Fecha_fin::depart:15=> IF fecha_ini.VALUE >= fecha_fin.VALUE AND NOT
fecha_fin.ISBLANK() THEN
Fecha_fin::depart:20=> IF self.RECORDSTATUS("New") THEN
Fecha_fin::depart:61=> IF self.RECORDSTATUS("Modified") THEN
Fecha_fin::depart:62=> IF MSGQUESTION("Confirmación","¿Acepta el
cambio?") = "Yes" THEN
Fecha_fin::depart:63=> self.ACTION(datapostrecord)
#FormData!:Entering method status.
#FormData!::status:4=> SWITCH
#FormData!::status:28=> eventinfo.SETSTATUSVALUE("Movimiento aceptado.")
#FormData!:Leaving method status.
Fecha_fin:Leaving method depart.

*** Apuntando con el mouse el botón de cancelar.

::mouseEnter:2=> MESSAGE("Dar de baja el registro actual.")
&&#amp; ERROR el mensaje es incorrecto debe ser "Cancelar movimiento.")

:Leaving method mouseEnter.
:Entering method mouseExit.
::mouseExit:2=> MESSAGE (" ")
:Leaving method mouseExit.

*** Modificar registro para después cancelar cambios.

Clave_cuenta:Entering method canDepart.
Clave_cuenta::canDepart:2=> IF self.ISBLANK() THEN
Clave_cuenta:Leaving method canDepart.
Nombre:Entering method canDepart.
Nombre::canDepart:2=> IF self.ISBLANK() THEN
Nombre:Leaving method canDepart.
Clave_grupo:Entering method canDepart.
Clave_grupo::canDepart:2=> IF self.ISBLANK() THEN
Clave_grupo:Leaving method canDepart.
Clave_oficina:Entering method canDepart.
```

```
Clave_oficina::canDepart:2=> IF self.ISBLANK() THEN
Clave_oficina:Leaving method canDepart.
Clave_agente:Entering method canDepart.
Clave_agente::canDepart:2=> IF self.ISBLANK() THEN
Clave_agente:leaving method canDepart.
:Entering method pushButton.
:pushButton:2=> self.ACTION(datacancelrecord)
#FormData!:Entering method status.
#FormData!:status:4=> SWITCH
#FormData!:status:12=> eventinfo.SETSTATUSVALUE("Movimiento
Cancelado.")

*** Apuntando con el mouse el botón de bajas.

:Entering method mouseEnter.
:mouseEnter:2=> MESSAGE("Dar de baja el registro actual.")
:Leaving method mouseEnter.

*** Oprimiendo el botón de baja.

:Entering method pushButton.
:pushButton:2=> IF MSGQUESTION("Confirmación","¿ Confirma la Baja ?") =
"Yes" THEN
:pushButton:3=> self.ACTION(datadeleterecord)
#FormData!:Entering method status.
#FormData!:status:4=> SWITCH
#FormData!:status:8=> eventinfo.SETSTATUSVALUE("Registro borrado")
:Leaving method pushButton.

*** Oprimiendo el botón de salir.

:Entering method pushButton.
:pushButton:2=> salir = TRUE
:pushButton:3=> CLOSE()
:Leaving method pushButton.
```

VI.4 Programación del sistema.

La programación de sistemas es un arte que depende de la capacidad individual de desarrollo de sistemas, la atención al detalle y el conocimiento de como utilizar eficientemente las herramientas disponibles de un lenguaje de programación de la mejor manera. La buena programación consiste en la producción de programas confiables, eficientes y fáciles de mantener e independiente del lenguaje de programación.

Como se mencionó en capítulos anteriores, el sistema SIGCE se desarrolló utilizando herramientas de programación orientadas a objetos bajo el DBMS Paradox 4.5, sin olvidar las técnicas y procedimientos establecidas por la programación estructurada convencional. A continuación presentamos algunas de las técnicas de programación así como el pseudocódigo utilizado en el desarrollo del sistema.

Programación descendente y ascendente.

El proceso de desarrollo de un programa a partir de un diseño de software puede enfocarse de dos formas: El desarrollo descendente y el ascendente. Si el sistema se considera como una jerarquía de componentes, entonces el desarrollo descendente implica emplear en el topo de la jerarquía y trabajar hacia abajo, mientras que el desarrollo ascendente empieza en el nivel más bajo y se prosigue hacia el nivel superior. Para sistemas jerárquicos pequeños, un enfoque descendente genera los programas más confiables, entendibles y modificables. Sin embargo los grandes sistemas de software suelen construirse con subsistemas conectados entre sí. Así pues, en la práctica no son apropiados los enfoques estrictamente descendentes ni ascendentes para la construcción de grandes sistemas. Un enfoque descendente estricto quizá conduzca a una duplicidad de módulos. Es más apropiado factorizar aquellos componentes que proporcionan servicios comunes a otros módulos.

El desarrollo de módulos SIGCE sigue un enfoque descendente, tal como lo muestra la carta de estructuras comprendida en el capítulo anterior, sin embargo ciertas funciones son comunes para la generación de información en diferentes módulos del sistema como son consultas y reportes.

Programación Estructurada y modular:

En un programa deben usarse construcciones de control para que el flujo del programa sea descendente. Al poder considerar los ciclos como una sola proposición compuesta, la ejecución debe empezar por la primera sentencia del programa, cada sentencia debe ejecutarse por turno y la ejecución debe terminar con la última proposición, esto es, las proposiciones deben tener un solo punto de entrada una sola salida.

Object Pal, el lenguaje de programación de Paradox 4.5, proporciona control sobre la visibilidad de los nombres de objetos que pueden ser accedidos desde un bloque externo al objeto, lo que permite diseñar funciones comunes a varios módulos.

Distribución de programas:

El lenguaje de programación proporcionado por Paradox 4.5 (Object Pal) es un lenguaje de formato libre, en el cual el significado de un programa no se afecta por la forma en que se distribuya en la página. La distribución afecta a la legibilidad de los programas. El uso discrecional de líneas en blanco, el énfasis en las palabras reservadas, la agrupación consistente en párrafos, y la documentación de programas con comentarios hacen que el programa sea más elegante, fácil de leer así como contar con la facilidad de mantenimiento del mismo.

Instrumentos de programación:

La práctica de la programación está relacionada con la manufactura; si se apoya la programación con instrumentos automatizados se pueden lograr grandes mejoras en la productividad del sistema.

Object Pal es un lenguaje orientado a objetos, lo cual permite proporcionarle al usuario un ambiente gráfico agradable en la presentación del sistema y al programador la facilidad del mantenimiento y desarrollo de programas al utilizar objetos ya definidos por el compilador o fabricar sus propios objetos.

Pseudocódigo:

En términos generales, en el desarrollo del sistema se tienen a grandes rasgos los siguientes módulos:

MENÚ PRINCIPAL DEL SISTEMA.

Inicio

Pedir acceso al sistema por medio de clave de seguridad

Si usuario autorizado

Llamar al menú principal

Seleccionar opción

Si opción es

Actualización

Validar nivel de seguridad del usuario al módulo de actualizaciones

Si usuario autorizado al módulo seleccionado

Llamar al menú de actualizaciones.

Consultas

Validar nivel de seguridad del usuario al módulo de consultas

Si usuario autorizado al módulo seleccionado

Llamar al menú de consultas

Reportes

Validar nivel de seguridad del usuario al módulo de reportes

Si usuario autorizado al módulo seleccionado

Llamar al menú de reportes

Utilerías

Validar nivel de seguridad del usuario al módulo de Utilerías

Si usuario autorizado al módulo seleccionado

Llamar al menú de Utilerías

Salir

Confirmar la salida definitiva del sistema

Si confirmación aceptada por el usuario

Se termina ejecución del sistema SIGCE.

MÓDULO DE ACTUALIZACIONES

Llamar al menú de Actualizaciones

Seleccionar opción

Si opción es

Carátulas

Llamar al programa de mantenimiento a carátulas.

Calendario de emisión

Llamar al programa de mantenimiento al calendario de emisión.

Estatus de Cuentas

Llamar al programa de mantenimiento a Estatus de cuentas especiales

Catalogo de Cuentas.

Llamar al programa de mantenimiento al Catalogo de cuentas especiales

Emisión de primas

Llamar al programa de mantenimiento de primas emitidas

Presupuestos

Llamar al programa de mantenimiento de presupuestos

Agentes

Llamar al programa de mantenimiento a agentes autorizados.

Oficinas

Llamar al programa de mantenimiento a oficinas existentes

Dividendos

Llamar al programa de mantenimiento de dividendos.

Salir

Regresar al menú principal de SIGCE.

MODULO DE CONSULTAS:

Llamar al menú de Consultas

Seleccionar opción

Si opción es

Carátulas

Llamar al programa de consultas a carátulas.

Calendario de emisión

Llamar al programa de consulta al calendario de emisión

Producción y Cobranza

Llamar al programa de consulta de Producción y cobranza

Resultado Técnico

Llamar al programa de consulta al resultado técnico anual

Estatus de Cuentas

Llamar al programa de consulta a Estatus de cuentas especiales

Catalogo de Cuentas.

Llamar al programa de consulta al Catalogo de cuentas especiales

Emisión de primas

Llamar al programa de consulta de primas emitidas

Dividendos

Llamar al programa de consulta de dividendos

Agentes

Llamar al programa de mantenimiento a agentes autorizados.

Oficinas

Llamar al programa de mantenimiento a oficinas existentes

Salir

Regresar al menú principal de SIGCE.

MODULO DE REPORTES

Llamar al menú de Reportes

Seleccionar opción

Si opción es

Carátulas

Llamar al programa de reporte a carátulas.

Calendario de emisión

Llamar al programa de reporte al calendario de emisión

Producción y Cobranza

Llamar al programa de reporte de Producción y cobranza

Resultado Técnico

Llamar al programa de reporte al resultado técnico anual

Catálogo de Cuentas.

Llamar al programa de reporte al Catálogo de cuentas especiales

Emisión de primas

Llamar al programa de reporte de primas emitidas

Dividendos

Llamar al programa de reporte de dividendos

Estatus de Cuentas

Llamar al programa de reporte a Estatus de cuentas especiales

Salir

Regresar al menú principal de SIGCE.

MODULO DE UTILERÍAS

Llamar al menú de Utilerías del sistema.

Seleccionar opción

Si opción es

Administración de usuarios

Llamar al programa de administración a niveles de usuario

Mantenimiento de datos

Llamar al submenú de mantenimiento de datos

Seleccionar opción del submenú de mantenimiento de datos

Si opción es

Respaldo

Llamar programa de respaldo de información

Recuperación

Llamar programa de restauración de información

Reparación de Tablas

Llamar al programa de reparación de tablas

Salir

Regresar al menú de Utilerías

Salir

Regresar al menú principal de SIGCE.

VII.
**Implementación
del sistema.**

VII. Implementación del sistema.

VII.1. Descripción del Hardware.

En este punto se describe el equipo necesario para la operación del Sistema Gerencial para Cuentas Especiales. Las descripciones de hardware y software operativo son minimas, y se obtuvieron a partir de la experiencia de la empresa con la base de datos Paradox, además de tomar en cuenta los resultados de las pruebas de desempeño que se describirán en el punto número 2 de este capítulo.

VII.1.1. Equipo servidor.

Para la operación del Sistema se requiere opcionalmente de un equipo, con algún sistema operativo que soporte el funcionamiento como servidor de archivos con clientes de tipo PC, para poder compartir entre varios usuarios la misma base de datos. Las características de dicho equipo dependen totalmente del software de red a utilizar. Algunos ejemplos de dicho software son los siguientes:

- Novell Netware.
- MS Windows NT.
- Lan Manager, etc.

Este equipo debe comunicarse a través de una red local con las Computadoras Personales que lleven a cabo la función de estaciones de trabajo.

En el caso de Grupo Nacional Provincial se utilizó un equipo servidor de archivos con procesador Pentium a 90 MHz, memoria RAM de 32 Mb y espacio en disco duro de 1 Gb; con sistema operativo Novell Netware versión 3.12 para 100 usuarios. El adaptador de red es de tipo Ethernet 10BaseT.

VII.1.2. Equipo PC (estaciones de trabajo).

Para que Paradox 4.5 funcione en un equipo PC, ya sea de manera compartida a través de un servidor de archivos o de manera independiente; es recomendable un equipo PC con procesador 486DX2 a 66 MHz, memoria RAM de 8Mb, espacio en disco duro y adaptador de red que dependen de la ubicación de la base de datos (PC o servidor).

El software instalado en el la PC debe ser el siguiente:

- Sistema operativo MS DOS 6.X
- Microsoft Windows 3.XX (para el caso en que no se instale en el servidor).
- Software cliente para el sistema operativo de red utilizado (en el caso de compartir la base de datos).
- Borland Paradox 4.5 para Windows (para el caso en que no se instale en el servidor).

Algunos de los equipos utilizados en Grupo Nacional Provincial son de las características antes mencionadas, y otros son con procesador Pentium a 90 MHz y memoria RAM de 16Mb. El adaptador de red de todos ellos es de tipo Ethernet 10BaseT. El software instalado es MS DOS 6.22, MS Windows 3.11 para trabajo en grupos, software cliente de Novell Netware 3.12 y Borland Paradox 4.5 para Windows.

VII.2. Pruebas de desempeño.

Existen diversas estrategias de prueba, las más comunes son las pruebas ascendentes y descendentes. El enfoque ascendente empieza por probar módulos individuales separadamente, luego los módulos individuales se combinan para formar unidades cada vez más grandes, que se probarán en masa, finalmente todos los componentes se unen para probarse. El enfoque de prueba descendente empieza con un esqueleto del sistema, la estrategia de prueba supone que se han desarrollado los módulos ejecutivos de alto nivel, pero que los de bajo nivel existen sólo como módulos vacíos, es decir, que no procesan nada, sino que simplemente termina luego de ser llamado, el propósito de estas pruebas es simplemente comenzar a ejercitar las interfaces entre los subsistemas principales. Las pruebas siguientes abarcan aspectos cada vez más profundos del sistema. Otros tipos de pruebas son:

- Prueba Funcional.** Su propósito es asegurar que el sistema realiza sus funciones normales de manera correcta. Los casos de prueba se desarrollan y alimentan al sistema, las salidas se examinan para verificar que sean correctas.
- Pruebas de recuperación.** Su propósito es asegurar que el sistema pueda recuperarse adecuadamente de diversos tipos de fallas. Las pruebas de recuperación pueden requerir que el equipo que realiza el proyecto simule o provoque fallas de corriente, fallas en el sistema operativo, etc.
- Pruebas de desempeño.** El propósito es asegurar que el sistema pueda manejar el volumen de datos y transacciones de entrada especificados en el modelo de implantación del usuario, además de asegurar que tenga el tiempo de respuesta requerido.
- Pruebas exhaustivas.** Idealmente se generarían casos de prueba para cubrir cada entrada posible de situaciones que el sistema pudiera enfrentar alguna vez, después se probarían de manera exhaustiva para asegurar que su comportamiento sea perfecto. Debido a lo cual no es común que se realicen este tipo de pruebas.

Como parte de las pruebas de desempeño para liberar el Sistema de Información Gerencial para Cuentas Especiales, se realizaron pruebas de volumen, es decir se prepararon los archivos de la base de datos para simular su funcionamiento con volúmenes reales de información, realizándose una serie de pruebas conjuntas como captura de datos, consultas e impresión de reportes.

Para actualizar la información se capacitó a tres usuarios en la captura de datos para SIGCE y se les pidió a los gerentes de operación y de servicio que hicieran pruebas durante una semana, reportando cualquier anomalía.

Previamente se les hizo una presentación del nuevo sistema, mostrando la información y los beneficios que se obtendrían al implantarlo. La capacitación de los usuarios de captura se realizó en sesiones de trabajo por las tardes, quedando como responsable el área de Cuentas Especiales.

El volumen de información que se escogió para realizar las pruebas de desempeño correspondió a un año de trabajo. Por ser un sistema que maneja información histórica es importante medir los tiempos de respuesta a las solicitudes de información.

El número de registros fue en promedio de 1500. Por ser un sistema gerencial y estar su alimentación de datos en función del calendario de emisión, se tienen las siguientes consideraciones:

Se generan de siete a nueve estadísticas en un mes, realizándose en promedio 100 estadísticas al año, cada cuenta consta de tres carátulas, dos dividendos por año con sus respectivos reportes de primas mensuales, un presupuesto anual, lo cual implica un volumen de información de alrededor de 1500 registros.

En SIGCE solo aparezca el concentrado de información no el detalle. En base a los comentarios recibidos y la pruebas de funcionamiento donde el sistema estuvo disponible en promedio de 7:20 de la mañana a 22:30 de la noche, durante una semana de pruebas el sistema estuvo funcionando correctamente, y se perfila como una herramienta de consulta para la toma de decisiones.

No existió comentario alguno en cuanto al funcionamiento del mismo considerándose entonces que SIGCE permitirá cubrir las necesidades de información y se convertirá en una herramienta de consulta a nivel ejecutivo de Cuentas Especiales.

VII.3. Implantación

Las formas más comunes de implantación son:

Implantación ascendente

Este tipo de implantación es típico de el ciclo de vida de los proyectos clásicos, en ella se espera que los programadores lleven a cabo primero sus pruebas modulares, luego las

pruebas del subsistema y finalmente las pruebas de sistema completo, como se muestra en la figura VII.1.

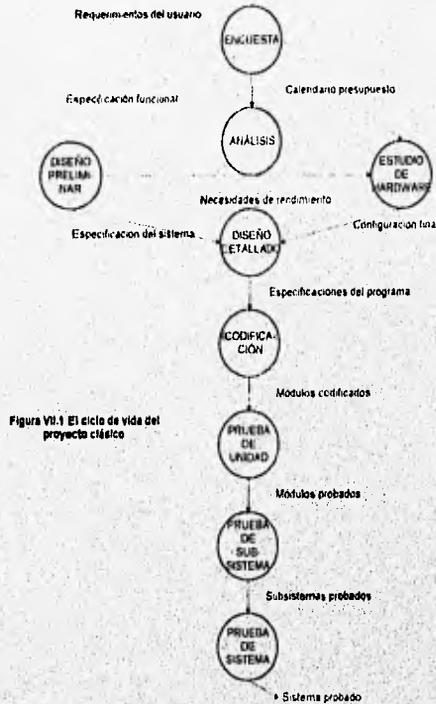


Figura VII.1 El ciclo de vida del proyecto clásico

Este enfoque se conoce también como ciclo de vida de cascada, es un enfoque muy útil en procesos como el montaje de automóviles en línea, pero en cuestiones de desarrollo de software presenta un gran número de dificultades:

- Nada está hecho hasta que todo está terminado. Esto significa que si no se ha terminado el proyecto no existe nada que presentarle al usuario, salvo papeles.
- Las fallas más triviales se encuentran al comienzo del periodo de pruebas y las más graves al final. Debido a que los programas modulares se prueban por partes, solo al juntar todo el sistema y realizar una prueba total, se descubren las fallas más graves.
- La eliminación de fallas suele ser extremadamente difícil durante las últimas etapas de prueba del sistema. A veces puede ser muy complicado localizar en qué módulo se encuentra una falla detectada en una prueba total del sistema.

- La necesidad de prueba con la computadora aumenta exponencialmente durante las etapas finales de prueba. Son necesarias muchas horas-máquina para probar el proyecto.

Implantación descendente.

Mientras que la implantación ascendente es parte del ciclo de vida clásico de un proyecto, la implantación descendente es parte del ciclo de vida semiestructurado. La secuencia de codificación, la prueba de módulos y prueba del sistema se reemplaza por una implantación de arriba hacia abajo, que es un enfoque en el cual los módulos de alto nivel se codifican y prueban primero, seguidos por los de bajo nivel, más detallados.

Para el sistema SIGCE se utilizará una implantación descendente, por la ventaja que ofrece de retroalimentación entre la codificación, la prueba y la eliminación de fallas.

VII.4. Capacitación del usuario.

El funcionamiento adecuado de un programa de capacitación a los usuarios de un sistema, depende de varias cosas, entre otras tenemos:

- El total respaldo de los altos directivos al programa. Este respaldo es fundamental cuando la capacitación incluye a personas de diferentes niveles dentro de la empresa. Los mejores resultados se obtendrán si todos se involucran y participan, activa y positivamente.
- La capacitación debe incluir a administradores y usuarios de todos los niveles. La capacitación no es solo para los niveles inferiores, sino para todos. Recordemos que los altos directivos deben ser los primeros en recibir la capacitación, de esa manera se confirma su decisión de brindar un desarrollo continuo.
- Las necesidades de capacitación varían. Estas necesidades varían de acuerdo con los puestos y los niveles jerárquicos que ocupan dentro de la organización, pero también de acuerdo a las personas, puesto que no todas tienen los mismos antecedentes, requerimientos, aspiraciones y capacidad de aprendizaje. En consecuencia, las actividades de capacitación deben diseñarse a la medida de estas necesidades.
- La teoría y la práctica deben ir de la mano. Esto es particularmente importante cuando se está hablando de la capacitación a usuarios de un sistema de cómputo, debido a que sin la práctica la teoría no sería comprendida, y las dudas surgirían en el momento en que ya el usuario ya administrara solo el sistema.

Por lo anterior el proceso de capacitación se llevará a efecto en cuatro fases, la primera se aplicará a los usuarios ejecutivos que solo realizarán consultas a consolidados, la segunda fase se aplicará a la gente que se dedicará a administrar el sistema, a estos usuarios se les explicará el funcionamiento total del sistema, las rutinas de seguridad, las

necesidades de hardware y software del sistema, así como la modificación de las rutinas para el mantenimiento del sistema. La tercera fase será la capacitación de los usuarios que capturarán la información de los archivos planos a el sistema, después en la última fase, se capacitará a los usuarios que accesarán el sistema para la elaboración de reportes.

La primera fase tendrá una duración de 6 horas, la segunda 19 horas, la tercera 8 horas y la última 6 horas.

VIII.
Conclusiones.

VIII. Conclusiones:

La importancia de la información.

Durante los años 90's y posteriores la información representará la principal herramienta de desarrollo dentro de un mundo capitalista que cambia a grandes pasos.

Sin duda empresas que cuenten con medios tecnológicos conjuntamente con sistemas informáticos eficientes podrán contar con información social, política y económica de primera mano, en base a la cual dirigirán su rumbo y su estrategia de competencia en el mercado. Así pues, la información se vuelve un elemento que si es adecuadamente empleado permite destacar dentro de un medio competitivo, y este concepto no solo aplica a empresas o grupos, la misma constante se puede emplear para los individuos y sus actividades diarias.

S.I.G.C.E.

Tomando como marco la importancia de la información se decidió llevar a cabo un sistema de información gerencial que fuera útil a Grupo Nacional Provincial durante la toma de decisiones, en especial aquellas relacionadas con clientes importantes conocidos como cuentas especiales. El sistema fue nombrado S.I.G.C.E. (Sistema de Información Gerencial para Cuentas Especiales) y cumple con las siguientes actividades:

- Concentra información histórica relevante sobre clientes importantes de Grupo Nacional Provincial.
- Permite realizar consultas sobre elementos de información relevantes para la toma de decisiones económicas de Grupo Nacional Provincial tal como lo son el catálogo de cuentas especiales, calendario de emisión de cuentas, cifras referentes a primas emitidas, dividendos, presupuestos anuales y carátulas de cuentas especiales.
- Emite reportes referentes a información destacada de las cuentas especiales como los son las primas emitidas, dividendos, etc.
- Cuenta con un módulo para la elaboración y mantenimiento de carátulas las cuales se forman a partir de datos concentrados referentes a cuentas especiales a cierta fecha de corte.
- Por medio del módulo de actualizaciones se puede dar mantenimiento al catálogo de cuentas especiales y al calendario de emisión.
- Administra al conjunto de agentes que manejan las cuentas especiales así como a las oficinas a las cuales éstos pertenecen.
- S.I.G.C.E. mantiene un control sobre el acceso a cada módulo por cada usuario que ingresa a él.
- El sistema cuenta con la funcionalidad de reparación de tablas.

- Por medio del sistema es posible realizar respaldo y recuperación del total de la base de datos del mismo.
- Presenta al usuario una interfaz gráfica lo cual hace agradable su uso.
- El sistema esta documentado por medio de un manual de usuario y un manual técnico.

Sumarizando el apartado referente al desarrollo del sistema el grupo considera que el software y la documentación de S.I.G.C.E. cumple con los objetivos planteados durante las etapas de análisis y diseño.

Conocimientos adquiridos por el grupo.

Durante la elaboración del presente trabajo fue muy importante distinguir la diferencia de utilizar o no una Metodología de Desarrollo de Sistemas para la construcción de la aplicación, además de la secuencia de actividades que se llevan a cabo desde la elección de la metodología hasta la implementación del mismo.

Podemos señalar que el desarrollo del sistema hubiese presentado grandes dificultades o hubiese sido casi imposible de no haber considerado el uso de una metodología para la consecución del objetivo.

Otro punto importante por hacer notar es que la elaboración de este trabajo nos permitió reafirmar conocimientos adquiridos durante nuestros estudios profesionales relacionadas con materias tan diversas como Ingeniería de programación, Base de datos, Programación Estructurada, Estructura de Datos, etc.

Por otra parte a excepción del integrante que propuso el tema de tesis, el resto del grupo no dominaba la herramienta Paradox for Windows, este hecho inicialmente provocó un estancamiento en la fase de codificación, sin embargo, después de un período de aprendizaje logramos adquirir la destreza necesaria para desarrollar en Paradox.

A la par con los conocimientos técnicos están los aspectos humanos. La experiencia indica que es difícil mantener la cohesión de un equipo, en nuestro caso logramos sostener la unión del mismo, y aunque existieron divergencias en ocasiones grandes, la mejor lección fue que cada elemento presentó una disciplina que sumada nos permitió concluir esta tesis.

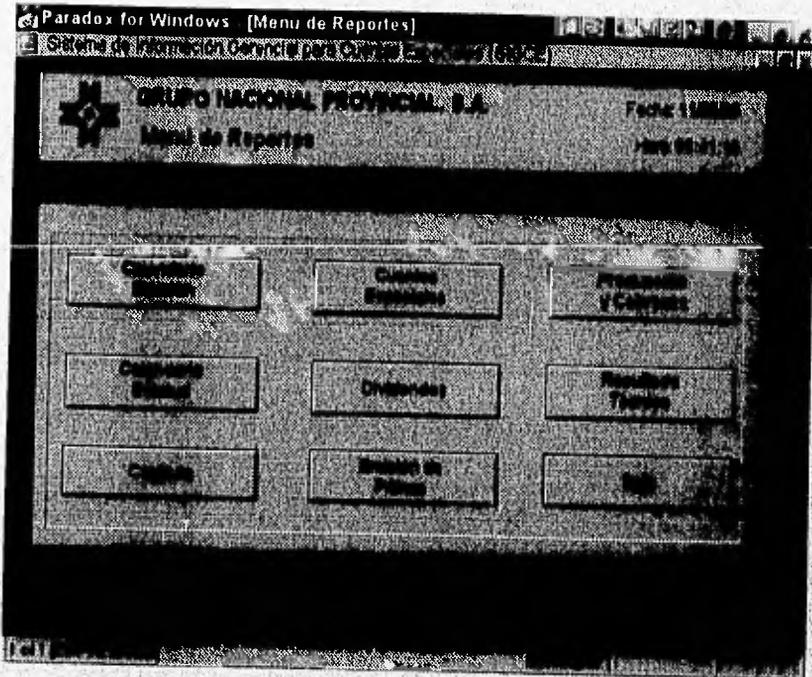
Bibliografía

Autor	Título	Editorial
Cochran Glenn, Robinson Celeste	<i>"Paradox for Windows 4.5 handbook"</i>	Random House, New York, 1993
Date C.J.	<i>"Introducción a los sistemas de bases de datos"</i>	Addison-Wesley, México, 1987
Fairley Richard	<i>"Ingeniería de Software"</i>	McGraw-Hill, México, 1985
Kendall Kenneth, Kendall Julie	<i>"Análisis y diseño de sistemas"</i>	Prentice-Hall, México, 1991
Korth Henry, Abraham Silberschatz	<i>"Fundamentos de bases de datos"</i>	McGraw-Hill, México, 1988
Pressman Roger S.	<i>"Ingeniería del software"</i>	McGraw-Hill, México, 1993
Yourdon Edward	<i>"Análisis estructurado moderno"</i>	Prentice-Hall, México, 1993
Yourdon Edward	<i>"Structured analysis workshop"</i>	Yourdon Inc., New York, 1992
Borland	<i>"Gula del programador objectpal"</i>	Borland International, Inc., México, 1992
López - Fuensalida	<i>"Metodologías de Desarrollo"</i>	Macrobít México, 1991

Apéndices

Apéndice A.

A continuación se presentan algunos ejemplos de reportes en Paradox generados por el Sistema de Información Gerencial para Cuentas Especiales (SIGCE).





GRUPO NACIONAL PROVINCIA, S.A.
Sistema Gerencial de Cuentas Especiales.

Fecha: 11/05/96
Hora: 7:32:18 PM

Reporte del Calendario de Emisión.

Página: 1

Grupo : CARRETERAS

Cuenta : **TIERRA COLORADA-ACAPULCO**

Fecha de corte	Fecha de elaboración	Fecha de entrega	Inicio	Fin	Tipo de estadística
31/03/94	24/03/96		11/11/95		N

Grupo : GRANDES NEGOCIOS

Cuenta : **SICREA**

Fecha de corte	Fecha de elaboración	Fecha de entrega	Inicio	Fin	Tipo de estadística
31/08/09	20/04/96		11/11/95		N
30/04/94	31/05/94		11/11/95		N

Cuenta : **VOLSKWAGEN CREDIT**

Fecha de corte	Fecha de elaboración	Fecha de entrega	Inicio	Fin	Tipo de estadística
31/03/94	24/03/96	29/01/94	11/11/95		N



GRUPO NACIONAL PROVINCIA, S.A.

Sistema Gerencial de Cuentas Especiales.

Fecha: 11/05/98
Hora: 7:32:18 PM

Reporte del Calendario de Emisión.

Página: 2

Grupo: **PLAN PISO**

Cuenta: **DISTRIBUIDORES FORD**

Fecha de corte	Fecha de elaboración	Fecha de entrega	Inicio	Fin	Tipo de estadística
31/03/94	24/03/96	29/03/94	11/11/95		S
28/02/95	24/03/96		11/11/95		N

Cuenta: **DISTRIBUIDORES MERCEDES BENZ**

Fecha de corte	Fecha de elaboración	Fecha de entrega	Inicio	Fin	Tipo de estadística
31/03/94	24/03/96		11/11/94		N

Cuenta: **FORD MOTOR CO. (IMPORTADOS)**

Fecha de corte	Fecha de elaboración	Fecha de entrega	Inicio	Fin	Tipo de estadística
31/03/94	24/03/96		11/11/95		N



GRUPO: 1 PLAN PISO

Clave:	Nombre	Tasa Udi	Tasa Gia	Tasa Costo Adquisición	Tasa Reserva Dividendo	Fecha Inicio Vigencia:	Fecha Fin Vigencia:
ADF	DISTRIBUIDORES FORD	1000.0 %	1000.0 %	1000.0 %	1000.0 %	11/11/95	
ADM	DISTRIBUIDORES MERCEDE	100.0 %	100.0 %	100.0 %	100.0 %	11/11/94	
CHD	CHRYSLER DAÑOS					11/11/95	
FMI	FORD MOTOR CO. (IMPORTA	1000.0 %	1900.0 %	1900.0 %	1000.0 %	11/11/95	

GRUPO: 2 GRANDES NEGOCIOS

Clave:	Nombre	Tasa Udi	Tasa Gia	Tasa Costo Adquisición	Tasa Reserva Dividendo	Fecha Inicio Vigencia:	Fecha Fin Vigencia:
SIC	SICREA	100.0 %				11/11/95	
VWC	VOLKSWAGEN CREDIT	1200.0 %				11/11/95	
VWL	VOLKSWAGEN LEASING	6700.0 %			6600.0 %	11/11/94	

GRUPO: 3 CARRETERAS

Clave:	Nombre	Tasa Udi	Tasa Gia	Tasa Costo Adquisición	Tasa Reserva Dividendo	Fecha Inicio Vigencia:	Fecha Fin Vigencia:
MOU	MUSE				4400.0 %	11/11/98	
TCO	TIERRA COLORADA-ACAPUL	200.0 %				11/11/95	

**GRUPO NACIONAL PROVINCIAL, S.A.**

Sistema de Cuentas Especiales.

Reporte de calculo de dividendos al corte del: 31/12/94
Cuenta Especial: SIC SICREA**PRIMAS**

	N\$8,374,871,828.00
Primas Neta pagada devengada	
(-) Menos uso de instalaciones	N\$795,612,824.00
prima Neta Devengada a considerar para Dividendos	

SINIESTRO OCCURRIDO

Siniestro ocurrido para calculo de dividendos	N\$462,214,158.00
Gasto indirecto de ajuste	
siniestro ocurrido con gastos indirecto de ajuste	
porcentaje de siniestralidad con gasto indirecto de ajuste	

DIVIDENDO

DIV = 0.9((0.75 PNPD) - (1.10 S.O))	N\$540,079,663.00
DIV = 0.9 ((0.75X75,792,590.04) - (1.10 X46,221,415.81))	N\$50,000,000.00
DIV = 0.9 (56,844,442.53 - 50,843,557.39)	
DIV = 0.9 (6,00,885.14)	
DIV = 5,400,796.63	



GRUPO NACIONAL PROVINCIAL, S.A.

Sistema Garancial de Cuentas Especiales

Reporte de Carátulas.

Fecha: 11/05/96

Hora: 7:27:35 PM

Página: 1

Corte al : 31/03/94

Nombre	Oficina	Póliza inicial	Póliza final
VOLSKWAGEN CREDIT	26	7453	
VOLSKWAGEN CREDIT	26	7701	7953
VOLSKWAGEN CREDIT	26	18230	18730

Fecha corte	Prim pagada	Prim. devengada	Importe siniestro	Gastos	Res. pendiente	Ingreso	Tasa ajuste
31/03/94	N\$100.00	N\$56.00	N\$12.00	N\$100.00	N\$12.00	N\$22.00	1.00
	N\$100.00	N\$56.00	N\$12.00	N\$100.00	N\$12.00	N\$22.00	1.00

Fecha de Elaboración : 05/11/96

Apéndice B.

A continuación se presentan algunos términos utilizados en el documento relacionados al área de seguros.

Prima Emitida:	Valor nominal de la póliza
Prima Pagada:	Es la cantidad den pesos que ingresa a la compañía por concepto de pago de seguro
Prima Devengada:	Es la cantidad en pesos que se devenga de la prima pagada o prima emitida en un determinado periodo de tiempo.
Siniestro pagado:	Es la cantidad en pesos que la compañía eroga por concepto del pago de un siniestro.
Siniestro Reservado:	Es la cantidad en pesos que la compañía guarda para el pago de un siniestro reportado.
Salvamento:	Es la cantidad que la compañía recuperra por concepto de un bien siniestrado.
Gastos:	Son las erogaciones que la compañía efectúa por conceptos de pagos asociados o indirectos por causa de un siniestro.
Siniestro Ocurrido (so):	Es la suma de + Siniestro Pagado + Siniestro Reservado - Salvamento + Gastos
Costos Netos de Adquisición(Comisión):	Son los costos asociados a la venta de un seguro. <ul style="list-style-type: none">• Comisiones• Bonos• Uso de Instalaciones (UDI)

Siniestralidad : Es la relación entre primas y siniestros que nos indica la productividad del ramo.

$\% \text{ Siniestralidad} = \text{S.O/P.D}$

$\% \text{ Siniestralidad} = \text{S.O/P.P}$ (Forma de Pago Fraccionada)

Resultado Técnico: Es el producto final de la operación de un ramo del seguro.

- + Prima devengada
- Costo Neto de Adquisición
- Siniestro Ocurrido
- Reserva de dividendos

Apéndice C.

**Sistema de Información Gerencial para
Cuentas Especiales.
SIGCE**

Versión 1.0

Manual Técnico

Contenido.

1. Requerimientos para la instalación del Sistema.
2. Configuración de Paradox para Windows en forma compartida.
3. Instalación de Paradox para Windows.
4. Instalación del Sistema Gerencial para Cuentas Especiales.
5. Notas sobre el menú "Utilerías del Sistema".

1. Requerimientos para la instalación del Sistema.

La instalación del Sistema Gerencial para Cuentas Especiales se puede llevar a cabo de dos formas:

1. Instalación en forma local:

Sin compartir la base de datos con otros usuarios.

2. Instalación en forma compartida:

Varios usuarios pueden acceder a la base de datos. El número de usuarios que pueden utilizar la misma base de datos depende de los recursos del equipo servidor de archivos y del software de red utilizado en el mismo.

Para cada uno de los modos anteriores, se requiere llevar a cabo las siguientes instalaciones:

Modo local.

Para instalar el Sistema en modo local, es necesario contar con una computadora personal con las características mínimas que previamente se señalaron.

En este equipo se debe instalar el siguiente software en el orden que se apunta:

1. Sistema operativo MS DOS.
2. MS Windows.
3. Borland Paradox (completo o solo módulo de "runtime").
4. Archivos del SIGCE.
5. Crear el icono del SIGCE en el administrador de programas de Windows.

Modo compartido.

Para instalar el Sistema en modo compartido, se debe contar con una PC por cada usuario, con las características mínimas que se señalaron en el punto anterior, conectadas a una red local que cuente con un servidor como el especificado anteriormente.

Para este tipo de operación, se debe decidir la configuración que Paradox tendrá en la red. De cualquier modo, los cinco pasos anteriores deben llevarse a cabo con la diferencia de que algunos serían instalados en la PC y otros en el servidor de archivos.

1. Sistema operativo MS DOS. (En las PC's)
2. MS Windows. (En el servidor o en las PC's)
3. Borland Paradox (En el servidor o en las PC's).
4. Archivos del SIGCE (En el servidor).
5. Crear el icono del SIGCE en el administrador de programas de Windows (En cada una de las PC's).

El SIGCE requiere de 9 Mb disponibles de espacio en disco para los archivos del mismo, además se debe disponer de espacio adicional para las tablas de la base de datos del sistema, ya que conforme vayan creciendo, ocuparán más espacio del que actualmente tienen (archivos con terminación .DB). Estas tablas son las siguientes:

Archivo:	Contenido:
AGENTE.DB	Agentes registrados.
CALENEMI.DB	Calendario de emisión de pólizas.
CARATULA.DB	Carátulas correspondientes a una cuenta especial.
CUENTA_A.DB	Relación entre agentes y cuentas especiales.
CUENTA_E.DB	Catálogo de cuentas especiales.
DIVIDEN.DB	Datos referentes a dividendos.
EMPRESA.DB	Perámetros de la empresa para presentación.
GRUPO_CU.DB	Grupos de cuentas especiales.
OFICINA.DB	Datos de las oficinas a las cuales pertenecen los agentes.
POLIZA.DB	Pólizas de seguros de las cuentas especiales.
PRESUPUE.DB	Presupuestos anuales por cuenta especial.
PRIMA_PE.DB	Primas emitidas por cada cuenta especial a un periodo determinado.
USUARIOS.DB	Usuarios del Sistema.

Además requiere de el espacio necesario para la instalación de Borland Paradox .

2. Configuración de Paradox para Windows en forma compartida.

Las instrucciones que se proporcionan en este punto están destinadas a los administradores o supervisores de red, es decir, a los usuarios que disponen de todos los derechos de acceso a los directorios de la red.

Configuraciones de red de Paradox

En Paradox existen tres tipos de configuración para la utilización compartida de datos en red, que se clasifican en este apartado de acuerdo con su velocidad de funcionamiento (del más rápido al más lento) y su costo (del más caro al más barato). Para aumentar el rendimiento de los dos últimos tipos de configuración.

1. Configuración local.

Windows y Paradox están instalados en los discos duros de las diferentes estaciones de trabajo, de modo que es posible ejecutar ambos programas a nivel local y acceder al mismo tiempo a los datos de Paradox contenidos en directorios de red compartidos.

2. Configuración combinada.

Windows está instalado en el disco duro local de cada una de las estaciones de trabajo, pero sólo hay una copia de Paradox con licencia para red local en el servidor. Estas estaciones ejecutan Windows a nivel local, pero acceden al programa Paradox y a sus datos a través de los directorios de red compartidos.

3. Configuración de servidor.

Las estaciones de trabajo utilizan una única copia de Windows y de Paradox con licencia para red local en el servidor, de forma que ejecutan ambos programas desde la red y acceden a los datos de Paradox a través de los directorios de red compartidos.

Salida incorrecta del programa.

Si un usuario que trabaja con Paradox en red sale del programa de forma incorrecta, ninguna otra persona puede ocupar su lugar. Este caso puede darse cuando, por ejemplo, un usuario apaga la estación de trabajo sin terminar la sesión con Paradox.

Para restaurar el número de usuarios con acceso autorizado, es preciso conectarse de nuevo a la red desde la misma estación de trabajo, sin volver a arrancar Paradox. Al realizar esta operación, el programa detecta que dicho usuario ya no lo utiliza y da acceso a otro.

Precaución: Si se sale de Paradox de forma incorrecta, como al reinicializar el ordenador o apagarlo, los datos pueden perderse o quedar en mal estado.

Archivos de control de red.

Paradox para Windows utiliza el archivo de control de red PDOXWIN.USR para regular el número de usuarios que trabajan con el programa en red y el archivo PDOXUSRS.NET para controlar la utilización compartida de tablas en la red.

Instalación del archivo PDOXUSRS.NET.

El directorio del archivo PDOXUSRS.NET de Paradox para Windows se especifica mediante la Utilidad de configuración ODAPI (en las instrucciones de instalación que se proporcionan más adelante se indica cómo efectuar esta operación). PDOXUSRS.NET debe incluirse en un directorio de datos compartido de la red único al que tengan derechos de acceso de lectura/escritura/creación todos los usuarios de Paradox para Windows.

PDOXUSRS.NET no debe instalarse en un disco duro local, ya que, en caso de hacerlo, es posible que los usuarios de otras estaciones de trabajo no puedan iniciar Paradox o acceder a los datos compartidos.

El archivo PDOXUSRS.NET es específico de cada sesión. Así, cuando el primer usuario inicia Paradox para Windows en la red, éste busca el archivo en la posición especificada y, si no lo encuentra, lo genera en dicha posición para que puedan acceder a él otros usuarios. Si el archivo se borra por error, los programas vuelven a crearlo.

Muchos sistemas de red permiten asignar varias letras de unidad a la misma unidad de disco duro compartida. Aunque en Paradox es posible acceder al archivo PDOXUSRS.NET desde varias unidades, la vía de acceso debe ser siempre la misma.

El archivo PDOXUSRS.NET debe encontrarse siempre en la misma posición, ya que, si hay varias copias de este archivo activos durante una misma sesión de Paradox, es posible que algunos usuarios no puedan acceder a los datos compartidos. Así, si se utilizan varios archivos PDOXUSRS.NET, los usuarios de un mismo archivo podrán compartir los datos entre sí, pero no con los de otro archivo.

Enlaces duales.

Paradox identifica las tablas de una red mediante su unidad, vía de acceso y nombre de tabla. Nunca deben utilizarse dos unidades lógicas para hacer referencia a la misma tabla durante una sesión, ya que, de lo contrario, se crea un enlace dual, lo cual significa que un mismo usuario ha utilizado dos nombres distintos para la misma tabla. Esto podría crear serios problemas.

Los usuarios de estaciones de trabajo diferentes pueden utilizar, sin problemas, cualquier unidad lógica para acceder a una tabla.

Directorios.

Antes de instalar Paradox y trabajar con él, conviene familiarizarse con estos tipos de directorios:

- El directorio de ODAPI (PDOXWIN\ODAPI, por defecto).
- El directorio del sistema de Paradox (PDOXWIN, por defecto).
- Los directorios de datos compartidos de Paradox (SIGCE, para alojar archivos del sistema).
- Los directorios personales.
- Los directorios de trabajo.

Directorio de ODAPI.

Cuando se instala Paradox, éste introduce de forma automática los archivos de ODAPI, incluido ODAPI.CFG, en C:\WINDOWS\SYSTEM. Estos archivos permiten utilizar los programas ODAPI de forma simultánea, entre ellos Paradox para Windows y Quattro Pro para Windows.

Si los archivos de ODAPI de una estación de trabajo no se encuentran en WINDOWS\SYSTEM o WINDOWS, es necesario especificar la vía de acceso a dichos archivos en la sentencia PATH.

Los usuarios de Paradox deben tener como mínimo derechos de acceso de lectura al directorio que contiene los archivos ODAPI. Si sólo se tiene acceso de lectura al archivo ODAPI.CFG, éste no puede modificarse con la Utilidad de configuración ODAPI.

Configuración local.

Si se realiza la configuración local, es probable que no se desee modificar la vía de acceso por defecto C:\WINDOWS\SYSTEM, ya que en este tipo de configuración WINDOWS\SYSTEM suele estar situada en la unidad C de todos los discos duros locales.

Los archivos de ODAPI pueden incluirse en una unidad y directorio distintos de C:\WINDOWS\SYSTEM, en cuyo caso es necesario especificarlos en la sentencia PATH de las estaciones de trabajo que no usan dicha vía de acceso.

Configuración combinada.

Cuando se realiza la configuración combinada, WINDOWS\SYSTEM suele situarse en la unidad C del disco duro local de todas las estaciones de trabajo. Sin embargo, al instalar Paradox con esta tipo de configuración, es necesario incluirlo primero en la red desde la estación de trabajo del administrador de red. Así, si C:\WINDOWS\SYSTEM se deja intacta, los archivos de ODAPI se incluyen en esta vía de acceso de la estación del

administrador de red para que después puedan transferirse a los demás discos duros locales.

Cuando se instale Paradox por primera vez en la red desde la estación de trabajo del administrador de red, incluir los archivos de ODAPI en un directorio de red y, a continuación, se deben copiar en los discos duros locales de las estaciones de trabajo. Los archivos ODAPI de la red sirven como copias de seguridad de los contenidos en las estaciones y el archivo ODAPI.CFG, como archivo de configuración por defecto.

Instalar los archivos de ODAPI en otro directorio de la red, cuyo nombre puede ser PDOXWIN\ODAPI. Una vez realizada la instalación, es posible que se desee definir el directorio de red que contiene los archivos de ODAPI por defecto y de copia de seguridad como de sólo lectura a fin de evitar que los usuarios modifiquen el archivo ODAPI.CFG por defecto mediante la Utilidad de configuración ODAPI.

Copie los archivos de ODAPI en la unidad y directorio C:\WINDOWS\SYSTEM de cada uno de los discos duros locales. Estos archivos pueden incluirse en otra unidad y directorio, pero es necesario especificarlos en la sentencia PATH de las diferentes estaciones de trabajo.

Una vez terminada la instalación desde la estación de trabajo del administrador de red, es preciso configurar todas las demás estaciones. El directorio de ODAPI.CFG se especifica mediante la Utilidad de Configuración local, para lo cual es necesario seguir las instrucciones que se indican más adelante.

Configuración de servidor.

Si se realiza la configuración de servidor, incluir los archivos ODAPI en el directorio de red WINDOWS, modificar la vía de acceso por defecto C:\WINDOWS\SYSTEM del cuadro de texto de ODAPI del programa de instalación y sustituir la unidad por defecto C por la unidad de red lógica asignada al directorio WINDOWS de la red. Como la versión para red de Windows se instala por defecto en el directorio WINDOWS, y no en WINDOWS\SYSTEM, también es necesario cambiar éste por WINDOWS.

Es posible instalar los archivos de ODAPI en un directorio de red distinto de WINDOWS. Si éste es el caso, y se desea que accedan a ellos todos los usuarios, se debe incluir la unidad y directorio correspondientes en la sentencia PATH de cada estación de trabajo.

También puede instalarse el archivo por defecto ODAPI.CFG en los discos duros locales a fin de proporcionar a los usuarios configuraciones independientes. En este caso, el archivo ODAPI.CFG de la red sirve como archivo de configuración por defecto y de copia de seguridad. Una vez realizada la instalación, es posible que se desee definir el directorio de red que contiene los archivos de ODAPI por defecto y los de copia de seguridad como de sólo lectura a fin de evitar que los usuarios modifiquen el archivo ODAPI.CFG por defecto con la Utilidad de configuración ODAPI. Sin embargo, si se precisa instalar otro programa ODAPI, como Quattro Pro para Windows, deberá tener todos los derechos de acceso al directorio correspondiente.

Si se copia el archivo ODAPI.CFG en los discos duros locales para proporcionar a los usuarios configuraciones independientes, es probable que se desee incluirlo en otro

directorio, por ejemplo, ODAPI. No se debe olvidar introducir la unidad y directorio de dicho archivo en la sentencia PATH de cada estación de trabajo.

Una vez terminada la instalación desde la estación de trabajo del administrador de red, es preciso configurar las demás estaciones. El directorio de ODAPI.CFG se especifica mediante la Utilidad de Configuración local, para lo cual es necesario seguir las instrucciones que se indican mas adelante.

Directorio del Sistema de Paradox, PDOXWIN.

Este directorio contiene los archivos del sistema o programa de Paradox. A continuación se muestra una descripción de cada una de las configuraciones:

- Si se realiza la configuración de tipo local, dejar la vía de acceso por defecto C:\PDOXWIN tal cual. Se puede sustituir la unidad C por otra unidad de disco duro local, como D o E, y cambiar el nombre del directorio por otro.
- Si se realiza la configuración combinada o de servidor, sustituir la unidad por defecto C por la unidad lógica de la red donde se va a incluir Paradox. También se puede definir como subdirectorio de un directorio compartido de aplicaciones de red, si se dispone de él, y cambiar el nombre de PDOXWIN por otro.

Directorio de datos compartidos, SIGCE.

Ya sea que se realice una configuración u otra, crear como mínimo un directorio de datos compartido al que tengan derechos de acceso de lectura/escritura/creación todos los usuarios del sistema. Este directorio es necesario para incluir el archivo de control PDOXUSRS.NET y los objetos compartidos del sistema.

Todos los usuarios deben tener derechos de acceso de lectura/escritura/creación al directorio de red que contiene PDOXUSRS.NET, ya que el primer usuario que inicia el programa crea este archivo y los demás realizan operaciones de lectura y escritura en él durante las sesiones con Paradox.

Directorios personales.

Los directorios personales se usan para guardar los objetos temporales de los usuarios del programa, por ejemplo, las tablas SOLUCION. Estos objetos tienen que incluirse en un único directorio, de lo contrario, los de un usuario pueden sobreescribir los de otro. Los usuarios de Paradox en red deben disponer de un directorio personal propio.

Cuando se realiza la configuración local, Paradox crea de forma automática un subdirectorio denominado PERSONAL en el directorio del sistema PDOXWIN, de forma que PDOXWIN\PERSONAL constituye el directorio personal por defecto en este tipo de configuración.

Cuando se trata de una configuración combinada y de servidor, Paradox detecta que la instalación se realiza en la red (no en una unidad local) y, por ello, no crea de forma automática el subdirectorio PERSONAL en el directorio PDOXWIN de la red. Si en este

tipo de configuración no se especifica un directorio personal de usuario, el programa utiliza el directorio temporal de Windows. Este proceso es aplicable a la configuración combinada, ya que cada estación de trabajo dispone de un directorio WINDOWS\TEMP local. En el caso de la configuración de servidor, es preciso especificar un directorio único personal para cada estación de trabajo, ya que Windows y el correspondiente directorio personal están situados en la red.

Crear los siguientes directorios personales para estos entornos de red:

- Si la estación de trabajo dispone de una unidad de disco local, utilizar CAPDOXWIN en el caso de la configuración combinada y de servidor y CAPDOXWINPERSONAL en el caso de la configuración local.
- Si la estación carece de disco duro local, utilizar el directorio personal de la red como directorio personal del usuario y asignarle derechos de acceso completos.

Se debe tener presente los siguientes puntos cuando se especifique un directorio personal:

- No es posible usar una unidad de disquetes como directorio personal.
- Si se realizan consultas en tablas de gran extensión, el directorio personal debe estar situado en una unidad con suficiente espacio en disco.

Los directorios personales de los usuarios se especifican mediante la Utilidad de Configuración local, cuyo uso se explica en las instrucciones de instalación que se proporcionan más adelante.

Directorios de trabajo.

Cuando se ejecuta Paradox, se tiene acceso a dos directorios de trabajo:

El cuadro de diálogo Propiedades del elemento de programa contiene los cuadros de texto Línea de comando y Directorio de trabajo. Si no se introduce la unidad y vía de acceso a PDOXWIN.EXE en el primer cuadro de texto, se deben incluir en el segundo cuadro.

El directorio de trabajo de datos de Paradox contiene los archivos de datos del programa con los que trabaja. Cuando se realiza una instalación de tipo local, Paradox crea de forma automática un subdirectorio denominado TRABAJO en el directorio del sistema PDOXWIN, de forma que PDOXWINTRABAJO es el directorio de trabajo por defecto en la configuración local.

Cuando se trata de la configuración combinada y de servidor, Paradox detecta que la instalación se realiza en la red (no en una unidad local) y, por ello, no crea de forma automática un subdirectorio TRABAJO en el directorio PDOXWIN de la red. Se deben especificar los directorios de trabajo de las distintas estaciones de trabajo mediante la Utilidad de Configuración local, cuyo uso se explica más adelante.

3. Instalación de Paradox para Windows.

Ya sea que se realice un tipo de configuración u otro, se debe instalar Windows primero y después Paradox. En este apartado se da por sentado que la estación de trabajo con la que se empieza a trabajar es la del administrador de red.

Instalación de Windows.

Seguir las instrucciones de instalación de la documentación de Windows:

- Si se realiza la configuración local o combinada, es necesario instalar una copia de Windows en el disco duro local de cada estación de trabajo.
- Si se realiza la configuración de servidor, instalar la versión de red de Windows en la red desde la estación de trabajo del administrador de red. Una vez realizada esta instalación, seguir las instrucciones de la documentación de Windows para configurar las diferentes estaciones de trabajo a fin de permitir el acceso a dicho programa.

Instalación de Paradox.

- Si se efectúa la configuración local, instalar una copia de Paradox en el disco duro local de cada estación de trabajo.
- Si se trata de la configuración combinada o de servidor, instalar una copia de Paradox en la red desde la estación de trabajo del administrador. Una vez instalado el programa, configurar cada una de las estaciones de trabajo para permitir su uso. Para ello, seguir las instrucciones que se indican a continuación.

Configuración combinada y de servidor.

Cuando se haya instalado Paradox en la red, conectar las estaciones de trabajo al programa. Si se realiza una instalación de tipo local, pase directamente al apartado "Ejecución de la Utilidad de Configuración local."

Creación de grupos de programa de Paradox en las distintas estaciones de trabajo.

Es posible crear en el programa Windows de cada estación de trabajo una ventana de grupo de Paradox (la estación del administrador de red ya contiene esta ventana porque el programa de instalación lo genera de forma automática). Si desea incluir Paradox en una ventana de grupo existente, pase directamente al apartado "Creación de Iconos de Paradox en las distintas estaciones de trabajo".

Arrancar Windows desde cada estación de trabajo y:

1. Elegir Archivo | Nuevo en el Administrador de programas de Windows.

2. Cuando aparezca el cuadro de diálogo Nuevo elemento, elegir Grupo de programas y después Aceptar.
3. Incluir en el cuadro de texto Descripción el título del grupo de programa de Paradox, por ejemplo, Paradox para Windows.
4. Seleccionar Aceptar. Aparece una ventana de grupo de Paradox vacía con el título introducido en el paso anterior.

Creación de iconos de Paradox en las distintas estaciones de trabajo.

Crear en la ventana de grupo del programa de cada estación de trabajo un icono de Paradox (la estación del administrador de red ya contiene este icono porque el programa de instalación lo crea de forma automática).

1. Una vez seleccionada la ventana de grupo de Paradox, elija Archivo | Nuevo en el Administrador de programas de Windows.
2. Elegir Elemento de programa y después Aceptar en el cuadro de diálogo Nuevo elemento.
3. Incluir en el cuadro de texto Descripción el título de Paradox, por ejemplo, Paradox para Windows.
4. Especificar la unidad y vía de acceso completa al archivo PDOXWIN.EXE de la red. Para ello, realizar uno de estos pasos:
 - Introducir en el cuadro de texto Línea de comando la unidad y vía de acceso completa al archivo PDOXWIN.EXE de la red, y el nombre de éste.
 - Incluir el nombre de archivo PDOXWIN.EXE en el cuadro de texto Línea de comando y la unidad y vía de acceso al directorio de dicho archivo en el cuadro Directorio de trabajo. Esta unidad y directorio (PDOXWIN, por defecto) también deben especificarse en la sentencia PATH correspondiente.

El directorio de trabajo definido en el cuadro de diálogo Propiedades del elemento de programa es diferente del directorio de datos de Paradox especificado mediante la Utilidad de Configuración local.

5. Elegir Aceptar. Aparece un cuadro de diálogo indicando que este programa está situado en un directorio de red y puede no estar disponible posteriormente.
6. Seleccionar Sí. El icono de Paradox aparece en la ventana de grupo del programa.

Asignación de iconos a las utilidades de Paradox en los distintos estaciones de trabajo.

Crear en la ventana de grupo de Paradox de cada estación de trabajo los iconos correspondientes a la Utilidad de configuración ODAPI, la Utilidad de Tablas y la Utilidad

de Configuración local. Seguir las mismas instrucciones del apartado anterior para cada utilidad. Para ello, basta con cambiar los valores de Descripción y Archivo.EXE.

Utilidad	Descripción	Archivo EXE
De configuración ODAPI	Utilidad de configuración ODAPI	ODAPICFG.EXE
De Tablas	Utilidad de Tablas	TUTILITY.EXE
De Configuración local	Utilidad de Configuración local	PWLOCAL.EXE
De No. de Serie	Utilidad de No. de Serie	PWUPDATE.EXE

Todos estos archivos EXE se encuentran en el directorio del sistema PDOXWIN, junto con PDOXWIN.EXE.

Ejecución de la Utilidad de Configuración local.

Especificar los directorios personales y de trabajo de los usuarios, así como el archivo de configuración de ODAPI, ODAPI.CFG, mediante la Utilidad de Configuración local. Una vez en la estación de trabajo:

1. Ejecutar la Utilidad de Configuración local, para lo cual es necesario hacer doble clic sobre el icono correspondiente de la ventana de grupo de Paradox.

Si se realiza la configuración combinada o de servidor, los cuadros de texto de todas las estaciones de trabajo, excepto la del administrador de red, estarán vacíos; si se efectúa la configuración local, el cuadro de texto Directorio de trabajo contiene la vía de acceso C:\PDOXWIN\TRABAJO, el cuadro Directorio personal incluye la vía C:\PDOXWIN\PERSONAL y el cuadro Archivo de configuración de ODAPI contiene la unidad y vía de acceso al archivo ODAPI.CFG, y el nombre de éste.

2. Teclar en el cuadro de texto Directorio de trabajo la unidad y vía de acceso completa al directorio local o de red que se va a utilizar como directorio de trabajo de los archivos de datos de Paradox. Este es distinto del directorio de trabajo del DOS, que debe ser el del sistema (PDOXWIN).

Si se efectúa una configuración de tipo local, el cuadro de texto Directorio de trabajo debe contener la unidad y vía de acceso completa al subdirectorio SIGCE incluido en el directorio del disco duro local donde se ha instalado Paradox.

Si se realiza la configuración combinada o de servidor, el cuadro de texto Directorio de trabajo de todas las estaciones de trabajo, excepto la del administrador de red, estará vacío, por lo que es necesario especificarlo de forma manual.

También conviene utilizar este directorio para generar y utilizar el archivo PDOXUSRS.NET de cada usuario. La posición de este archivo se especifica con la Utilidad de configuración ODAPI.

3. Teclar en el cuadro de texto Directorio personal la unidad y vía de acceso completa al directorio local o de red que va a usar como directorio personal por defecto de la estación de trabajo. Es preciso asignar a cada usuario de red de Paradox un directorio personal único en el que guardar los objetos temporales, como las tablas SOLUCION (nunca debe asignarse el mismo directorio personal a varios usuarios).

Si se realiza una configuración a nivel local, el cuadro de texto Directorio personal debe contener la unidad y vía de acceso completa al subdirectorio PERSONAL creado en el directorio donde se ha instalado Paradox. Si la instalación se ha efectuado en la unidad y directorio por defecto, dicho cuadro de texto incluirá la vía C:\PDOXWIN\PERSONAL, que puede dejarse tal cual, ya que se encuentra en el disco duro local.

Si se realiza una configuración combinada o de servidor, el cuadro de texto Directorio personal de todas las estaciones de trabajo, excepto la del administrador de red, estará vacío. Si la estación de trabajo dispone de un disco duro local, incluir en él un directorio denominado PDOXWIN y definirlo como personal. Si no se dispone de él, utilizar el directorio de usuario de la red como directorio personal y definirlo con todos los derechos de acceso.

4. Incluir en el cuadro de texto Archivo de configuración de ODAPI la unidad y vía de acceso completa al archivo ODAPI.CFG, y el nombre de éste.

Si se realiza una configuración de tipo local, el cuadro de texto Archivo de configuración de ODAPI debe contener la unidad y vía de acceso completa al directorio del disco duro local donde se han instalado los archivos de ODAPI. Si ha efectuado la instalación en la unidad y directorio por defecto, dicho cuadro de texto incluirá la vía C:\WINDOWS\SYSTEM\ODAPI.CFG.

Se puede dejar la vía de acceso C:\WINDOWS\SYSTEM\ODAPI.CFG intacta. Si se modifica, se deberá cambiar la posición del archivo ODAPI.CFG, según sea necesario.

Si se realiza una configuración combinada o de servidor, el cuadro de texto Archivo de configuración de ODAPI de todas las estaciones de trabajo, excepto la del administrador de red, estará vacío.

Teclar la unidad y vía de acceso completa el archivo ODAPI.CFG de la red, si éste ha de ser el único archivo ODAPI.CFG empleado por los usuarios. Si se copia este archivo en el disco duro local de las distintas estaciones de trabajo a fin de que los usuarios puedan adaptar la configuración por defecto a sus propias necesidades, teclar la unidad y vía de acceso completa a dicho archivo, y el nombre de éste. Comprobar que también se incluyen en la sentencia PATH de las estaciones.

Configuración de Paradox para la utilización compartido de datos de red.

Una vez instalado Paradox, es preciso configurar las estaciones de trabajo a fin de permitir el acceso a los datos de la red. Para ello, ejecutar la Utilidad de configuración ODAPI con objeto de especificar la posición del archivo PDOXUSRS.NET.

La vía de acceso al directorio de red del archivo PDOXUSRS.NET debe ser la misma en todas las estaciones de trabajo. No especificar un directorio de disco duro local. Realizar los siguientes pasos en cada una de las estaciones de trabajo:

1. Ejecutar la Utilidad de configuración ODAPI, para lo cual es preciso hacer doble clic sobre su icono en la ventana de grupo de Paradox. Aparece en pantalla el cuadro de diálogo Utilidad de configuración ODAPI.

2. Teclar en el cuadro de texto Directorio de archivos de control de red la unidad y vía de acceso al archivo PDOXUSRS.NET de la red. Almacenar este archivo en el directorio

principal de datos compartidos del programa y asignarle el nombre SIGCE. En este paso se da por sentado que ya se ha creado este directorio y que los usuarios tienen acceso de lectura/escritura/creación al mismo.

Ejecución de Paradox en los estaciones de trabajo.

Ya sea que se realice una configuración u otra, ejecutar Paradox en todas las estaciones de trabajo tras instalar el programa y configurarlo para permitir el acceso a los datos compartidos de la red. Si el programa no puede iniciarse, realizar de nuevo los pasos descritos en los apartados anteriores, en caso necesario.

4. Instalación del Sistema Gerencial para Cuentas Especiales.

Para la instalación de los archivos del Sistema Gerencial para Cuentas Especiales, se debe insertar el diskette de instalación en una unidad de disco flexible (de un equipo conectado a la red local en su caso) y ejecutar el siguiente comando en el prompt de DOS:

instalar -d < directorio de instalación >

En donde "directorio de instalación" es la unidad y ruta completa del directorio de datos compartidos de Paradox (por ejemplo: F:\SIGCE), para el caso de configuración combinada o de servidor; o la unidad de disco duro y ruta completa del directorio de trabajo de Paradox para el caso de configuración local (por ejemplo: C:\PDOXWIN\TRABAJO o C:\SIGCE).

El comando "instalar" invoca la descompresión del archivo instaler.exe, el cual fué generado con las utilerías de compresión y descompresión de archivos PKWARE.

En el momento en que los archivos del SIGCE se encuentran en el directorio de trabajo, se debe crear el icono correspondiente para cada una de las estaciones de trabajo, cualquiera que sea la configuración que se use para el funcionamiento de Paradox. El icono se crea de la misma forma que se procedió para la creación de iconos del sistema de Paradox que se describió anteriormente, con la variante de que en el cuadro de texto Línea de Comando se debe colocar el siguiente comando:

<ruta del directorio de sistema de Paradox>\PDOXWIN.EXE <ruta del directorio de trabajo>\SIGCE.SSL

ejemplo:

C:\PDOXWIN\PDOXWIN.EXE F:\SIGCE\SIGCE.SSL

5. Notas sobre el menú "Utilerías del Sistema".

El menú de Utilerías del SIGCE, contiene las siguientes dos opciones:

1. Administración de usuarios.
2. Mantenimiento de datos.

La opción de Administración de usuarios tiene como propósito el limitar el acceso a las cuatro opciones del menú principal a diferentes usuarios del sistema, además de agregar o borrar más usuarios.

Al acceder al Sistema, este muestra el nombre asignado al usuario actual del sistema operativo (a través de la tabla USUARIOS,DB), el cual puede ser cambiado a través del botón Establecer Usuario. Se hace notar que sólo los usuarios anotados en dicha tabla tendrán acceso a determinada opción (de las cuatro iniciales del menú principal), de acuerdo a lo asignado en esta pantalla del sistema (para cambiar estos privilegios se debe tener acceso al menú de Utilerías del SIGCE).

En la parte de mantenimiento de datos se muestran la subopciones:

- Reparar tablas: Que invoca la utilería de Paradox para la reparación de tablas, como en el caso de una salida irregular del sistema que dañe una tabla.
- Respaldo: Que invoca la utilería de compresión de archivos PKZIP, la cual genera un archivo de datos comprimido que se puede recuperar con la siguiente opción.
- Recuperación: Descomprime a través de la utilería PKUNZIP el archivo que genera la opción Respaldo. Esto lo hace directamente en el directorio de trabajo del SIGCE, por lo que se debe tener precaución para no sobrescribir una versión actualizada por error.

Apéndice D.

**Sistema de Información Gerencial para
Cuentas Especiales.
SIGCE**

Versión 1.0

Manual del Usuario

INTRODUCCIÓN

El sistema de Información Gerencial para Cuentas Especiales (SIGCE), está programado en Paradox Versión 4.5 para Windows. Este manual del usuario le explicará la forma de acceder al sistema y sus diferentes módulos. Es recomendable que el usuario este familiarizado con Windows, pero no es indispensable.

1.1. Convención Tipográfica.

Los manuales de SIGCE utilizan fuentes especiales para establecer la distinción entre las teclas que deben pulsarse, los nombres de los objetos, los comandos, menús o el texto que introduce el usuario

Convención Tipográfica	Se aplica a	Ejemplo
Negrita	Cualquier mensaje mostrado por SIGCE	SIGCE muestra el mensaje: El registro está bloqueado.
<i>Cursiva</i>	Nombres de objetos de SIGCE, termino de glosario, elementos de ejemplo, o palabras con significado o énfasis especial.	<i>Tabla maestra</i> , base de datos relacional.
MAYÚSCULAS	Directorios y archivos del DOS, palabras reservadas, operadores, tipos de consultas.	Directorio SIGCE CLIENTES.DBF C:\WINDOWS
Inicial Mayúscula	Campos, comandos de menú, nombres de objetos.	Campo Precio, Comando Edición Cortar.
Fuentes de Teclado	Teclas de la computadora	F1: Intro
Fuentes de texto	Texto que debe introducir el usuario	No. de Cliente

El directorio de Trabajo del SIGCE es C:\SIGCE, dentro de este se encuentran las tablas de trabajo.

1.2. Iconos Utilizados.

Los iconos y botones utilizados en todo el sistema son los siguientes:

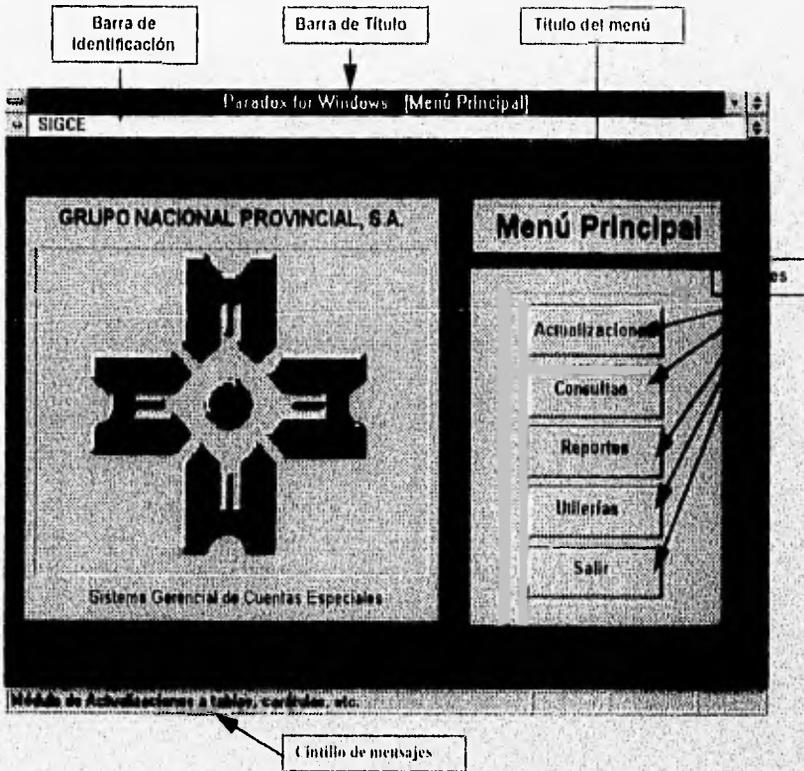


Cancelar. Aborta el movimiento.

1.3. Salir. *Regresa a la pantalla anterior, confirmando la salida.*

Características de las pantallas de SIGCE.

Las pantallas de menú de SIGCE están conformadas por:



- Barra de Título.** En esta barra aparecerá el nombre de la pantalla que se está ejecutando en ese momento, además del nombre del paquete en que fue programa el sistema.
- Barra de Identificación.** En esta aparecerá el nombre del sistema, en este caso SIGCE.
- Titulo del menú.** Es el nombre el menú o del módulo que se esta accedendo.
- Botones.** Son los objetos por medio de los cuales, al seleccionarlos, el sistema sabrá que opción se ejecutará.

Cintillo de mensajes. En este lugar aparecerán los mensajes de error, las leyendas de cada botón, etc.

Cada módulo tiene botones especiales, que se verán a detalle en cada caso.

1.3.1. Pantallas de Consulta

Las pantallas de consulta, tienen como en el caso de las de menú

Barra de Título. En esta barra aparecerá el nombre de la pantalla de consulta que se está ejecutando en ese momento, además del nombre del paquete en que fue programado el sistema.

Barra de Identificación. En esta aparecerá el nombre del sistema, en este caso SIGCE.

Título del menú. Es el nombre del menú o del módulo que se está accediendo.

Botones. Son los objetos por medio de los cuales, al seleccionarlos, el sistema sabrá que opción se ejecutará.

Los botones de esta pantalla, en orden son ::

Primero. Este botón nos posicionará en el primer registro de la tabla que cumpla con las características de la consulta.

Anterior. Este botón nos posicionará en el registro anterior al que este seleccionado.

Este botón nos posicionará en el primer registro de la tabla que cumpla con las características de la consulta.

Siguiente. Este botón nos posicionará en el registro siguiente al que este seleccionado.

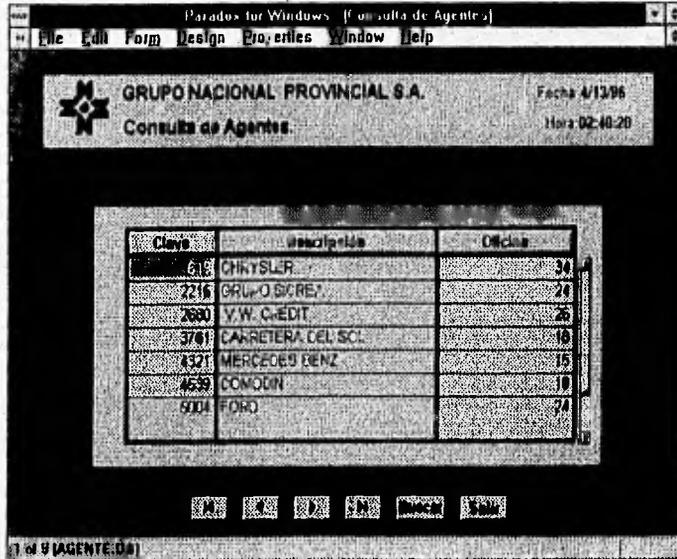
Último. Este botón nos posicionará en el último registro de la tabla que cumpla con las características de la consulta.

Buscar. Este botón desplegará el cuadro de diálogo de búsqueda, para localizar el registro o registros que cumplan con una condición dada, y posicionará el cursor en el primer registro que cumpla con la condición de búsqueda.

Salir. Este botón cancela el último movimiento y nos regresa al menú o pantalla anterior.

Cintillo de mensajes. En este lugar aparecerán los mensajes de error, las leyendas de cada botón, etc.

Ejemplo:



1.3.2. Pantallas de Actualización

Las pantallas de consulta tienen:

Barra de Título. En esta barra aparecerá el nombre de la pantalla de consulta que se está ejecutando en ese momento; además del nombre del paquete en que fue programado el sistema.

Barra de Identificación. En esta aparecerá el nombre del sistema, en este caso SIGCE.

Título del menú. Es el nombre del menú o del módulo que se está accediendo.

Botones. Son los objetos por medio de los cuales, al seleccionarlos, el sistema sabrá que opción se ejecutará.

Los botones de esta pantalla, en orden son ::

Alta. Este botón nos posicionará en un registro en blanco al final de la tabla.

Baja. Este botón dará de baja el registro en el que estemos posicionados.

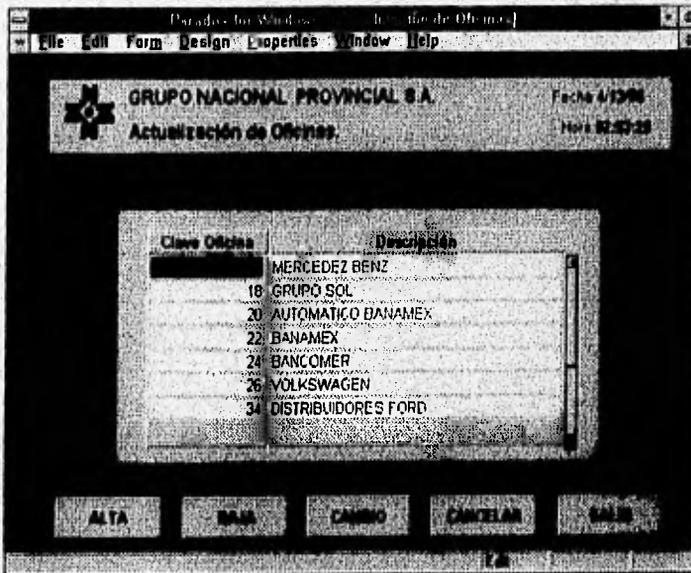
Cambio. Este botón desplegará el cuadro de diálogo de búsqueda, para localizar el registro a modificar y posicionará el cursor en el registro que cumpla con la condición dada.

Cancelar. Este botón deshará el último movimiento efectuado, cancelará la baja, el alta o el cambio.

Salir. Este botón confirma el último movimiento y regresa a la pantalla anterior o menú.

Cintillo de mensajes. En este lugar aparecerán los mensajes de error, las leyendas de cada botón, etc.

ejemplo:

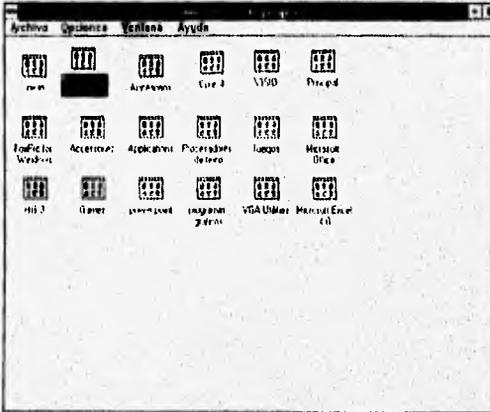


ACCESO AL SISTEMA

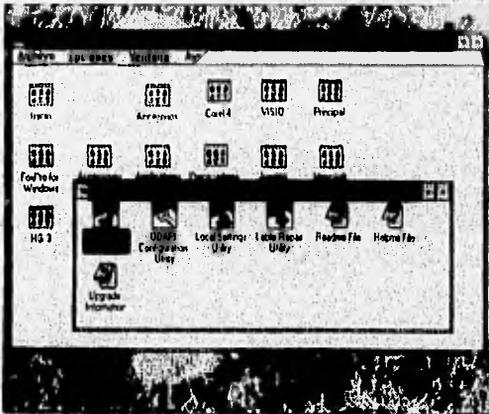
Se accesa a una sesión de Windows, desde el sistema operativo, tecleando:

C:\>WIN

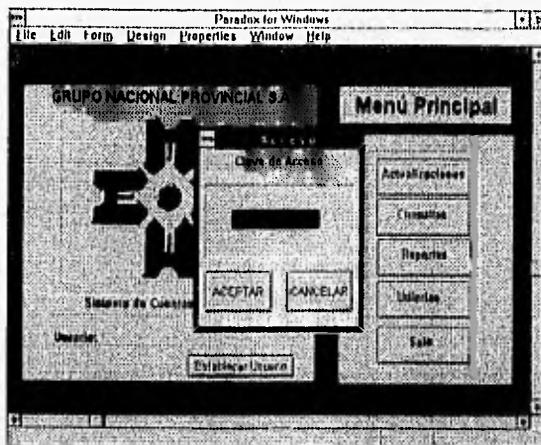
Después de abre la ventana de Paradox para Windows del Administrador de Programas. Como se muestra a continuación:



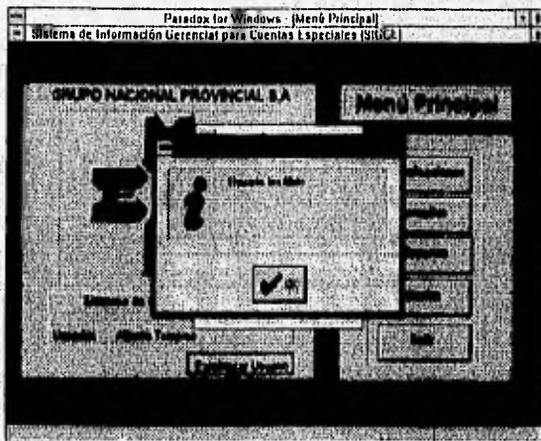
Se abre la ventana y se selecciona el icono de SIGCE, con doble clic se accesa al programa:



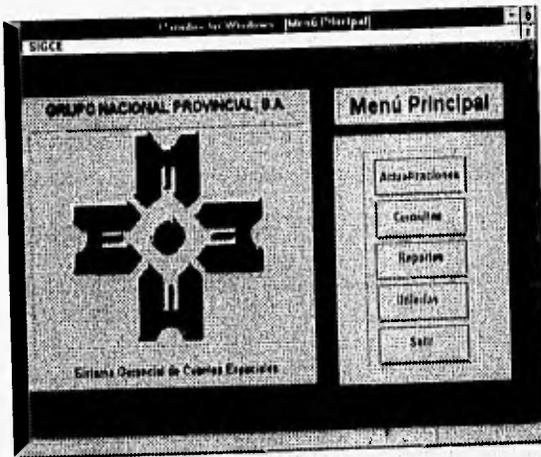
La pantalla que aparece es la pantalla principal del sistema SIGCE. En la cual se pedirá al usuario para poder acceder al sistema, recordando, que de acuerdo a el usuario que teclee serán los módulos a los que tendrá acceso.



En caso de que el usuario no exista, se desplegará el cuadro de diálogo siguiente, en donde el sistema nos avisa que es un usuario inválido, se deberá teclear OK para regresar nuevamente a la petición de usuario.

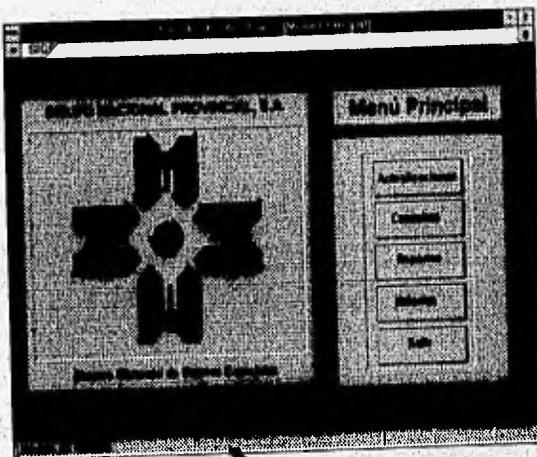


Después de teclear el usuario correcto, y dar clic en el botón de ACEPTAR quedara la pantalla principal como sigue:



En ella podemos elegir las opciones de Actualización, Consulta, Reportes y Utilerias. Al seleccionar alguno de estos módulos, el sistema nos posicionará en el menú de cada uno de ellos.

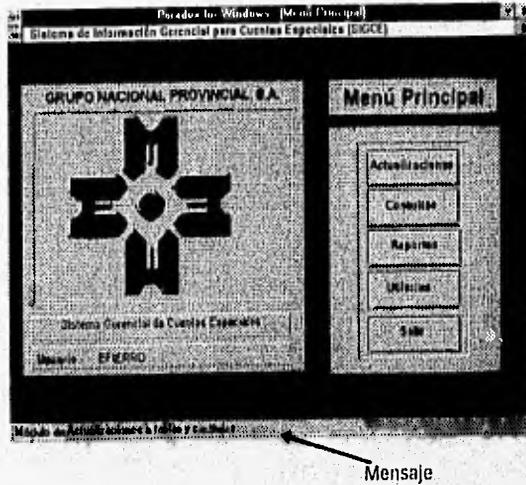
Al colocar el Mouse sobre cada uno de los botones del menú principal o de los diferentes menús, aparecerá en la parte inferior izquierda, una leyenda relativa al módulo que se va a acceder. En este lugar aparecen también los mensajes de error, entre otros.



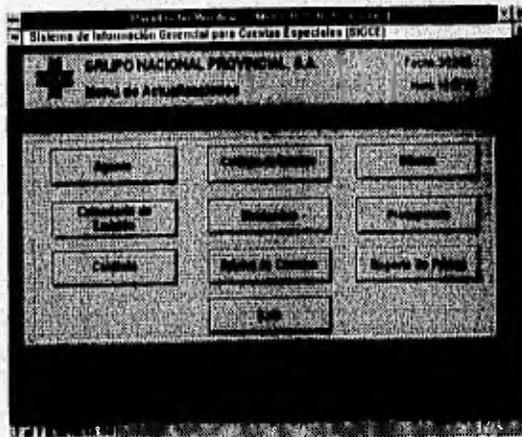
Mensaje al colocar el puntero en el botón de Utilerias.

MÓDULO DE ACTUALIZACIONES

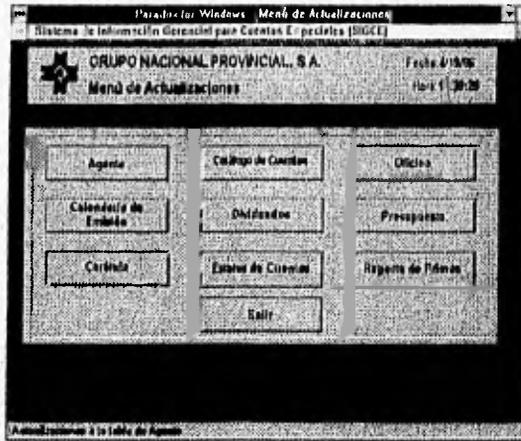
Como ya hemos mencionado al colocar el Mouse sobre el botón de Actualizaciones aparecerá una leyenda en la parte inferior izquierda que dice: Módulo de Actualizaciones a tablas, carátulas, etc. como se muestra a continuación:



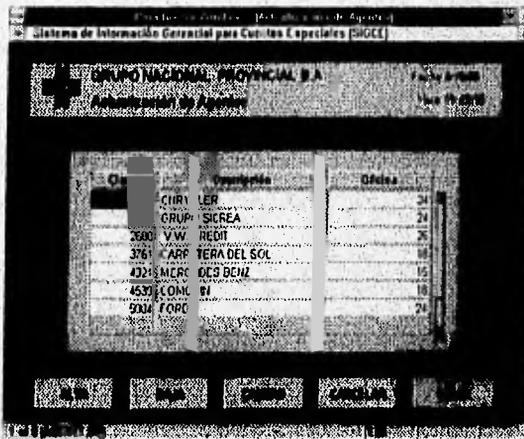
Para acceder este módulo se da clic sobre el botón y el menú de Actualizaciones aparecerá:



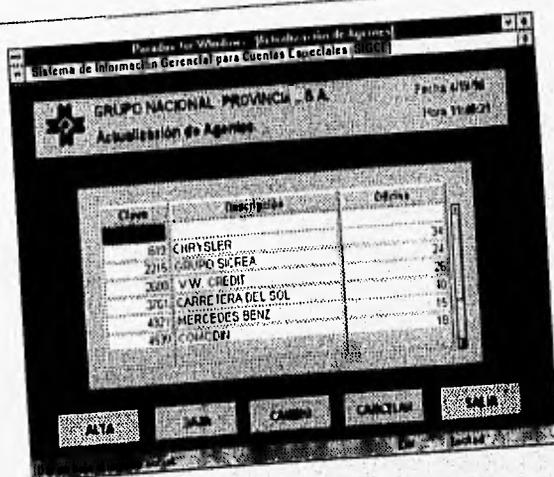
En este módulo contamos con nueve actualizaciones. Al seleccionar la cualquier opción se desplegará en el Cintillo de mensajes la leyenda correspondiente, como ejemplo tenemos el mensaje al colocar el Mouse sobre el botón de Agente:



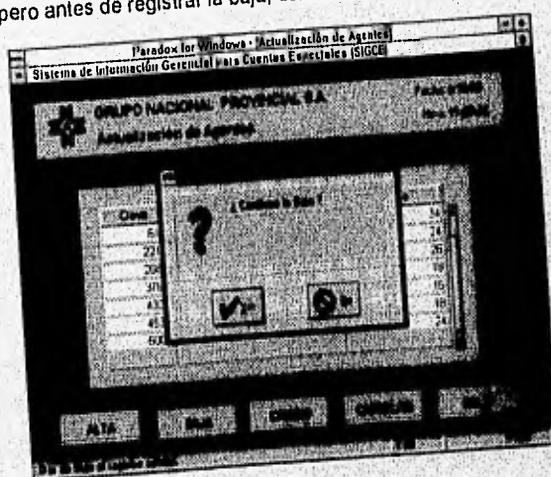
Al seleccionar la opción de Agentes, se abrirá la siguiente pantalla, en la que podremos actualizar la tabla de agentes:



Al seleccionar el botón de Alta, el sistema abrirá un registro en blanco al principio de la tabla, en donde podremos teclear los datos del nuevo agente, como se muestra a continuación:

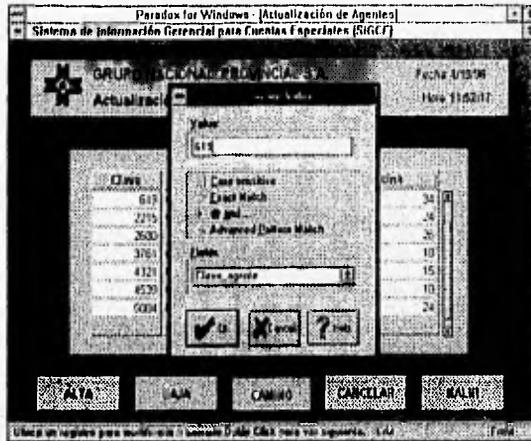


Al seleccionar el botón de baja, el sistema eliminará el registro en el que estamos posicionados, pero antes de registrar la baja, confirmará el movimiento:

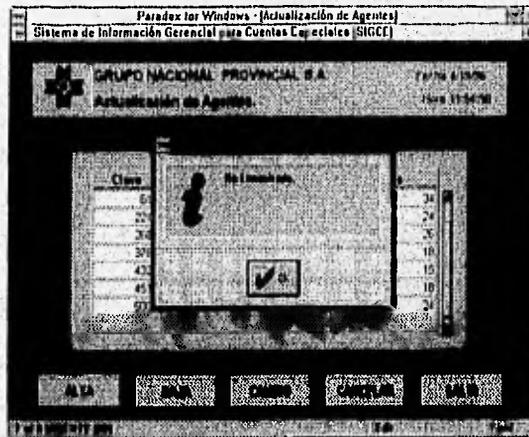


Si seleccionamos Yes, se confirmará la baja y se realizará el movimiento, si seleccionamos no, se descartará el último movimiento.

Si seleccionamos el botón de Cambio, aparecerá la siguiente caja de diálogo, en donde teclaremos el valor a buscar, recordando que buscará por el campo en el que estamos posicionados:



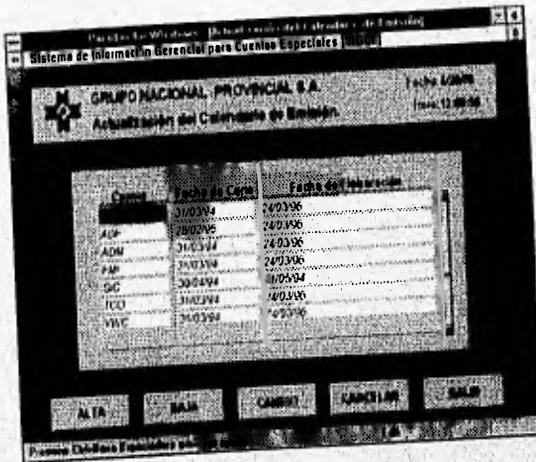
Si no existe ningún registro que coincida con la búsqueda se desplegará el siguiente mensaje :



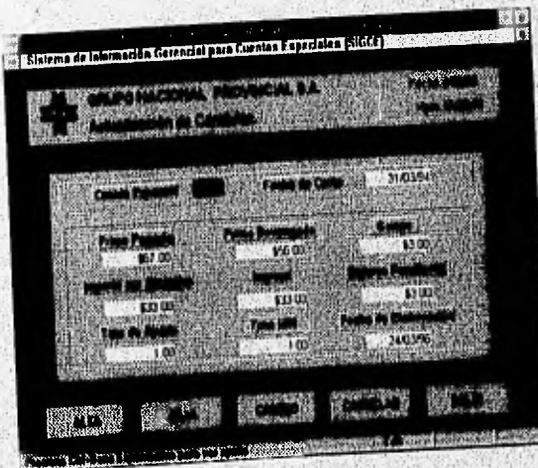
Si al realizar la búsqueda se localizaran mas de un registro que coincida con lo tecleado, para avanzar al registro siguiente que cumple con la condición, bastará colocar el puntero del Mouse sobre el botón de Cambio y dar doble clic con el botón izquierdo.

Las siguientes opciones del menú de Actualizaciones son las siguientes:

Calendario de Emisión:



Carátula



Catálogo de Cuentas:

Sistema de Información Gerencial para Cuentas Especiales (SIGCE)

GRUPO NACIONAL PROVINCIAL S.A. Fecha: 02/09/98
Actualización del Catálogo de Cuentas Hora: 12:07:28

Clase: Nombre: DISTRIBUIDORES FORD

Grupo: Tipo adm.: Tipo (porve):

Oficina: 34 Tipo adm.: Tipo gra.:

Agencia: 013 P.M. Inicial: 34 P.M. Final:

Inicio: 11/11/93 Fin: 11/11/93

ALTA BAJA CAMBIO CANCELAR SALIR

Dividendos:

Estatus de Cuentas:

Sistema de Información Gerencial para Cuentas Especiales (SIGCE)

GRUPO NACIONAL PROVINCIAL S.A. Fecha: 02/09/98
Actualización del Catálogo de Cuentas Hora: 12:07:28

Clave de la cuenta: Transferencia:

Fecha de inicio: 01/03/94 Operación:

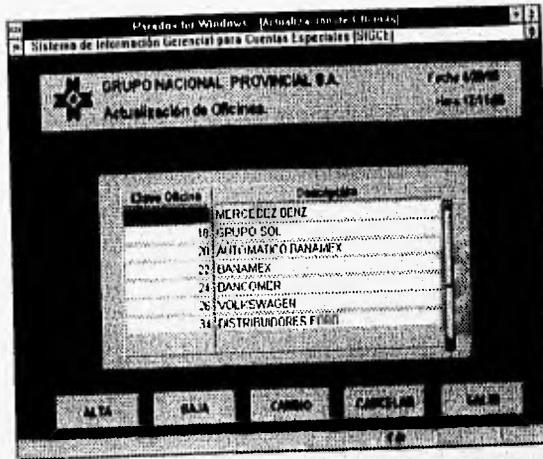
Fecha de fin: 30/09/94 Reserva:

Tipo de operación: Comisión:

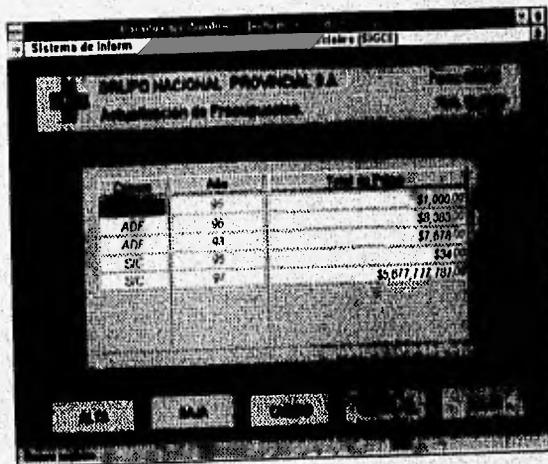
Interés:

ALTA BAJA CAMBIO CANCELAR SALIR

Oficina:



Presupuesto:



Reporte de Primas:

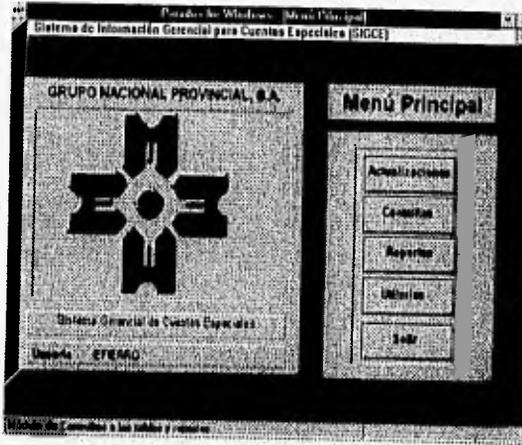
The screenshot shows a Windows application window titled "Actualización de Primas por Período" for "GRUPO NACIONAL PROVINCIAL S.A.". The window contains a form with the following fields and values:

Field	Value
Período actual:	11/11/04
Período:	11/11/04
Prima propia:	198.00
Prima asegurado:	179.00
Clave cambio:	CHD

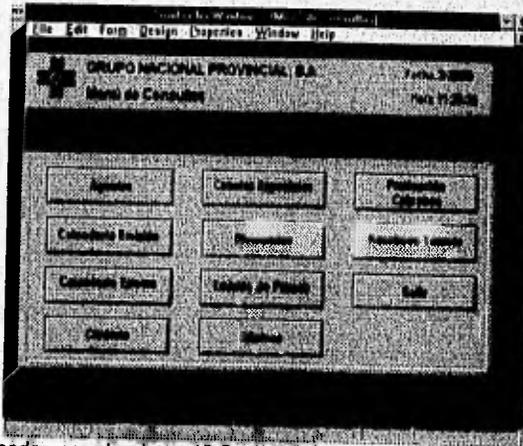
At the bottom of the form, there are five buttons: "AJUSTAR", "BORRAR", "CAMBIAR", "ACTUALIZAR", and "IMPRIMIR".

MODULO DE CONSULTAS

Al colocar el Mouse sobre el botón de Consultas del menú principal aparecerá la leyenda Módulo de Consultas a tablas y reportes, como se muestra a continuación:

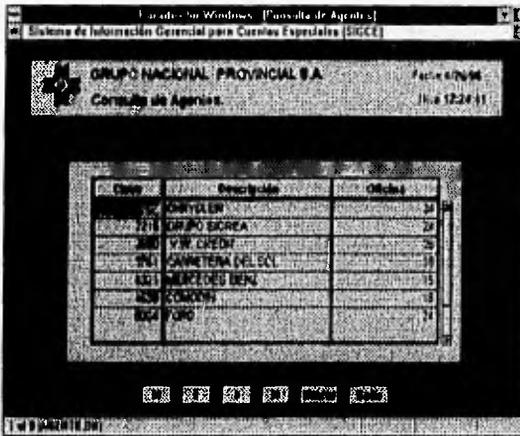


Al seleccionar con doble clic el botón de Consultas, aparecerá el menú del módulo de consultas:



En este menú podemos seleccionar 10 Opciones:

Agentes:

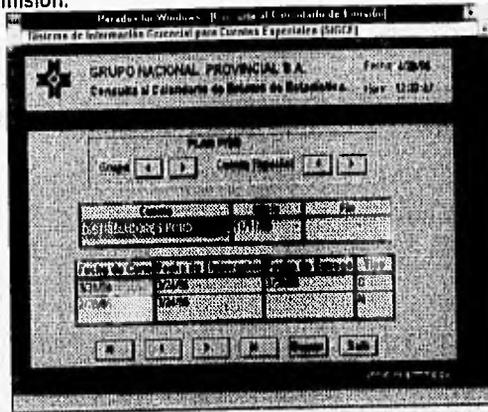


donde los botones son:

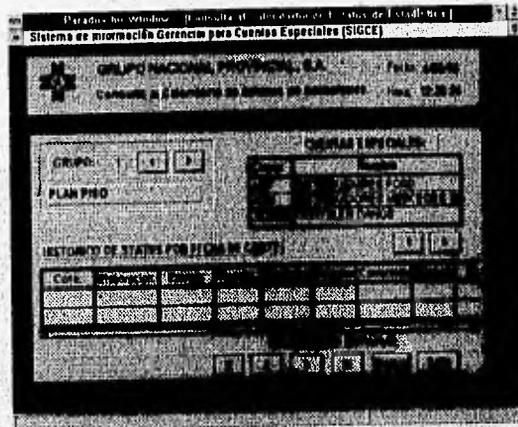
- Primero.** Este botón nos posicionará en el primer registro de la tabla que cumpla con las características de la consulta.
- Anterior.** Este botón nos posicionará en el registro anterior al que este seleccionado.

Este botón nos posicionará en el primer registro de la tabla que cumpla con las características de la consulta.
- Siguiente.** Este botón nos posicionará en el registro siguiente al que este seleccionado.
- Último.** Este botón nos posicionará en el último registro de la tabla que cumpla con las características de la consulta.
- Buscar.** Este botón desplegará el cuadro de diálogo de búsqueda para localizar el registro o registros que cumplan con una condición dada, y posicionará el cursor en el primer registro que cumpla con la condición de búsqueda. En caso de que varios registros cumplan con la condición, bastará con colocar el puntero en el botón de Buscar y dar doble clic con el botón izquierdo para pasar al siguiente registro.
- Salir.** Este botón cancela el último movimiento y nos regresa al menú o pantalla anterior.

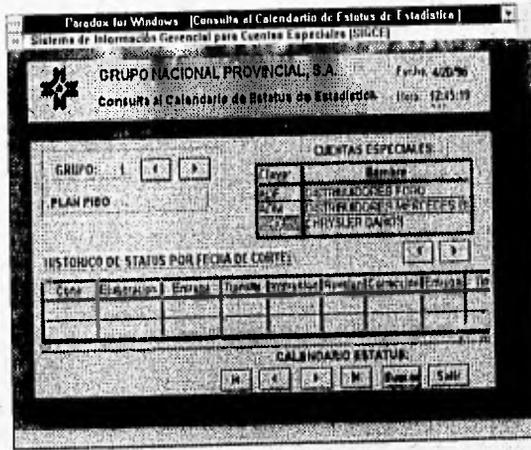
Calendario de Emisión.



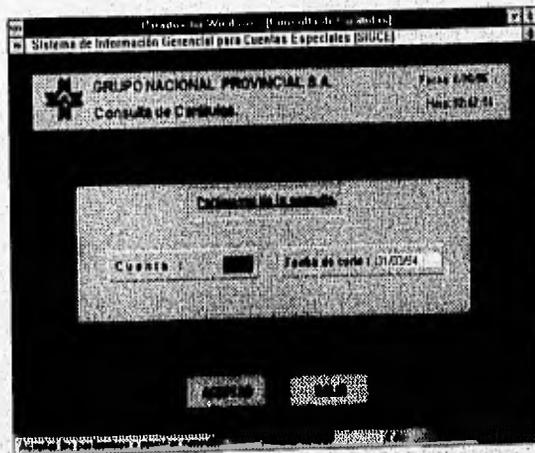
Existen en esta forma botones adicionales, para moverse a lo largo de las cuentas especiales y a través de los diferentes grupos.



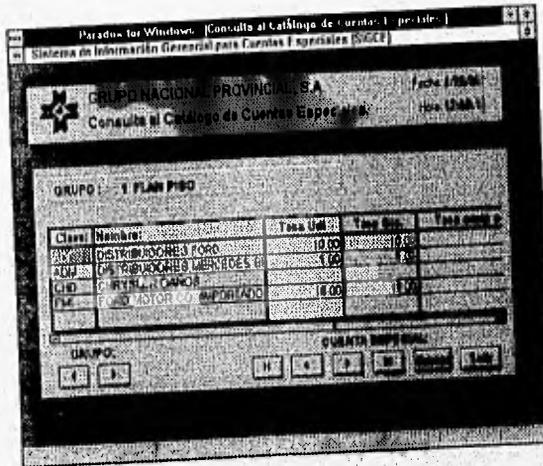
Calendario de Estatus



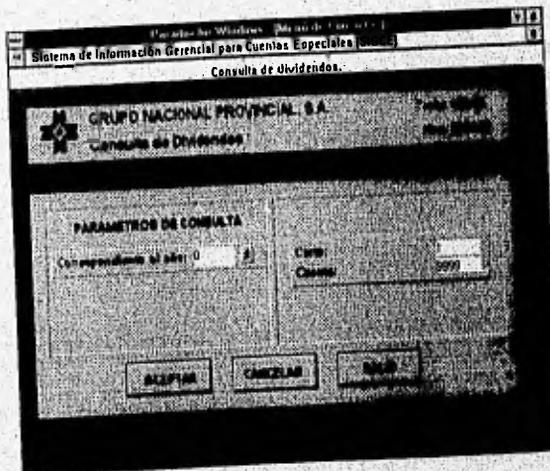
Carátula



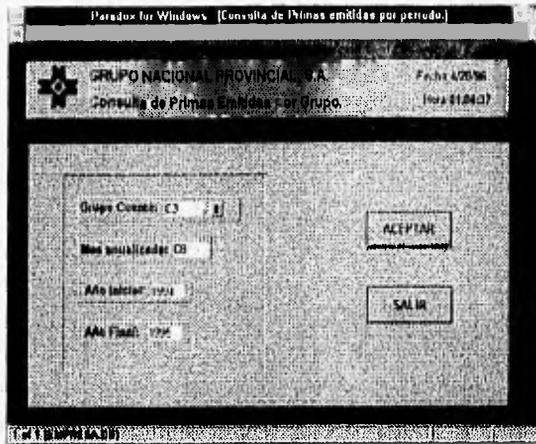
Cuentas Especiales.



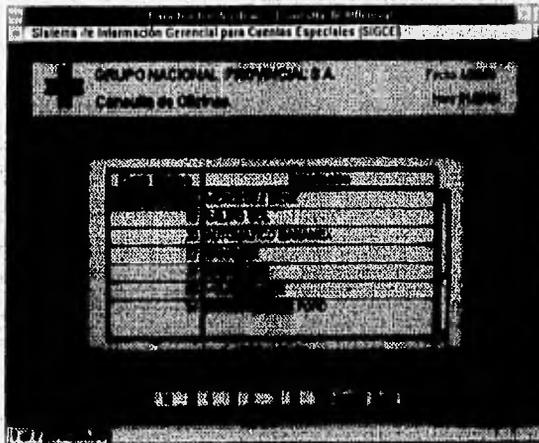
Dividendos



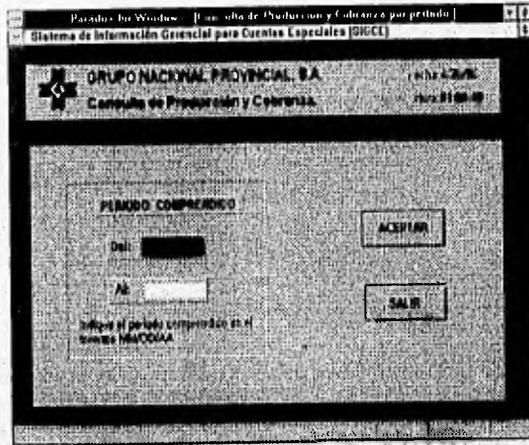
Emisión de Primas



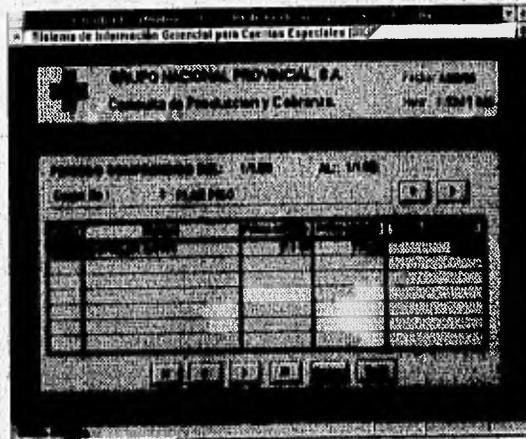
Oficinas



Producción y Cobranza

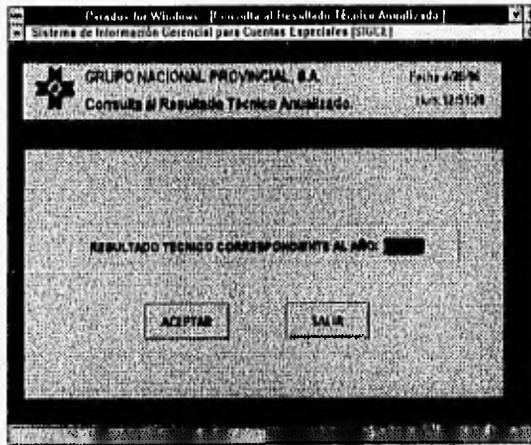


Al teclear los datos de la consulta y dar clic en Aceptar, se desplegará la siguiente forma:

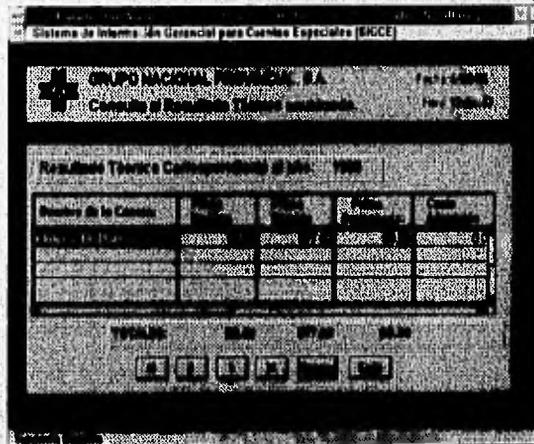


Al seleccionar el botón de Salir, regresaremos a la pantalla anterior donde, se no piden los parámetros de la consulta.

Resultado Técnico



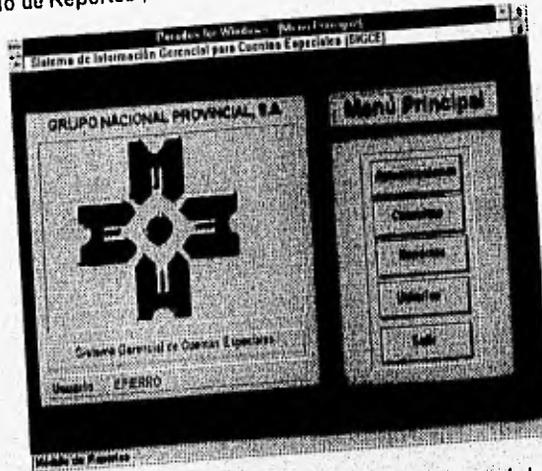
En esta forma se tecldea el año del que se desea la consulta al resultado Técnico anualizado, y se da clic en Aceptar. Con lo que aparecerá la siguiente forma:



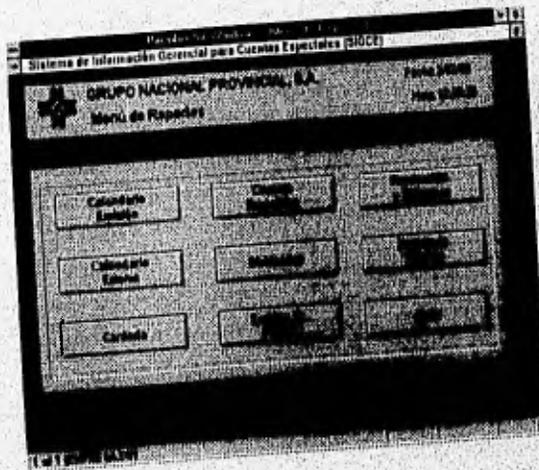
Al seleccionar Salir, el sistema nos regresará a la forma que requiere el año del que se desea el reporte técnico.

MODULO DE REPORTES

Al colocar el Mouse sobre el botón de Reportes del menú principal aparecerá la leyenda **Módulo de Reportes**, como se muestra a continuación:



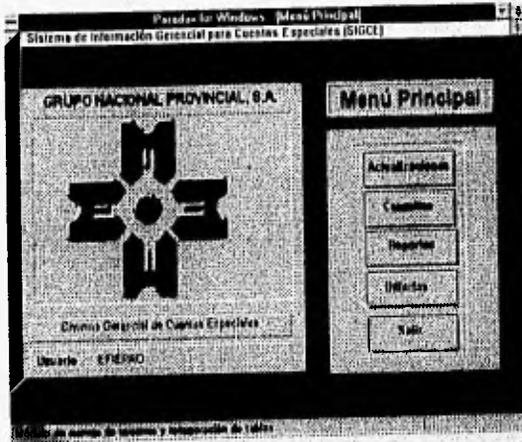
Al seleccionar con doble clic el botón de Reportes, aparecerá el menú del módulo de Reportes:



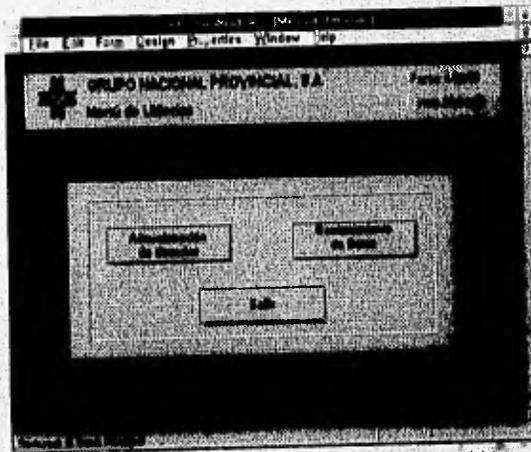
El sistema genera 8 Reportes diferentes, los cuales a continuación se presentan a continuación.

MÓDULO DE UTILERÍAS

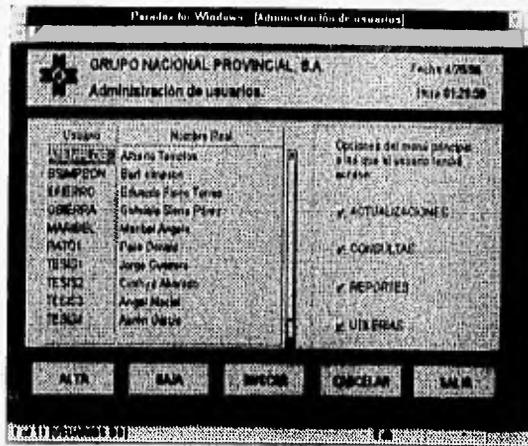
Al colocar el Mouse sobre el botón de Utilerías del menú principal aparecerá la leyenda Módulo de manejo de Usuarios y recuperación de tablas, como se muestra a continuación:



Al seleccionar con doble clic el botón de Utilerías, aparecerá el menú del módulo de Utilerías:

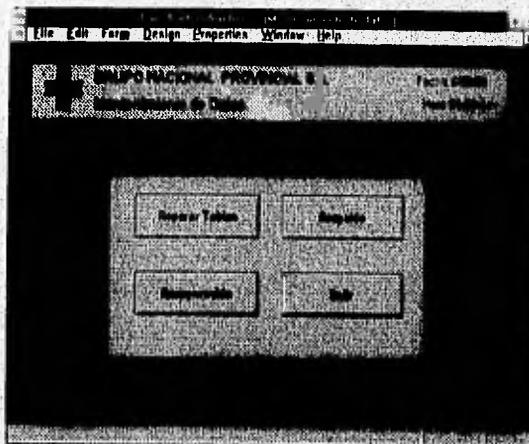


El sistema cuenta con dos Utilerías, la Administración de Usuarios, en de la cual podremos dar de alta o de baja un usuario, modificar sus capacidades dentro del sistema, etc.

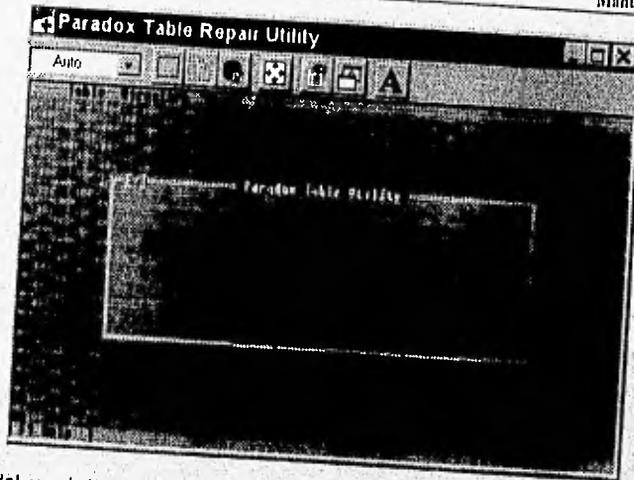


En esta forma, al seleccionar el registro del usuario que deseamos modificar, seleccionamos los módulos que deseamos que pueda acceder. Para las altas, bajas y cambio funciona igual que el módulo de actualizaciones.

La otra utilería es el Mantenimiento de los datos, a través de la cual respaldamos, recuperamos y reparamos las tablas del sistema.



Al seleccionar Reparar Tablas, se despliega la siguiente pantalla.



A través del menú de File, seleccionamos la tabla a reparar y Paradox la repara.