



22
UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGÓN

SISTEMA DE COBRANZA POR TERCEROS.

T E S I S
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN.

P R E S E N T A

LEONARDO LUIS ARCOS.



ENEP ARAGON

ASESOR:
M. EN I. JUAN CARLOS ROSA BEIZA.

MÉXICO

280145

2000





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 Dios, por guiarme por su camino de luz.

A mis padres, Leonardo César Luis Sibaja y Natividad Arcos de Luis por darme su cuidado, su amor, inculcarme el bien y apoyarme en todo momento.

A mi Esposa, Georgina Natalia Barrera de Luis por brindarme su amor y confianza.

A mis hijos, Leonardo, Guillermo y Natalia, quienes son mi fuente de inspiración para seguir adelante.

A mi hermana, Edith Luis Arcos por su apoyo y cariño.

A mis abuelos, pilares de grandes familias:

Segundo Arcos Hernández

Leonardo Luis Torres †

Maria Luna Mancilla

Valeriana Sibaja Santos †

A todos mis familiares por ser siempre unidos.

A mis amigos por estar siempre con nosotros.

Al Programa de Apoyo a la Titulación de la Facultad de Ingeniería de la UNAM y en especial al asesor de esta tesis **Juan Carlos Roa Beiza**.

ÍNDICE TEMÁTICO

CAPITULO 1.

1. MARCO TEÓRICO.

1.1. Conceptos básicos de facturación.	2
1.1.1. Proceso de facturación.	2
1.1.2. Categorías de inventarios.	2
1.1.3. Compras.	8
1.1.4. Inventarios.	14
1.1.5. Contabilidad y finanzas.	19
1.2. Teoría de bases de datos orientada a objetos.	25
1.2.1. Sistema de gestión de bases de datos.	25
1.2.2. Sistema de gestión de bases de datos orientada a objetos.	27
1.2.3. Modelo de datos orientado a objetos.	28
1.2.4. Conceptos básicos.	28
1.3. Metodología de diseño de bases de datos orientada a objetos.	41
1.3.1. Elección de la metodología optima.	64
1.4. Conceptos básicos de contabilidad financiera.	65
1.4.1. Origen y definición.	65
1.4.2. Importancia y utilidad.	67
1.4.3. Principios de contabilidad generalmente aceptados.	69
1.4.4. Registro de operaciones.	72
1.4.5. Ecuación básica de la contabilidad.	73
1.4.5.1. Definición y clasificación de activo.	74
1.4.5.2. Definición y clasificación de pasivo.	80
1.4.5.3. Definición y clasificación de capital contable.	84
1.4.6. Estados financieros.	86
1.4.6.1. Objetivo y utilidad.	86
1.4.6.2. Balance general.	87
1.5. Sistemas operativos.	89
1.5.1. Unix.	89
1.5.2. Windows 9X/windows NT.	105

1.6. Tipos de redes y protocolos de comunicación.	121
1.6.1. Antecedentes.	121
1.6.2. Redes de computadoras y tipos de configuraciones.	123
1.6.2.1. Redes LAN.	123
1.6.2.2. Redes MAN.	126
1.6.2.3. Redes WAN.	127
1.6.2.4. Ethernet.	128
1.6.2.5. Configuración estrella.	129
1.6.2.6. Configuración anillo.	132
1.6.2.7. Configuración bus.	135
1.6.2.8. Configuración malla.	137
1.6.2.9. Dispositivos de conectividad.	138
1.6.3. Comunicaciones.	157
1.6.3.1. Modelos de interconexión.	157
1.6.3.2. Modelo OSI.	158
1.6.3.3. Arquitectura TCP/IP.	163
1.6.3.4. Pasado y presente.	165
1.6.3.5. Servicios en modo conexión y en modo sin conexión.	166
1.6.3.6. Direccionamiento.	168
1.6.3.7. Ruteo.	170
1.6.3.8. Servicio de nombres.	175
1.6.3.9. Suite del protocolo TCP/IP.	177
1.6.3.10. Aplicaciones y utilidades.	178
1.6.3.11. Protocolo de control de transferencia.	180
1.6.3.12. Protocolo de transferencia de archivos.	183
1.6.3.13. Protocolo TCP/IP.	184
1.6.3.14. Protocolo básico de gestión de red.	187
1.7. Características ventajas y desventajas de Visual Foxpro 5.0.	189
1.7.1. Ventajas.	189
1.7.2. Desventajas.	191
1.7.3. Microsoft Visual Foxpro 5.0	192

CAPITULO 2.

2. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN.	
2.1. Situación actual.	207
2.2. Requerimientos del usuario.	210
2.3. Recopilación y análisis de la información.	212
2.4. Planteamiento del problema.	218
2.5. Descomposición funcional.	219
2.6. Opciones de solución.	225
2.7. Elección de la solución optima.	239

CAPITULO 3.

3. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA.	
3.1. Aplicación de la metodología elegida (Back End para cada módulo).	245
3.1.1. Diagrama de contexto.	275
3.1.2. Diagrama de eventos.	288
3.1.3. Diagrama dinámico.	290
3.1.4. Diccionario de datos.	369
3.1.5. Diagrama de entidad relación.	374
3.1.6. Normalización.	376
3.2. Generación de código para el procesamiento de la información.	381
3.3. Diseño y construcción del Front End.	396
3.4. Diseño e implementación de rutinas de diagnostico y evaluación de datos.	418
3.5. Pruebas e integración del sistema.	457
3.6. Factibilidad técnica y operativa.	472

MANUAL TÉCNICO.	504
MANUAL DE USUARIO.	519
CONCLUSIONES.	560
GLOSARIO.	568

BIBLIOGRAFÍA.	576
APÉNDICE A	
Generación de código de Back End.	A-1
APÉNDICE B	
Generación de código de Front End.	B-1

CAPITULO I

MARCO TEÓRICO

1.1 CONCEPTOS BÁSICOS SOBRE LOS PROCESOS DE FACTURACIÓN

1.1.1 Proceso de Facturación

Introducción

En este capítulo definiremos los conceptos de todas aquellas partes o funciones que nos permiten se lleve a cabo el proceso de facturación, como son: Ventas, Compras, Inventarios, Contabilidad y Finanzas; que una empresa debe tener a fin de que esta sea productiva y tenga una calidad total en todos sus productos y servicios.

1.1.2 Ventas

Introducción

El proceso que sigue el comprador para tomar decisiones en la venta, se desarrolla en una secuencia determinada.

Esta secuencia es la que marca la diferencia entre una venta tradicional y una venta compleja, y hay que entender cómo llegan las personas a tomar la decisión de realizar determinadas adquisiciones.

La Venta Compleja

Las características fundamentales de la venta son, entre otras, las siguientes:

- Implica decisiones de valor estratégico.
- Involucran a gente de distintos niveles.
- El ciclo del proceso no es corto.
- Las decisiones tienen impacto organizativo.
- Implican una alta probabilidad de continuar y ampliar el negocio.
- Se realizan en un mercado altamente competitivo.

¿Que significa la expresión "venta compleja"?

El término en sí mismo evoca la imagen de un laberinto con numerosos puntos de decisión. Cada uno de estos puntos de decisión constituirá, o bien un obstáculo que podrá impedirle continuar avanzando, o bien representará una pequeña victoria que le llevará a un escalón mas arriba. Seleccionar el camino correcto podrá llevarle al establecimiento de una larga y provechosa relación de negocio con el cliente. Sin embargo, si se elige el camino equivocado, podrá perderse y terminar en un callejón sin salida. En la figura 1.1.2.1 se muestra lo que ocurre en el caso real

¿Que necesita saber un vendedor para lograr el éxito en una venta compleja?

En primer lugar, debe tener un conocimiento perfecto tanto de sus productos y/o servicios como de aquéllos que ofrecen sus competidores. También se debe saber, de que modo la aplicación

de esos productos y/o servicios puede solucionar las necesidades del cliente.

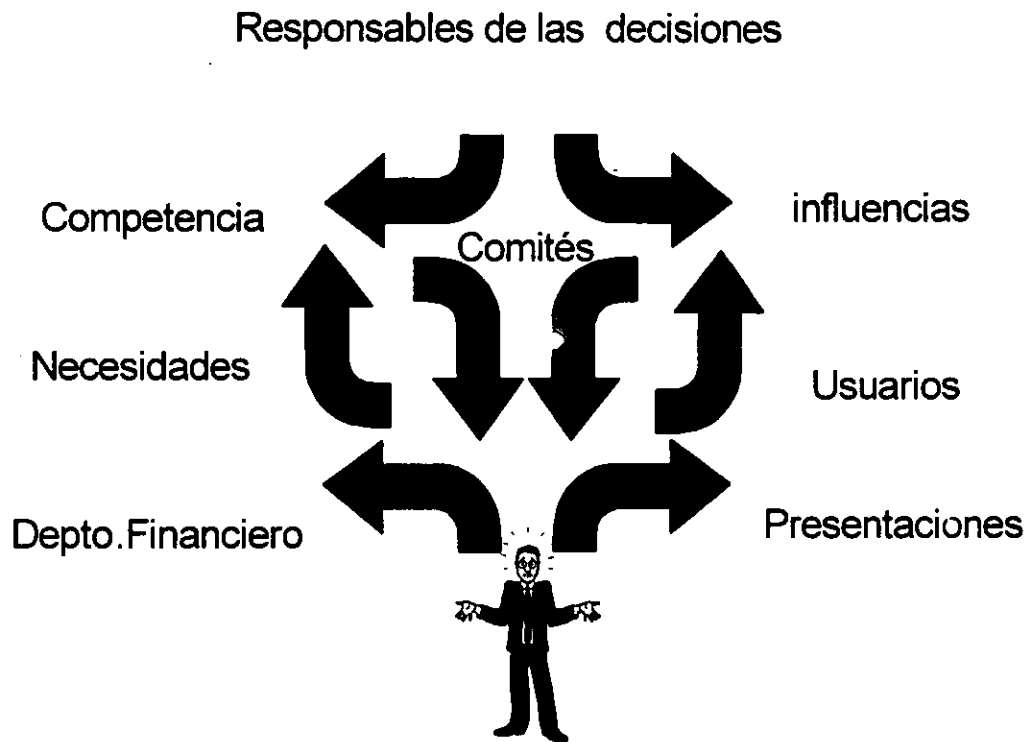


Figura 1.1.2.1 Laberinto de venta compleja.

El vendedor ha de tener la capacidad para observar cada uno de los pasos del proceso de venta a través de los ojos del comprador, con el fin de poder anticiparse a los problemas, los callejones sin salida y los demás riesgos inherentes a un proceso complejo.

En resumen: las ventas realmente eficaces empiezan por haber llegado a entender como se compra; y las estrategias, que se desarrollen como resultado de haber adoptado la perspectiva del comprador, servirán como la guía más eficaz para sortear con éxito

las dificultades que presenten en el camino. En la figura 1.1.2.2 se muestra el proceso de venta.

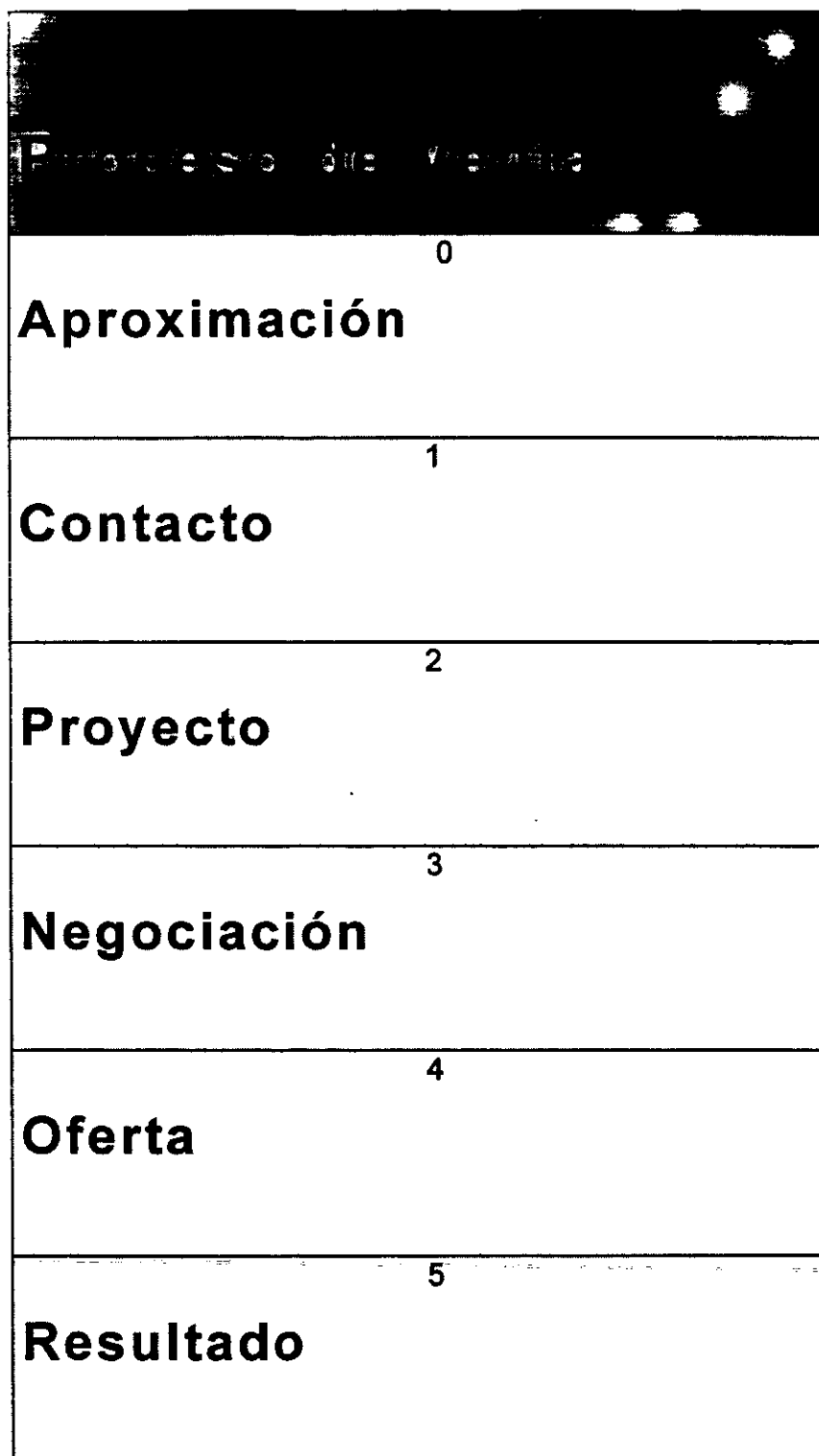


Figura 1.1.2.2 Esquema del proceso de venta

Aproximación

Es la fase 0 del proceso de venta. Su objetivo es realizar un buen contacto (fase 1). Engloba diferentes acciones, tales como búsqueda y manejo de información, creación de un anteproyecto etc. Esta aproximación puede realizarse a través de una serie de acciones encaminadas a obtener información sobre recursos, competencia, decisiones, personas, tipos etc. Es decir se debe de generar todo un Plan de Venta.

Contacto

Es la fase 1 del proceso y su objetivo es identificar un Proyecto (fase 2) o, bien directamente, obtener un resultado del negocio. Se entiende que se procede cuando tiene lugar la entrevista personal a un nivel adecuado con un interlocutor que ejerce algún tipo de influencia. Dentro de la fase entran diversas acciones a emprender como: entrevista(as) personal(es) que permitan conocer los objetivos del cliente, identificar un aliado y desarrollo de una estrategia específica.

La fase no queda cubierta mientras no haya un Proyecto, un Aliado y una estrategia.

Proyecto

Representa la fase 2 del proceso y comprende tanto el desarrollo del proyecto en sí, como el inicio de las acciones tácticas propias

de la estrategia establecida.* Los proyectos deben ajustarse a un formato preestablecido.

Negociación

En esta fase 3 se sitúan las diferentes entrevistas a realizar, con distintos interlocutores que tienen diversas actitudes, y que son necesarias para eliminar las debilidades del producto o servicio y apoyándonos en sus fortalezas. Esta fase queda cubierta cuando el cliente solicita una oferta (fase 4).

Oferta

Es la fase 4 del proceso. El vendedor no debe presentar ninguna oferta que el cliente no haya solicitado. Por tanto, no debe considerarse ningún proyecto en esta fase hasta que el cliente pida la propuesta formal. Las acciones son las propias de la presentación de la oferta en todos sus términos y condiciones. El objetivo es cerrar la venta. Las ofertas deben ajustarse a un formato preestablecido.

Resultado

Es la fase 5 y última del proceso. Se llega a ella cuando se ha logrado un objetivo de venta, lo cual no significa que se haya logrado el mayor resultado.* Por eso, es una fase clave, ya que permite detectar nuevas oportunidades.

El aprovechamiento de estas nuevas oportunidades dará lugar a un nuevo proyecto y, en definitiva, llevara a fidelizar al cliente.

1.1.3 Compras

Proceso de Compras

Las compras son la adquisición de bienes o servicios con intercambio de fondos. La figura 1.1.3.1 ilustra el proceso, el cual difiere de acuerdo con el tipo de artículo. Los artículos en gran volumen generalmente se abastecen bajo una orden amplia de compra, la cual establece un precio de la empresa, pero permite al comprador emplear fechas de entrega diferidas.

Las compras están a cargo de compradores profesionales que tienen conocimiento especializado acerca de las líneas seleccionadas de productos y están familiarizados con especificaciones de ingeniería, contratos legales, reglamentaciones de embarques y otros factores. A continuación se delinearán las principales responsabilidades del personal del departamento de compras.

- 1. Identificar y desarrollar fuentes de abastecimiento.**
- 2. Seleccionar proveedores y negociar contratos.**
- 3. Mantener relaciones de trabajo y controlar la eficiencia de los proveedores.**
- 4. Evaluar economías de oferta-demanda e iniciar estudios de costos de fabricar los artículos que se compran.**
- 5. Mantener una base de datos del sistema de aprovisionamiento.**

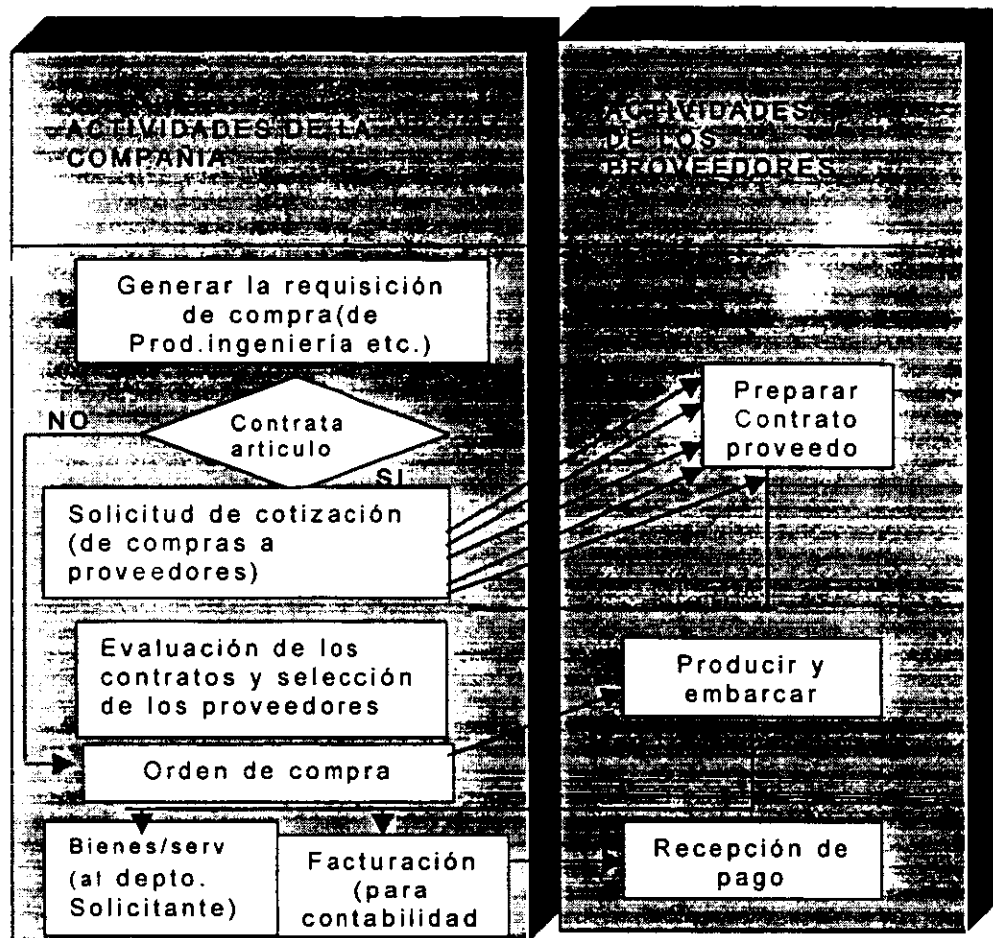


Figura 1.1.3.1 Proceso de compra.

La empresa necesita proveedores confiables si están disfrutando de los beneficios de inventarios disponibles bajos. Las variables importantes que deben considerarse en la selección de proveedores incluye:

1. Precio.
2. Cantidad.
3. Servicio.
4. Soporte técnico.
5. Condiciones de compra.
6. Tiempos de entrega.
7. Calidad.
8. Mantenimiento.
9. Estabilidad financiera.

En algunos casos, los proveedores están calificados con base en diferentes criterios, a los cuales se le asignan calificaciones de importancia de 1 a 10.

Calificación esperada = (calificación importancia x valor criterio)

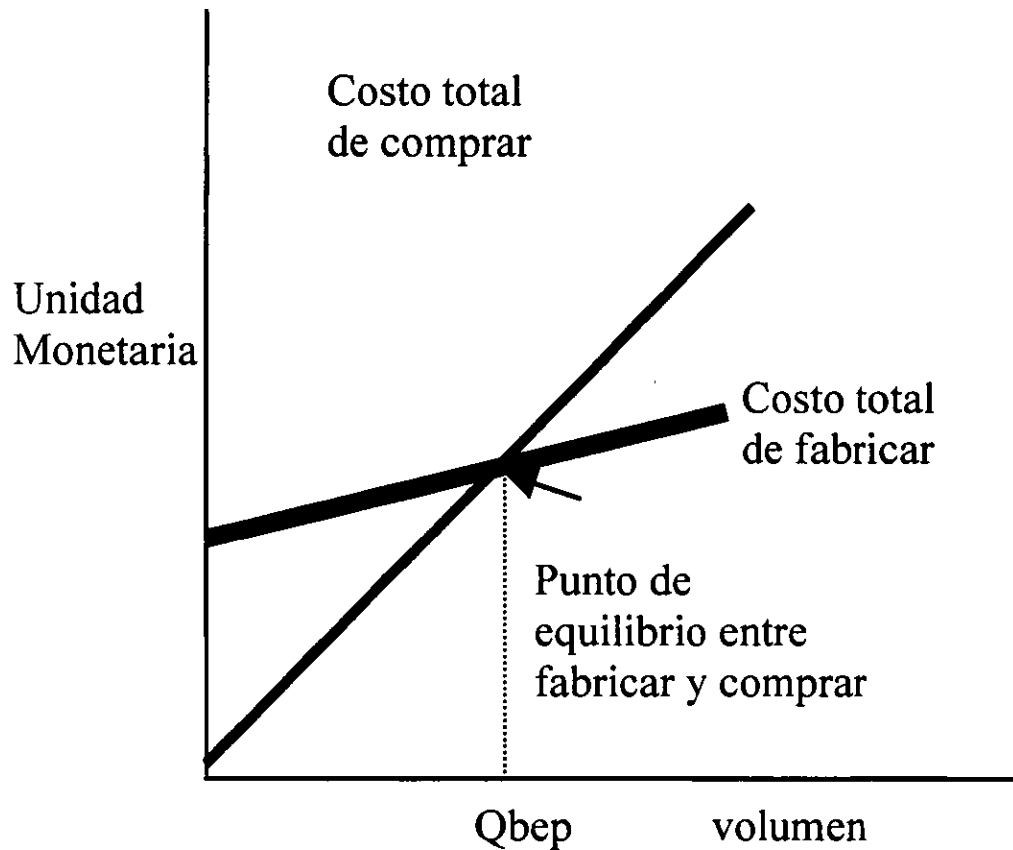


Figura 1.1.3.2 Factores económicos y no económicos que influyen en las decisiones de fabricar contra comprar. *

Factores menos económicos

Revisión:

- Disponibilidad de fondos y personal capacitado.

- Disponibilidad y volumen de oferentes.
- Deseo de fuentes alternativas de oferta.

Procesamiento:

- Preferencias de los empleados e interés en la estabilidad.
- Deseo de desarrollar facilidades de I & D.
- Necesidad de controlar secretos de negocios.
- Deseo de expandir una línea nueva de productos.
- Necesidad de controlar los tiempos de entrega.
- Impacto de la flexibilidad de producción.

Resultados:

- Necesidad de controlar la calidad o la confiabilidad.
- Efecto de la imagen y la reciprocidad en los clientes.

Decisiones Sobre Fabricar o Comprar

Las decisiones concernientes a fabricar o comprar componentes implica tanto consideraciones económicas como no económicas. Económicamente, un artículo puede ser fabricado por la empresa si ésta tiene suficiente capacidad y si el valor del componente es suficientemente alto para cubrir todos los costos variables de la producción y además hace alguna contribución a los costos fijos. Los volúmenes bajos de uso favorecen la compra, la cual tiene pocos o ningunos costos fijos. Como se muestra en La figura 1.1.3.2

Decisiones Sobre la Cantidad de Compra: Modelo de Periodo Único

Deben comprarse y mantenerse materiales hasta que el ingreso marginal de adquirir y de mantener sean iguales al costo marginal (IM-CM). Pero el ingreso marginal incremental (IM) de cada artículo durante un solo periodo es una función de la oportunidad o probabilidad, de que él artículo sea demandado $P(D)$ durante ese periodo. Por tanto:

$$(IM)P(D) = CM \quad \text{Ec. 1.1.3.1}$$

Resolviendo $P(D)$ se obtiene una medida de la probabilidad de venta, donde la ganancia incremental de vender un artículo solo compensa el costo incremental de adquirir y mantener él artículo para la venta. Reconociendo que el ingreso marginal incluye ambos, costo marginal (CM) y utilidad marginal (UM), en un periodo único.

$$P(D) = CM/IM = CM/CM + UM \quad \text{Ec. 1.1.3.2}$$

Donde $P(D)$ representa la probabilidad acumulada de necesitar la siguiente unidad.

Ejemplo: Producir un artículo cuesta \$6; se vende en \$10 y tiene una distribución de probabilidad acumulada de demanda durante el siguiente periodo que se muestra en la figura 1.1.3.3

Unidad número	1	2	3	4	5	6	7	8
P(vender esta unidad)	1.00	.92	.82	.75	.62	.40	.15	.10

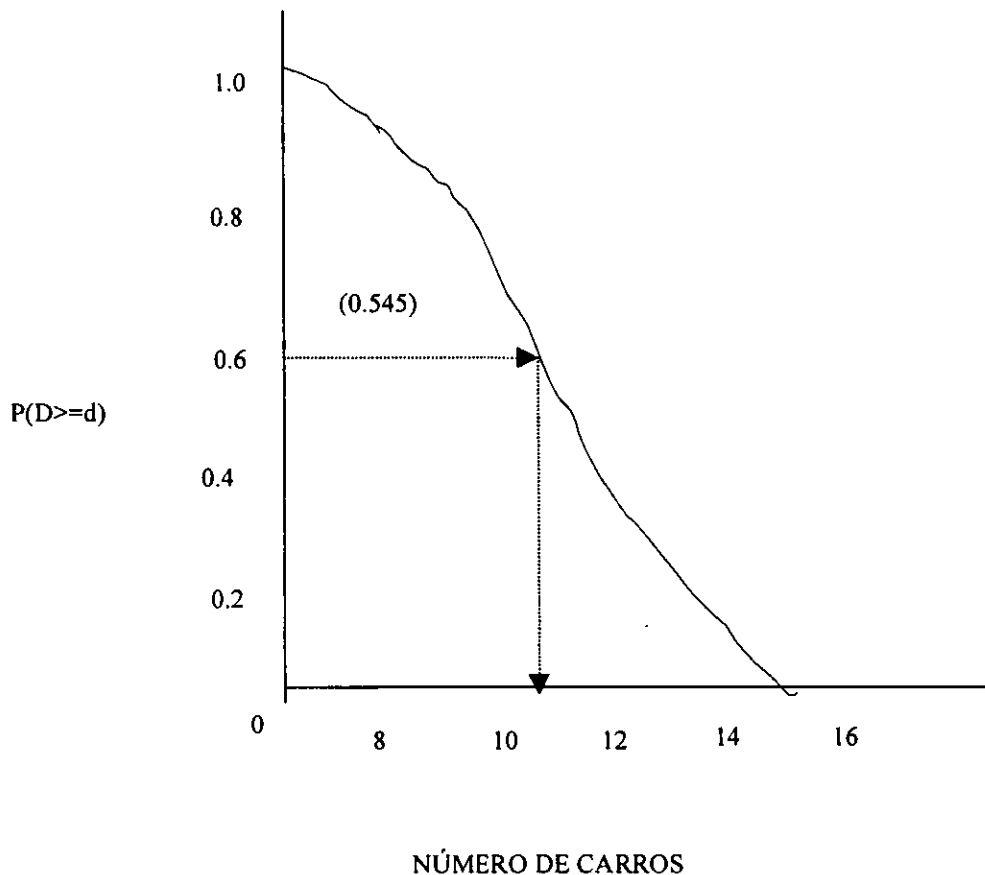


Figura 1.1.3.3

¿Cuántas unidades deben ser ordenadas?

$$P(D) = \frac{CM}{CM + UM}$$

donde $CM = \$6$

$$UM = \$10 - \$6 = \$4$$

$$P(D) = \frac{\$6}{\$6 + \$4} = .60$$

Por lo tanto, procede ordenar cinco unidades cuando $P(\text{vender} \geq 5 \text{ unidades})$ de .62 es un poco mayor que la probabilidad de 0.60. Esto parece intuitivamente correcto dado que el costo (\$6) representa 60% del valor del artículo. La probabilidad de venta $P(D)$, también debe ser igual a 60%. Si el costo fuera una mayor

proporción (por ejemplo, 82%), se justificaría pocas unidades (3) debido a que el costo marginal es mas alto.

1.1.4 Inventarios

Propósito de los Inventarios

Los inventarios son recursos ociosos que poseen un valor económico. Las empresas generalmente clasifican sus inventarios como:

- 1) Materias primas.
- 2) Producto en proceso.
- 3) Producto terminado.

Todos los inventarios representan una inversión designada para facilitar las actividades de producción y servir a los consumidores. Sin embargo, mantener inventarios consume capital de trabajo, el cual puede no estar proporcionando un ingreso o un rendimiento en una inversión y puede ser requerido urgentemente en cualquier momento. Consecuentemente, el problema de la administración de inventarios es el mantenimiento de niveles de inventarios adecuados, pero no excesivos. A continuación se describen algunas de las principales razones para mantener inventarios.

1. Proporcionar servicio a clientes con demandas variables (inmediatas o estacionales).
2. Protegerse contra los errores de los proveedores, la escasez y los faltantes.

3. Mejorar el nivel de las actividades de producción, estabilizando el empleo y mejorando las relaciones de trabajo.

4. Desfasar las etapas sucesivas en operaciones para que las fallas no detengan el sistema.

5. Facilitar la producción de diferentes productos en las mismas instalaciones.

6. Proporcionar un medio de obtener y mejorar materiales en lotes de tamaño económico y obtener descuentos por cantidad.

7. Proporcionar un medio para eliminar los riesgos de incertidumbre sobre los precios futuros y las entregas, tales como huelgas, incrementos de precio e inflación.

Demanda Dependiente e Independiente

Un inventario de demanda dependiente está compuesto por las materias primas, los componentes, y los subensambles que son usados en la producción de artículos que sirven para la fabricación de otros artículos o para la fabricación de productos finales. Por ejemplo, la demanda de teclados de computadora depende del artículo original, las computadoras. El inventario de producción es muy dependiente y predecible. Los requerimientos de todos los componentes encarados con otros componentes son fijados por el diseño, y las cantidades de producción son dictadas por la programación maestra de la empresa.

Los inventarios de demanda independiente constan de los productos terminados, las partes de servicio, y otros artículos cuya demanda aumenta mas directamente del ambiente incierto de mercado. Por esto, la distribución de inventarios generalmente tiene una demanda altamente incierta e independiente. Las demandas dependientes normalmente pueden calcularse, mientras que las demandas independientes usualmente requieren alguna clase de pronostico.

Costo de Inventario y la Ecuación CEP

Los principales costos asociados con acopiar y mantener inventarios son los siguientes:

- Costos de ordenar e iniciar la producción para colocar pedidos, expedición, inspección, y cambio o establecimiento de instalaciones para producir internamente.
- Costos de mantener en inversión de capital, manejo, almacenamiento, seguros, impuestos, obsolescencia, deterioro y procesamiento de datos
- Costos de compra incluyendo el precio pagado o la mano de obra, los materiales y los cargos indirectos necesarios para producir el artículo.

En la figura 1.1.4.1 podemos observar la relación existente entre los costos relevantes de ordenar y mantener.

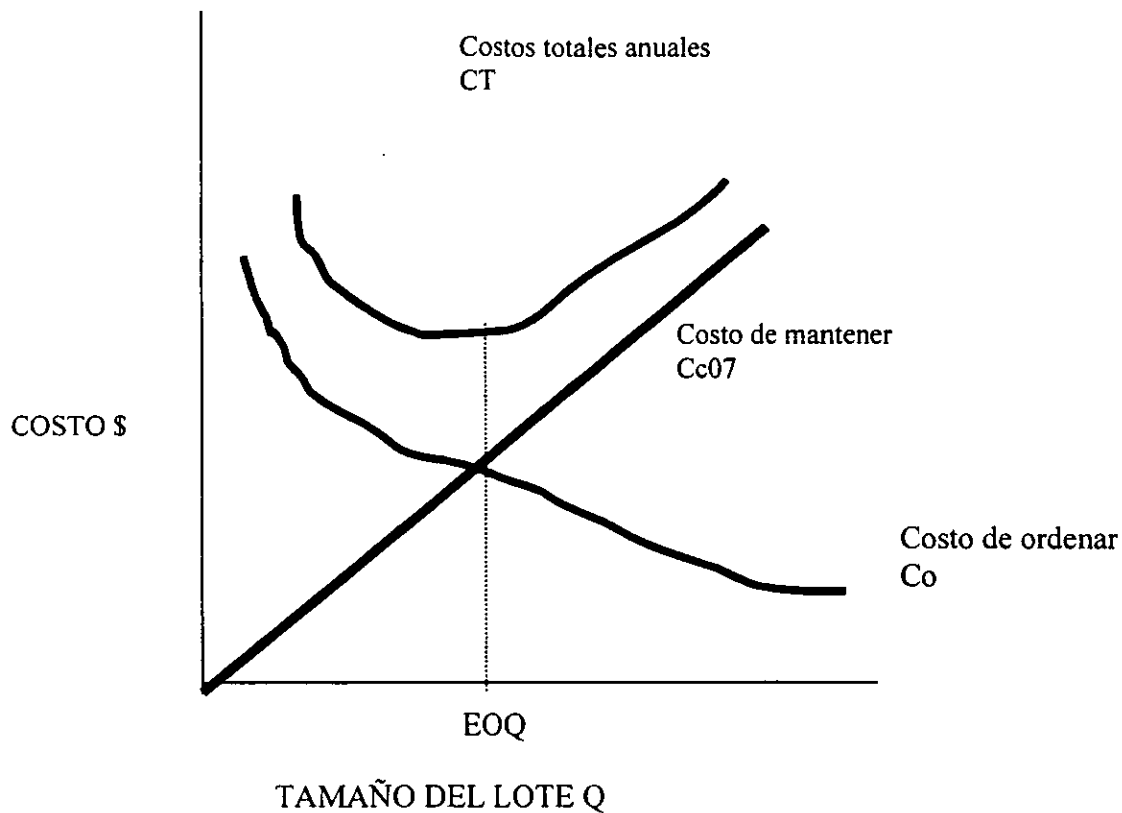


Figura 1.1.4.1 Cantidad económica de pedido (CEP).

El costo total (CT) de un inventario es la suma del costo de ordenar, más el costo de mantener y los costos de comprar. Si d es igual a la demanda de unidades en una base anual, C^o es igual al costo de preparar o hacer un pedido, $Cç$ es igual al costo de mantener una unidad de inventario durante un periodo de tiempo dado, P es igual al costo de compra, Q es igual al tamaño del lote, y $Q/2$ equivale al inventario promedio, entonces las relaciones pueden ser expresadas matemáticamente:

$$\text{Costo total} = \text{costo de ordenar} + \text{costo de mantener} + \text{costo de comprar}$$

donde,

costo de ordenar = $(C^{\circ}\$/orden) (orden/Q unidades)(D unidades/año)$

costo de mantener = $(C_{\zeta}\$/unidad-año) (Q unidades/2)$

costo de compra = $(P\$/unidad)(d unidades/año)$

Entonces,

$$CT = C^{\circ}D/Q + C_{\zeta}Q/2 + PD \quad (1.1.4.1)$$

Diferenciando respecto a la cantidad ordenada Q se obtiene la pendiente de la curva CT.

$$dCT/dQ = C^{\circ}D/Q + C_{\zeta}Q/2 + 0$$

Estableciendo esta primera derivada igual a cero identifíquese el punto donde CT es mínimo.

$$0 = C^{\circ}D/Q + C_{\zeta}Q/2 + 0$$

Esto es, $Q = CEP = 2C^{\circ}D/C_{\zeta} \quad (1.1.4.2)$

La ecuación (1.1.4.2) es la cantidad económica de pedido (CEP) o tamaño económico del lote (TEL). Ésta proporciona la cantidad que se ordenara para satisfacer la demanda calculada al menor costo total. Cuatro supuestos básicos delimitan el modelo CEP; son los siguientes:

1. La demanda y el tiempo de entrega son conocidos y constantes.

2.El reemplazo es instantáneo tras la expiración del tiempo de entrega.

3.Los costos de compra no varían con la cantidad ordenada.

4.Las expresiones de costos de ordenar y mantener incluyen todos los costos relevantes y esos costos son constantes.

1.1.5 Contabilidad y Finanzas

Introducción

Las operaciones comerciales son operaciones económicas, la compra y venta, operaciones típicas de una empresa comerciales, son operaciones económicas, pues el objetivo de las operaciones comerciales es intercambio de bienes económicos satisfactorios de necesidades humanas. La compra y la venta constituyen las fases de la "comercialización".

La "producción" operación típica de las empresas industriales, también es una operación económica, pues de la producción depende que podamos disponer de Ropa, Muebles, Libros y muchos otros "productos industriales" que necesitamos para satisfacer nuestras necesidades. La contabilidad de las empresas industriales es el objetivo primordial de este tema.

Concepto de Contabilidad

La contabilidad es la técnica por los métodos, procedimientos e instrumentos aplicables para llevar acabo el registro, la

clasificación y resumen de los efectos financieros que provocan las operaciones realizadas por las empresas, con el objetivo de efectuar el control de los recursos de que dispone, y de las operaciones que realiza, y obtener información financiera para elaborar estados financieros.

La contabilidad es necesaria en toda empresa porque es necesario controlar los recursos de que disponen y las operaciones que realizan, y obtener información financiera que permita conocer y evaluar la situación financiera en que se encuentra, los resultados de las operaciones realizadas y otros aspectos relacionados con la obtención y el uso de dinero. En el inciso 1.4 abundamos mas sobre aspectos primordiales de la contabilidad financiera.

Finanzas

El administrar una empresa, ya sea pequeña o corporativa, implica muchas funciones. La mercadotecnia y el servicio a los clientes son decisivos para el éxito de una compañía. Pero es mediante las finanzas que todo esto sucede; éstas hacen funcionar todo lo demás. Sin capital para satisfacer las necesidades de la compañía, ya sea financiar el desarrollo de las operaciones cotidianas no se podría incrementar y probar nuevos productos, diseñar campañas de mercadotecnia comprar maquinaria etc. La función del administrador financiero consiste en garantizar que este capital esté disponible en las cantidades y en el momento correcto, así como al mas bajo costo. De lo contrario una compañía no puede subsistir.

El campo de las finanzas es amplio y dinámico. Afecta directamente las vidas de cada persona y cada organización, financiera o no financiera, privada o pública, grande o pequeña, lucrativa o no lucrativa. Existen numerosas áreas de estudio en la cuáles se encuentran disponibles diversas oportunidades de desarrollo dentro del campo financiero.

¿Qué son las Finanzas?

Las finanzas pueden ser definidas como el arte y ciencia de administrar el dinero. Prácticamente todos los individuos y todas las organizaciones ganan u obtienen dinero, y gastan o invierten dinero. Las finanzas se ocupan del proceso, las instituciones, los mercados e instrumentos relacionados con la transferencia de dinero entre individuos, empresas y gobierno.

Función de la Administración Financiera

La administración financiera está íntimamente relacionada con la economía y la contaduría, pero es bastante diferente a estas.

Puesto que la mayor parte de las decisiones empresariales se miden en términos financieros, el administrador financiero desempeña un papel fundamental en la operación de la empresa. El personal de todas las áreas de responsabilidad de la compañía – contabilidad, producción, comercialización, recursos humanos, investigación etc.- Debe interactuar con el personal financiero para realizar sus tareas. Todos tienen que justificar requerimientos del personal, negociar presupuestos de operación, preocuparse por apreciaciones del desempeño financiero y vender propuestas, cuando menos parcialmente, por méritos financieros para obtener

recursos de la alta administración. Desde luego, el personal financiero, para hacer pronósticos y tomar decisiones correctas, debe entablar comunicación con todos los demás miembros de la empresa.

Relación con la Contaduría

Las actividades financieras y las contables se encuentran, con frecuencia, bajo el control del vicepresidente de finanzas, como se muestra en la figura 1.1.5.1

Dichas funciones están relacionadas de manera estrecha y, por lo general, se superponen; en realidad la administración financiera y la contaduría suelen ser difíciles de distinguir. En las empresas pequeñas el contralor, con frecuencia, es quien realiza la función financiera, y en las corporaciones numerosos contadores se encuentran involucrados íntimamente con diversas actividades financieras. De cualquier manera, existen dos diferencias básicas entre las finanzas y la contaduría; una se refiere a el tratamiento que se da a los fondos y la otra, a la toma de decisiones. En el inciso 1.4 se describen en forma más amplia los estados financieros de la empresa.

En las siguientes paginas se muestran la figura 1.1.5.1 que es la organización de una corporación y función de las finanzas. La función financiera se identifica con la pantalla. La figura 1.1.5.2 muestra el proceso de facturación integro y completo.

ORGANIZACIÓN DE UNA CORPORACIÓN

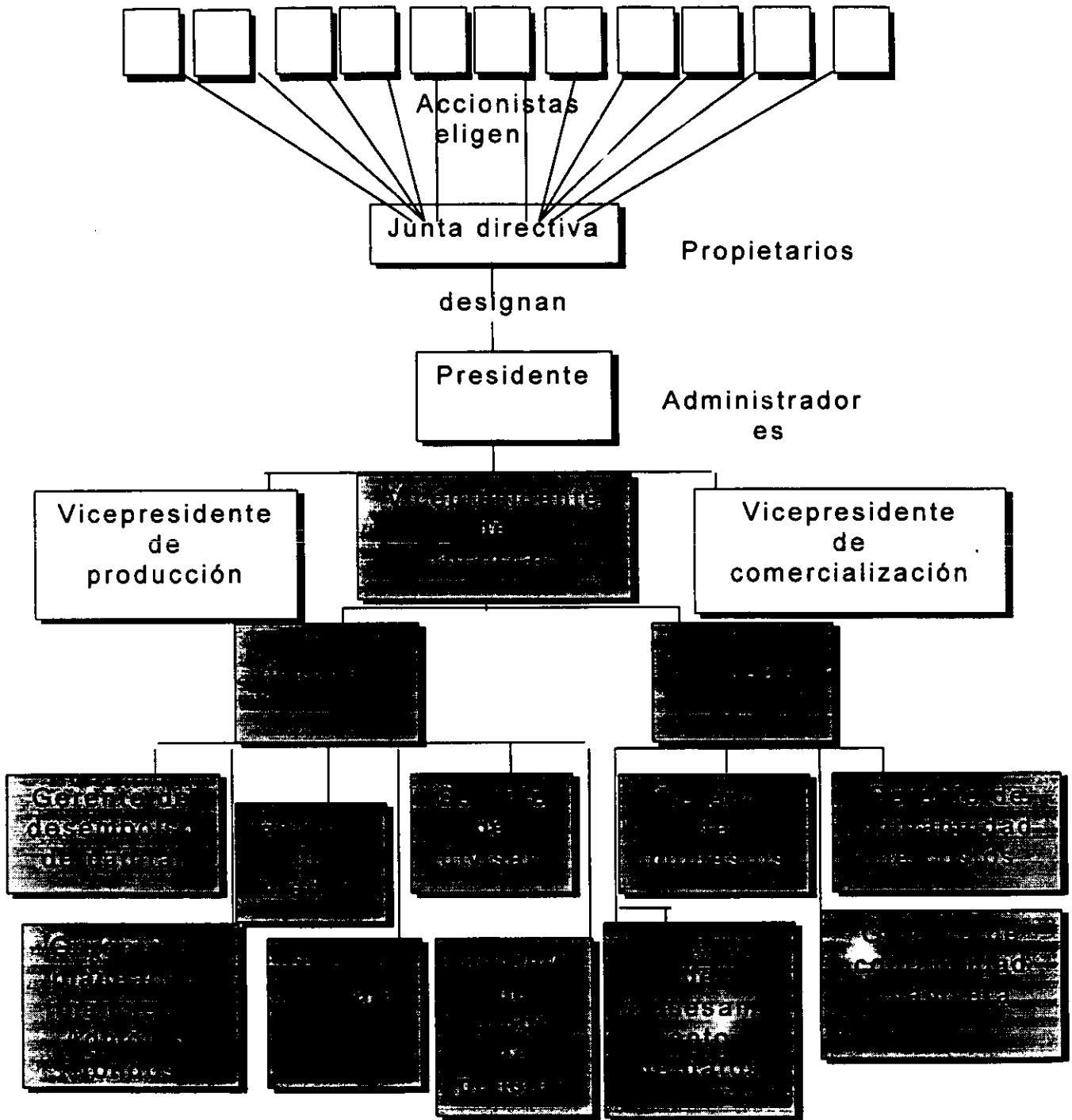


Figura 1.1.5.1. Organización de una corporación y función de las finanzas.*

Facturación



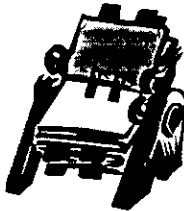
Ventas



Compras



Administración
y Finanzas



Inventarios

Figura 1.1.5.2. Proceso de facturación integro.*

1.2 TEORÍA DE BASES DE DATOS ORIENTADA A OBJETOS

Introducción

En este capítulo se ofrece una breve descripción del dominio de la tecnología de las bases de datos y de las tendencias actuales con vistas a averiguar las causas que subyacen al desarrollo de las bases de datos orientadas a objetos. En particular, analizaremos los rasgos principales de las aplicaciones avanzadas que requieren el desarrollo de nuevas técnicas para hacer posible la ejecución de las tareas de manipulación de datos.

1.2.1 Sistemas de Gestión de Bases de Datos

En cualquier tipo de organización se dedican considerables recursos a la recolección, clasificación, procesamientos e intercambio de datos basados en procedimientos bien establecidos con vistas a alcanzar objetivos específicos. En los años recientes, debido a significativos cambios en la tecnología de computación y debido a la disminución subsecuente de los costos, se ha tenido un gran crecimiento en el número de procesadores electrónicos para facilitar y desarrollar las posibilidades del procesamiento de datos.

Los primeros sistemas de bases de datos estuvieron basados en el uso de archivos separados. Partiendo de esta tecnología, se pasó a la integración de los datos en una única colección (base de datos). La gestión o manipulación de estos es llevada a cabo por los **DataBase Management System** (Sistemas de Gestión de Base de Datos).

Los sistemas de gestión de base de datos son sistemas de software centralizados o distribuidos que ofrecen facilidades para la definición de la base de datos, para la

selección de las estructuras de datos necesarias para el almacenamiento y búsqueda de los datos, lo mismo interactivamente que mediante un lenguaje de programación. Las operaciones asociadas con el modelo de datos definen las estructuras de datos que representan las entidades del dominio de la aplicación que se desea modelar en la base de datos, la forma de acceder a ellas para la recuperación de los datos y la forma de usarlas para llevar a cabo las actualizaciones.

Las diferentes operaciones que ofrece un Sistema de Gestión de Base de Datos se expresan por medio de uno o varios lenguajes. Normalmente, un Sistema de Gestión de Base de Datos ofrece un **Data Definition Language** (Lenguaje de Definición de Datos) que define el esquema de la base de datos. Además, un Sistema de Gestión de Base de Datos incluye un **Data Manipulation Language** (Lenguaje de Manipulación de Datos). Muy a menudo, la componente de la manipulación de datos, que permite las operaciones de acceso, se conocen como lenguaje de consulta. Además, un Sistema de Gestión de Base de Datos proporciona un conjunto de funciones cuyo propósito es asegurar la calidad y seguridad de los datos, así como un acceso fácil y eficiente de los mismos.

De este modo un Sistema de Gestión de Base de Datos esta equipado con mecanismos para el control de concurrencia, lo que permite que varios usuarios tengan acceso a los datos al mismo tiempo. También tiene mecanismos de recuperación que aseguran la consistencia de la Base de Datos si el sistema falla o si los usuarios cometen ciertos errores. Un Sistema de Gestión de Base de Datos contiene también estructuras auxiliares de acceso para asegurar el acceso eficiente de los datos y un subsistema para optimizar las operaciones de consulta.

Tendencias actuales de la Tecnología de Bases de Datos

Con vistas a satisfacer los requisitos impuestos por las nuevas aplicaciones, la investigación y desarrollo de las Bases de Datos siguen diferentes tendencias, las

cuales muy a menudo involucran la integración de la tecnología de las Bases de Datos con la tecnología de los lenguajes de programación, tales como los lenguajes de programación orientados a objetos, hay una tendencia común a aumentar el poder expresivo de los modelos de datos y de los lenguajes de gestión de datos; una de las tendencias es:

1.2.2 Sistemas de Gestión de Bases de Datos Orientadas a Objetos

Estos sistemas integran la tecnología de las Bases de Datos con el paradigma orientado a objetos que se han desarrollado en el área de los lenguajes de programación y de la ingeniería de Software. Esta tendencia en su mayor parte ha estado controlada por los desarrollos industriales.

En general aunque las diversas tendencias se basan en aproximaciones diferentes, tales como la integración de diferentes funciones de los Sistemas de Gestión de Bases de Datos con modelos de programación muy variados, se puede prever con bastante facilidad que la mayoría de las próximas generaciones de sistemas de gestión de bases de datos tendrán un conjunto de características comunes que incluirán la posibilidad de definir y manipular objetos complejos, ciertas formas de jerarquías de tipos, mecanismos para manejar reglas deductivas y restricciones de integridad.

Las direcciones de las tendencias indican a los **Object Oriented Database Management Systems** (Sistemas de Gestión de Bases de Datos Orientadas a Objetos) como la tecnología más prometedora para la próxima generación de Sistemas de Gestión de Bases de Datos. Como el grupo de trabajo de ANSI para las bases de datos orientadas a objetos. Los grandes fabricantes en el Hardware se están involucrando en estas iniciativas y en los intensos esfuerzos de investigación, no solo a nivel académico. Algunos de estos fabricantes colaboran en iniciativas con productos de Sistemas de Gestión de Bases de Datos Orientadas a Objetos.

El modelo orientado a objetos es hoy en día una de las vías más prometedoras en el desarrollo del Software. Podemos decir que en la metodología orientada a objetos, se puede prever una reducción drástica en la cantidad de código y un aumento en la reutilización que tendrá efecto sobre la reducción de los costos e incluso de los tiempos. Esto es debido a que el modelo de objetos puede integrar diferentes tipos de sistemas más fácilmente.

1.2.3 Modelos de datos orientados a objetos

No hay un modelo común a utilizar como punto de referencia, muchas de las ideas de la programación orientada a objetos se derivan del lenguaje SIMULA en 1966, pero sólo más tarde comenzó este modelo a ser ampliamente utilizado, como resultado de la introducción de Smalltalk. Desde entonces se han desarrollado lenguajes como son C++ y FoxPro.* La clave de la programación orientada a objetos es considerar a un programa como si estuviera compuesto de objetos independientes, agrupados en clases, los cuales se comunican con otros objetos por medio de mensajes. Las bases de datos requieren un modelo propio de datos, y, en ausencia de un estándar, ciertos conceptos concernientes al modelo de datos se han agrupado en un modelo central o modelo básico. Obviamente, el modelo central aunque poderoso, no captura las restricciones de integridad y las relaciones semánticas que son tan importantes. Tales restricciones incluyen, por ejemplo, la unicidad de los valores de un atributo, la posibilidad del valor nulo para un atributo, el rango de valores que un atributo puede asumir, y otros conceptos similares.

1.2.4 Conceptos Básicos

Objetos e identidad

- Cada entidad del mundo real se modela con un objeto. Cada objeto esta asociado con un identificador único.

Objetos complejos

- Un conjunto de atributos está asociado a cada objeto; el valor de un atributo puede ser un objeto o un conjunto de objetos. Esta característica permite definir objetos arbitrariamente complejos en términos de otros objetos.

Los objetos se manifiestan ellos mismos de alguna de las formas mostradas en la Figura 1.2.4.1. Los objetos pueden ser:

- Entidades externas (dispositivos, otros sistemas, personas) que producen o consumen información
- Cosas (Cartas, informes) que son parte del dominio de la información del problema.
- Lugares (planta de producción) que establece el contexto del problema para su solución
- Estructuras (computadoras, vehículos de cuatro ruedas) quien define una clase de objetos o. en casos externos, clases relacionadas a objetos.
- Unidades organizacionales (equipo, grupo, conjuntos) que son relevantes en una aplicación.
- Papeles o roles (director, ingeniero) desempeñados por personas que interactúan con el sistema.
- Eventos (la terminación de una serie de movimientos de un robot) que ocurren dentro del contexto de operación del sistema.

Estado de un objeto

- El estado de un objeto abarca todas las propiedades del objeto, y los valores actuales de cada una de sus propiedades. Las propiedades de los objetos suelen ser estáticas, los valores de las propiedades cambian con el tiempo.

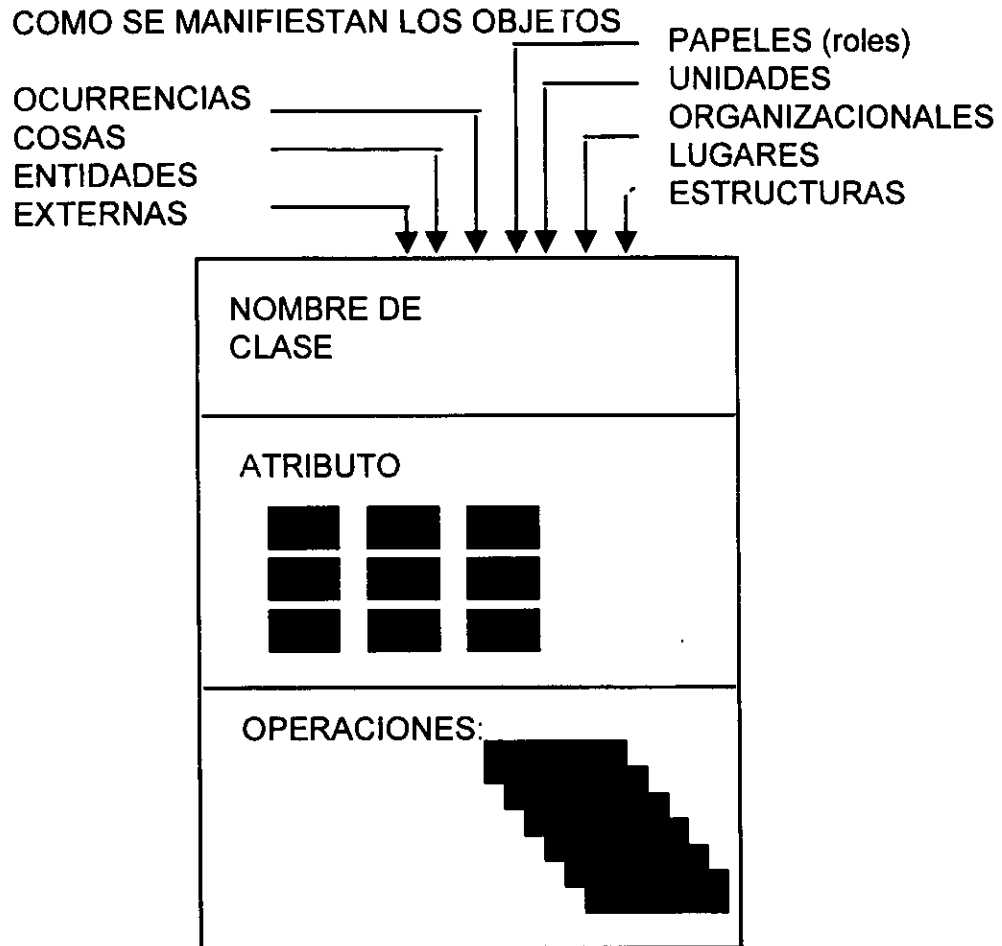


Figura 1.2.4.1 Manifestación de los objetos

Evento

Un evento es un cambio en el estado de un objeto. Un evento ocurre cada vez que un sistema orientado a objetos y un sistema externo u operador intercambian información. En la Figura 1.2.4.2 se muestra un ejemplo.

Un evento sirve como un identificador de los instantes en que ocurren los cambios de estado de un objeto. Es decir el tiempo en que se da el cambio de estado del objeto.

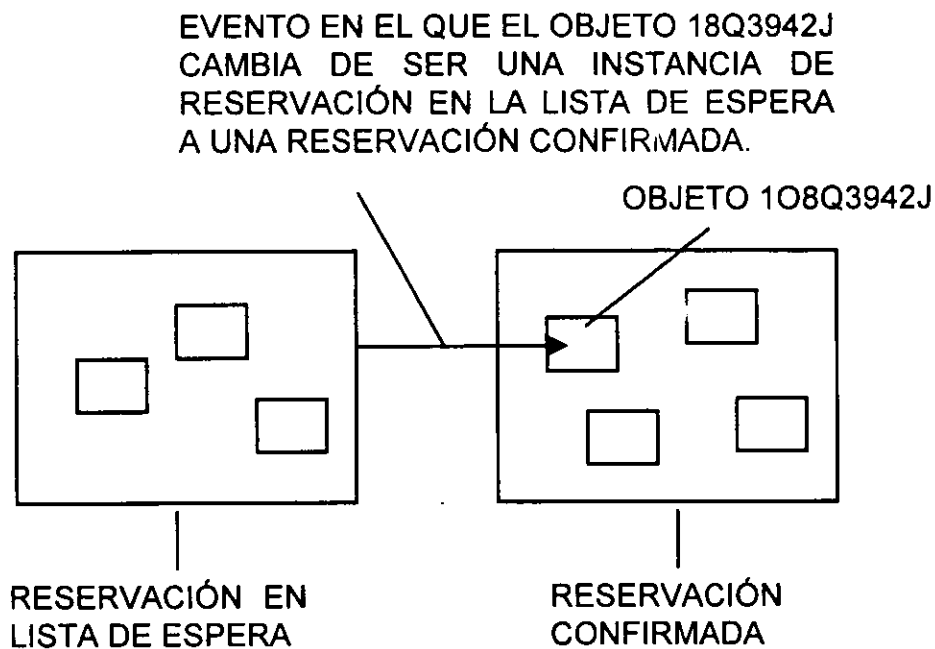


Figura 1.2.4.2 Estado de un objeto que cambio. *

Atributos

Definen las propiedades de un objeto o clase y se puede usar para:

- Nombrar una ocurrencia de un objeto
- Describir la ocurrencia
- Hacer referencia a otra ocurrencia

Los atributos son quienes describen al objeto y califican lo que se representa con el objeto en el contexto del espacio del problema Ver Figura 1.2.4.3

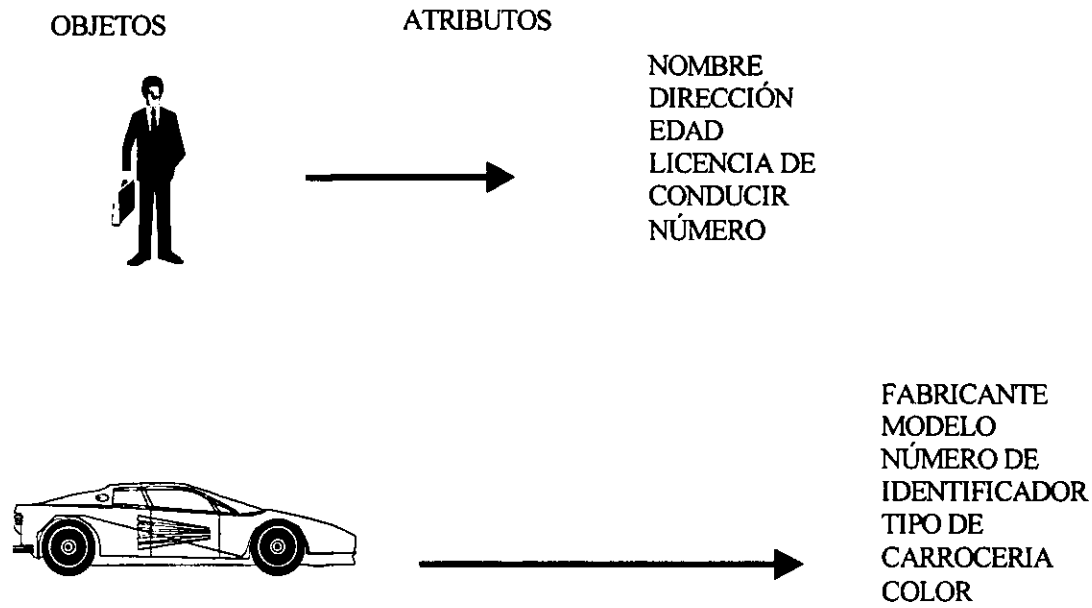


Figura 1.2.4.3 Objetos y atributos. *

Clases

Todos los objetos que comparten el mismo conjunto de atributos y de métodos se agrupan en clases. Cada objeto pertenece a alguna clase. La clase define una colección de objetos similares

Clases y Tipos

Los sistemas orientados a objetos pueden clasificarse en dos categorías principales: los que manejan la noción de clase y los que manejan la noción de tipo. En investigación hay mucha discusión sobre la distinción entre clases y tipos. Aunque no hay una línea clara de demarcación entre éstos, los dos conceptos son esencialmente diferentes.

Un tipo modela los rasgos comunes a un conjunto de objetos que tienen las mismas características y se corresponden con la noción de tipo de dato abstracto. En los

lenguajes de programación, los tipos son una herramienta para aumentar la productividad del programador, asegurando la corrección de los programas. Si el sistema de verificación de tipos se diseña cuidadosamente, los tipos pueden ser controlados durante la compilación; de otra manera, algunas partes se podrán procesar sólo en tiempo de ejecución. En general, en un sistema basado en tipos, los tipos no son objetos en el verdadero sentido de la palabra y no pueden ser modificados dinámicamente.

A menudo los conceptos de tipo y clase se utilizan indistintamente. Sin embargo, cuando ambos están presentes en el mismo lenguaje, el tipo se utiliza para indicar la especificación de la interfaz de un conjunto de objetos, mientras que la clase es una noción de implementación. Por tanto, un tipo es un conjunto de objetos que tienen un mismo comportamiento que se puede observar desde afuera.

Esto significa que el tipo al cual un objeto pertenece depende de qué operaciones pueden invocarse sobre el objeto, cuál es el orden y tipo de sus argumentos y cuál es el tipo del resultado.

Por otro lado una clase es un conjunto de objetos, los cuales tienen exactamente la misma estructura interna y, por lo tanto, los mismos atributos y los mismos métodos. La clase define la implementación de un conjunto de objetos, mientras que un tipo describe como pueden utilizarse tales objetos. Un tipo puede estar implementado por varias clases. Inversamente, una clase puede implementar varios tipos; si una clase implementa un tipo automáticamente implementa todos los supertipos de ese tipo.

Una clase encapsula las abstracciones de datos y procedimientos que se requieren para describir el contenido y comportamiento de la vida real. Una clase es la implantación de un tipo de objeto. Especifica la estructura de datos y los métodos operativos permitidos que se aplican a cada uno de sus objetos. Ver Figura 1.2.4.4

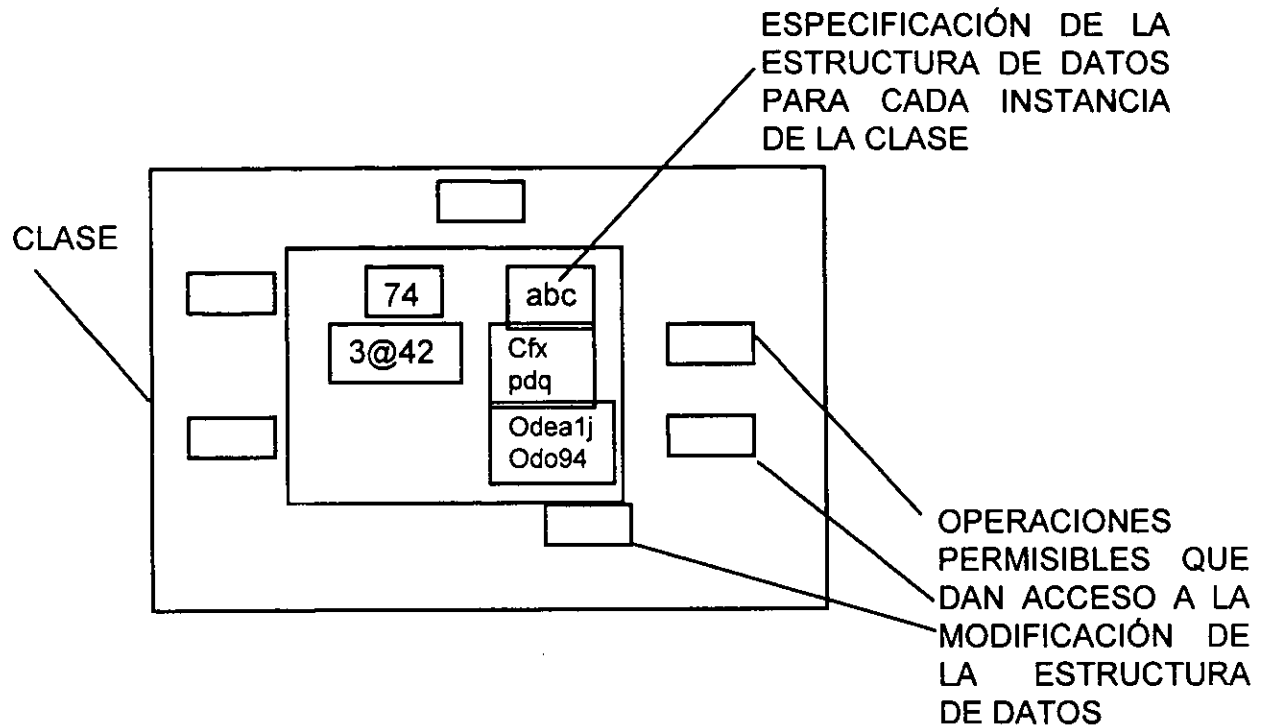


Figura 1.2.4.4 Cada clase especifica una estructura de datos, las operaciones permisibles y los métodos operativos.

La especificación de como se llevan a cabo las funciones de una clase se llaman método. Los objetos se pueden utilizar exclusivamente con métodos específicos.

Dentro de la estructura de clases se encuentran las superclases (colección de clases) y la subclase (concepto genérico de las clases).* Todos los objetos que existen dentro de una clase heredan sus atributos y las operaciones disponibles para su manipulación.

Estas definiciones implican la existencia de una jerarquía (clasificación u ordenación de la relación que se produce entre clases), en la cual los atributos y operaciones de la superclase son heredadas por subclases que pueden añadir, cada una de ellas, atributos privados y métodos. Como se puede observar en la Figura 1.2.4.5. que se muestra a continuación.

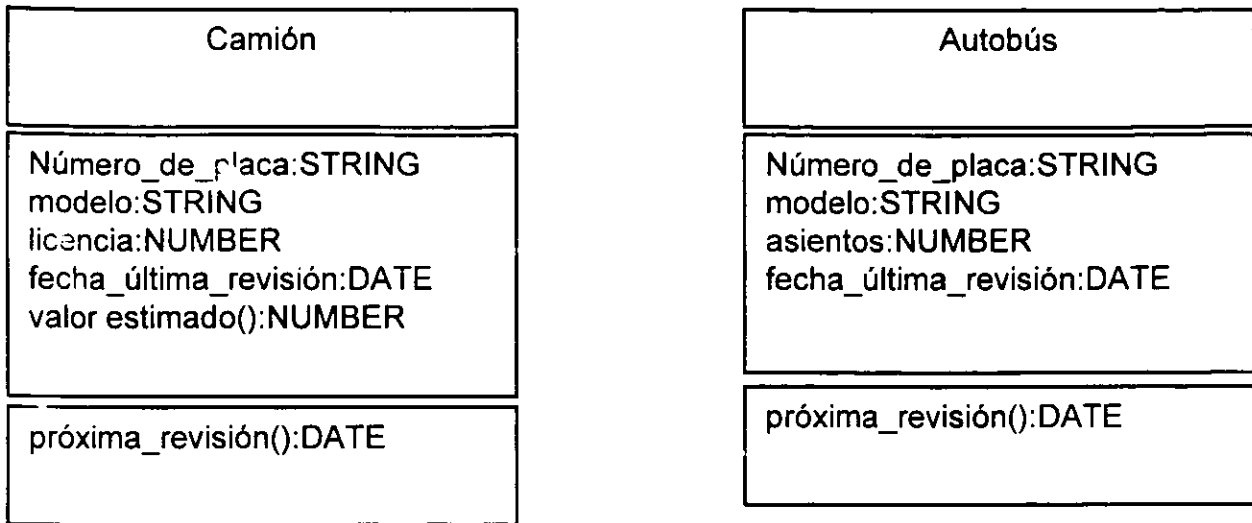


Figura 1.2.4.5 Representación de una jerarquía de clase

Herencia

Una clase se puede definir como especialización de una o más clases existentes, y heredará los atributos y métodos de tales clases. La clase así definida suele llamarse subclase, mientras que las clases a partir de las que ha sido definida se denominan superclases

Herencia de clase

Cuando se aplica una herencia, el conjunto de los atributos de las instancias de una clase consiste en la unión de los atributos heredados y de los atributos específicos de la clase.

La herencia de clase hace que la estructura de datos y operaciones estén disponibles para su reutilización por parte de sus subclases. La herencia de las operaciones de una superclase permite que las clases compartan el código en lugar de volver a definirlo. Como se muestra en la Figura 1.2.4.6. La herencia de estructura de datos permite revitalizar la estructura.

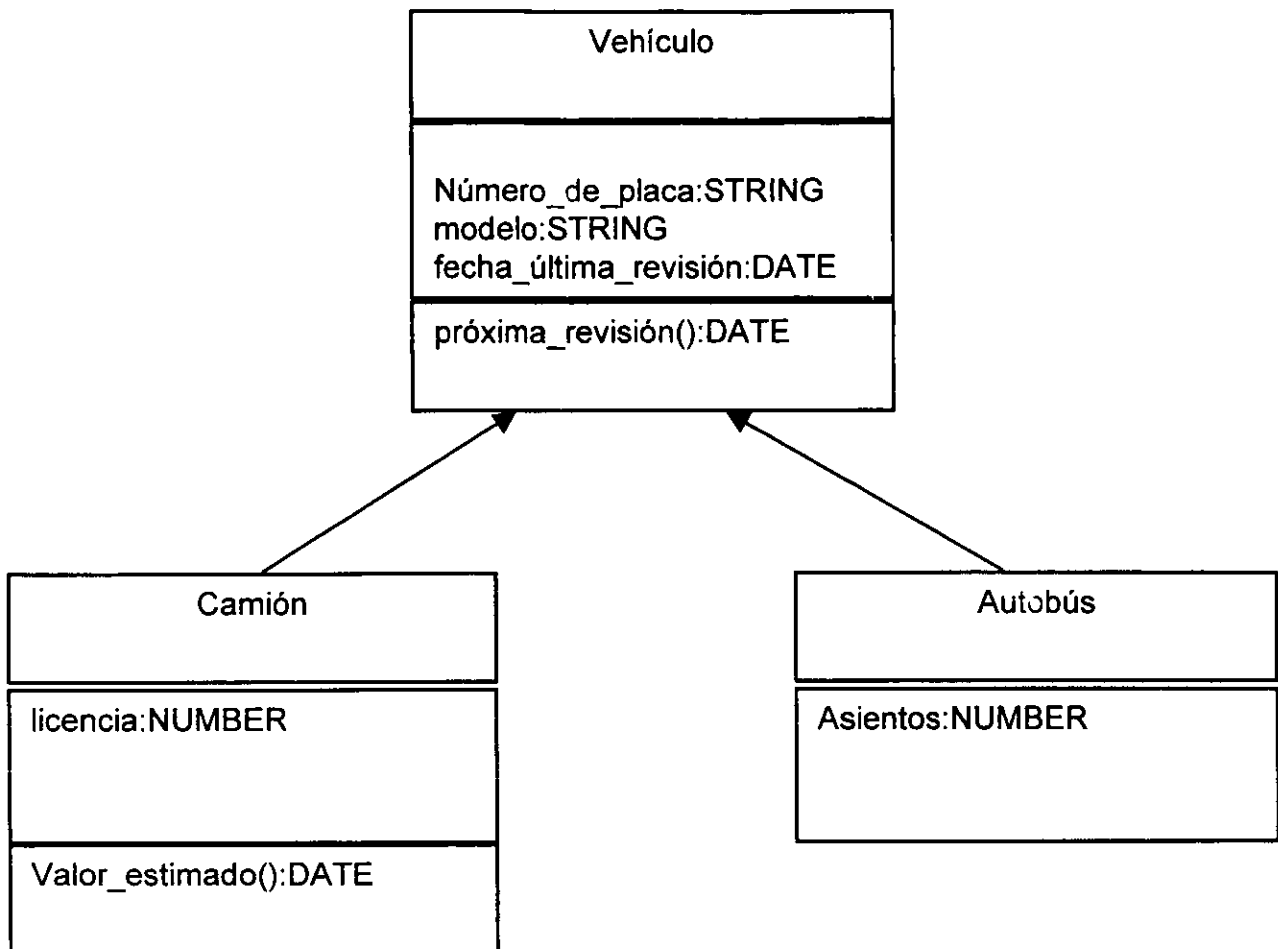


Figura 1.2.4.6 Herencia de operaciones de clase a un objeto.

Herencia simple.- Es aquella en la que una clase puede heredar la estructura de datos y operaciones de una superclase. En la Figura 1.2.4.7 se muestra un ejemplo de la herencia simple. Como podemos observar se van heredando los métodos de una clase a otra, aun cuando las siguientes clases ya cuentan también con sus propios métodos, los métodos heredados se siguen conservando.

En la figura 1.2.4.8 se muestra un ejemplo de la herencia múltiple que hereda a las tres clases inferiores los atributos de la primera y a su vez, las dos siguientes heredan sus atributos a la última.

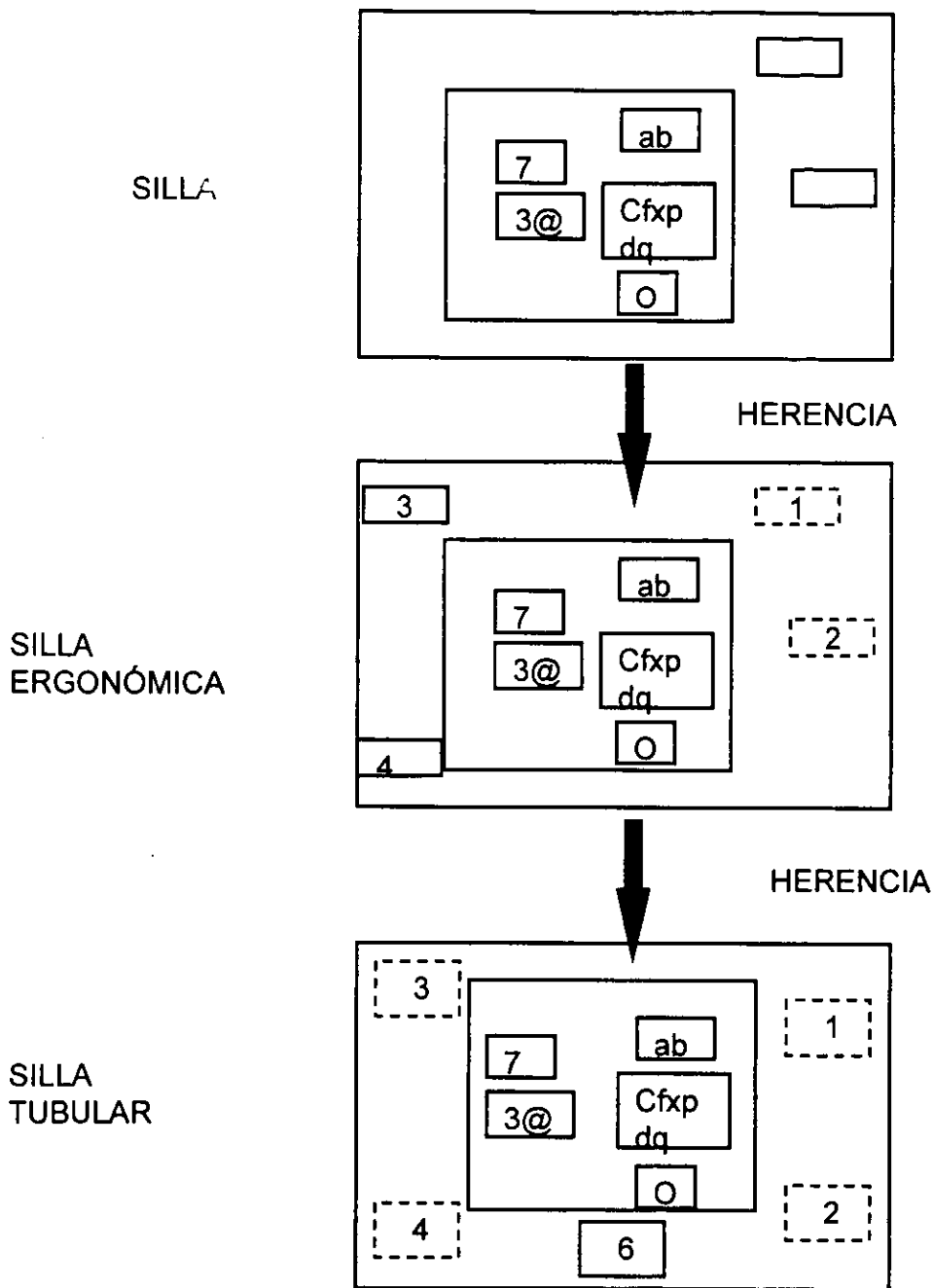


Figura 1.2.4.7 Herencia. La clase silla ergonómica hereda los métodos 1 y 2 de silla. Silla ergonómica tiene sus propios métodos 3 y 4. La clase silla tubular hereda los métodos 3 y 4, de silla ergonómica, además de tener su propio método 6

Herencia múltiple.- Una clase puede heredar estructuras de datos y operaciones de mas de una superclase. En la siguiente figura 1.2.4.8 se muestra un ejemplo:

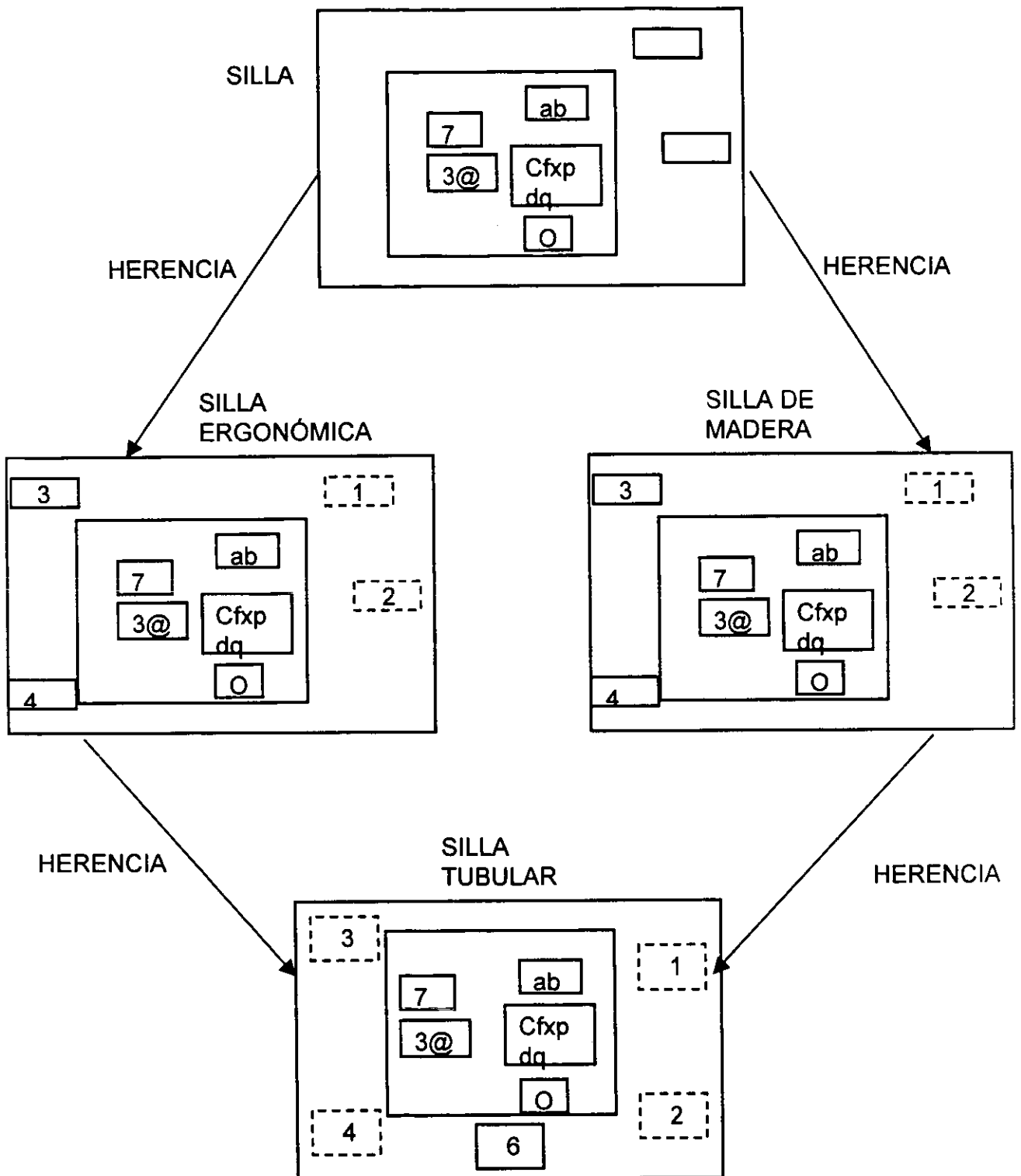


Figura 1.2.4.8 Las tres clases inferiores heredan los métodos de silla. Silla tubular hereda los métodos de silla ergonómica y silla de madera.

Encapsulamiento

Cada objeto contiene y define los procedimientos y las interfaces mediante las cuales se puede acceder a él y otros objetos pueden manipularlo. La interfaz de un objeto consiste en un conjunto de operaciones que pueden ser invocadas sobre el objeto. El estado de un objeto es manipulado mediante los métodos invocados por las operaciones correspondientes. El encapsulamiento es el proceso de compartir las características esenciales que distinguen a un objeto de otros tipos de objetos. El encapsulamiento permite separar las interfaces de una abstracción y su implantación. Ver Figura 1.2.4.9.

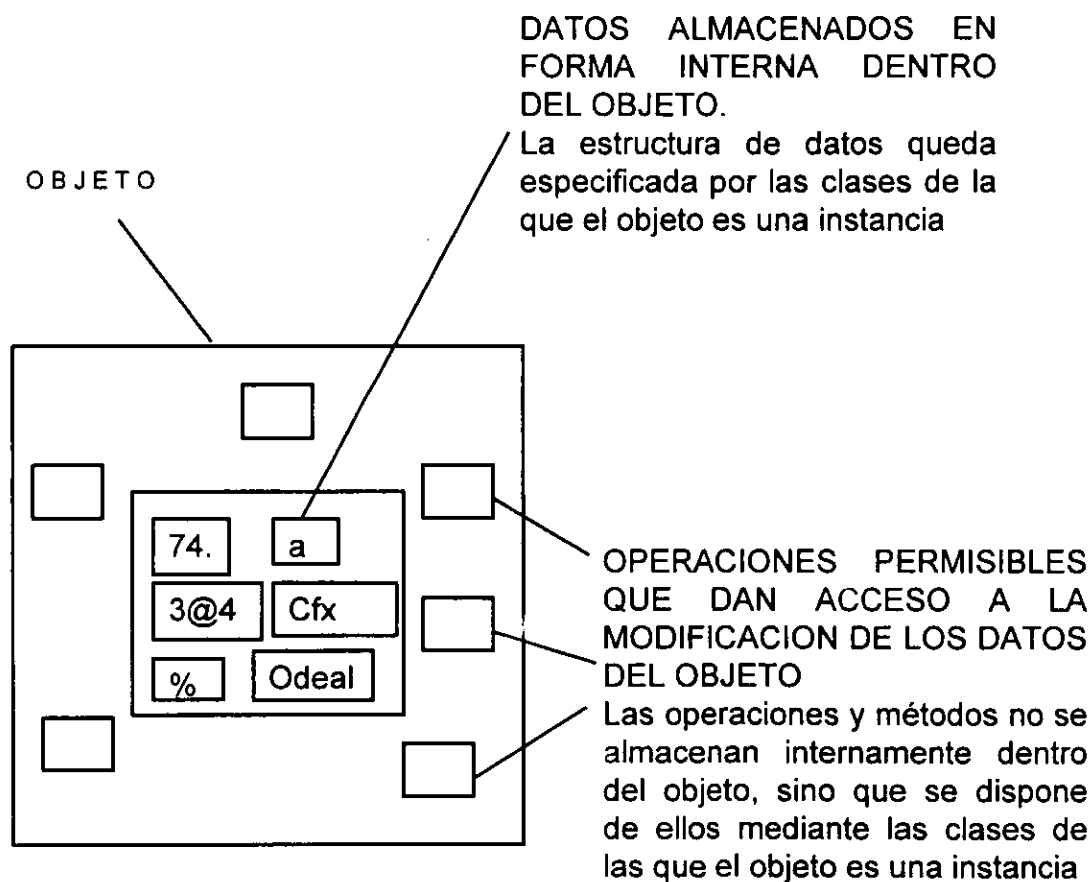


Figura 1.2.4.9 El encapsulamiento oculta la implantación de la estructura de datos y métodos de cada objeto. Los usuarios solo conocen las operaciones permisibles del objeto: las interfaces.

Polimorfismo

Se puede definir como la posibilidad de construir varios métodos con el mismo nombre, pero relacionados a la clase que pertenece cada uno, con comportamientos diferentes. Esto conlleva la habilidad de enviar un mismo mensaje a objetos de clase diferentes. Estos objetos recibirían el mismo mensaje global pero responderían a él en formas diferentes. Ver Figura 1.2.4.10

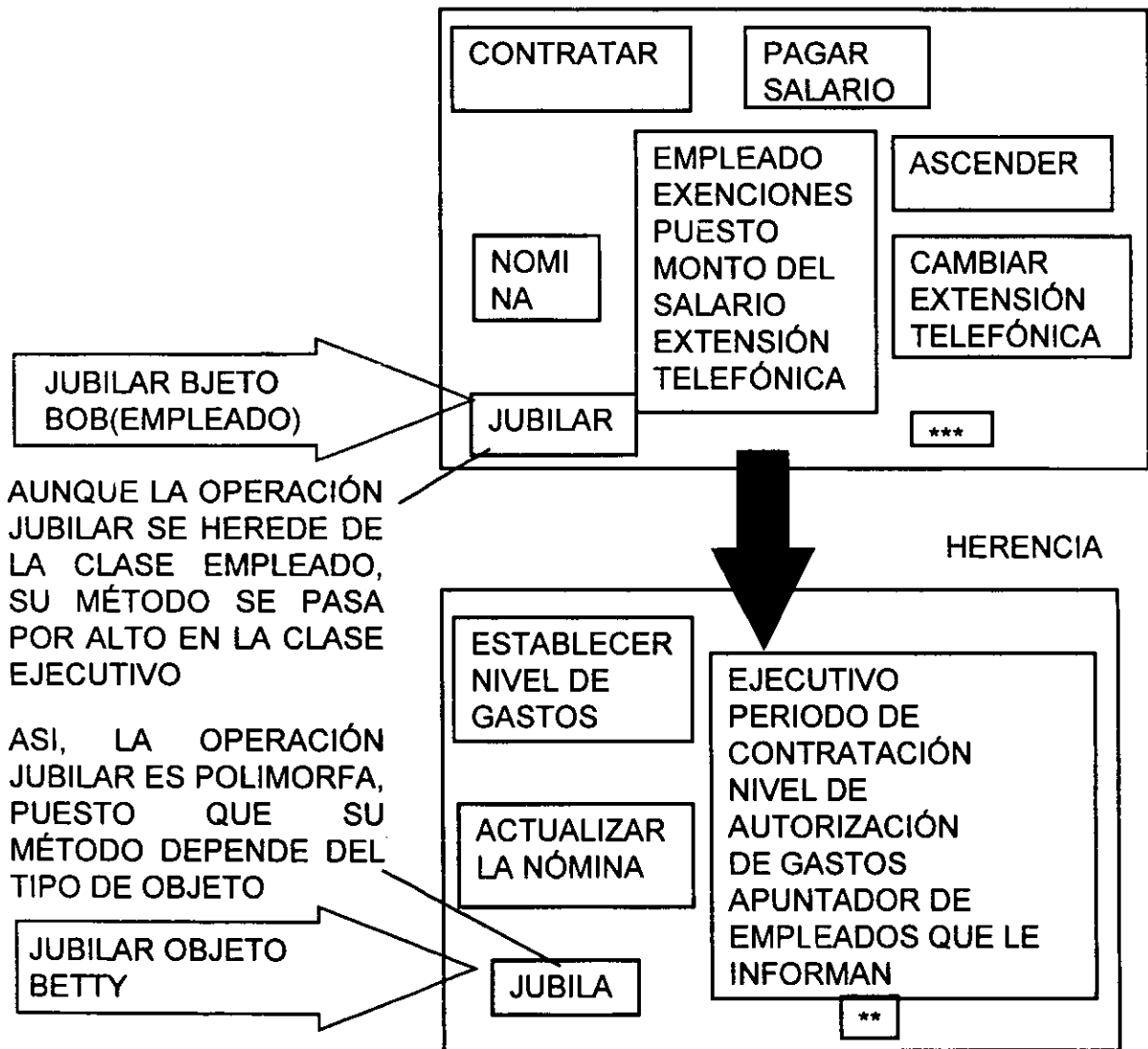


Figura 1.2.4.10 El método se puede implantar de manera diferente en una clase y su subclase. Las subclases pueden implantar la misma operación con métodos diferentes.

Esto se llama polimorfismo.

1.3 METODOLOGÍAS DE DISEÑO DE BASES DE DATOS ORIENTADAS A OBJETOS

Introducción

En los métodos clásicos de programación estructurada (orientada a procedimientos), un programa describe una serie de pasos que se realizarán en un orden establecido, es lo que se conoce como algoritmo. En la programación orientada a objetos, un programa describe un sistema de objetos que interactúan entre sí.

La programación orientada a objetos, implica una serie de conceptos claves. Estos conceptos son: abstracción de datos, encapsulamiento, clases y jerarquía de clases. La abstracción hace la escritura de grandes programas más sencilla. El encapsulamiento facilita el cambio y mantenimiento de un programa. La jerarquía de clases (herencia) es una herramienta que permite la extensibilidad de un programa. Estos conceptos fundamentales, que son soportados por los lenguajes de programación orientados a objetos, convenientemente aplicados, facilitarán el diseño (metodología de diseño) y la posterior programación de aplicación, así como su mantenimiento y adecuaciones a requerimientos posteriores.

El desarrollo de un sistema orientado a objetos requiere de los conocimientos fundamentales de orientación a objetos, la utilización de un método específico de diseño y el conocimiento del modo de implementación en un lenguaje de programación orientado a objetos. Todo lo anterior con el propósito de obtener un producto que satisfaga las necesidades y requerimientos de la entidad o los usuarios que así lo hayan requerido.

Este capítulo se enfoca al análisis y diseño orientado a objetos, en donde se describe una propuesta para el diseño de programas orientados a objetos basado en un método estándar, influenciado por las metodologías de Booch y Yourdon/Coad. Se describen también las etapas y las notaciones de Yourdon/Coad, Booch y OMT Rumbaugh.

Desarrollo de un sistema orientado a objetos

El diseño orientado a objetos no se realiza en términos de tareas o procesos (subrutinas), ni en términos de datos como lo hace el diseño estructurado, sino que se analiza el problema como un sistema de objetos que interactúan entre sí.

Existen numerosos métodos de desarrollo orientado a objetos, por mencionar algunos, se encuentran:

- Booch'91, Booch'93
- Yourdon/Coad
- Martín
- Shlaer & Mellor
- OMT Rumbaugh

Cada uno de estos métodos utiliza una serie de etapas que permiten pasar de las tradicionales etapas de análisis y diseño, a las fases de implementación, pruebas y mantenimiento. Cada método está sustentado por una metodología, que a su vez soporta una notación específica.

En general, al igual que un proyecto estructurado, un proyecto orientado a objetos se compone de las siguientes etapas:

Diseño orientado a objetos

Diseño estructurado

Análisis orientado a objetos

Diseño de datos

Diseño orientado a objetos

Diseño arquitectónico

Programación orientada a objetos

Diseño de procedimientos

No siempre están bien delimitadas las etapas o fases de análisis y diseño en orientación a objetos; por ello sintetizamos las ideas claves de las diferentes metodologías en una secuencia de pasos que denominamos diseño o desarrollo orientado a objetos. Otros métodos de diseño orientado a objetos contienen las etapas siguientes:

- Identificar clases y objetos
- Asignar atributos y comportamiento
- Encontrar relaciones entre clases
- Organizar las clases en jerarquías
- Implementar las especificaciones de diseño

Estas etapas con mayores o menores modificaciones suelen seguirse por la mayoría de los métodos del diseño orientado a objetos. De forma general un problema desarrollado con técnicas orientadas a objetos requiere primeramente, de identificar las clases (basadas en los objetos del programa), sus atributos y comportamiento, así como las relaciones entre ellas y su posterior implementación en un lenguaje de programación

Identificar clases y objetos

Las clases se determinan en base a los objetos que tiene el programa, los objetos son instancias de clases, son las entidades fundamentales de un programa orientado a objetos, las clases son entonces un conjunto de objetos.

Existen diversos criterios para identificar objetos o candidatos a objetos. Shlaer y Mellor, junto con Coad y Yourdon, son algunos de los métodos que mejor resumen las características que deben cumplirse para ser candidatos a objetos, en donde se localizan nombres o frases con nombres y verbos o frases con verbos de la definición o descripción de especificaciones del problema. Los nombres son buenos indicadores de existencia de objetos en el modelo orientado a objetos y los verbos son candidatos a métodos o funciones miembro. Sin embargo no todos los nombres identificados son buenos candidatos a clases. Después de encontrar todos los nombres, se examina la lista y se decide que nombres son realmente clases en nuestro sistema. Por ejemplo un nombre que no puede ser una clase es un número del Seguro Social, contiene datos pero ninguna acción sobre el cálculo. Los problemas que se pueden encontrar en un sistema son:

- Encontrar demasiados objetos
- Algunos nombres son atributos, no objetos

Algunos criterios (Shlaer y Mellor) para clasificar objetos son:

- Cosas tangibles (mesa, despacho, automóvil, base de datos)
- Interacciones (matrimonio, ventas, compras)
- Papeles o roles (director, propietario, cliente)
- Incidentes (viaje, transacción, llegada)
- Unidades organizacionales (departamentos, divisiones)
- Lugares, posiciones geográficas o físicas (ciudad, camino)
- Especificaciones (descripciones o estándares)

Es fácil identificar clases para un programa que modele objetos físicos. Por ejemplo, las clases avión y pasajero en un programa de líneas aéreas, la clase Dispositivo que represente unidades de disco e impresoras en un sistema operativo. Muchos programas no manejan entidades físicas, en estas situaciones se deben identificar las entidades conceptuales que manipulan el programa. Por ejemplo, la clase Proceso en un sistema operativo. Los sucesos (cosas que suceden a un objeto) e interacciones (cosas que suceden entre objetos) son también candidatos para clases. Por ejemplo, la clase transacción puede representar operaciones como depósitos, préstamos, transferencias en un programa bancario. Otros candidatos a clase son las estructuras de datos, como son pilas, colas, listas, etc.

Asignación de atributos y comportamiento

Una vez que se ha identificado una clase, la siguiente tarea es determinar que responsabilidades tiene. Las responsabilidades dentro de una clase pertenecen a dos categorías:

- Atributos
- Comportamientos

Cada clase tiene atributos que son las propiedades o características que lo describen y un comportamiento, que son las operaciones que un objeto puede realizar o se pueden realizar sobre el mismo.

En la metodología Yourdon/Coad los atributos describen el estado de un objeto, mientras que los métodos (funciones miembro) describen el comportamiento activo del objeto.

Los atributos describen la información relativa a los datos, es decir, la parte de datos de una clase.

Una vez que se han identificado los atributos y el comportamiento de una clase, se tienen candidatos a funciones miembro (métodos) de la clase. El comportamiento que se ha identificado implica las funciones miembro. Un estado de un objeto consta de los valores actuales de sus atributos.

Los métodos definen como se comporta el objeto. La diferencia entre un método y una función tradicional reside en que los métodos operan sobre datos de objetos, mientras que una función tradicional puede actuar sobre cualquier dato que se elija adecuadamente. Los atributos son manipulados por métodos en las clases y objetos. Por ejemplo, la clase rectángulo puede tener como atributos altura y anchura, la clase archivo puede tener el nombre demo.txt, modo de acceso de solo lectura y tamaño de 15 bytes. Algunos atributos nunca cambian mientras que algunos cambian constantemente. Cada clase tiene un comportamiento, que es el modo en como un objeto interactúa con los demás objetos y como cambia su estado durante esas interacciones. Por ejemplo, un usuario puede meter o quitar elementos de un objeto pila que cambia su estado interno.

Relaciones entre clases y objetos

Esta etapa trata de decidir el modo en que las clases utilizan otras clases; aunque algunas clases pueden existir aisladas, la mayoría no pueden y deben cooperar unas con otras. En esta etapa se analizan las relaciones entre clases y el modo en que estas se convierten en una implementación.

Las dos formas típicas de relaciones (jerarquía de clases) son:

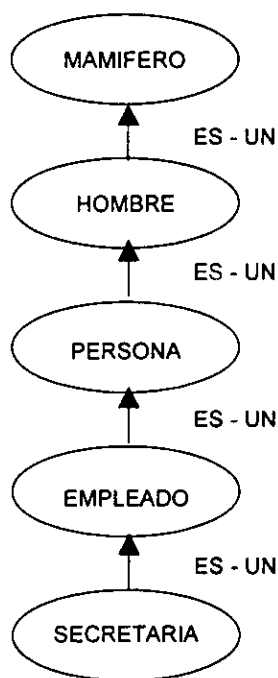
- es-un (is – a)
- tiene un (has – a) o parte de (part – of)

Una relación es – un se presenta cuando una clase es un subtipo de otra (a veces esta relación se denomina a – kind –of “un tipo de”). Por ejemplo, un oso es – un mamífero; una rosa es – una flor, etc.

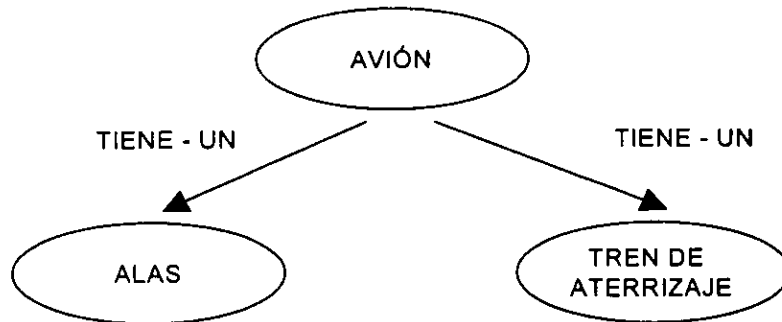
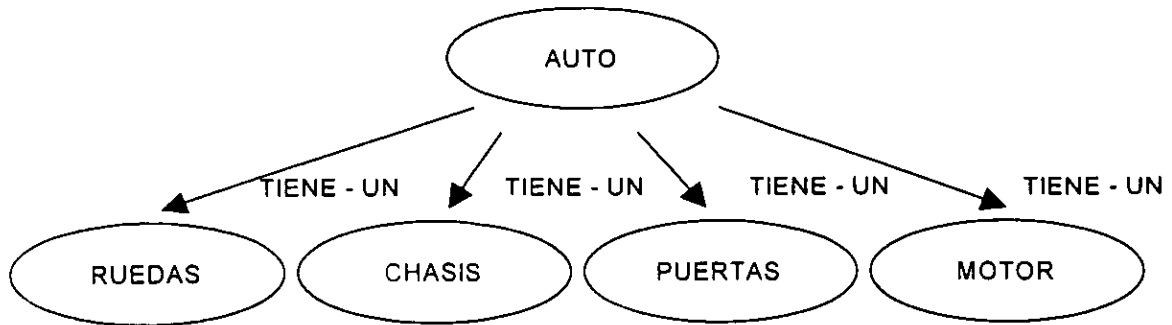
Una relación tiene – un o parte – de se representa cuando una clase contiene otras clases. Se denominan clases compuestas o agregadas. Por ejemplo, un automóvil se compone de puertas, motor, ruedas y chasis.

La relación es – un se conoce como generalización/especialización y la relación tiene – un se conoce como agregación/composición.

La relación es – un es únicamente una relación de clases, mientras que la relación tiene – un define una relación entre dos clases, entre una clase y un objeto o entre dos objetos. La relación es – un define una jerarquía de clases-subclases (herencia), mientras que la relación tiene –un describe datos que se deben mantener dentro de una clase.



(a)



(b)

Figura 1.3.1.1 Relaciones entre clases: (a) es – un (b) tiene - un

Las relaciones de agregación/composición (tiene – un) y generalización/especialización o también conocida como herencia (es- un) permiten a una clase reutilizar el código de otra clase, pero implican distintas relaciones. La composición se utiliza cuando una clase tiene (contiene) otra clase, mientras que la herencia solo se utiliza cuando una clase es un tipo de otra. Por ejemplo, un círculo no es un tipo de punto; tiene o contiene un punto, una secretaria es – un empleado.

Interfaz e implementación de las clases

La mayoría de los lenguajes que soportan el diseño orientado a objetos proporcionan mecanismos de abstracción de datos. El mecanismo para la definición de clases proporciona un medio para designar las operaciones a las que pueden acceder los

usuarios de la clase. El conjunto de operaciones proporcionadas para cada clase es la interfaz de la clase.

La interfaz de una clase proporciona su vista externa, con lo que acentúa la abstracción, mientras oculta su estructura y los comportamientos, consta de las declaraciones de todas las operaciones aplicables a instancias de esta clase, pero puede incluir también las declaraciones de otras clases, constantes, variables y excepciones que sean necesarias.

La implementación de una clase es su vista interna, abarca los comportamientos de las clases, consta de la implementación de todas las operaciones definidas en la interfaz de la clase.

La interfaz de una clase se puede dividir en tres partes:

- Pública.- Una declaración pública es accesible a todos los clientes
- Protegida.- Una declaración protegida es accesible solo a la propia clase, sus subclases y sus amigas
- Privada.- Una declaración privada es accesible solo a la propia clase y a sus amigas

Las características principales que deben seguirse en el diseño de la interfaz de cada clase son:

- Identificar las operaciones requeridas por usuarios de cada clase. Estas operaciones definen la interfaz pública de la clase.
- En el caso de una jerarquía de clases, identificar operaciones que son dependientes del tipo.

- Identificar las posibles funciones de ayuda requeridas por la clase para soportar las operaciones de su interfaz pública. Estas operaciones serán privadas o protegidas en una jerarquía de clases.
- Identificar la representación de cada clase

A continuación se describen las ventajas y etapas de algunas de las metodologías más utilizadas para el diseño o desarrollo orientado a objetos.

Booch

Esta metodología esta soportada por una variedad de herramientas automatizadas.

El método Booch abarca un micro proceso y un macro proceso de desarrollo, el nivel de micro define un conjunto de tareas de análisis que se reaplican en cada etapa en el macro proceso.

Booch considera que las etapas del diseño orientado a objetos son:

- Identificar las clases y objetos en un nivel dado de abstracción
 - Proponer objetos candidatos
 - Conducir el análisis de comportamiento
 - Identificar escenarios relevantes
- Definir atributos y operaciones para cada clase
- Identificar la semántica de las clases y objetos
 - Seleccionar y analizar escenarios
 - Asignar responsabilidades para alcanzar el comportamiento deseado
 - Dividir las responsabilidades para equilibrar el comportamiento

- Seleccionar un objeto y enumerar sus papeles y responsabilidades
Definir operaciones para satisfacer las responsabilidades
Buscar colaboraciones entre objetos
- Identificar las relaciones entre clases y objetos
Definir las dependencias que existen entre objetos
Describir el rol (papel) de cada objeto
Validar los escenarios por revisión completa
- Realizar una serie de refinamientos
Producir diagramas apropiados para el trabajo realizado en los puntos anteriores
Definir jerarquías de clases apropiadas
- Crear agrupamientos basados en clases comunes
- Especificar la interfaz y la implementación de las clases y objetos

Yourdon/Coad

Este método es considerado como uno de los más sencillos de aprender.

La notación del modelado es relativamente simple y las reglas para desarrollar el modelo de análisis son evidentes.

Las etapas que considera esta metodología son:

- Identificar objetos usando el criterio de ¿Qué buscar?
- Definir la estructura de generalización especificación
- Definir la estructura de todo – parte

- Identificar temas (representación de componentes de subsistemas)
- Definir atributos
- Definir servicios

Rumbaugh (OMT)

La técnica de Modelado de Objetos (OMT) fue desarrollada por Rumbaugh para el análisis, diseño del sistema y diseño del nivel de objetos.

La actividad de análisis crea tres modelos:

- Modelo de objeto (representación de objetos, clases y relaciones)
- Modelo dinámico (representación del comportamiento del sistema y los objetos)
- Modelo funcional (representación a alto nivel del flujo de información del sistema similar al diagrama de flujo de datos)

Las etapas que considera este método son las siguientes:

- Desarrollar la declaración del ámbito del problema
- Desarrollar el sistema de objetos
 - Identificar clases relevantes al problema
 - Definir atributos y asociaciones
 - Definir enlaces de objetos
 - Organizar las clases de objetos usando la herencia
- Desarrollar un modelo dinámico

Definir eventos y desarrollar una traza de eventos para cada escenario
 Construir un diagrama de flujo de eventos
 Desarrollar un diagrama de estados
 Revisar el comportamiento para comprobar consistencia y complejidad

- Desarrollar un modelo funcional para el sistema
- Identificar entradas y salidas
 Usar diagramas de flujo de datos para representar transformaciones de flujo
 Desarrollar la Especificación del Proceso (EP) para cada función
- Especificar criterios de restricciones y optimización

Comparación de metodologías de diseño o desarrollo orientado a objetos

Fichman y Kemerer comparan diferentes métodos de análisis convencional y orientado a objetos basándose en once dimensiones de modelado (etapas de modelado) y en diez componentes de modelado.

Etapa	Booch	Yourdon / Coad	Rumbaugh (OMT)
Identificación/Clasificación de entidades	✓	✓	✓
Relaciones de entidad general a específica y del todo a partes	✓	✓	✓
Descripción de atributos de entidades	✓	✓	✓
Estados y transiciones entre estados	✓	✓	✓
Especificación detallada de funciones	✓	✓	✓

Comunicación entre entidades (a través de mensajes o eventos)	✓	✓	✓
Identificación de servicios exclusivos	✓	✓	
Partición de modelos a gran escala		✓	✓
Descomposición descendente (top – down)		✓	
Otras relaciones de entidad	✓		
Secuencias de procesamiento de fin –a fin			

Tabla 1.3.1 Comparación de metodologías por etapas de modelado

Componente	Booch	Yourdon / Coad	Rumbaugh (OMT)
Representación de jerarquías de módulos	✓	✓	✓
Especificación de definiciones de datos	✓	✓	✓
Definición de clases y jerarquías	✓	✓	✓
Especificación de la lógica de procedimientos	✓	✓	✓
Representación de estados de objetos y transiciones	✓	✓	
Asignación de operaciones a clases		✓	✓

Especificación de conexiones de mensajes		✓	✓
Indicación de secuencias de procesos fin - fin		✓	✓
Identificación de servicios exclusivos		✓	
Definición detallada de las operaciones			

Tabla 1.3.2 Comparación de metodologías por componentes de modelado

Las metodologías de diseño orientadas a objetos cumplen con la gran mayoría de las etapas y componentes marcados por Fichman y Kemerer.

La terminología y etapas del proceso para cada una de los métodos tienen diferencias, sin embargo los procesos generales son en realidad muy similares.

Notaciones gráficas

Notación de Booch'93

La notación de Booch'93 incorpora dos tipos distintos de diagramas: diagramas de clases y diagramas de objetos. Cada diagrama tiene su propia notación y un conjunto de iconos.

Los diagramas de clases documentan la estructura de las clases, sus relaciones con otras y las relaciones generales de sus instancias.

Las relaciones entre clases que contempla la notación de Booch son:

- Generalización/especialización (hereda, es – un)

- Agregación/composición (tiene - un)
- Asociación
- Uso

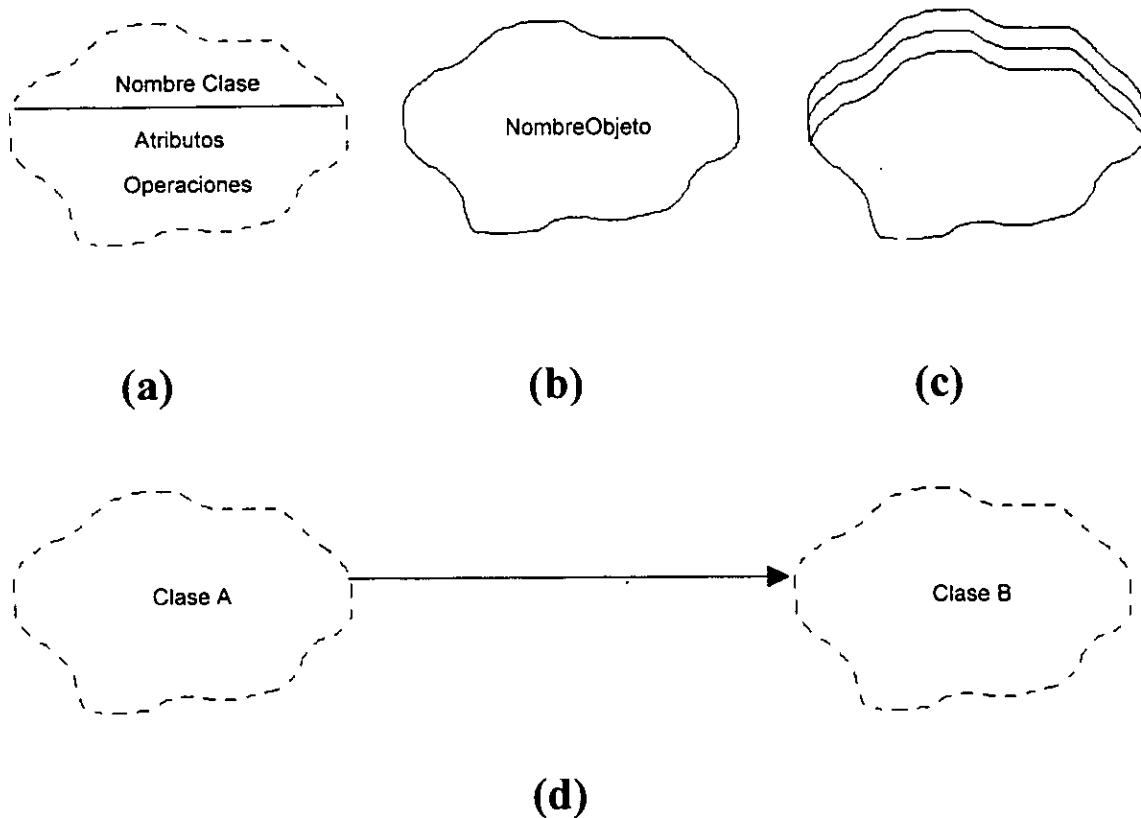
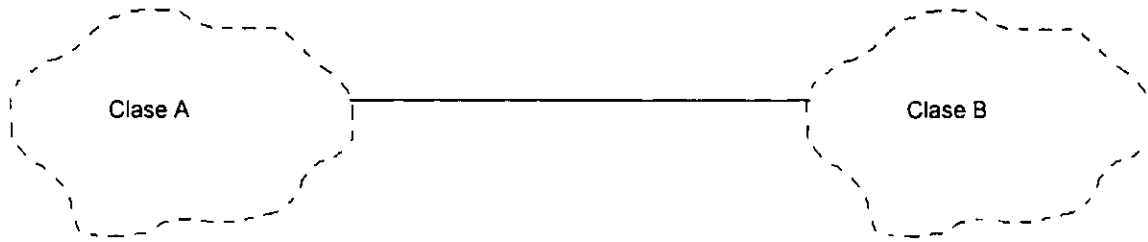
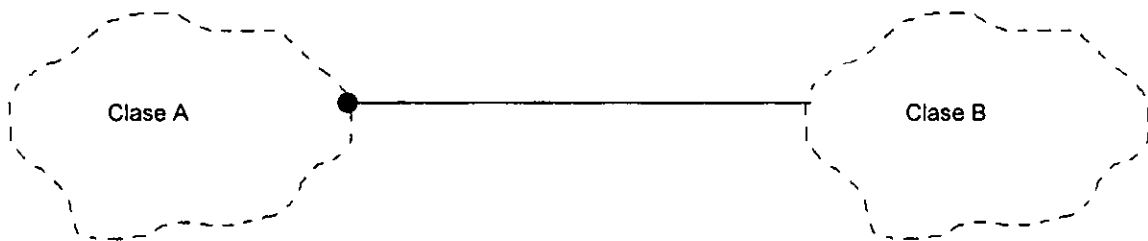


Figura 1.3.1.2 Iconos de: (a) clases, (b) objeto (c) objetos múltiples (d) Relación hereda (es -un)

La relación de herencia entre clases muestra que la subclase comparte la estructura o comportamiento definido de uno o más superclases. La flecha apunta hacia la clase base.



(a)



(b)

Figura 1.3.1.3 (a) Relación de agregación (tiene – un) (b) Relación de asociación

La relación de agregación muestra una relación entre clases todo y partes (parte – de, tiene – un). El objeto agregado está construido físicamente a partir de otros objetos.

La relación de asociación representa una conexión semántica entre dos clases. Las asociaciones son bidireccionales; son las relaciones más generales de todas y las más débiles semánticamente.

La relación de uso entre dos clases muestra que la fuente depende de clase destino para proporcionar servicios, tales como:

- La clase fuente accede a un valor (constante o variable) definido en la clase destino.
- Operaciones de la clase fuente llaman a operaciones de la clase destino.

El círculo abierto designa la clase fuente y el otro extremo es la clase destino.

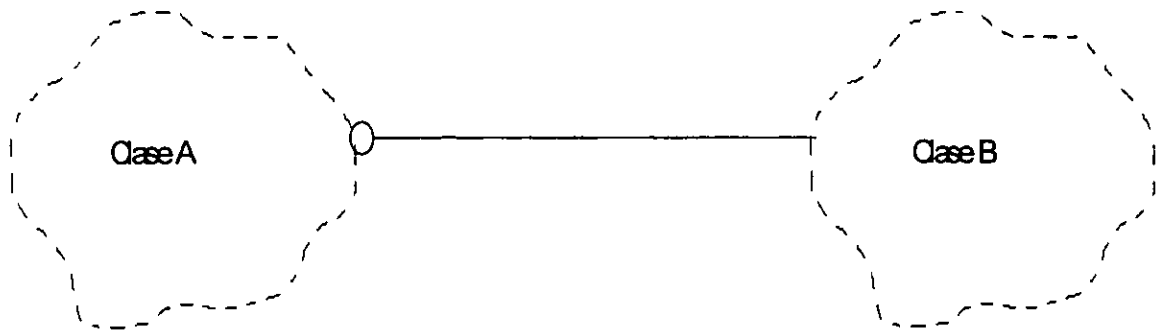
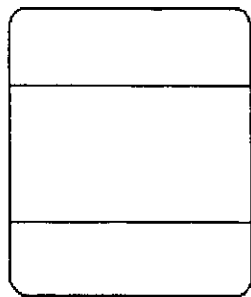


Figura 1.3.1.4 Relación de uso

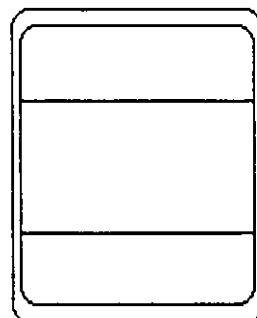
Notación de Yourdon

Los iconos que representan a clases y objetos se diferencian por la línea más tenue que circunda al icono interior.

El icono objeto se representa internamente con atributos y métodos.



(a)



(b)

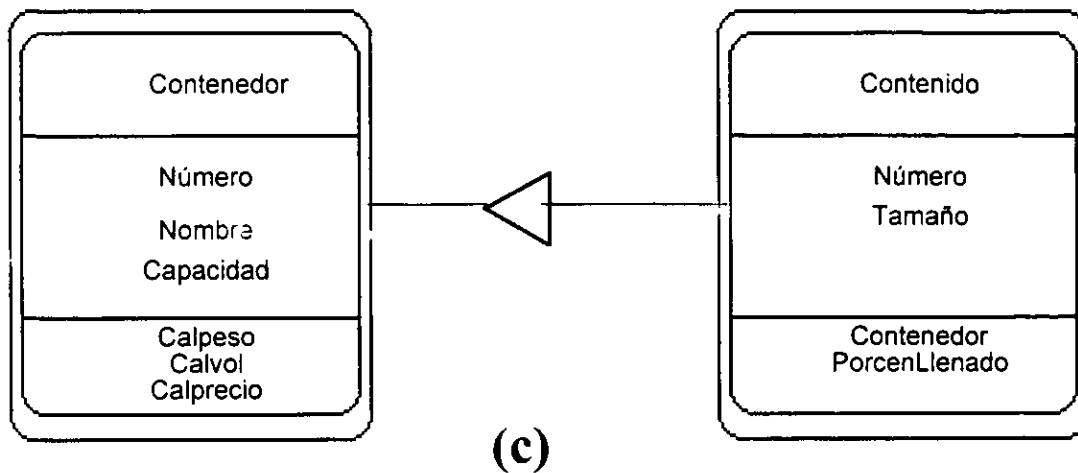
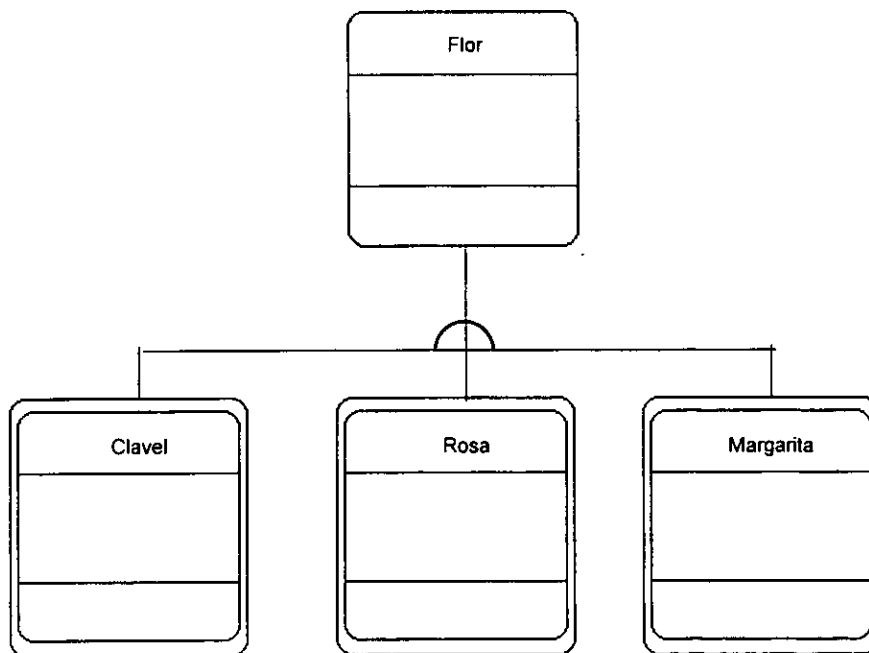
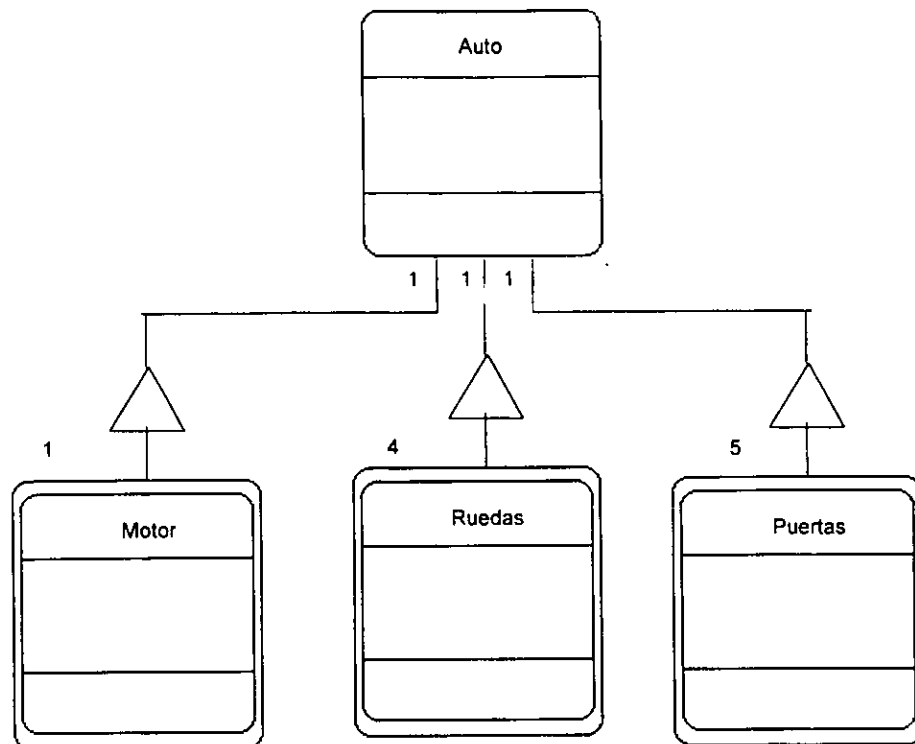


Figura 1.3.1.5 Iconos: (a) clase; (b) clase y objetos (c) clase y objetos (atributos y métodos)

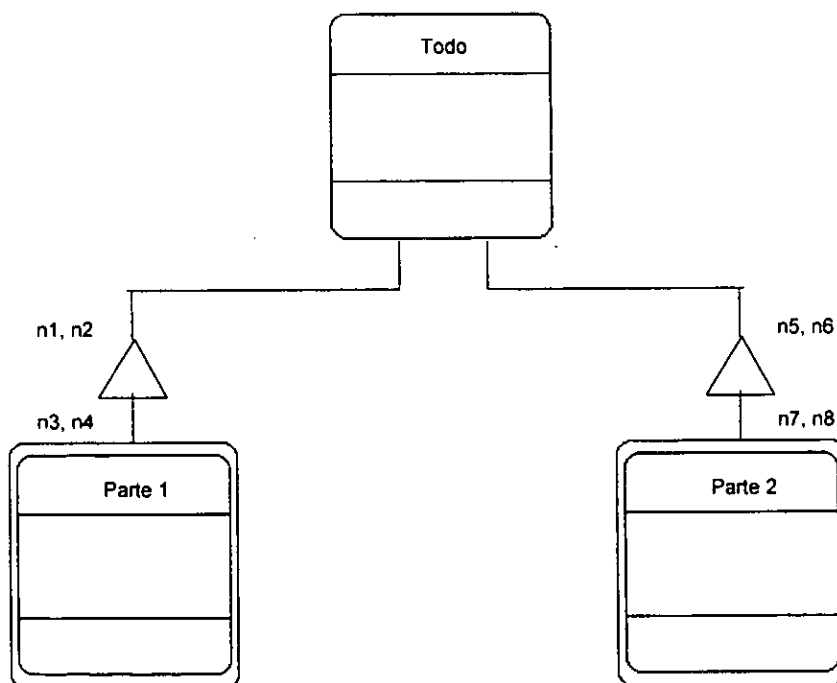
La relación de generalización se representa con un icono semicircular, mientras que la relación de agregación se representa con un icono triangular.



(a)



(b)



(c)

Figura 1.3.1.6 Relaciones: (a) generalización; (b) y (c) agregación

Notación de Rumbaugh (OMT)

Los iconos de las clases y objetos, como instancias de la clase.

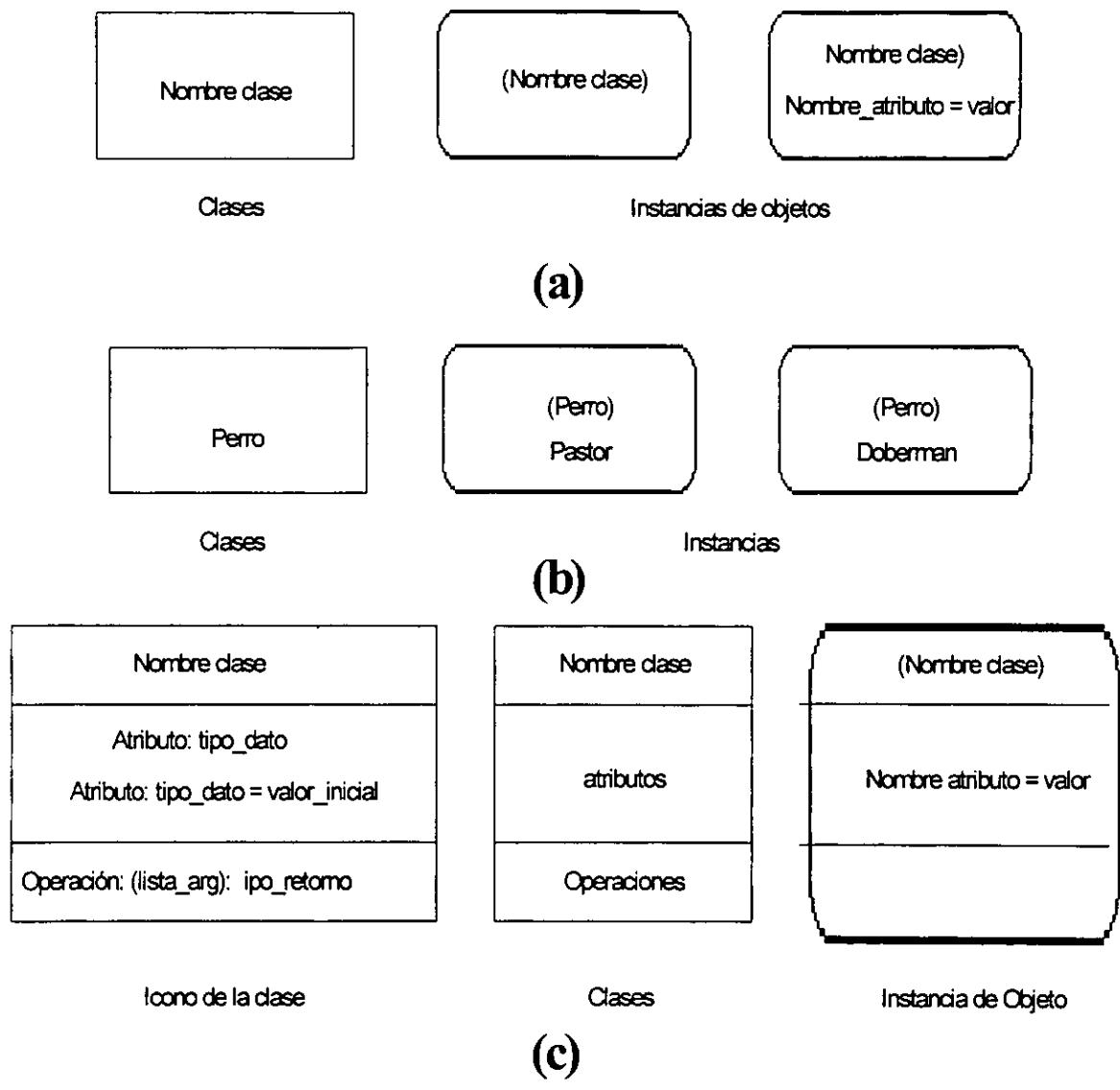
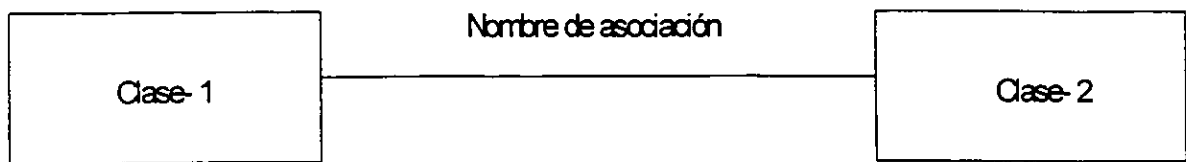
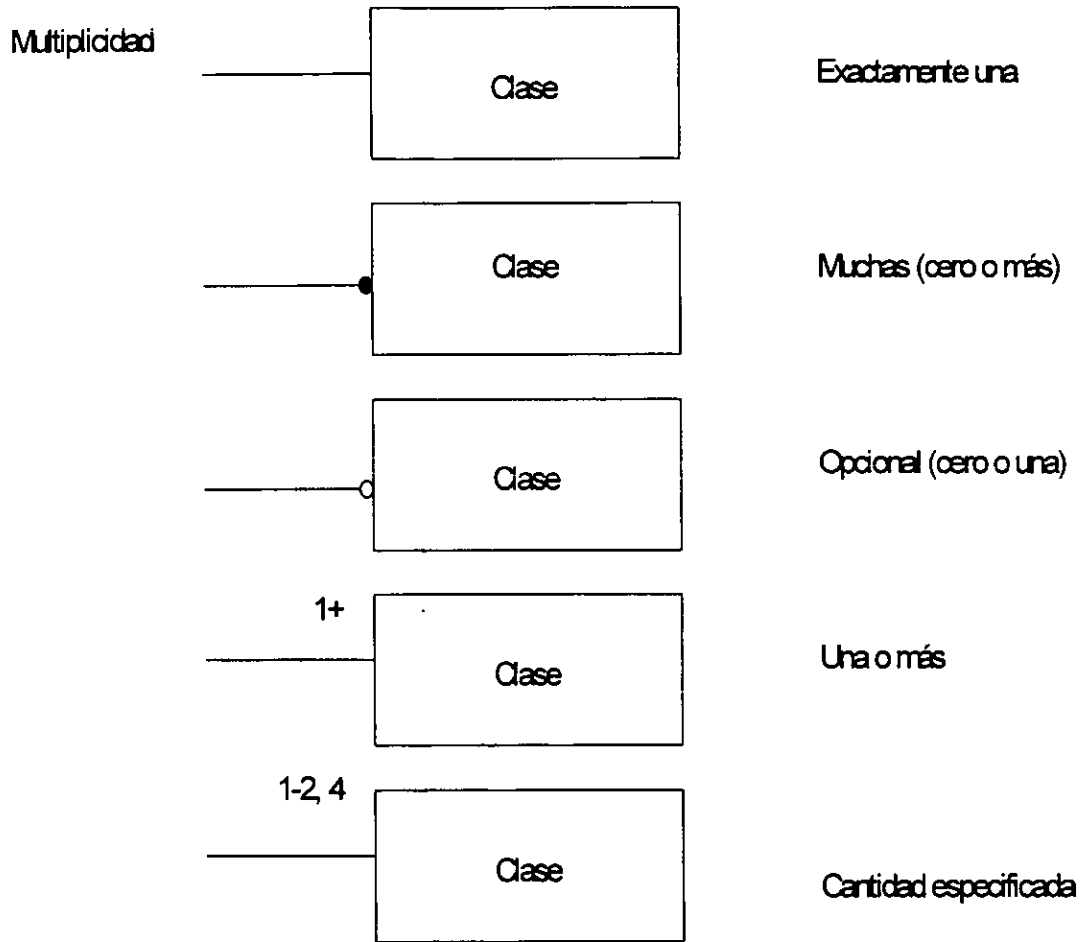


Figura 13.1.1.7 (a) Clases y objetos (b) Atributos y operaciones de una clase

Las relaciones de asociación entre clases se representan con los iconos de la Figura 13.1.1.8 y pueden representarse diagramas de clases y diagramas de instancias.



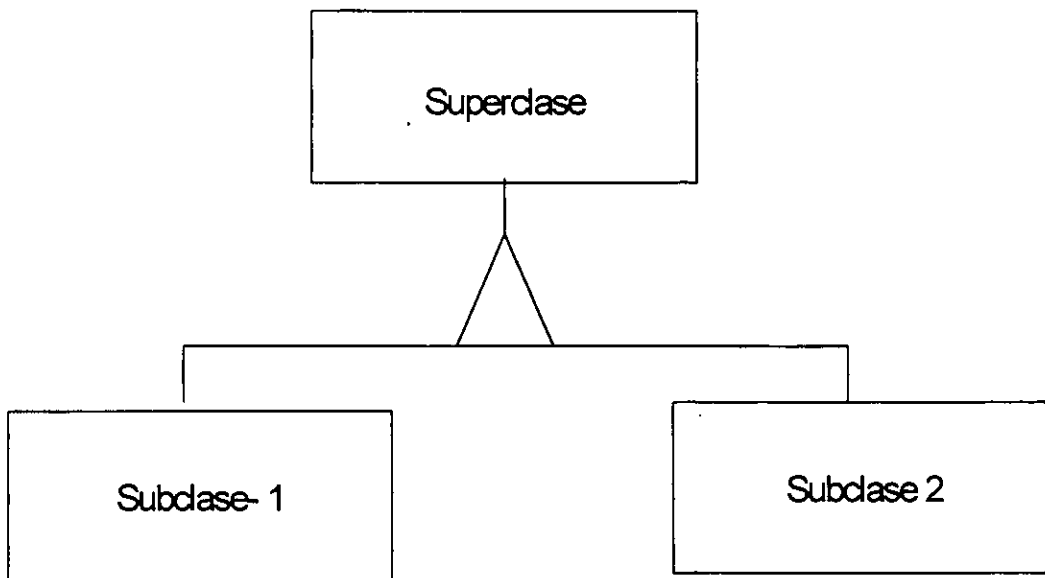
(a)



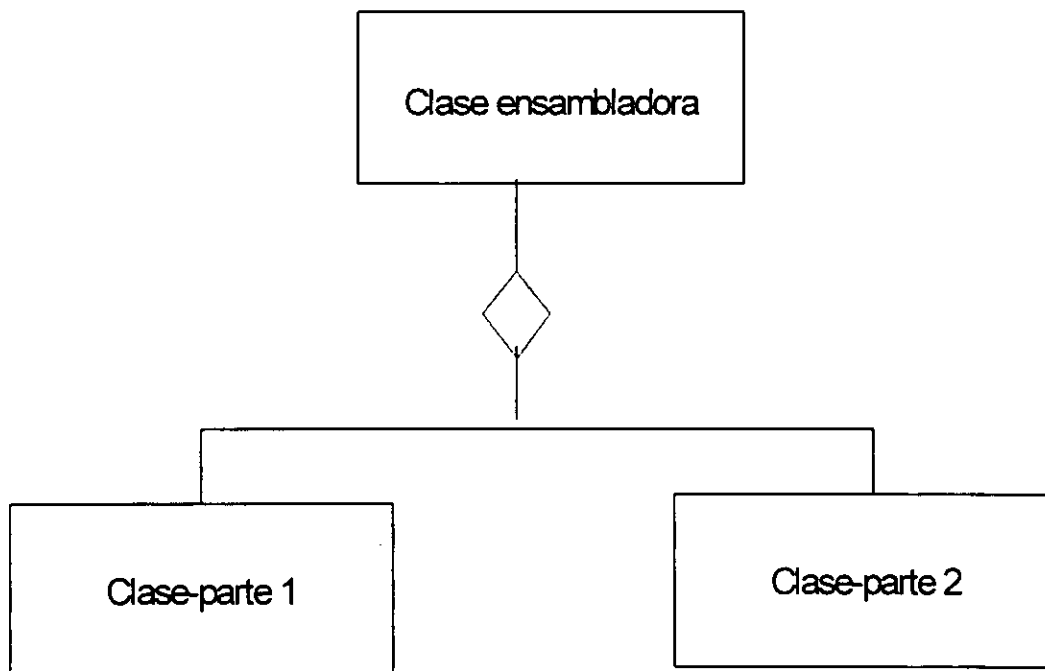
(b)

Figura 13.1.1.8 Relaciones de asociación; (a) iconos (b) multiplicidad

Las relaciones de generalización y agregación se representan con iconos en forma de triángulo, rombos y con notaciones jerárquicas similares a la notación de Coad/Yourdon.



(a)



(b)

Figura 13.1.1.9 (a) Relación de generalización; (b) Relación de agregación

1.3.1 Elección de la metodología óptima

La metodología a emplear, plantea las siguientes fases:

- **Análisis de la aplicación.** En esta etapa se darán a conocer los requerimientos funcionales y no funcionales de las áreas usuarias, las características fundamentales con las que deberá contar el sistema con el fin de obtener el conocimiento preciso de todos los aspectos que involucren el sistema.
- **Diseño de la aplicación.** En esta fase se diseñará la base de datos, apoyándonos en la metodología de diseño de bases de datos orientadas a objetos propuesta por **Yourdon/Coad**. Esta decisión es tomada con base en las características que presenta este método (sencillez y facilidad de aprendizaje y aplicación). También se diseñarán las relaciones entre los diferentes grupos de información, la funcionalidad del sistema y las interfaces con los diferentes usuarios.
- **Implementación de la aplicación.** Con las especificaciones de procesos del sistema y con los requerimientos funcionales, se genera la infraestructura de sistemas que será la base para la estandarización. Se desarrollarán los programas de los módulos identificados generándose las pruebas individuales y modulares.
- **Realización de pruebas.** Dependiendo del módulo que se trate y las condiciones, se pueden realizar las pruebas integrales del sistema. Las revisiones de los usuarios son fundamentales para la conclusión de esta fase.
- **Liberación del sistema.** Los módulos serán revisados y aprobados por el área de sistemas. La capacitación del área usuaria es vital para que la operación del sistema sea inmediatamente después de concluida esta fase.

1.4 CONCEPTOS BÁSICOS DE CONTABILIDAD FINANCIERA

1.4.1 Origen y Definición

Para definir la contabilidad financiera de una manera correcta y satisfactoria, es necesario partir de la idea de que ha evolucionado a través del tiempo y que de un registro empírico o inspirado tan solo por la necesidad de llevar un orden de cuenta y razón y de las operaciones realizadas, evoluciono hacia una serie de reglas, métodos, leyes y principios, lo que dio origen a que, al definirla en diferentes puntos del tiempo, se utilizaran diferentes puntos de vista y criterios, mismos que en el futuro también evolucionaran, cambiando su significado y la manera de definirla.

La contabilidad financiera tiene su origen en la necesidad de cuantificar las medidas de la posición financiera, el ingreso de las entidades de negocios y las variantes que constituyen las medidas de la posición financiera. La justificación para el estudio de la contabilidad financiera la encontramos en la importancia de estas medidas como base para evaluar tanto la actuación de los directores de negocios, como la salud de la entidad. Lo anterior nos indica que los resultados de la contabilidad financiera son utilizados para la toma de decisiones ya que se relaciona con:

- Registro de transacciones financieras.
- Sumarización y presentación de información financiera en reportes
- Análisis de los reportes financieros.

La contabilidad financiera se ubica como un ente mediante el cual los interesados pueden medir, evaluar y darle seguimiento al progreso, estancamiento o retroceso en la situación financiera de alguna entidad, resultado de la lectura de los datos económicos y financieros, cuantitativos, reunidos, transformados y resumidos en informes denominados estados financieros, mediante los cuales y dependiendo de su utilidad y confiabilidad, permiten tomar decisiones relacionadas con dicha empresa.

El IMPC (Instituto Mexicano de contadores públicos) define la contabilidad financiera de la siguiente Manera: “La contabilidad financiera es una técnica que se utiliza para producir sistemática y estructuradamente información cuantitativa expresada en unidades monetarias de las transacciones que realiza una entidad económica y de ciertos eventos económicos identificables y cuantificables que la afectan, con el objeto de facilitar a los diversos interesados el tomar decisiones en relación con dicha entidad económica. Figura 1.4.1.1.

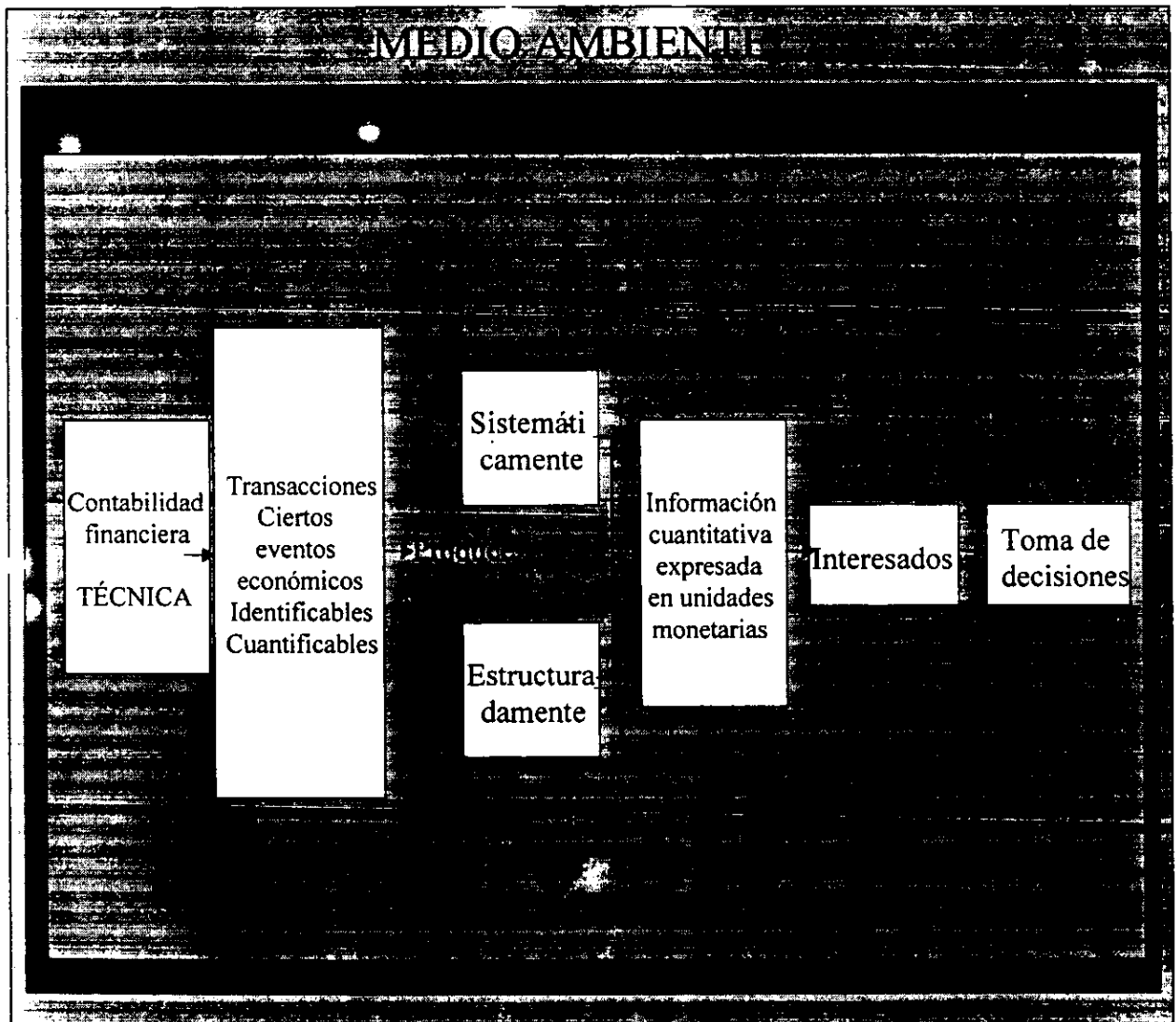


Figura. 1.4.1.1 Contabilidad Financiera

1.4.2 Importancia y utilidad

Por lo descrito en el punto anterior, se dice que la finalidad de la contabilidad financiera es proporcionar información útil, confiable y comprensible, que sirva de base para la toma de decisiones. También comprende dentro de su utilidad los siguientes tres objetivos generales:

1. Proporcionar información útil para los actuales y potenciales inversionistas y acreedores, y para otros usuarios que han de tomar decisiones racionales de inversión y de crédito.
2. Preparar información que ayude a los usuarios a determinar los montos, la oportunidad y la incertidumbre de los proyectos de entrada de efectivo asociados con la realización de inversiones dentro de la empresa, con el propósito de incrementar el capital.
3. Informar acerca de los recursos económicos de una empresa, los derechos sobre estos y los efectos de las transacciones y acontecimientos que cambien esos recursos y los derechos sobre aquellos.

Además de lo anterior la contabilidad financiera puede ser utilizada como parte e instrumento de control interno, para informar sobre como hacer más eficiente los bienes y recursos de la empresa en las actividades que esta misma realiza, como se puede observar en la figura 1.4.2.1.

De los puntos anteriores, se desprende información conformada en diferentes tipos de estados financieros y que ayuda a tener una visión más amplia de los recursos de la empresa para la toma de decisiones, tales estados financieros los listamos a continuación:

- a) Estado de posición financiera.

- b) Estado de resultados.
- c) Estado de cambios en la situación financiera.
- d) Estado de cambios o modificaciones en las cuentas de capital contable.

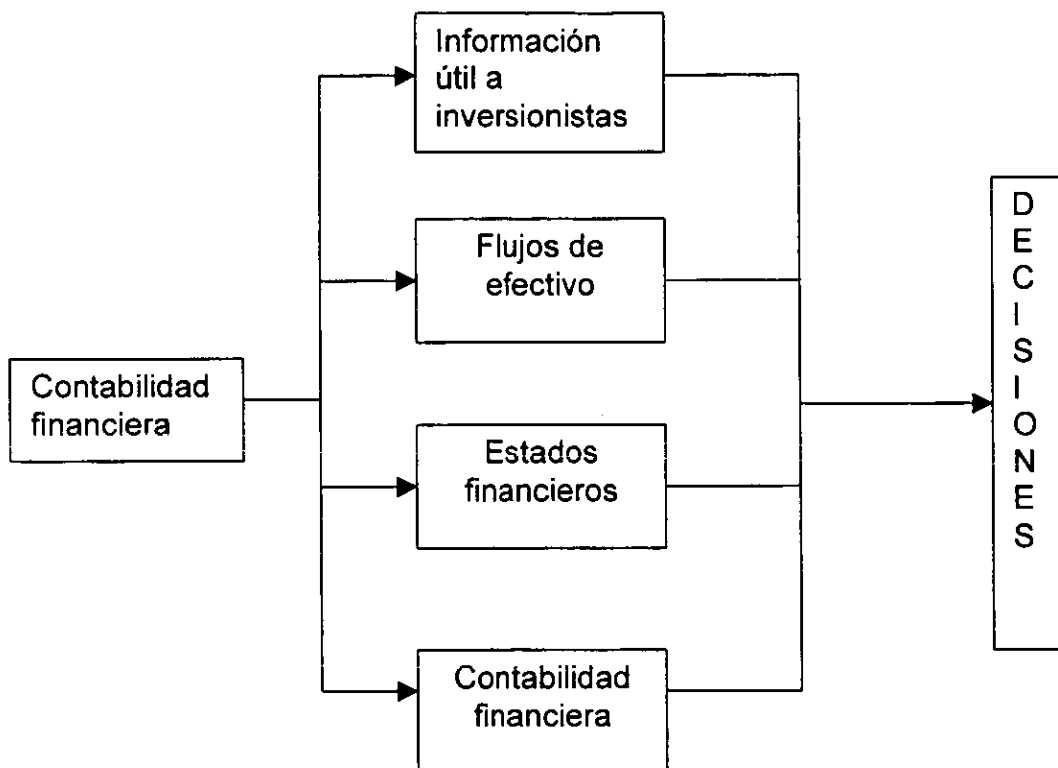


Figura 1.4.2.1 Finalidades de la contabilidad financiera.

Finalmente, la contabilidad financiera que corresponde a las definiciones que se han presentado y cuyo objetivo es presentar información financiera de propósito y usos generales, tanto en beneficio de la gerencia de la empresa, sus dueños o socios y a todos aquellas personas externas que se interesen en conocer dicha información financiera de la empresa, con el fin de tener un conocimiento más amplio o detallado de esta situación; este tipo de contabilidad nos proporciona, prepara y presenta estados financieros con el propósito de que puedan ser utilizados por agentes externos o auditorías.

1.4.3 Principios de contabilidad generalmente aceptados

La globalización en todos los ámbitos de la economía, ha llevado al mundo de los negocios a impulsar y al mismo tiempo conocer las normas, los principios y las políticas contables aceptadas en el ámbito internacional, con el propósito de obtener información financiera útil a los usuarios internos y externos para tomar decisiones sobre la entidad emisora de los estados financieros.

Se nos definen los principios de contabilidad generalmente aceptados, como conceptos básicos que establecen la identificación y delimitan el ente económico, las bases de cuantificación de las operaciones y la presentación de la información financiera cuantitativa por medio de los estados financieros. Los principios que delimitan e identifican al ente económico y a sus aspectos financieros son:

- **Entidad:** el objetivo principal de este principio es el de determinar al sujeto de la contabilidad, es decir, el ente económico que realiza actividades de negocios y que, por consecuencia y necesidad, debe llevar contabilidad. Así como delimitar perfectamente la diferencia entre la personalidad jurídica de la empresa y de los socios o dueños de la misma, con el fin de presentar información financiera de un solo ente diferente de los demás.
- **Realización:** este principio nos indica que todo lo que se realizó debe aparecer en la contabilidad y en los estados financieros, y viceversa, es decir, que todo lo que aparece en los estados financieros y la contabilidad, debió haberse realizado.
- **Periodo contable:** este principio surge de la necesidad del inversionista de poder tomar decisiones, con frecuencia, sobre el o los resultados de una empresa o negocio en marcha y con una existencia continua, por lo que su existencia se divide en periodos o lapsos convencionales de tiempo, denominados periodos contables.

Los principios de contabilidad que establecen las bases para la cuantificación de operaciones del ente económico y su presentación son:

- **Valor histórico original:** cuando se habla de valor histórico se acostumbra a expresarlo en términos de precio de costo, costo histórico o simplemente costo. Lo anterior se define de la siguiente manera “Las transacciones y eventos económicos que la contabilidad cuantifica se registran según las cantidades de efectivo que se afecten o su equivalente o a la estimación razonable que de ellos se haga al momento que se consideren realizadas contablemente”.
- **Negocio en marcha:** se dice que el negocio esta en marcha o bien que la vida de la entidad se presume en existencia permanente, salvo especificación de lo contrario por lo que las cifras de sus estados financieros representan valores históricos o modificaciones de los mismos, obtenidos de manera sistemática.
- **Dualidad económica:** esté principio nos habla principalmente de los recursos y su procedencia, con los que la entidad cuenta para realizar sus fines. De tal manera que la dualidad económica se integra con:
 1. Los recursos de los que dispone la entidad para la realización de sus fines.
 2. Las fuentes de dichos recursos. Como se ejemplifica en la figura 1.4.3.1.

El principio que hace referencia a la información es:

- **Revelación suficiente:** este principio nos indica que debemos evitar en lo posible la presentación de datos que sean confusos o que nos expresen información contraria a la que necesitamos, tampoco debemos incluir

demasiada información, que lejos de ayudarnos, nos entorpece el análisis, sin embargo se deben resaltar aquellos datos o acontecimientos cuya significación es de vital importancia para la gerencia y, además, que toda la información sea presentada y valuada correctamente.

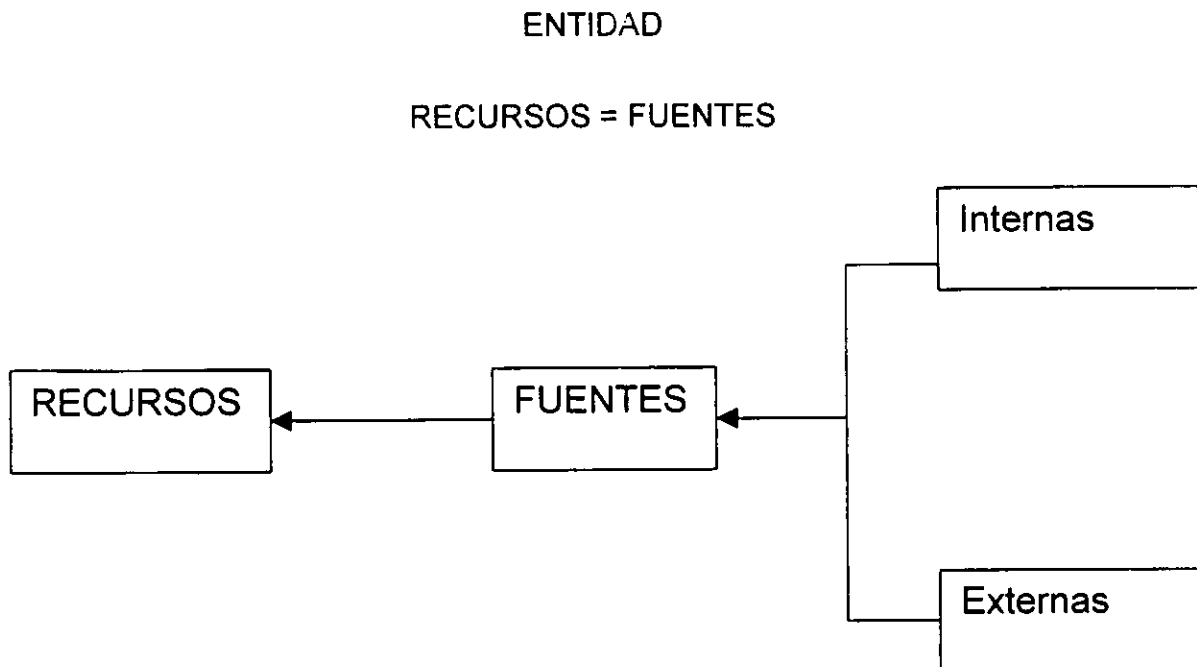


Figura 1.4.3.1 Dualidad económica

Y los principios que conforman las clasificaciones anteriores como requisitos generales del sistema son:

- **Importancia relativa:** Este principio se refiere básicamente al efecto que tienen o tendrán las partidas en la información financiera, es decir que toda partida cuyo monto o naturaleza sea significativo, debe de ser procesada y presentada para no distorsionar los objetivos de la información financiera.
- **Comparabilidad:** este principio establece que para lograr una adecuada comparación de los estados financieros de la entidad y poder conocer su evolución o, mediante la comparación de otros estados financieros de otras

entidades, conocer su posición relativa, es necesario que dicha información sea obtenida mediante la aplicación de los mismos procedimientos, principios y reglas particulares. Existen situaciones especiales que afectan la comparabilidad como son:

- a) Cambios en las reglas particulares por elección.
- b) Cambios de una regla particular establecidos por una nueva regla particular.
- c) Cambios en estimaciones contables.
- d) Correcciones a errores en la información financiera de ejercicios anteriores.
- e) Otras circunstancias como partidas especiales, extraordinarias y operaciones discontinuadas.

1.4.4 Registro de operaciones

Para hablar de los procedimientos para el registro de operaciones es necesario que en la entidad, tengamos presente ante todo las características de la contabilidad financiera de utilidad, de apegarse a los propósitos del usuario para poder brindarle la información que le satisfaga.

Por lo que al establecer o diseñar el procedimiento para registrar operaciones realizadas, debemos tener presente lo siguiente:

- a) Giro de la empresa.
- b) Volumen de operaciones.

- c) Capacidad económica.
- d) Tipo de mercado.
- e) Información deseada.
- f) Claridad en el registro y en la información.

Con la evolución y continuos cambios de las economías, los procedimientos para el registro de operaciones nos dan un ejemplo claro de tal evolución y de la adecuación de los sistemas contables a la solución de las necesidades de información de los usuarios. Los principales procedimientos que a la fecha existen son:

1. Procedimiento global de mercancías generales.
2. Procedimiento analítico o pormenorizado, llamado también como de inventarios periódicos (casi sin uso).
3. Procedimiento de inventarios perpetuos o continuos

Por otra parte, sabemos que de la confiabilidad de la información financiera depende, en mucho, de las decisiones que se puedan tomar, ya que dicha información cuantitativa se convierte en unidades monetarias.

1.4.5 Ecuación Básica de la contabilidad

Como veremos a continuación, en todos los casos en que al definir cualquiera de los conceptos que conforman la estructura financiera de la entidad (activo, pasivo y capital contable) se hizo referencia al principio de dualidad económica, que es de alguna manera de donde se derivan las formulas básicas de la contabilidad cuyos conceptos se describen enseguida.

1.4.5.1 Definición y clasificación de activo

Se ha definido al activo como la integración de todos aquellos recursos de que dispone una entidad para la realización de sus fines, los cuales representan beneficios económicos futuros esperados y controlados por una entidad económica, provenientes de transacciones o eventos realizados, identificables y cuantificables en unidades monetarias. Algunas características del activo son las siguientes:

1. El concepto se refiere a todos aquellos bienes de que dispone una entidad para la realización de sus fines.
2. Aquellas partidas que representan un derecho específico a recibir beneficios económicos futuros o servicios potenciales y que se espera generen flujos de efectivo. Por lo que debe existir un fundamento legal para disponer de esos recursos en su momento, por lo tanto todos aquellos recursos bienes o servicios cuyos derechos han expirado no se deben considerar como activo, puesto que el potencial generador de recursos o beneficios económicos futuros ha desaparecido.
3. Los activos son el resultado de acciones realizadas por una entidad, por lo que, estos recursos sólo pueden acumularse a favor de una entidad económica, ya que son recursos bajo el control de esta entidad.
4. Los recursos, derechos, bienes o servicios potenciales futuros, deben poseer el atributo de ser razonablemente medibles en términos monetarios, una vez que hayan sido identificados como tales.

El activo es clasificado principalmente en la disponibilidad de las partidas o de los recursos y en su facilidad de convertirlos en efectivo, por lo anterior el activo se clasifica en: Activo circulante y activo no circulante y algunas sub-clasificaciones que se basan en las circunstancias y necesidades de información que se presenten. Figura 1.4.5.1.1.

A continuación describiremos brevemente cada uno de los conceptos que se muestran en la figura 1.4.5.1.1. Así como en las figuras 1.4.5.1.2. y 1.4.5.1.3. se muestran algunos ejemplos de los servicios, bienes y recursos integrados en las diferentes formas de activos.

- **Activo circulante:** fundamentados en lo anterior podemos decir que el activo circulante son todas las partidas que representan efectivo o que se espera en forma razonable se conviertan en efectivo; los bienes o recursos que se espera brinden un beneficio económico esperado, ya sea por su venta, uso, consumo o servicio, normalmente en el plazo de un año o en el ciclo financiero a corto plazo, el que sea mayor.
- **Activo circulante disponible:** este es representado por el efectivo en caja, bancos, y las inversiones temporales en valores negociables de disponibilidad inmediata.
- **Activo circulante realizable:** Esta integrado por aquellos recursos que representan derechos de cobro, se pueden vender, usar o consumir generando beneficios futuros esperados en el plazo de un año del ciclo financiero a corto plazo.
- **Activo no circulante:** Son todas las partidas que se espera en forma razonable se conviertan en efectivo; los bienes o recursos que se espera brinden un beneficio económico, ya sea por su venta, uso, consumo o servicio, normalmente en el plazo mayor a un año o en el ciclo financiero a corto plazo.
- **Activo fijo tangible:** Son todos aquellos bienes tangibles que tiene por objeto el uso o usufructo en beneficio de la entidad, la producción de artículos para su venta o para el uso de la propia entidad, y a la prestación de servicios a la entidad, a su clientela o al público en general.
- **Activo intangible:** Son todas aquellas partidas que representan la utilización de servicios o el consumo de bienes pero que, en virtud de que se espera que

producirán directamente ingresos específicos en el futuro. También son las partidas cuya naturaleza es la de un bien incorpóreo, que implica un derecho o privilegio y, en algunos casos, tienen la particularidad de poder reducir costos de producción, mejorar la calidad de un producto o promover su aceptación en el mercado.

- **Otros activos:** Son todos aquellos recursos, bienes, servicios, etcétera, que por sus características no cumplen con los requisitos de los anteriores

El activo se define con la siguiente fórmula o ecuación. Esta igualdad se conoce también como fórmula del balance

$$\text{ACTIVO} = \text{PASIVO} + \text{CAPITAL CONTABLE} \quad (A = P + C)$$

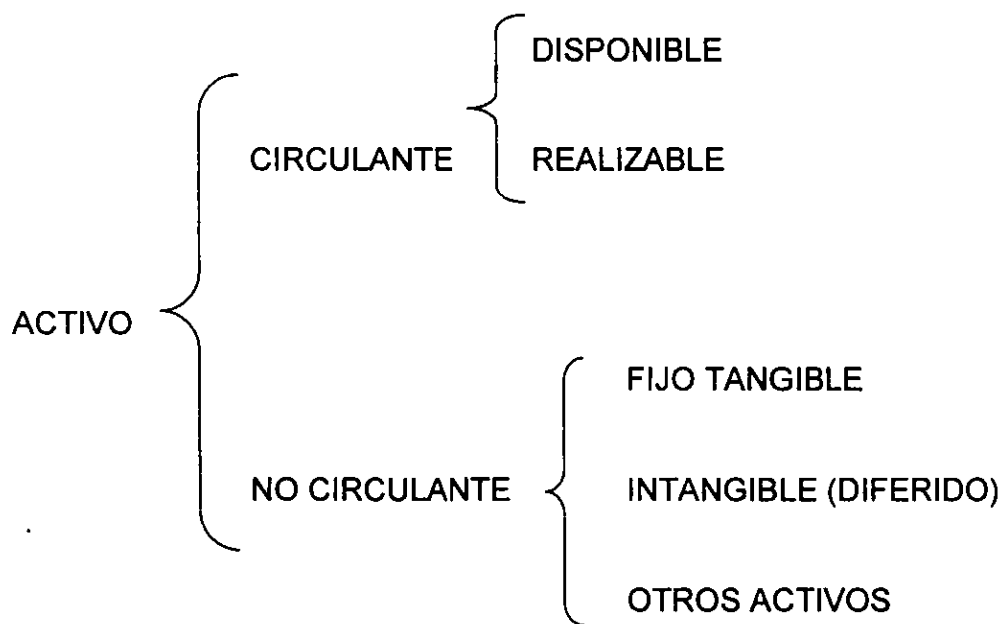


Figura 1.4.5.1.1 Clasificación del activo.

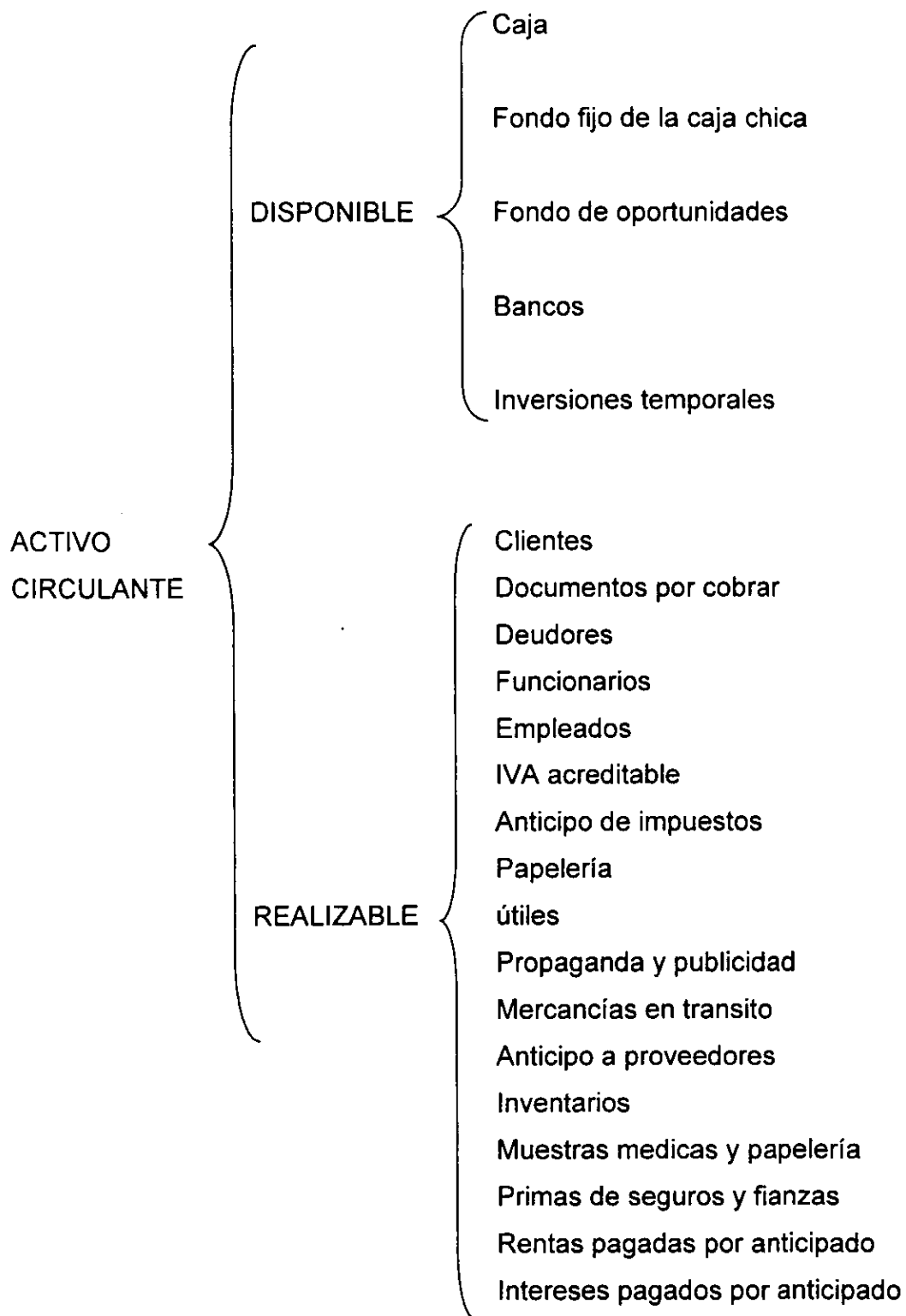


Figura 1.4.5.1.2 Clasificación del activo circulante (ejemplos)

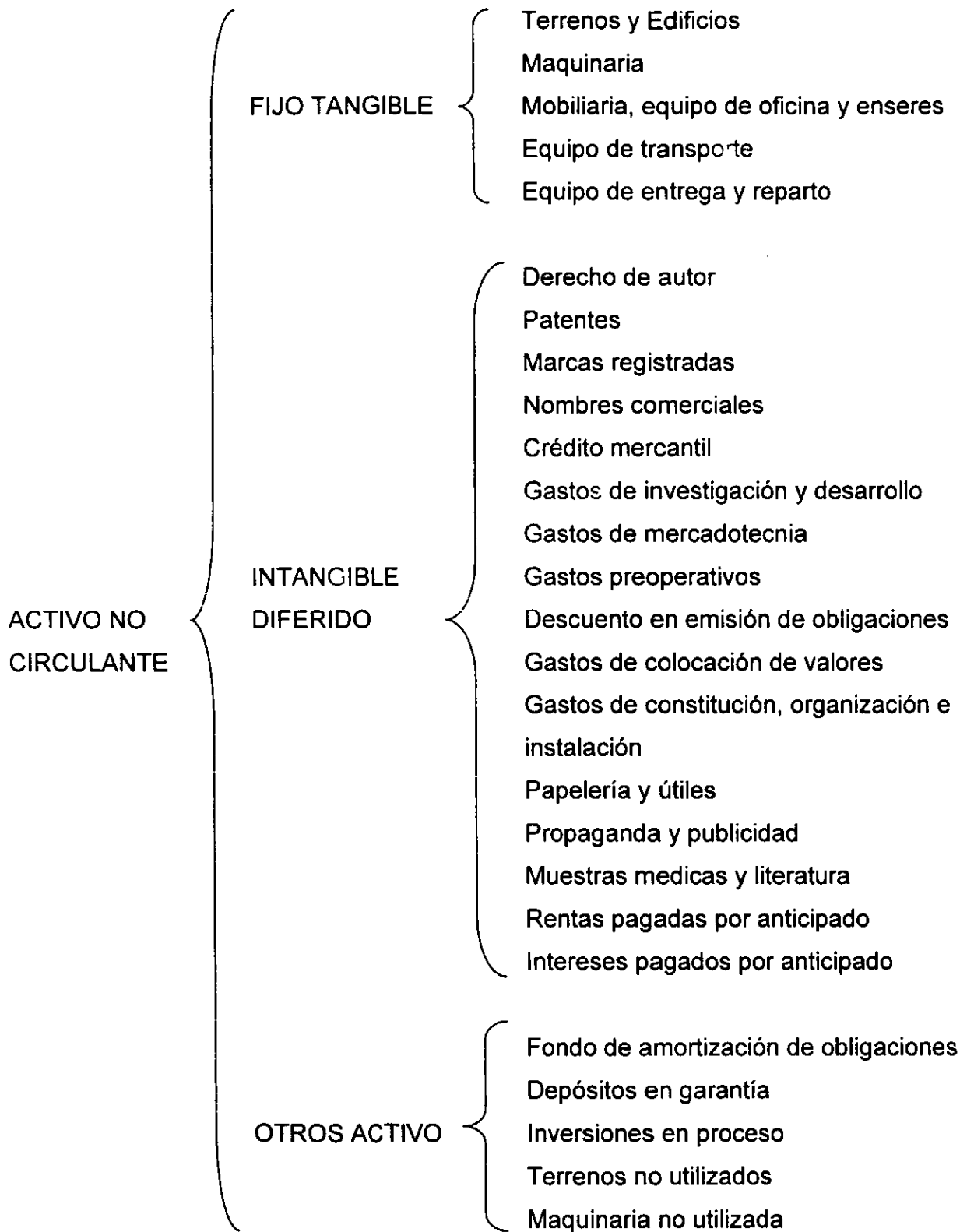


Figura 1.4.5.1.3 Clasificación de activo no circulante (ejemplos)

<u>ACTIVO</u>		
<u>CIRCULANTE</u>		
Caja	\$5.00	
Bancos	20.00	
Clientes	9.00	
Almacén	25.00	
Propaganda	4.00	
Papelería y útiles	<u>2.00</u>	\$ 65.00
<u>NO CIRCULANTE</u>		
Terrenos	100.00	
Edificios	90.00	
Equipo de reparto	30.00	
Gastos de constitución	<u>.50</u>	220.50
<u>TOTAL ACTIVO</u>		<u>\$285.50</u>
<u>PASIVO</u>		
<u>CIRCULANTE</u>		
Proveedores	\$30.00	
Acreedores	10.00	
Rentas cob x anticipado	<u>2.50</u>	\$42.50
<u>NO CIRCULANTE</u>		
Acreedores hipotecarios	\$22.50	
Obligaciones en circulación	<u>15.00</u>	\$37.50
<u>TOTAL PASIVO</u>		<u>\$80.00</u>
<u>CAPITAL CONTABLE</u>		
<u>CAPITAL CONTRIBUIDO</u>		
Capital social		\$150.00
<u>CAPITAL GANADO</u>		
Reserva legal	\$5.50	
Utilidades acumuladas	<u>50.00</u>	\$55.50
<u>TOTAL CAPITAL CONTABLE</u>		<u>\$205.50</u>

Figura 1.4.5.1.4 Estado financiero.

Y en la figura 1.4.5.1.4 se muestra un ejemplo, de como se aplica esta fórmula. Se utilizará este mismo ejemplo para los siguientes conceptos que son el pasivo y el capital

Con los datos obtenidos en el ejemplo, se sustituye en la igualdad y tenemos

$$\begin{array}{rccccr} \mathbf{A} & = & \mathbf{P} & + & \mathbf{C} \\ 285.50 & = & 80.00 & + & 205.50 \end{array}$$

1.4.5.2 Definición y clasificación del pasivo

Pasivo = Activo – Capital Contable. ($P = A - C$).

En esta igualdad conocida como formula del pasivo, se restan el total de recursos de la entidad (activo) y el total de las fuentes internas (Capital). Tomando el ejemplo de la figura 1.4.5.1.4. tenemos:

$$\begin{array}{rccccr} \mathbf{P} & = & \mathbf{A} & - & \mathbf{C} \\ 80.00 & = & 285.50 & - & 205.50 \end{array}$$

Como hemos visto en el punto anterior, los recursos de que dispone una entidad para lograr sus fines los definimos como activo. De la procedencia de algunos de esos recursos obtenemos lo que es el pasivo, que representa todos los recursos de que dispone la entidad para realizar sus fines, que han sido aportados por fuentes externas a la entidad; por las cuales surge la obligación con los acreedores de efectuar un pago, ya sea en efectivo, especie, bienes o servicios.

El pasivo comprende operaciones presentes que provienen de operaciones o transacciones pasadas, como pueden ser: la adquisición de mercancías o servicios, pérdidas o gastos en que se ha incurrido, o por la obtención de préstamos para el financiamiento de los bienes que constituyen el activo. Los principales conceptos que se comprenden en el pasivo son los siguientes:

- a) Adeudos por la adquisición de bienes y servicios, inherentes a las operaciones principales de la empresa, como son obligaciones derivadas de la compra de mercancías para su venta o procesamiento.
- b) Cobros anticipados a cuenta de futuras ventas de mercancías o prestaciones de servicios.
- c) Adeudos provenientes de la adquisición de bienes o servicios para consumo o beneficio de la misma empresa.
- d) Pasivos provenientes de obligaciones contractuales o impositivas como son sueldos, comisiones, impuestos, etc.
- e) Préstamos de instituciones de crédito, de particulares, filiales, accionista, etc. Que pueden ser préstamos obtenidos por contrato de crédito.
- f) Ciertos casos de obligaciones provenientes de operaciones bancarias de descuento de documentos.

De igual manera que el activo el pasivo esta clasificado en pasivo circulante y en pasivo no circulante y esto es de acuerdo a su fecha de liquidación o vencimiento En la figura 1.4.5.2.1 se muestra la clasificación del pasivo.

Pasivo circulante o a corto plazo. Este esta integrado por aquellas deudas y obligaciones a cargo de una entidad económica, cuyo vencimiento es en el ciclo de un año o del ciclo financiero a corto plazo, el que sea mayor.

Pasivo no circulante o a largo plazo. Este esta integrado por todas aquellas deudas u obligaciones a cargo de una entidad económica, cuyo vencimiento es un plazo mayor de un año o del ciclo financiero a corto plazo.

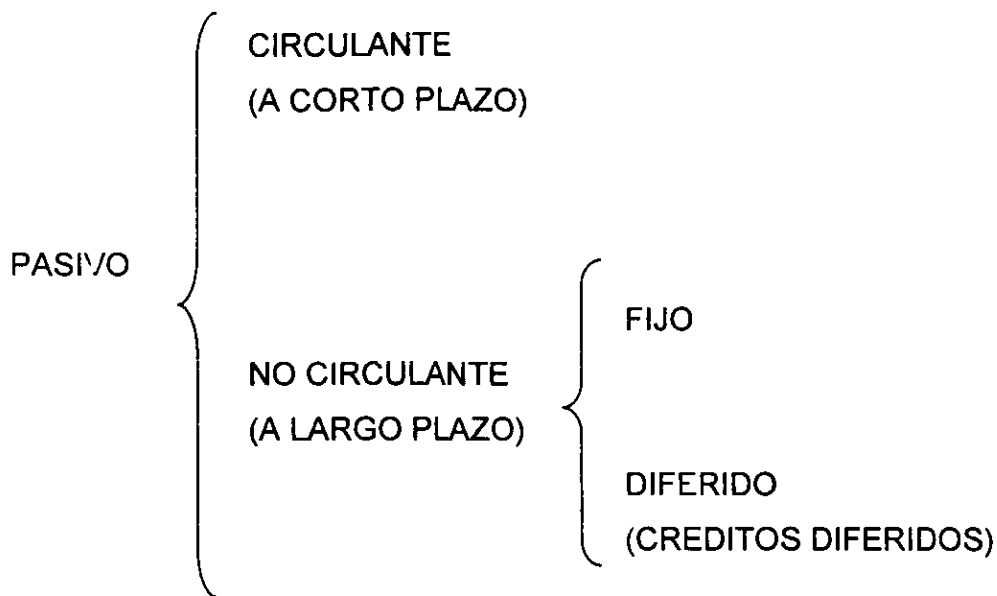


Figura 1.4.5.2.1 Clasificación de pasivo

A sí mismo como se mostró en la figura 1.4.5.2.1 el pasivo no circulante se subdivide en dos grupos que son, el pasivo diferido y el pasivo fijo, que a continuación se definen brevemente.

- **Pasivo fijo.** Este incluye todas aquellas deudas y obligaciones a cargo de la entidad, pagaderas en un plazo mayor de un año o del ciclo financiero a corto plazo.
- **Pasivo diferido o créditos diferidos.** Este está integrado por los cobros anticipados, mediante los cuales se adquiere la obligación de dar o proporcionar algún servicio en beneficio de terceras personas en un plazo mayor a un año o del ciclo financiero a corto plazo.

En las siguientes figuras se muestran ejemplos de los servicios, obligaciones y recursos que comprenden tanto el pasivo circulante como el no circulante, con el propósito de tener una idea más clara de lo que representan estos conceptos en un estado financiero.

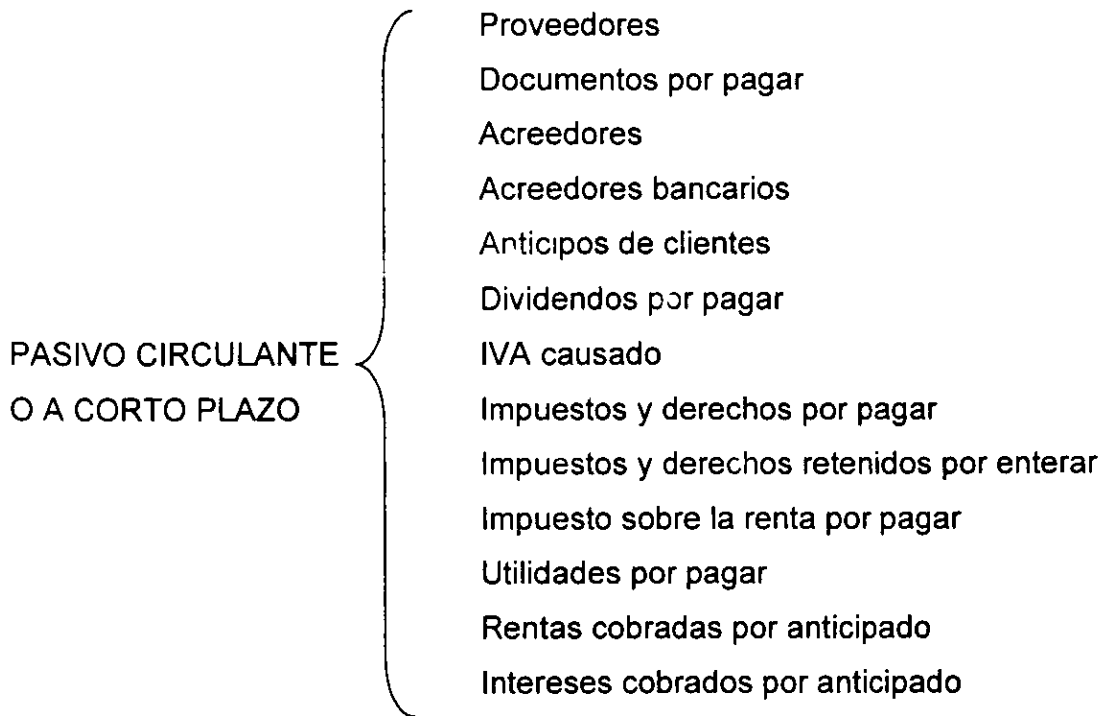


Figura 1.4.5.2.2 Clasificación del pasivo circulante o a corto plazo

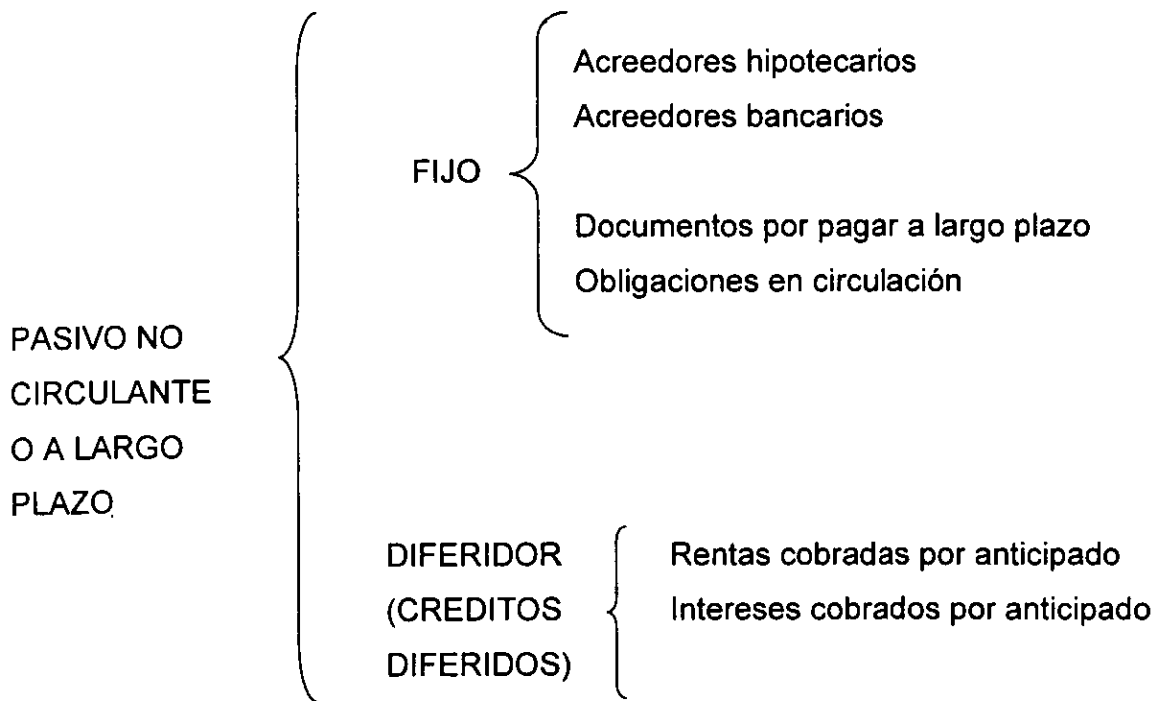


Figura 1.4.5.2.3 Clasificación de pasivo no circulante (ejemplos)

1.4.5.3 Definición de capital contable

Capital contable = Activo – Pasivo. ($C = A - P$).

En esta formula se resta la totalidad de los recursos de que dispone la entidad (activo) al importe de las aportaciones hechas por las fuentes externas. Y nuevamente basándonos en la figura 1.4.5.1.4. queda como sigue.

P	=	A	-	C
80.00	=	285.50	-	205.50

Podemos definir al capital contable como los recursos de que dispone la entidad para la realización de sus fines, los cuales han sido aportados por fuentes internas representadas por los propietarios o dueños, y los provenientes de las operaciones realizadas y otros eventos económicos y circunstancias que lo afecten. Los propietarios adquieren un derecho sobre los activos netos, el cual se ejerce mediante reembolso o distribución.

De acuerdo al origen el capital contable se clasifica de la siguiente manera, en capital contribuido y en capital ganado, se definen como sigue, y se mencionan algunas características.

- **Capital contribuido.** Esta formado por las aportaciones del dueño o los dueños y las donaciones recibidas por la entidad, así como también el ajuste a estas partidas por la repercusión de los cambios de precios.
- **Capital ganado (déficit).** Esta conformado por el resultado de las actividades operativas de la entidad y de otros eventos o circunstancias que le afecten. Igualmente el ajuste a que por la repercusión de los cambios de precios se tengan que hacer a este concepto.

En la figura 1.4.5.3.1 se muestra la clasificación del capital contable así como los conceptos que forman parte de él.

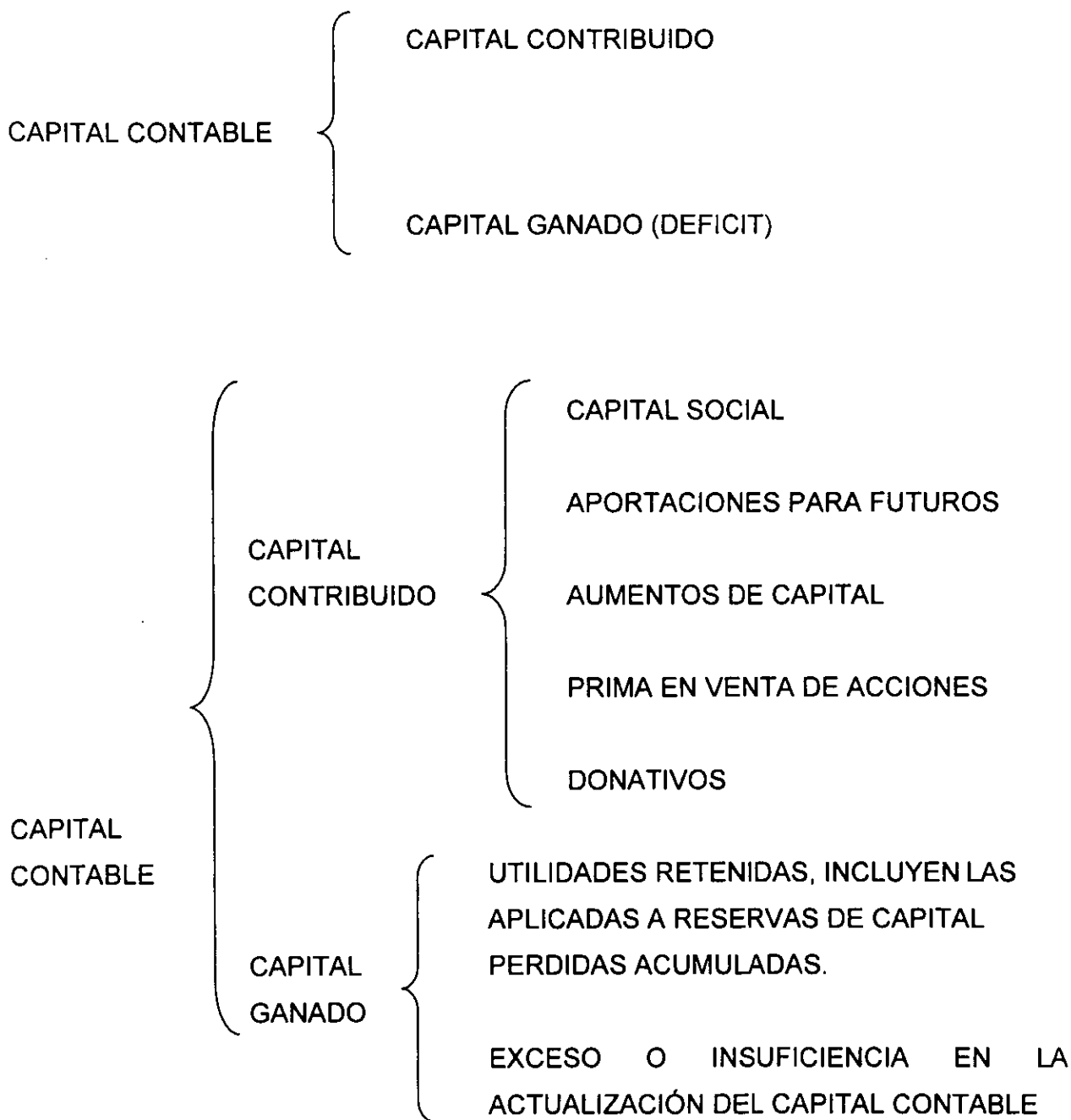


Figura 1.4.5.3.1. Clasificación del capital contable

1.4.6 Estados financieros

1.4.6.1 Objetivo y utilidad

El objetivo de los estados financieros, es que deben proporcionar información básicamente sobre:

- a) La situación financiera de la entidad en determinada fecha.
- b) Los resultados de sus operaciones en un periodo.
- c) Los cambios en la situación financiera por el periodo contable terminado en dicha fecha.
- d) Los cambios en la inversión de los propietarios durante el periodo.

Dado que los estados financieros son un medio para transmitir información financiera su utilidad consiste en que deben procurar, o ser capaces de emitir información satisfactoria para todos aquellos que lo requieran y que a su vez tengan los conocimientos técnicos adecuados para poder dar una opinión y emitir un juicio sobre:

- a) El nivel de rentabilidad. Para poder tomar decisiones de inversión y crédito, con el propósito de aportar financiamiento o crédito a la empresa. Por lo que se interesa medir la capacidad de crecimiento y redituabilidad para asegurar la recuperación de la inversión y de un rendimiento.
- b) La posición financiera. Esto es para poder medir la solvencia y liquidez de la empresa, así como su capacidad generadora de recursos.
- c) Capacidad financiera de crecimiento. Se evalúa el origen y las características de los recursos financieros, así como el rendimiento de los mismos.
- d) Flujo de fondos. Es una manera de evaluar la función de la administración por medio de una evaluación global de la forma en que se maneja la rentabilidad, solvencia y capacidad de crecimiento de la empresa.

- e) Habilidad para obtener utilidades. Implica el uso óptimo de recursos para obtener ingreso derivado de las operaciones realizadas por la entidad.
- f) Cambio de capital. Se refiere a los movimientos que se generan en la inversión de los propietarios.

1.4.6.2 Balance general

Como parte de los estados financieros tenemos al balance general, el cual nos muestra los recursos de que dispone una entidad para la realización de sus fines (activo) y las fuentes externas e internas de dichos recursos pasivo y capital contable, a una fecha determinada. De su análisis e interpretación podemos conocer la situación financiera y económica, la liquidez y rentabilidad de una entidad.

La elaboración del balance general depende tanto de la forma de presentación como del número de cuentas o grupo de cuentas que formen la contabilidad de una entidad, lo importante es que la información se presente de una manera objetiva, clara y comprensible para hacer un uso adecuado de la información contenido en dicho balance.

Dado que el balance se prepara en una fecha determinada se menciona que este es una fotografía del estado de la empresa y que dicha información puede cambiar drásticamente de un día para otro sin que siquiera se hayan realizado o celebrado otras transacciones, esto se debe principalmente a las variables económicas que en ese momento se modifiquen y que afecten directamente a los valores del activo, pasivo o capital con lo que sufrirían modificaciones.

A continuación, y con fines ilustrativos, en las figuras siguientes se mostraran algunos ejemplos esquematizados de la presentación de un balance general.

EL PORVENIR. S.A.
BALANCE GENERAL AL 19 DE FEBRERO DE 19X_

ACTIVO			
A CORTO PLAZO			
Caja		10 000	
Bancos		200 000	
Clientes		150 000	
Almacén		340 000	
Primas de seguro		<u>6 000</u>	706 000
A LARGO PLAZO			
Edificios		400 000	
Equipo de reparto		<u>80 000</u>	480 000
DIFERIDO			
Gastos de constitución		5 000	
Primas de seguro		<u>6 000</u>	<u>11 000</u>
TOTAL ACTIVO			<u>1 197 000</u>
PASIVO			
A CORTO PLAZO			
Proveedores		320 000	
ISR por pagar		40 000	
PTU por pagar		<u>10 000</u>	<u>370 000</u>
TOTAL PASIVO			<u>370 000</u>
CAPITAL CONTABLE			
CAPITAL CONTRIBUIDO			
Capital social	500 000		
Donaciones	<u>70 000</u>	570 000	
CAPITAL GANADO			
Utilidad neta del ejercicio	50 000		
Utilidades acumuladas	200 000		
Reserva legal	<u>7 000</u>	<u>257 000</u>	<u>827 000</u>
TOTAL PASIVO + CAPITAL CONTABLE			<u>1 197 000</u>

LAS NOTAS QUE SE ACOMPAÑAN SON PARTE INTEGRANTE DE ESTE ESTADO FINANCIERO

AUTORIZADO POR

ELABORADO POR

SR. ROBERTO AGUILAR AGUILAR
GERENTE GENERAL

C.P. VIRGINIA LOPEZ REYES
CONTADOR GENERAL

1.5 SISTEMAS OPERATIVOS

1.5.1 Unix

Introducción a Unix

Unix es un sistema operativo desarrollado por Bell Telephone Laboratories de la American Telephone and Telegraph. Históricamente, UNIX evolucionó a partir de un proyecto pequeño que Thompson Ken inició en Bell Labs. a finales de los 60's. UNIX tiene algunas características cualitativamente distintas de los sistemas operativos desarrollados por los fabricantes de computadoras que han tenido gran aceptación tanto en el mundo académico como en el industrial. Dicho sistema operativo puede ser usado tanto en microcomputadoras como en equipos grandes, proporciona un entorno compartido, eficaz y de fácil uso para el desarrollo de programas de aplicaciones y de sistemas en lenguajes de alto nivel como Fortran 77, Pascal, Basic, Cobol, etc. Así como proporcionar facilidades en cuanto al uso de editores que normalmente no son parte de otros sistemas operativos y que es muy útil.

Definición de Sistema operativo

Es un programa que maneja todos los recursos de la computadora, incluyendo la unidad de control de proceso, memoria, dispositivos de almacenamiento externo como discos y cintas, también controla el uso de periféricos tales como son terminales e impresoras.

Diseño de Unix

“Un ambiente de trabajo para Desarrollo de Programas”

“Un Sistema Operativo diseñado por Programadores para Programar” Ken Thompson

Metas de Diseño

- Portabilidad a través de diferentes plataformas de hardware
- Manejo de los dispositivos como archivos, sin importar modelo y tipo
- Combinación de programas existentes para llevar a cabo tareas complejas
- Diseñado para ser un sistema multiusuario y multitarea

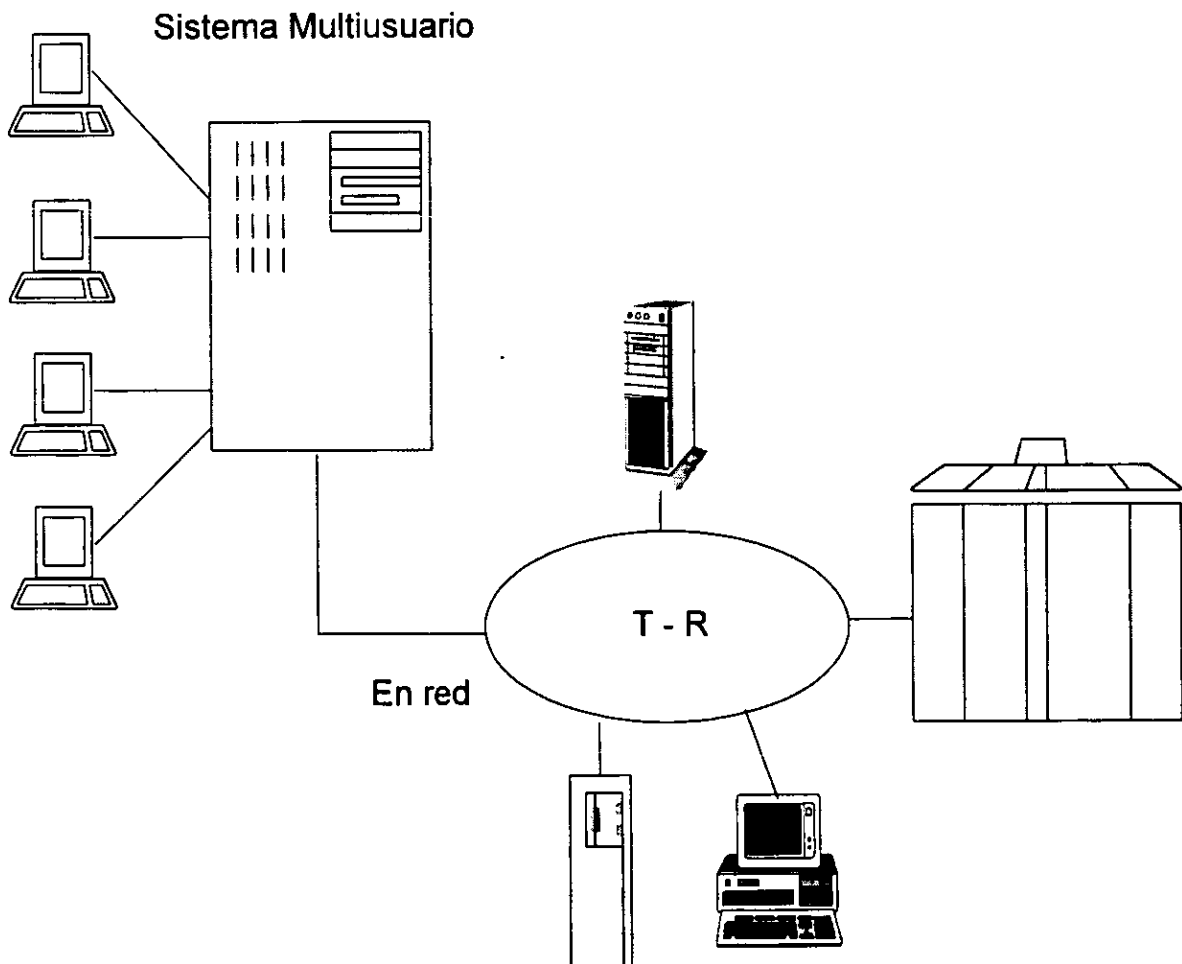


Figura 1.5.1.1 Sistema multiusuario

Principales componentes de Unix

- Kernel
- Shell
- Sistema de Archivos
- Procesamiento de texto
- Herramientas de programación
- Administración del sistema
- Documentación en línea

El Kernel

El Kernel es el nombre dado al corazón central del sistema operativo, maneja múltiples procesos los cuales pueden correr en el sistema en cualquier momento, controla a muchos usuarios y la seguridad del sistema. Controla los sistemas de entrada y salida de la máquina, por ejemplo, terminales e impresoras. Maneja los discos conectados al sistema y controla el acceso a los archivos.

El Kernel esta escrito en su mayoría en lenguaje C, con pequeñas cantidades de código dependiente de la máquina desarrollado en lenguaje ensamblador. Una vez que un compilador de C ha sido desarrollado para un ambiente de hardware en particular, entonces portar Unix a este ambiente involucra compilar la parte de C del Kernel y desarrollar las rutinas de ensamblador asociadas.

- Maneja múltiples procesos, usuarios y la seguridad del sistema

- Interface con dispositivos de E/S
- Administra los archivos en disco

El Shell

El Shell es el nombre que se le da a la interface básica de comunicación entre el usuario y el Kernel del sistema operativo Unix, es decir, entre el usuario y los programas que se utilizan para realizar tareas específicas. Unix trae consigo varios Shells para que el usuario seleccione con cual trabajar, algunos Shells interpretan los comandos de los usuarios e inician procesos para los requerimientos de los usuarios. Otros Shells proporcionan una interface de pantalla completa donde los usuarios pueden seleccionar comandos desde menús.

Un Shell puede definirse también como un interprete de las instrucciones entradas por pantalla. En otros sistemas operativos e incluso en equipos grandes, las instrucciones entradas por pantalla que son interpretadas por el sistema operativo son mucho más limitadas que en Unix. Por ejemplo, algunos sistemas operativos como máximo interpretan una secuencia de instrucciones cargadas en un archivo.

El Shell de Unix no es simplemente un conjunto limitado y definido de instrucciones que el sistema operativo interpreta y en algunos casos puede ser usado como lenguaje de programación, proporcionando facilidades como son variables y control de flujo, por ejemplo con estructuras de control y bucles o loops (if-then-else, case, do while, etc.) que junto con los programas de aplicación de utilidad de Unix, permite desarrollar muy rápidamente programas de aplicación bastante potentes.

El Shell es una parte esencial del sistema operativo Unix ya que permite utilizar gran parte de la potencia de UNIX para crear, manipular e imprimir archivos en general, ejecutar procesos y determinar el estado del sistema. Muchos Shells permiten al usuario iniciar procesos en segundo plano (background), así como en primer plano

(foreground). En este caso, el Shell está disponible inmediatamente para que el usuario introduzca más comandos mientras sus procesos en segundo plano siguen en ejecución. Tan pronto como el usuario se conecta al sistema, es ubicado dentro del Shell, el cual es llamado Shell de login, muchas de las veces es el mismo Shell que el sistema tiene por omisión.

- Interfacé de usuario entre usuario y Kernel
- Varios Shells disponibles:
Korn, Bourne, C
- Interprete de comandos
- Capacidades de programación
- Soporte para ejecutar procesos en primer y segundo plano

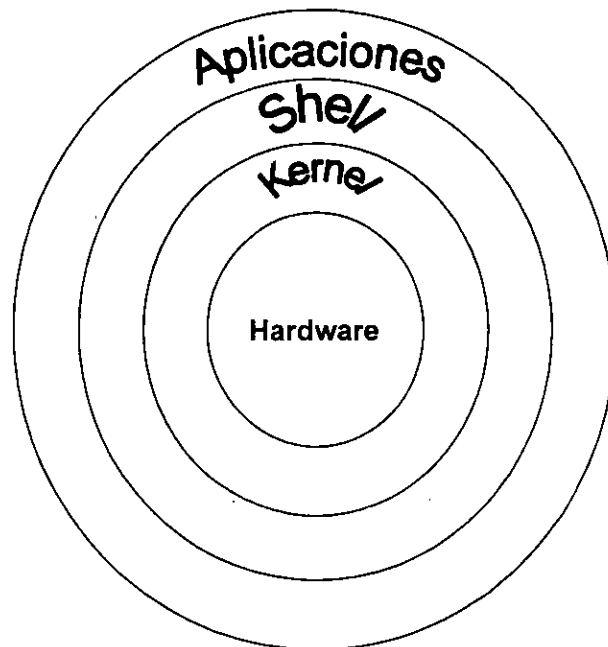


Figura 1.5.1.2 Componentes de Unix

El sistema de archivos

La estructura del sistema de archivos Unix es la forma en que son organizados estos sobre los discos del sistema. Las características más importantes del sistema de archivos son la estructura jerárquica en árbol de los directorios y archivos, y la facilidad de encontrar automáticamente cualquier archivo con una sola instrucción de búsqueda.

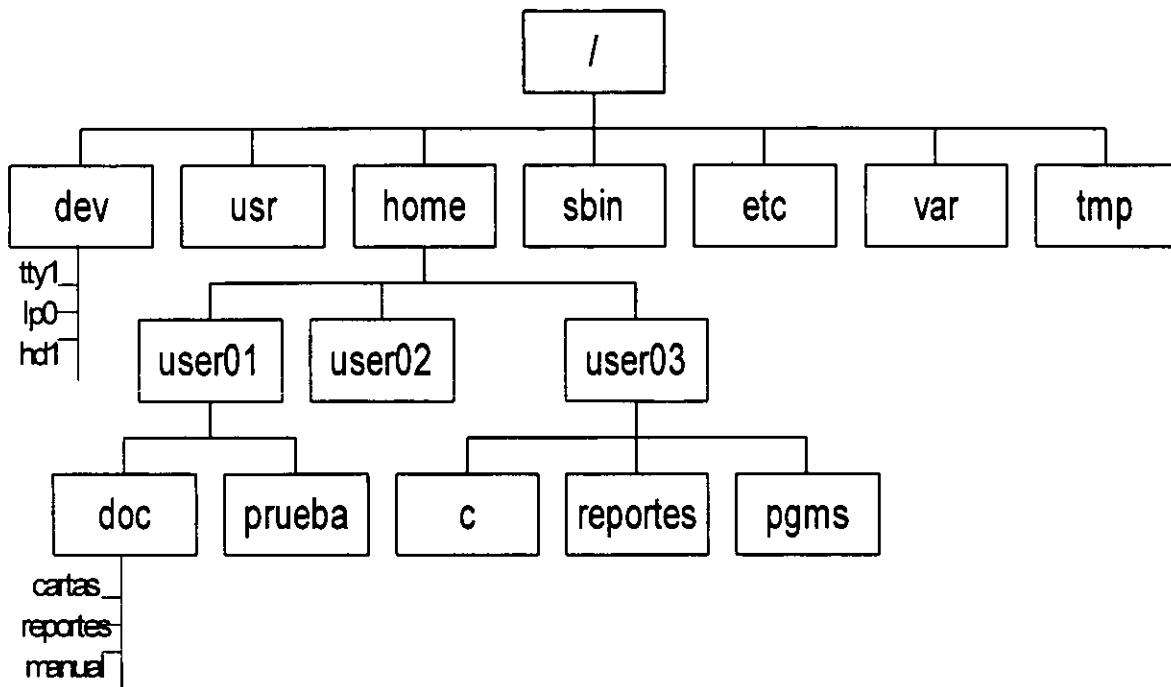


Figura 1.5.1.3 Sistema de Archivos *

Existen tres tipos principales de archivos:

- Archivos ordinarios normalmente conteniendo texto
- Directorios conteniendo una lista de apuntadores a otros archivos. Su propósito principal es permitir al usuario percibir un nivel de organización impuesto a los archivos sobre el sistema. Por concepto un directorio es un archivo que contiene información sobre otros archivos y subdirectorios.

El directorio root (raíz) es el origen de todos los demás y se denota con el símbolo “/”.

El árbol de Unix es una estructura abierta, es decir, el usuario puede crear y eliminar directorios del sistema operativo con toda flexibilidad y libertad, aunque por seguridad no puede eliminar aquellos que no hayan sido creados por él mismo. Dos o más archivos pueden tener el mismo nombre pero su path (trayecto) es diferente, es decir, se encuentran en distintos directorios.

- Archivos especiales son normalmente apuntadores a dispositivos unidos al sistema. Por ejemplo, el enviar datos a un archivo especial llamado /dev/lp0, realmente implica que dichos datos serán impresos en alguna de las impresoras del sistema.

Procesamiento de Texto

Existen editores usados para crear y modificar archivos de texto ASCII. Estos editores no proporcionan las mismas funciones de formato como lo haría un procesador de palabras, pero tienen facilidades para ubicar y modificar texto.

- Editores
- Procesamiento de documentos
- Verificación de Ortografía
- Impresión

Estos son algunos editores de Unix:

“Ined” y “vi” son editores de pantalla completa para tareas grandes de edición

“ed” es un editor de línea para tareas pequeñas

“nroff” y “troff” son herramientas para dar formato a textos, incluyen creación de documentos, herramientas para dar formato, diccionario para verificación de ortografía y facilidades para dar formato e imprimir la salida a una variedad de impresoras.

Herramientas de programación

El sistema operativo Unix, tiene varias herramientas para la asistencia en el desarrollo, control y mantenimiento de programas, a continuación se mencionan algunas.

- Los editores del sistema pueden ser utilizados para crear código fuente.
- El Source Code Control System SCCS (Sistema de Control de Código Fuente), es una ayuda muy utilizada para grandes proyectos de desarrollo. SCCS permite mantener versiones separadas de programas sin almacenar copias completas por separado de cada versión. Su uso puede reducir requerimientos de almacenamiento y puede ayudar en el rastreo del desarrollo de un proyecto que requiere guardar muchas versiones de un gran número de archivos.
- El depurador (dbx) permite al programador monitorear un programa en ejecución y analizarlo para encontrar errores.
- El verificador de código (lint) y otras herramientas, corrigen sintaxis de tal forma que hacen los programas más fáciles para leer y mantener.
- El Xwindows Development Environment XDE (Ambiente Gráfico) herramienta gráfica para monitorear y manipular programas durante el desarrollo.
- Editores

- SCCS- Sistema de control de Código Fuente
- Depuradores
- Verificadores de código

Administración del sistema

Hay muchas tareas sobre el sistema que requieren ser administradas, usualmente involucran desarrollar procedimientos muy complejos, para simplificar estos procesos Unix proporciona herramientas, como por ejemplo SMITH, la cual es una interface de comandos orientada a pantalla completa e interactiva que pide al usuario la información necesaria para construir comandos y le presenta elecciones apropiadas o valores por omisión disponibles, también ayuda al usuario haciéndole más fácil la administración del sistema y evitándole que cometa errores, al automatizar tareas que se hacen con regularidad.

El usuario "root" o cualquier otro usuario con suficientes permisos (llamado también super usuario), puede desarrollar tareas sobre el sistema que están restringidas para los usuarios normales, Algunas de esas tareas se enlistan enseguida;

- Instalación de software
- Definición de cuentas de usuario
- Configuración de dispositivos periféricos
- Almacenamiento en disco
- RespalDOS de información

- **Monitoreo del rendimiento del sistema**
- **Determinación de problemas del sistema**

Documentación en línea

Muchos sistemas Unix proporcionan su documentación en línea en la forma de páginas de manual, esta muestra los comandos y llamadas al sistema, así como dispositivos, archivos y procedimientos de administración del sistema. Existen facilidades limitadas para la búsqueda de una palabra clave (keyword) en particular, la cual indica en que páginas del manual puede ser encontrada dicha keyword.

La base de datos InfoExplorer es un sistema de documentación en línea, se presenta como una interface interactiva con menús de los cuales se pueden realizar búsquedas y controlar la información presentada. Los datos son mostrados en dos ventanas (una de navegación y otra de lectura de información). En modo gráfico (X-windows) son posibles diferentes vistas de la información utilizando las facilidades de búsqueda para seleccionar la información deseada, seleccionando por título de un libro, por tarea o por nombre de comando.

- **Páginas de manual**
 - Comandos y llamadas al sistema**
 - Una página a la vez**
 - Búsqueda de palabras clave**
- **InfoExplorer**
 - Manuales en línea**
 - Interface de pantalla completa interactiva**
 - Texto o gráfico (X-Windows)**
 - Facilidades de búsqueda (libro, tarea, comando)**

Sistemas Operativos Comerciales Unix:

- Santa Cruz Operation SCO
- Hewlett Packard UX
- Digital Equipment Corporation Ultrix
- Sun Microsystems Solaris
- IBM AIX
- Digital VMS
- AT&T System V
- Bull Bos
- Microsoft Xenix

El sistema operativo Unix ha encontrado amplia aceptación en el mundo comercial, debido a que es un sistema abierto que hace posible su portabilidad entre diversas computadoras. Unix puede observarse en una amplia gama de computadoras, desde Mainframes hasta PC's.

El sistema operativo puede utilizarse en diferentes tipos de computadoras, está escrito de manera que es sencillo de modificar, ofreciendo en consecuencia la posibilidad de traslado de una marca de computadora a otra. Dicha característica puede ahorrar tiempo y dinero ya que al conmutar los sistemas, los usuarios no necesitan aprender a utilizar el sistema nuevo porque la única diferencia es el hardware y podrán seguir usando los mismos programas a los que están acostumbrados.

Debido a que existen diversas marcas de productos en el mercado enfocados a trabajar bajo ambiente operativo Unix con diferentes equipos en hardware y software, se analizarán los requerimientos de hardware para el sistema operativo Unix The Santa Cruz Operation (SCO), que se ha convertido rápidamente en un estándar alternativo para la PC y algunos expertos creen que la firma puede llegar a ser el líder a seguir en el terreno de este sistema operativo

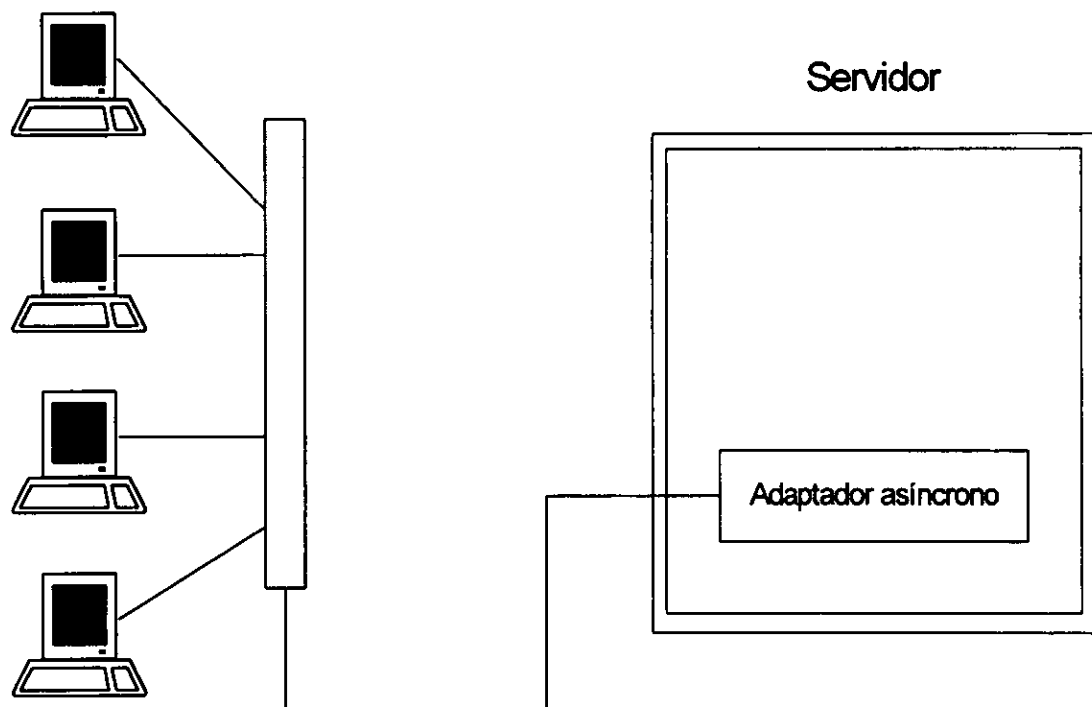
Requerimientos de hardware para instalar Unix SCO

Para poder trabajar en una red local bajo un ambiente de Unix SCO es necesario contar con un servidor Pentium 333 Mhz., hasta equipos altamente sofisticados como Mainframes o Minis H.P.9000. Al ser un sistema multiprocesos y multitareas proporciona la facilidad de operarlo según sean las necesidades de uso de la aplicación. Si son aplicaciones que requieren de gran interconectividad y manejo de grandes bases de datos a una velocidad muy dinámica y con una plataforma de red inmensa es necesario emplear Minis con terminales o PC`s. En caso de que se utilicen aplicaciones de menor requerimiento de usuarios e interconectividad sofisticada basta con una PC como servidor y periféricos adicionales de acuerdo al tipo de plataforma que se requiera, también podrían ser terminales y PC`s con diferentes características.

Características del servidor

Se requiere de un servidor con capacidad de crecimiento, es decir que a parte de que cuente con un procesador Pentium hacia arriba y con velocidades de 333 Mhz. hacia arriba, tenga el número de ranuras (slots) de expansión disponibles para los dispositivos que se requieran como podrían ser: controladores de discos duros, tarjetas de red, tarjetas controladores para concentradores, tarjetas de video, etc. Dicho servidor también deberá contar con un mínimo de 64 MB de memoria RAM, puertos seriales y paralelos y dispositivos para obtener el mejor rendimiento y confiabilidad en la red.

- Capacidad de crecimiento
- Procesador Pentium; se recomienda superior
- Velocidad mínima de 333 Mhz; se recomienda superior
- Suficientes ranuras de expansión (slots), mínimo 10
- Puertos seriales y paralelos
- Memoria mínima de 64 Mb, se recomienda más



Terminales asíncronas

(a)

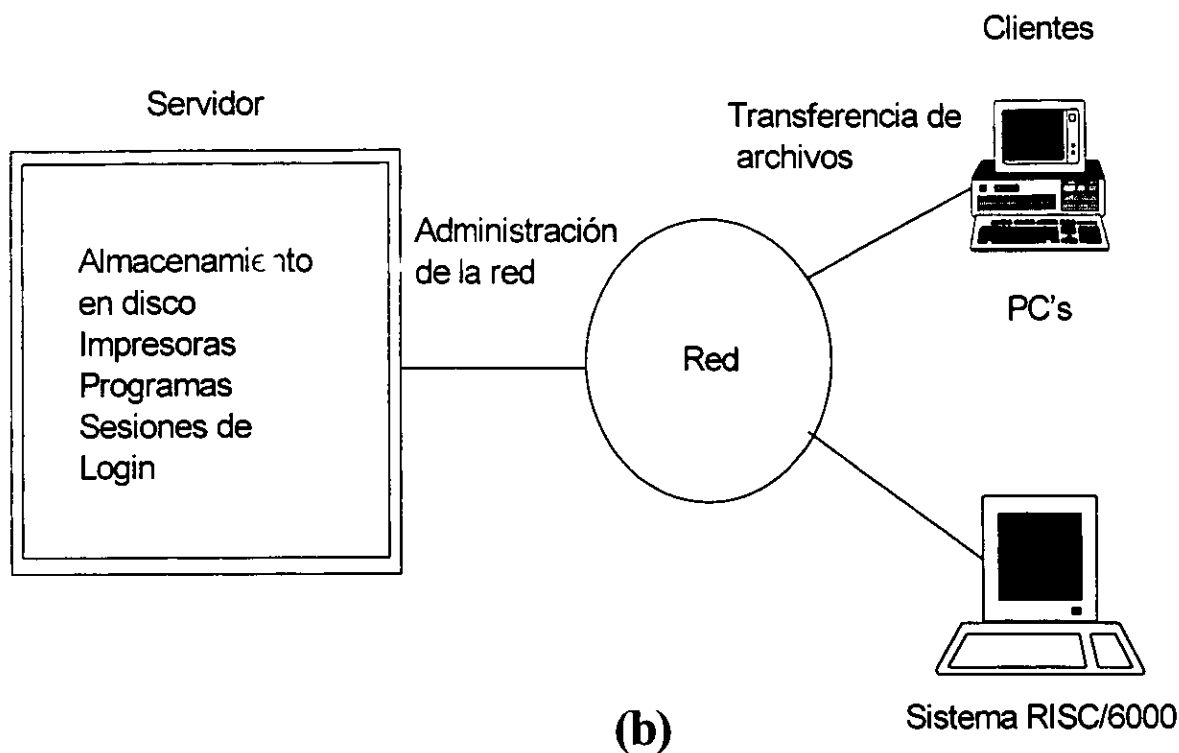


Figura 1.5.1.4 Configuraciones como servidor (a) Sistema multiusuario (b) Sistema en red

Para las estaciones de trabajo en caso de PC's:

- Procesador Pentium 266 Mhz., se recomienda superior
- 16 MB de memoria RAM, se recomienda más
- Puerto serial para comunicación asíncrona con su respectiva interface
- Sistema operativo (Windows) o software que emule algún tipo de terminal

En caso de terminales:

- Se requiere que sea configurado el tipo de terminal que se empleará, es decir si será terminal que emule gráficos o simple video, el tipo de teclado, etc.

- Realizar el cable o interface de la terminal según el tipo de concentrador, ya que hay con RJ-45, RJ-11 y DB-25 para diferentes modelos.
- Configurar la terminal en velocidad, paridad, bits de paraJa, protocolo

LINUX

LINUX es una versión gratuita de UNIX, desarrollada primeramente por Linus Torvalds en la universidad de Helsinki en Finlandia. LINUX fue desarrollado con la ayuda de muchos programadores de UNIX que navegan a través de Internet. El kernel de LINUX no usa código de AT&T u otro propietario, la gran mayoría del software disponible para LINUX fue desarrollado por proyectos de la fundación de software gratuito en Cambridge, Massachusetts. A continuación se presentan algunas características principales de LINUX

- Es un sistema multitarea y multiusuario
- Es un sistema gráfico tipo Windows 9X, por lo que ya no hay necesidad de teclear comandos.
- Tiene capacidad de instalar memoria virtual. Algunas veces se requiere utilizar más memoria RAM de la que ya se tiene, de manera física. La memoria virtual es un espacio en la unidad de disco duro que se comporta como memoria RAM extra cuando se requiere, ese espacio en LINUX puede ser de hasta 256 Megabytes. A diferencia de otros Sistemas Operativos LINUX mueve los procesos que no están en uso a un espacio conocido como "swap" que es un espacio de intercambio para liberar RAM física, con el fin de que ahí sean atendidas las tareas que se hagan en ese momento.
- Soporta consolas virtuales que permiten la conexión entre múltiples sesiones desde la consola en modo texto.

- Puede generar programas de acceso a bibliotecas de códigos que ya existen en el propio sistema. De esta forma el tamaño de los programas se mantiene reducido y se ahorra espacio en el disco duro y en la RAM.
- Tiene la capacidad de administrar su propia memoria. El kernel implementa una memoria para uso de programas y en este sentido esta memoria es utilizada en vez de la memoria caché, la cual es reducida cuando se corren programas largos.
- Provee una implantación de redes TCP/IP, puesto que lo utiliza para su conectividad en red. Incluye adaptadores de tarjetas Ethernet, SLIP (Serial Line Internet Protocol), PLIP (Parallel Line Internet Protocol), PPP (Point to Point Protocol), NFS (Network File System).
- Ostenta los estándares de UNIX IEEE POSIX.1, System V y BSD que lo hacen portable.

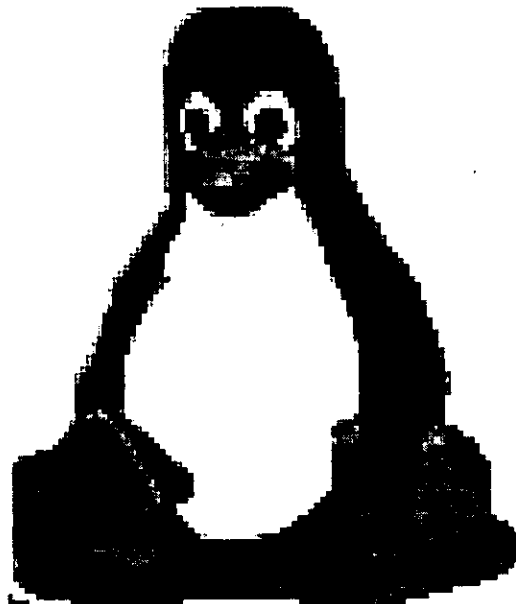


Figura 1.5.1.5 Logotipo de LINUX

1.5.2 Windows 9X/Windows NT

Introducción a Windows

Las primeras computadoras personales fueron introducidas hace poco más de 20 años, a mediados de la década de 1970. Windows ha estado presente más o menos la mitad de ese tiempo, habiéndose introducido la primera versión del producto en 1985. Desde ese momento, Windows ha desarrollado un árbol genealógico que incluye varios sistemas operativos de escritorio. Sin embargo, cabe señalar que Windows no comenzó como un sistema operativo, sino como un shell gráfico para el sistema operativo existente en ese momento, DOS.

Windows en la actualidad

Desde el punto de vista de Microsoft es una familia de productos. El objetivo que Microsoft ha estado persiguiendo es una amplia gama de sistemas operativos basados en Windows. Estos sistemas se ejecutarían en diferentes plataformas de hardware, pero compartirían los mismos fundamentos básicos.

- Elementos comunes del sistema operativo
- Interfaz de usuario común
- Plataforma de programación común

Actualmente se tienen las siguientes versiones disponibles: Windows 9X (95-98) y Windows NT (Server y Workstation). Estas versiones del sistema operativo se diferencian por el hardware donde se ejecutan. No existe una división firme respecto a cuál sistema operativo se ejecuta en cuál hardware. Windows 9X puede ejecutarse en el mismo hardware que Windows NT (no necesariamente a la inversa) y el factor determinante son las necesidades del usuario.

El basarse en una interfaz común tiene grandes implicaciones para empresas medianas a grandes. Básicamente significa que sólo se tiene que entrenar a los empleados una vez y sabrán como usar cualquier computadora personal con Windows, desde la portátil mas chica hasta el servidor más grande, aunque hay matices basados en la versión de Windows que se este usando y las tareas que se estén realizando, pero la interfaz de usuario es la misma, significando que la curva de aprendizaje disminuye en forma considerable.

Principales características de Windows 9X

Interfaz de usuario fácil de usar

Windows incorpora una interfaz de usuario que simplifica y aumenta la eficiencia en la utilización de la computadora personal por parte de los usuarios.

Pulsando un solo botón (botón de inicio), los usuarios tienen acceso a casi todo lo que necesiten, desde ejecutar programas y documentos, hasta cambiar los parámetros de la computadora y conseguir ayuda. Este botón proporciona un punto de partida a los usuarios para las funciones que realizan diariamente.

El escritorio simplifica la ejecución de tareas y la búsqueda de conexiones a recursos. El icono de Entorno de Red proporciona un punto central donde se muestran los servidores de red y desde donde se pueden establecer conexiones con otros equipos o servidor y utilizar sus recursos.

- Interfaz de fácil uso
- Botón de inicio
- Escritorio

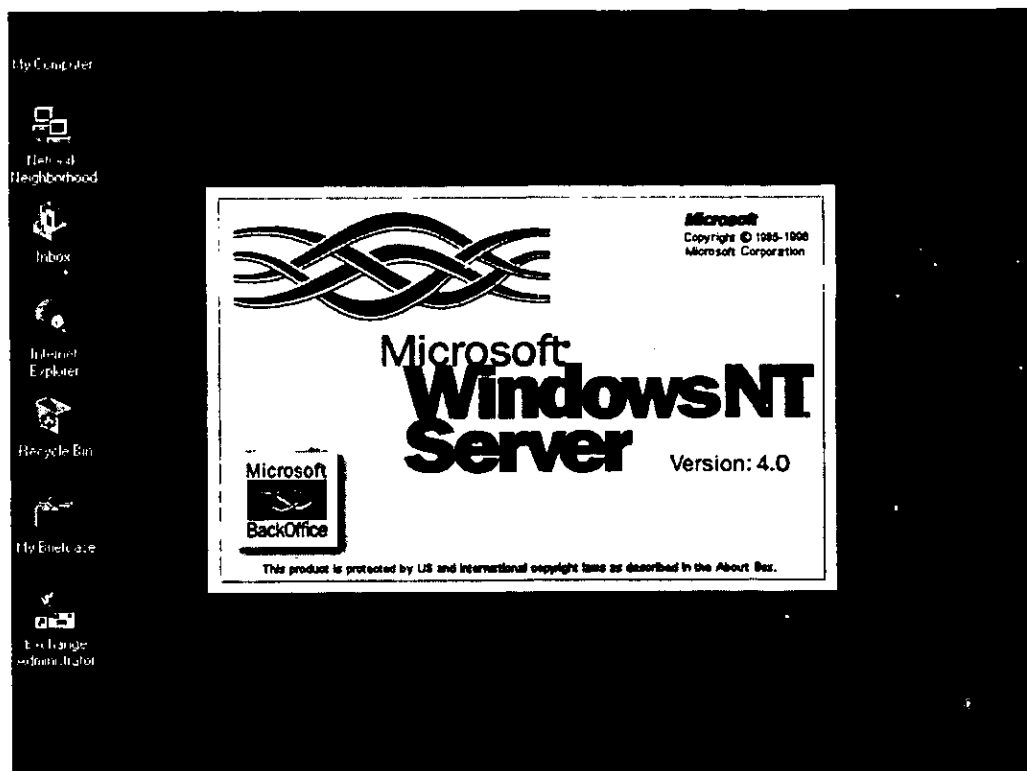


Figura 1.5.2.1 Pantalla principal de Windows NT Server

Tecnología Plug & Play

La tecnología Plug & Play (conecta y listo) proporciona un medio lógico y consistente para la autoidentificación de dispositivos y para configurar sus parámetros en la computadora. Con conecta y listo Windows simplifica la instalación, edición y eliminación de hardware y software.

Esta tecnología permite que muchos de los dispositivos de hardware se autoconfiguren de manera automática en la computadora, eligen sus propios parámetros, interrupciones o canales, sin necesidad de que el usuario los especifique.

También se simplifica la instalación y la configuración de dispositivos que no son compatibles con conecta y listo. El asistente para agregar nuevo hardware detecta los dispositivos legados y mueve sus parámetros de hardware al registro de

configuraciones durante la instalación, asegurando que la configuración del dispositivo legado no se sobrescriban cuando se agreguen los dispositivos conecta y listo. Los usuarios pueden añadir o quitar dispositivos mientras la computadora está en funcionamiento. La computadora detecta automáticamente que el estado del hardware ha cambiado y ajusta los parámetros del sistema de acuerdo con el nuevo estado, además de todo lo anterior el sistema operativo cuenta con los drivers de casi todos los dispositivos y se pueden utilizar estos o los proporcionados con el hardware que se instalara.

- Hardware conecta y listo
- Simplificación de instalación de dispositivos legados (no compatibles)

Arquitectura de sistema operativo de 32 bits

La arquitectura de 32 bits y el manejo de recursos, reduce el tiempo improductivo y las llamadas de soporte, proporcionando un entorno de sistema operativo más estable, dando mayor fiabilidad y rendimiento del sistema.

Las aplicaciones de 32 bits se ejecutan en su propio espacio de direcciones por lo que el cerrar un proceso errático no afecta al resto de los programas (modo protegido). Los recursos del sistema expandidos permiten al usuario ejecutar varias aplicaciones sin que se produzca un error por falta de memoria, una vez que una aplicación con error finaliza, Windows realiza una labor de seguimiento de los recursos de los programas y de limpieza. La liberación de los recursos del sistema una vez que una aplicación se ha cerrado, significa que el rendimiento del sistema es menos propenso a degradarse en el tiempo.

- Subsistemas de 32 bits en modo protegido
- Recursos del sistema expandidos

- Limpieza de aplicaciones

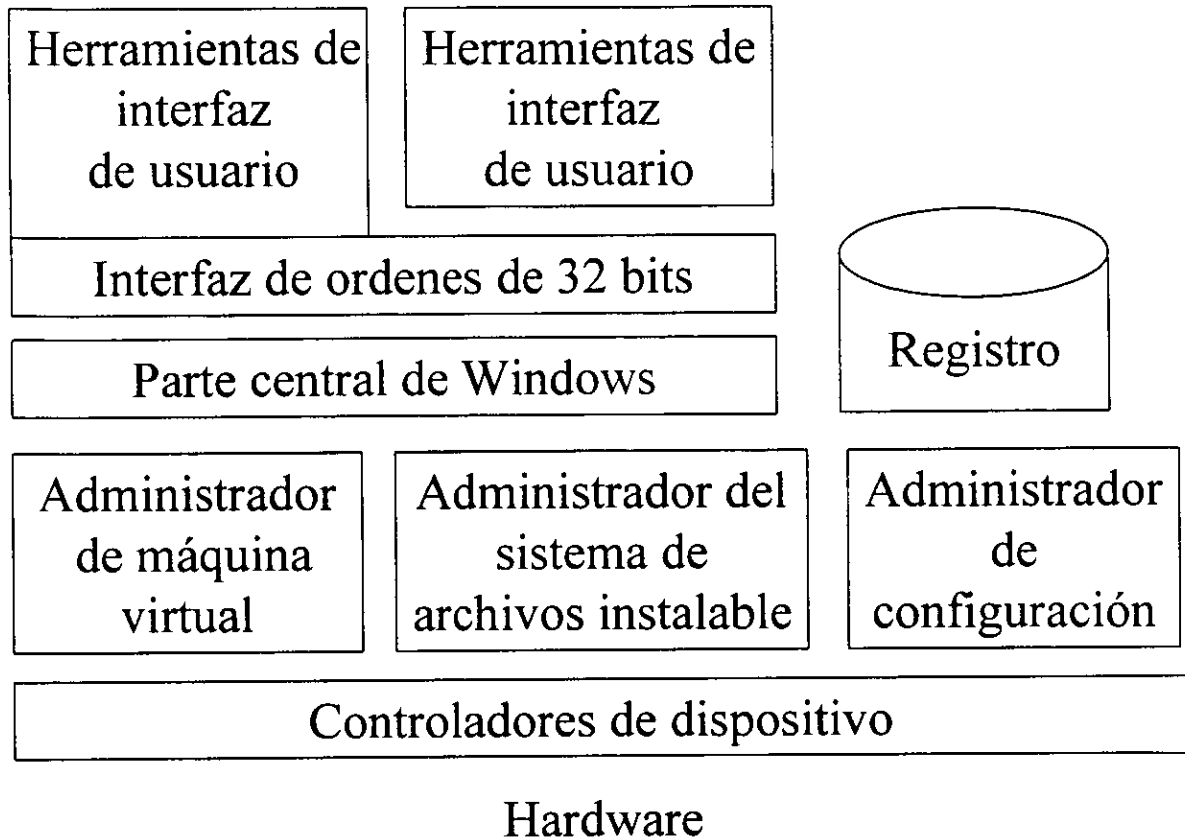


Figura 1.5.2.2 Subsistemas de Windows

Red incorporada

Windows utiliza componentes de red de 32 bits que permiten trabajar sin problemas con las principales redes. El soporte para componentes de 32 bits incluye, el protocolo, el adaptador de red y los servicios de compartición de archivos e impresoras. Estos componentes fueron diseñados para ejecutarse en entornos multitarea, no utilizan memoria en modo real y ofrecen un funcionamiento en red rápido y estable. Windows incluye las versiones de 32 bits del protocolo compatible con IPX/SPX y el protocolo TCP/IP. También puede trabajar con diversos protocolos y clientes de red de 16 bits, por lo que la conectividad es relativamente sencilla y casi con cualquier equipo conectado a la red.

Con el entorno de red, los usuarios pueden explorar los servidores de red de manera rápida y sencilla.

- Componentes de red de 32 bits
IPX/SPX
TCP/IP
- Entorno de red

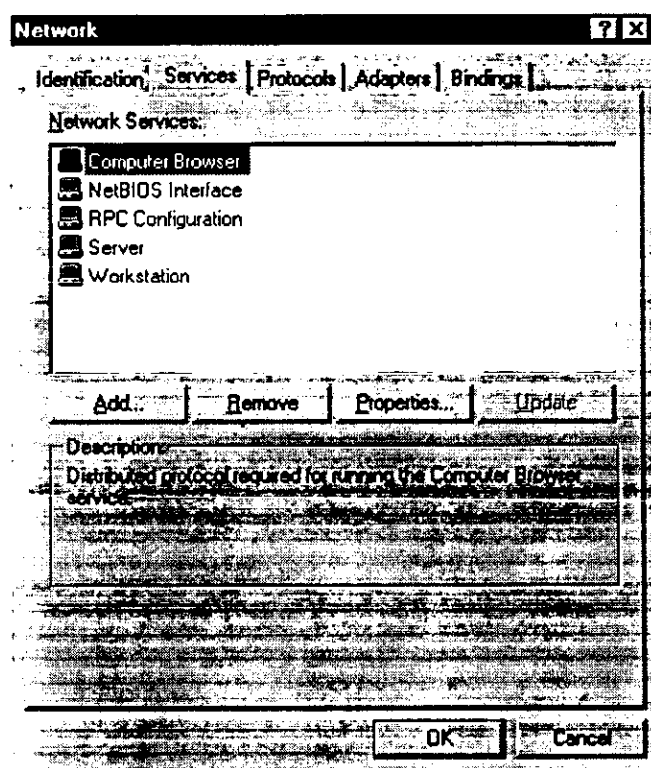


Figura 1.5.2.3 Pantalla de Red

Seguridad centralizada

Windows admite seguridad delegada basada en servidor para redes Netware y Windows NT, permitiendo a la computadora cliente accionar el sistema de seguridad existente. Esto hace que la implementación de la seguridad de la red sea más sencilla y

eficiente mediante la utilización de las cuentas de usuario existentes, permitiendo un inicio de sesión válido en el servidor antes de que el usuario pueda utilizar Windows en un entorno de red. Utilizando la información de las cuentas de usuario, se puede habilitar la seguridad en función a una especificación del usuario para todos los recursos de la red, incluyendo los servicios opcionales de compartición de archivos e impresoras.

Windows proporciona seguridad adicional para algunas tareas como acceso telefónico a redes, admite contraseñas de llamada codificadas y opciones de rellamada, además de dispositivos de hardware de otros vendedores.

- Validación de inicio de sesión
- Seguridad a nivel de usuario
- Seguridad adicional

Planes del sistema (administración y configuración del sistema)

Los planes del sistema permiten definir y controlar de una manera centralizada el acceso de los usuarios a la red y la funcionalidad de escritorio, como por ejemplo, la posibilidad de compartir datos y editar parámetros del sistema. El administrador puede especificar estas restricciones para el usuario, la computadora o el grupo, y así delimitar los accesos.

Se cuenta con un editor de planes del sistema, es una herramienta de administración que se puede utilizar para especificar los derechos y restricciones de usuarios específicos, de computadoras y para crear planes que definan valores generales predeterminados. Estos planes se pueden utilizar para controlar el acceso a la red, para configurar el escritorio y para evitar que los usuarios puedan modificar las

aplicaciones. También puede limitarse a los usuarios para que sólo puedan ejecutar una lista específica de aplicaciones.

Además del editor de planes del sistema, Windows proporciona soporte para agentes que consultan y administran el registro de configuraciones en la computadora cliente. Como resultado, el administrador puede escribir su software de administración del sistema o utilizar el software de otros vendedores para conseguir una administración de la red más potente.

- Editor de planes del sistema
- Agentes de administración del sistema

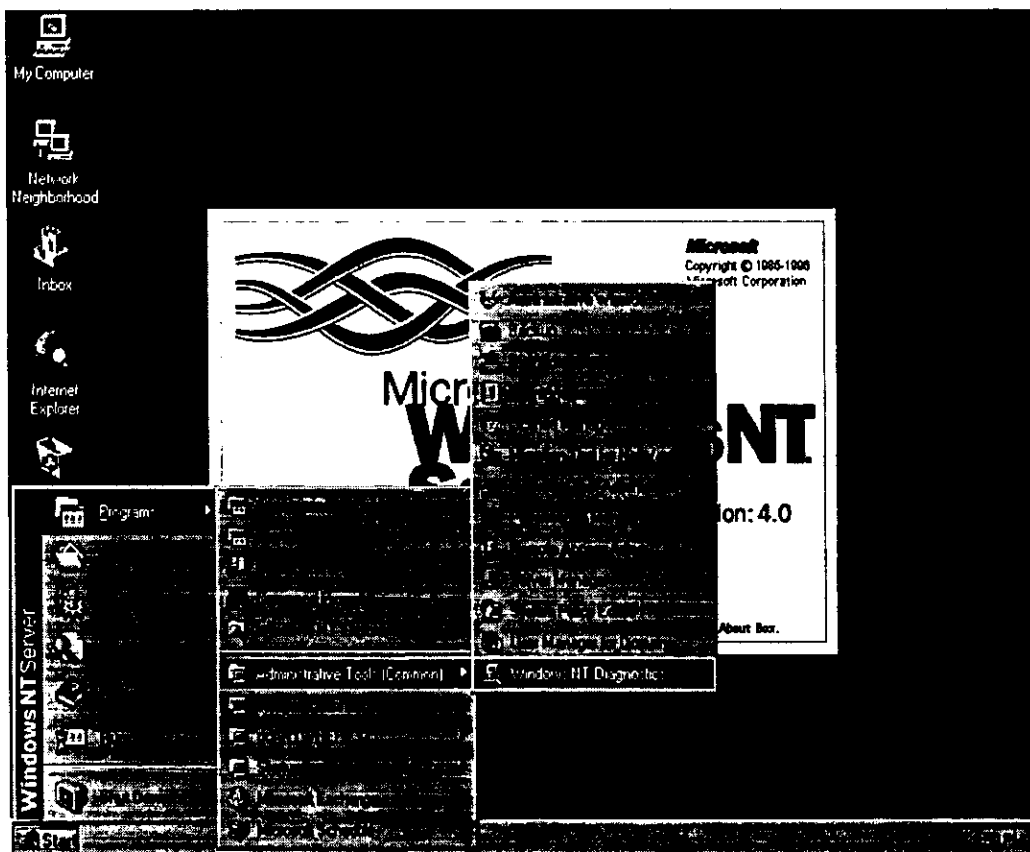


Figura 1.5.2.4 Herramientas de Administración

Perfiles de usuario y de hardware

Los perfiles de usuario describen información específica del usuario o de la computadora, como parámetros y preferencias de software. Los perfiles de hardware determinan los parámetros de hardware actuales en la computadora. Mediante los perfiles, los usuarios pueden trabajar en un entorno consistente y personalizado. Lo que facilita la utilización y la administración de la computadora.

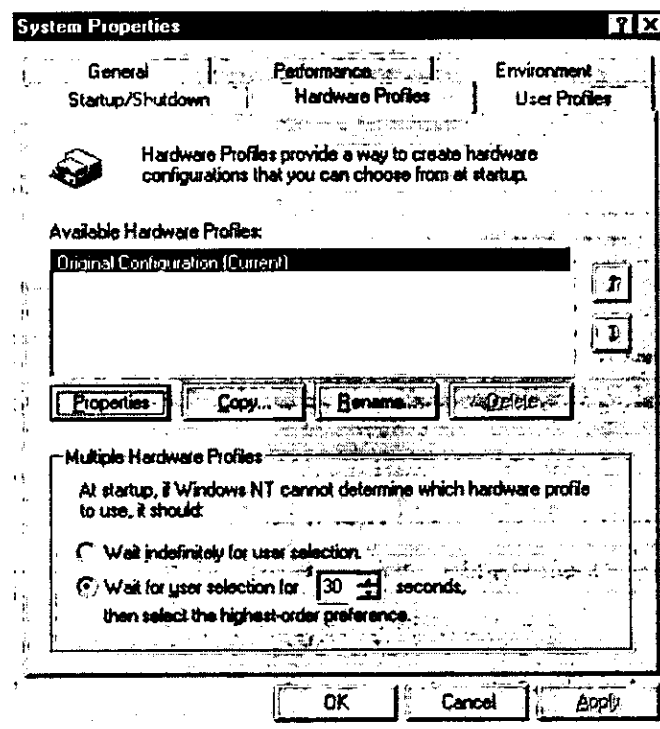


Figura 1.5.2.5 Pantalla de Propiedades del Sistema

Administración remota y copias de seguridad

Windows cuenta con herramientas de administración remota, Monitor de sistema, Editor del registro de configuraciones y Monitor de red, además de agentes para la copia de seguridad que se basan en servidor.

El monitor de sistema facilita mediciones gráficas del tráfico de la red, del rendimiento del sistema de archivos y de otras actividades de la computadora. Mediante el Editor

del registro de configuraciones, el administrador puede editar de manera remota el registro de configuraciones de una computadora en particular.

El monitor de red permite visualizar y desconectar de manera remota las conexiones de red, así como controlar los servicios de compartición de archivos e impresoras de cualquier computadora que ejecuta Windows con el servicio de registro de configuraciones remoto.

Windows incluye agentes de copia de seguridad basados en servidor, con el software de servidor adecuado, se puede realizar una copia de seguridad de Windows sobre otro servidor que disponga de estos agentes.

- Monitor de sistema
- Editor de Registro de configuraciones
- Monitor de red
- Agentes de copia de seguridad

Computación acelerada

Los subsistemas de 32 bits para impresoras y gráficos entre otros, aceleran las operaciones de las tareas más comunes. La impresión es más rápida, tanto en términos del tiempo de regreso a la aplicación, como en términos de velocidad de salida de la impresión.

El sistema de archivos de 32 bits permite que las operaciones que acceden al disco duro se realizan más rápidamente. Las operaciones de red son también más rápidas gracias a los componentes de red de 32 bits que proporcionan mejoras en la velocidad, aunque esta configuración de 32 bits también puede tener sus desventajas.

Multitarea para aplicaciones de 32 bits.

Windows, permite y habilita una clase de aplicaciones de 32 bits con capacidad para multitarea con derecho preferente, que permiten que la computadora realice más de una tarea de forma simultánea, es decir, el usuario no se tiene que esperar a que finalice una determinada tarea, como la carga de un archivo largo desde la red, para poder comenzar a ejecutar otra tarea distinta.

Las aplicaciones de 32 bits, admiten capacidades como nombres de archivos largos y caminos en cuadros de diálogo comunes, facilitando con esto el manejo de archivos a los usuarios.

Acceso telefónico a redes

Windows proporciona varias capacidades para los usuarios remotos y móviles, como son el acceso telefónico a redes y la sincronización de archivos.

El acceso telefónico a redes permite a los usuarios conectarse a los recursos de una red, como archivos o correo electrónico, utilizando un protocolo punto a punto o paquetes de llamadas basados en servidor.

Mi maletín es una herramienta que permite a los usuarios móviles, de una forma sencilla, seguir la pista y actualizar las copias de los archivos almacenados en dos o más computadoras, normalmente una computadora de oficina y una computadora portátil.

- Acceso telefónico a redes
- Mi maletín

Mensajería incorporada

Se incorpora en Windows el soporte necesario para diversos servicios de mensajería, incluyendo correo electrónico y fax, además del acceso a Internet y servicios de conexión, permitiendo a los usuarios comunicarse de manera sencilla con los demás.

La utilidad de Microsoft Exchange, proporciona una bandeja de entrada para todos los servicios de mensajería compatibles, en donde el usuario puede dirigirse para recoger toda su información de correo electrónico y fax.

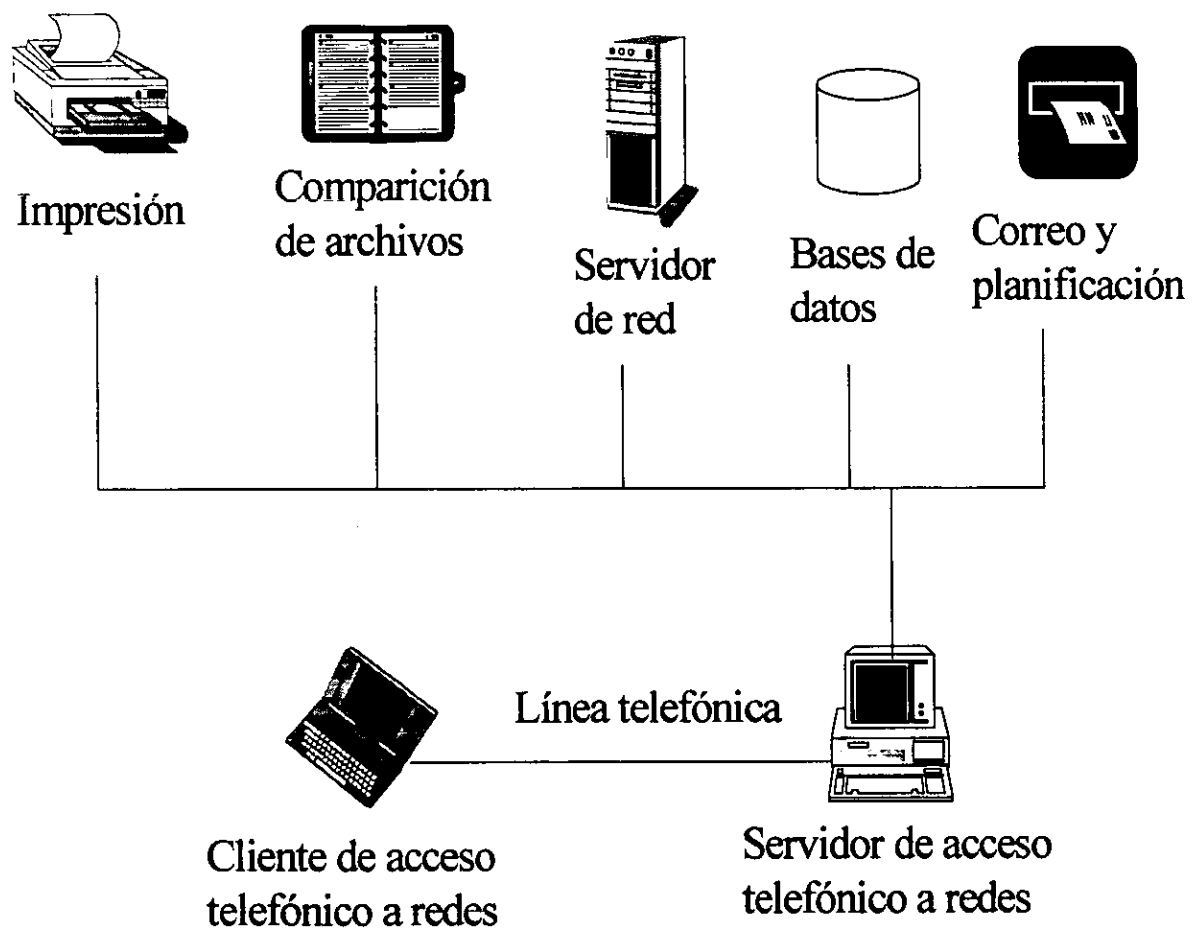


Figura 1.5.2.6 Acceso telefónico a redes

Windows incluye el software de módem y todos los protocolos necesarios para acceder a Internet. Las utilidades de soporte básico para FTP, Telnet y para el soporte avanzado como Mosaic, permiten al usuario introducirse y navegar fácilmente en Internet.

Windows incluye servicios de conexión con lo que se introduce a los usuarios en las facilidades de conexión como, información de productos, foros de conversación y algunos otros como:

- Correo electrónico y fax
- Acceso a Internet
- Servicios de conexión

Instalación basada en servidor.

Windows incluye un programa que automatiza la instalación y permite una fácil personalización cuando se instala Windows desde archivos fuente de red. Este programa de instalación copia automáticamente los archivos de Windows sobre la ruta del servidor seleccionado, de manera que es posible instalar Windows en una computadora individual directamente desde la red, sin necesidad de acceder a las unidades de disco o CD.ROM. Para una mayor seguridad y manejabilidad, Windows se puede instalar de manera que se ejecute desde un servidor, ya sea parcial o totalmente.

Guiones de instalación personalizados

La utilización de guiones para el proceso de instalación, los parámetros predeterminados y las respuestas a las indicaciones de la instalación se pueden leer de manera automática desde un archivo individual, con lo que se reduce significativamente el tiempo de instalación. La instalación basada en servidor tiene opción para crear

guiones de instalación. Mediante las opciones del Servidor de Administración del sistema y un guión de instalación, es posible instalar Windows en una computadora sin necesidad de visitar físicamente su ubicación, tan pronto como el usuario inicia la sesión en red, la instalación se ejecuta automáticamente.

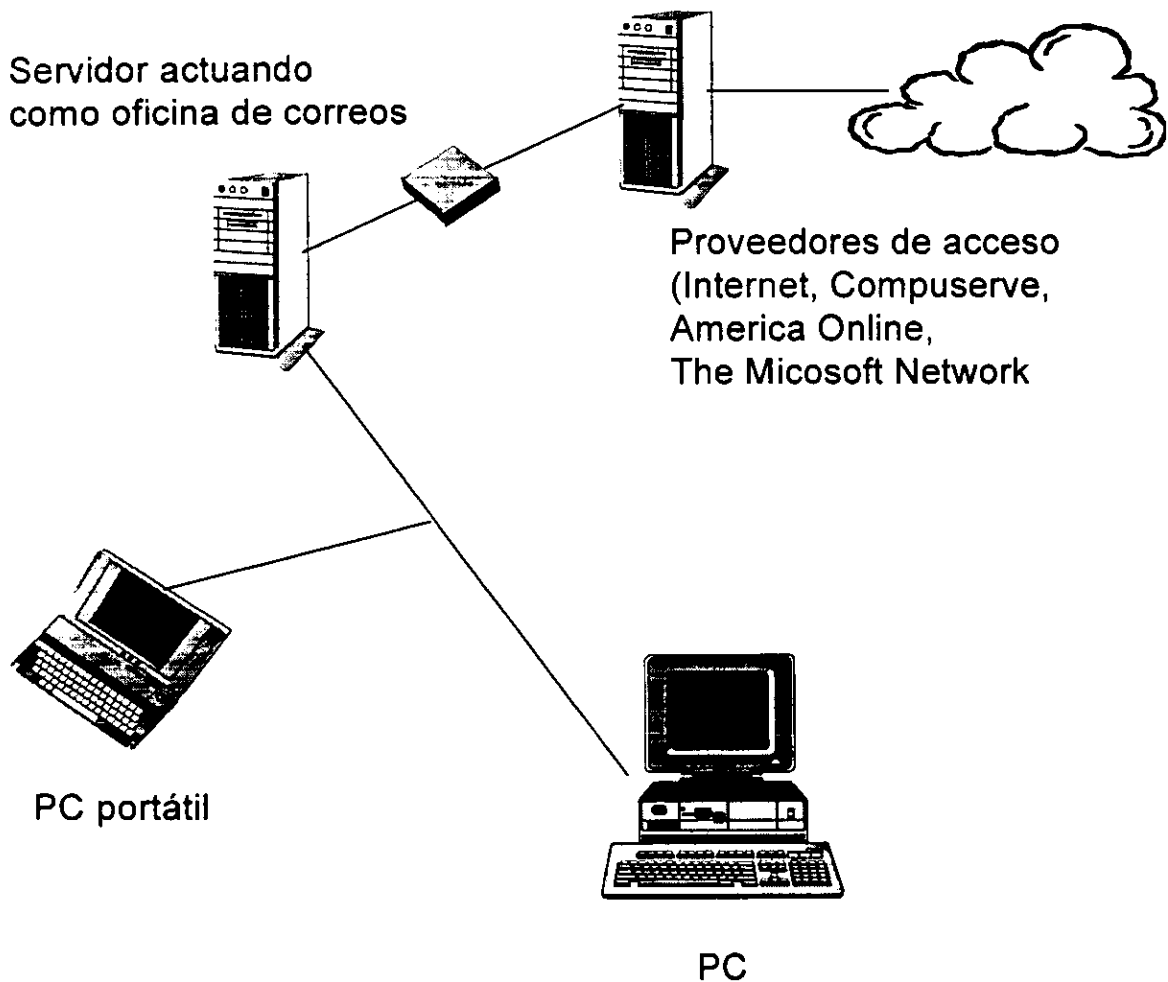


Figura 1.5.2.7 Servicios de Mensajería

Nueva Ayuda y Tutorial

Windows incluye una ayuda basada en tareas y un tutorial que ayudan al usuario a obtener información que necesitan para realizar una tarea específica. La mayoría de los procedimientos se han llevado de la documentación a la ayuda interactiva, donde se

puede acceder mientras se esta trabajando. Asimismo la ayuda está orientada a tareas y a líneas de flujo, simplificando su utilización.

Windows incluye un tutorial que facilita al usuario el aprendizaje de las tareas más comunes mostrando los pasos a seguir mediante un enfoque gráfico, incluye también pistas y ejemplos, Además se cuenta con la Ayuda de contexto que proporciona ayuda específica en un cuadro de diálogo.

- Ayuda interactiva
- Tutorial de introducción al sistema
- Ayuda de contexto

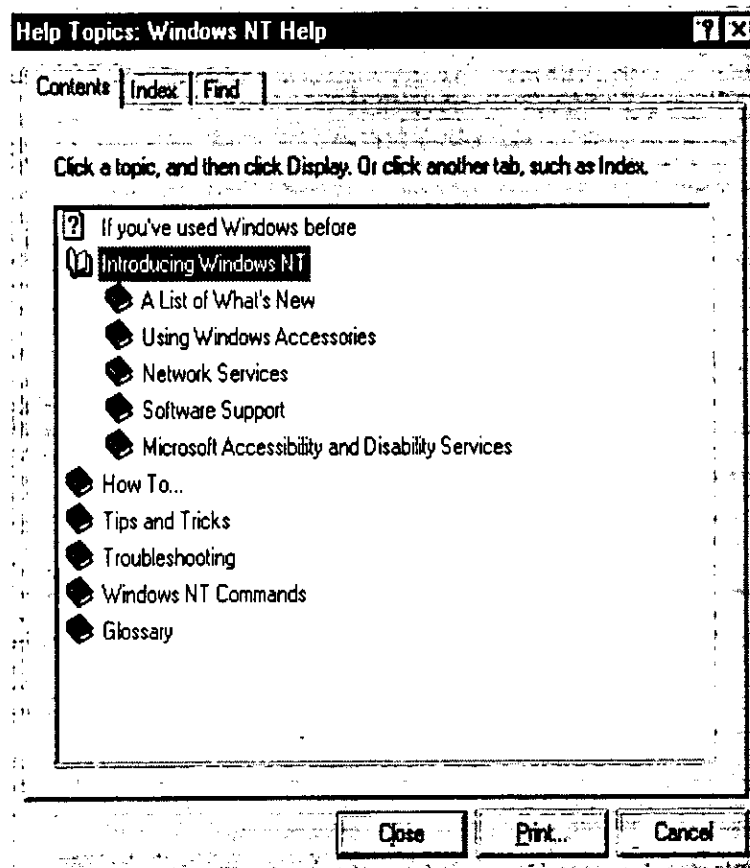


Figura 1.5.2.8 Pantalla de ayuda

Requerimientos de hardware para Windows 9X/Windows NT

Componente	Windows 9X	Windows NT Server	Windows NT Workstation
Computadora	Procesador 80386; recomendable superior	80486; recomendable Pentium	80486; recomendable Pentium
Memoria	16 Mb; se recomiendan 16 MB o más	32 Mb o más; preferible 64 Mb	32 Mb o más;
Espacio en disco	65 MB	158 Mb	124 MB
Componentes	Unidad de disco de 3 ½" y 1.44 Mb Monitor de resolución VGA o superior Mouse o apuntador compatible	Unidad de disco de 3 ½" y 1.44 Mb Monitor de resolución SVGA o superior Mouse o apuntador compatible Unidad de CD-ROM Unidad de respaldo	Unidad de disco de 3 ½" y 1.44 Mb Monitor de resolución SVGA o superior Mouse o apuntador compatible
Opcionales	Fax-Modem de clase 1, 2 o que cumpla con la especificación de aplicaciones de comunicación o acceso a través de la red Unidad de CD-ROM Tarjeta de audio	Fax-Modem de clase 1, 2 o que cumpla con la especificación de aplicaciones de comunicación o acceso a través de la red Tarjeta de audio	Fax-Modem de clase 1, 2 o que cumpla con la especificación de aplicaciones de comunicación o acceso a través de la red Unidad de CD-ROM Tarjeta de audio

Tabla 1.5.2.1 Requerimientos de hardware para Windows 9X/NT

1.6 TIPOS DE REDES Y PROTOCOLOS DE COMUNICACIÓN

Debido a la importancia que han adquirido en la actualidad y, a su diversificación, no podemos sustraernos a la importancia que las redes de computadoras tienen actualmente en la vida de la mayoría de las empresas y en lo importante que serán en un futuro. Es por eso que trataremos en el presente capítulo de ejemplificar de una manera concisa y clara los tipos de redes de computadoras que existen en la actualidad y su funcionamiento, así como los protocolos de comunicación más comunes que se pueden utilizar con las mismas, de igual manera veremos el modelo OSI, mediante el cual se definen las normas y los estándares internacionales de comunicaciones, con el propósito de tener un panorama completo de los sistemas su funcionamiento y conectividad.

1.6.1 Antecedentes

Comenzaremos por definir lo que es una red de computo. Una red esta definida por un conjunto de elementos unidos, comunicados o interconectados entre sí a través de un medio común, es decir que tenemos una relación física entre los elementos que la componen con el propósito de tener un medio colectivo de trabajo, mediante el cual podamos compartir los recursos de **hardware (dispositivos, periféricos y equipos)** y los elementos de **software (archivos, programas, bases de datos, sistemas operativos, etcétera)** con los que cuenta la red.

El uso de las computadoras y la manera de verlas como simples equipos de escritorio, cambio radicalmente con la introducción de las redes, ya que con estas se introducía a los usuarios a un ambiente en el cual podrían compartir recursos e información de una manera más rápida, sencilla y optimizando su trabajo y desempeño. En los siguientes incisos se describen los tipos de redes más comunes, las cuales, básicamente se clasifican de la siguiente manera:

- Redes LAN (Redes de área local)

- Redes MAN (redes de área metropolitana)
- Redes WAN (redes de área amplia)

A parte de compartir recursos y optimizarlos, al contar con una red se obtiene gran confiabilidad dado que tenemos disponibilidad de información y equipo, además de ser un poderoso medio de comunicación, como se muestra en la figura 1.6.1.1. La clasificación se hace principalmente por la distancia que media entre las terminales, como observaremos en los siguientes puntos.

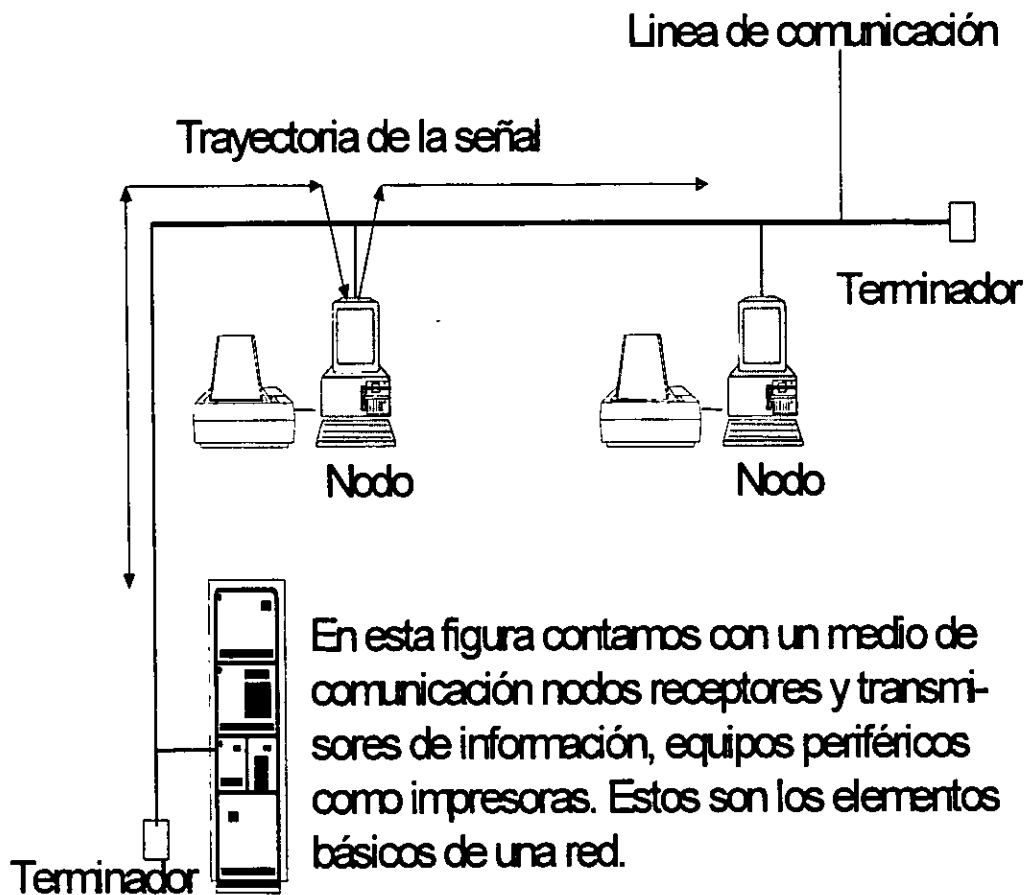


Figura 1.6.1.1 Elementos básicos de una red

1.6.2 Redes de computadoras, tipos y configuraciones

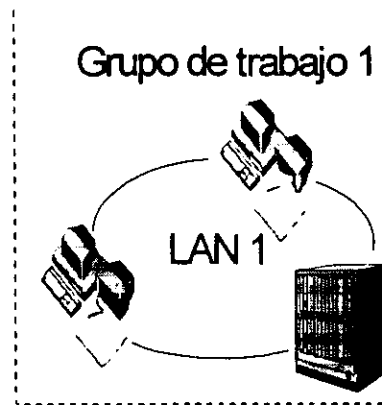
Como observamos una red nos permite la interacción de varios usuarios con los diferentes elementos que rodean su medio ambiente de trabajo logrando de esta forma que no solamente se tenga acceso a los datos e información de quien lo necesite, sino que también se podrán compartir los recursos de hardware y software como son impresoras, plotters, espacio en disco duro, paquetería de diseño, administrativa, presentación, correo electrónico, etcétera; con lo que se reducen costos al compartir recursos. De igual forma se comparte información generada por los mismos usuarios que es de utilidad para otros usuarios u otras áreas esto se puede realizar a través de bases de datos o sistemas de información que se pueden compartir en un momento dado, con lo que optimizamos la atención a los clientes o para la pronta elaboración de algún proyecto.

Por otro lado los diferentes tipos de redes se conectan de diferentes formas, es decir, ocupan configuraciones o topologías diferentes, así como medios de transmisión de datos diferentes también. Esto dependerá de los requerimientos y necesidades de las empresas y/o de los usuarios

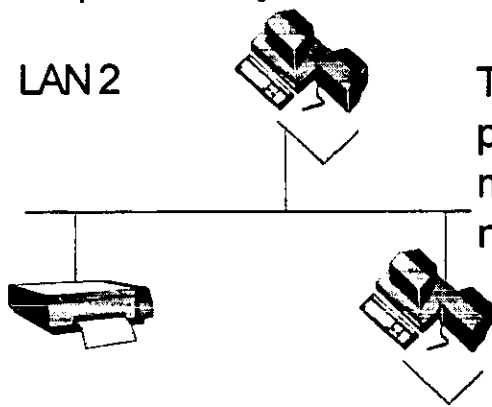
1.6.2.1 Redes LAN

Entendemos como una red local o LAN a un sistema de transmisión de datos que nos permite, compartir recursos e información por medio de computadoras (ordenadores), mini-computadoras, terminales, o cualquier otro dispositivo periférico que se encuentre dentro del espacio físico de un mismo centro. Estas se han diseñado para facilitar la interconexión de una gran variedad de equipos de tratamiento de información dentro de un centro o un área determinada (normalmente oficina, piso o edificio). Este concepto incluye tanto al hardware como al software necesario para el funcionamiento de la misma. En la figura 1.6.2.1. se muestra una típica y básica red LAN en una oficina, donde encontramos Equipos de usuario, un equipo principal (Servidor) e impresora, que funciona como dispositivo de salida.

Podemos observar dos redes lan que pueden estar en oficinas contiguas o pisos de un mismo edificio.



Grupo de trabajo



También podemos observar que pueden tener diferentes formas y maneras de conectarse o comunicarse.

Figura 1.6.2.1.1 Red típica de área local (LAN)

Las principales características con que podemos definir una red local se describen a continuación:

- Un medio de comunicación común mediante el cual todos los dispositivos pueden compartir información, programas y equipo, sin importar donde se ubiquen tanto el usuario como el dispositivo, normalmente dentro del mismo edificio.
- Una velocidad de transmisión muy elevada, de 1 a 100 mb/s (depende tanto de las características de la red, como de los equipos y de los medios de transmisión), con el fin de adaptarse a las necesidades de los usuarios y del equipo.

- Una distancia entre estaciones de trabajo relativamente corta, que va desde unos metros hasta no más de 10 km. dependiendo del medio de transmisión. Esta característica depende también tanto del lugar como de las trayectorias que se sigan para la instalación de los cableados.
- Conexión sencilla, funcionamiento de los equipos de manera individual y colectiva y fiabilidad tanto del equipo como de la información puesto que el usuario administra y controla su propio sistema.

Una red local bien diseñada va siempre dirigida a proporcionar servicios a los usuarios. Estos servicios están definidos por el lugar que ocupa el usuario dentro de una organización, los trabajos que desarrolla y la necesidad de obtener y proporcionar información.

Como una herramienta de tratamiento de la información y con los avances tan acelerados en sistemas y equipo, las redes locales son aplicables en casi todos los campos, por ejemplo: la industria, los negocios, la ciencia, la educación, etcétera. Económicamente se gana en la optimización de todos los recursos, tanto humanos como materiales.

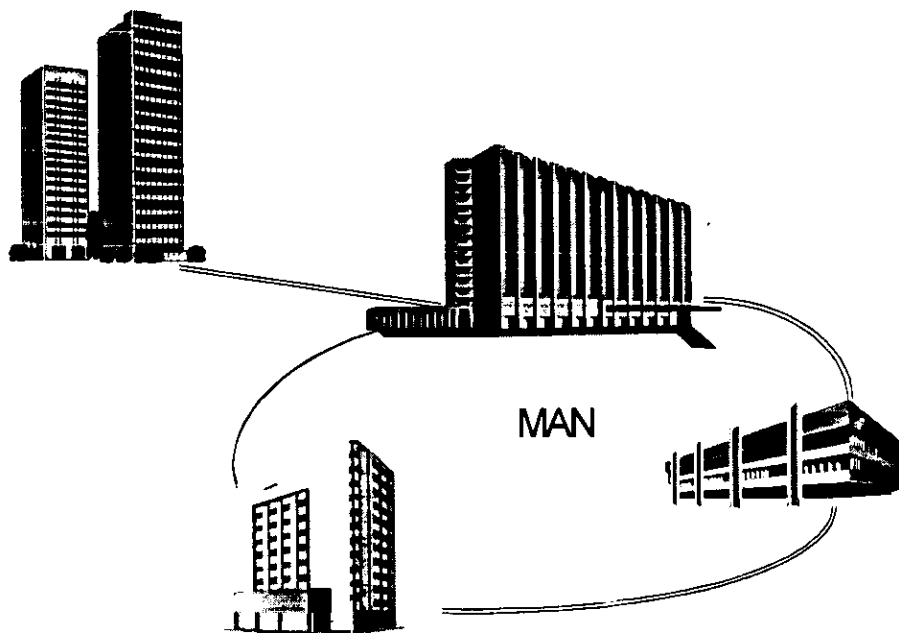
Enseguida se enumeran Algunas de las principales ventajas de las redes locales como son las siguientes:

- Aumento de la productividad.
- Reducción de costos.
- Aumento del nivel de comunicación.
- Simplicidad de gestión.

1.6.2.2 Redes MAN

Este tipo de redes se encuentra en un punto intermedio entre las redes LAN y las redes WAN; estas redes pueden cubrir un área geográfica que puede estar entre 10 y 50 km., es decir se utiliza principalmente para comunicar entidades dentro de una misma ciudad o campus. Como se ve en la figura 1.6.2.2.1.

Este tipo de redes normalmente sigue la norma de acceso 802.6 del **IEEE (Instituto de ingenieros eléctricos y electrónicos)**, que es la tecnología de **DQDB (bus dual de cola distribuida)**. Que son dos cables de fibra, normalmente, proporcionando transmisión en direcciones opuestas al mismo tiempo. Un nodo perteneciente al bus dual puede enviar datos en ambas direcciones y se incorpora en una topología en anillo.



Una MAN esta compuesta por una serie de redes conectadas entre si que se encuentran ubicadas en un mismo campus. Por ejemplo las escuelas dentro de la universidad, una cadena de centros comerciales en la ciudad de México, etcétera.

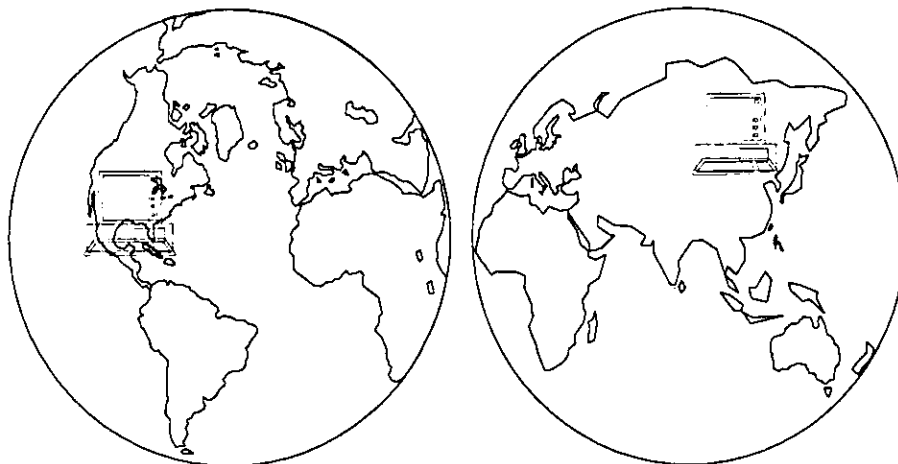
Figura 1.6.2.2.1 Red de área metropolitana (MAN)

De igual forma, las MAN pueden estar integradas por varias LAN, lo que las hace universalmente aplicables con un alto desempeño y alta velocidad. Se puede transmitir voz, datos y vídeo a través de sistemas digitales.

Son redes que pueden utilizar cable coaxial de banda ancha, aunque su transmisión es media y se enlazan también por microondas, fibra óptica y vía satélite.

1.6.2.3 Redes WAN

A este tipo de redes se conectan tanto LAN's como MAN's a través de dispositivos que permiten su conectividad local y remotamente, aunque las redes que se vayan a interconectar tengan diferentes topología y/o protocolos. Estos dispositivos de interconectividad pueden usar diferentes medios de transmisión que pueden ser públicos o privados.



El internet es el mejor ejemplo que tenemos en la actualidad de una red WAN ya que a través de ella podemos comunicarnos prácticamente a cualquier parte del mundo y tener acceso a la información de escuelas, empresas y oficinas gubernamentales entre otras.

Figura 1.6.2.3.1 Red de área amplia (WAN)

Las redes de área amplia ocupan un área geográfica bastante grande, con radios que superan los 50 km. e incluso conexiones intercontinentales. Este tipo de redes tan grandes ha avanzado rápidamente y en la actualidad se ha convertido en sector estable. La interconexión de este tipo de redes se realiza a través de equipos de comunicación con tecnología de punta, muy sofisticados. El mejor ejemplo de este tipo de redes lo tenemos actualmente en lo que se conoce como la super carretera de la información "INTERNET". Figura 1.6.2.3.1.

1.6.2.4 Redes Ethernet

Las redes ethernet son relativamente nuevas y en las ultimas décadas han alcanzado gran popularidad gracias a las topologías que utilizan y a su gran capacidad de transmisión ya que pueden conectarse hasta 8000 estaciones de trabajo en una sola LAN. Ethernet presenta un rendimiento de 10 Mb/seg.; y utiliza un método de acceso sensible a la señal portadora, mediante el que las estaciones de trabajo comparten un cable de red, pero solo una de ellas puede usarlo en determinado momento. Este método de acceso múltiple con detección de portadora y detección de colisiones (CSMA/CD) es utilizado para arbitrar el acceso al cable.

En la siguiente lista se relacionan todas las topologías utilizadas en las redes ethernet, donde el primer número se refiere a la velocidad en Mb/seg.; y el ultimo a los metros que admite un segmento (multiplicado por 100) Base y Broad hacen referencia al método de transmisión, banda base y banda ancha, respectivamente.

- 10Base-5. Cable coaxial con longitud máxima de segmento de 500 metros.
- 10Base-2. Cable coaxial con longitud máxima de segmento de 185 metros.
- 10Base-T. Cable de par trenzado con longitud máxima de segmento de 100 metros.

- 1Base-5. Cable de par trenzado con longitud máxima de segmento de 500 metros y máxima velocidad de transmisión de 1 Mb/seg.
- 10Broad-36. Cable coaxial con longitud máxima de segmento de 3600 metros.
- 10Base-F. Sirve de soporte a redes, soporte de fibra óptica de hasta 4 km. con transmisión a 10 Mb/seg.
- 100base-X. Esta es una nueva norma ethernet que presenta un rendimiento de 100 Mb/seg. y utiliza el método de acceso CSMA/CD sobre configuraciones jerárquicas de cableado de par trenzado.
- 100VG-AnyLAN. Al igual que la anterior es una nueva norma con rendimiento de 100 Mb/seg. y utiliza el método de acceso de prioridad bajo demanda sobre configuraciones jerárquicas de cableado de par trenzado.

Es una tecnología de transmisión de datos de alta velocidad con una topología de bus lineal que usa canal compartido de comunicación.

1.6.2.5 Configuración estrella

En una configuración en estrella cada estación de trabajo se encuentra conectado a un nodo central a través de un canal punto a punto dedicado. El nodo o servidor central recibe los mensajes de las estaciones y los retransmite a las estaciones a las que van dirigidos.

El control de la red puede ser asignado de las siguientes tres formas:

- El control lo tiene la estación central, que es el que realiza la retransmisión de los mensajes. En este nodo se pueden procesar los datos recibidos o se envían a otra estación para que los procese.

- El control se entrega a una de las estaciones exteriores, en lugar de la estación central.
- Por ultimo, el control se puede distribuir entre todas las estaciones. El nodo se utiliza para enviar mensajes a sus destinos y para resolver las solicitudes de conexiones conflictivas entre estaciones de trabajo.

En cualquiera de los casos anteriores el nodo es la estación principal y si esta falla se suspenderá el servicio de red. Además este mismo nodo es el punto lógico para conectar directamente los recursos compartidos más importantes.

De la potencia de la estación central, depende directamente la capacidad de la red. Esta topología nos elimina la necesidad de que las estaciones tomen decisiones de retransmisión.

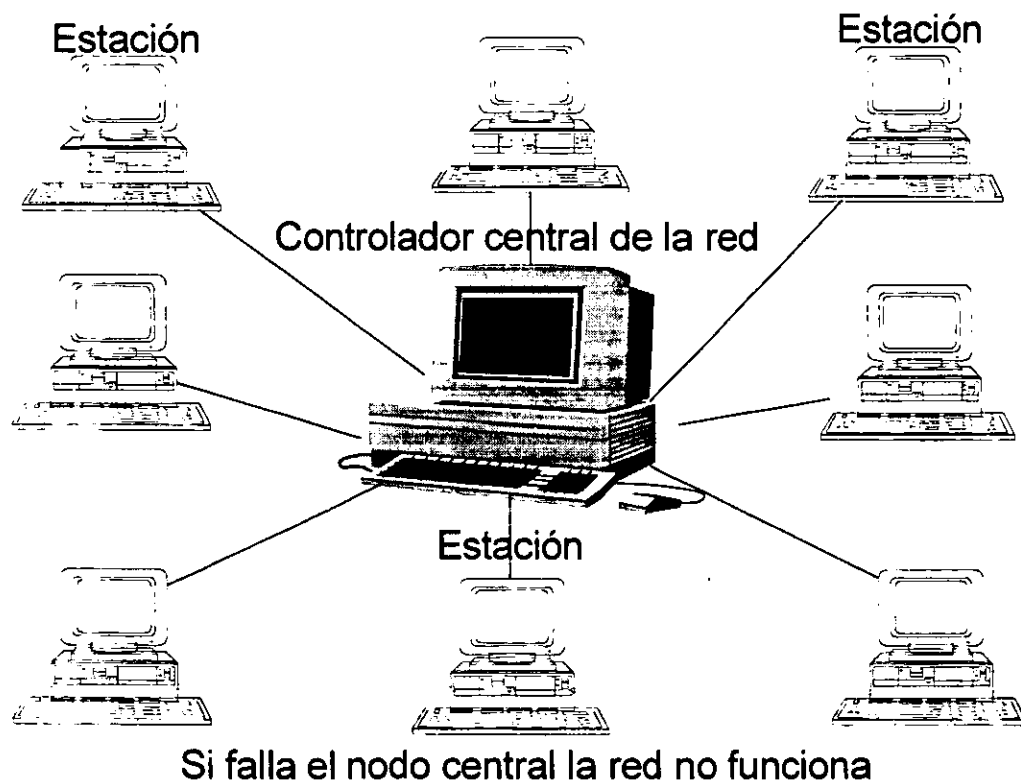


Figura 1.6.2.5.1 Topología en estrella

La topología en estrella puede ser evaluada por los siguientes factores:

- Aplicación. Actualmente esta topología nos proporciona la mejor forma de integrar servicios de datos y voz.
- Complejidad. La complejidad de esta topología depende proporcionalmente de la cantidad de estaciones conectadas y de cuantas de estas funcionan como nodos centrales de otras estaciones o se conectan a enlaces remotos.
- Respuesta. La respuesta en este caso será buena mientras no exista una carga muy grande, normalmente la estación central debe ser dedicada.
- Vulnerabilidad. El que la red sea confiable depende directamente de la estación central.

De lo anterior se desprende algunos puntos sobre las ventajas y desventajas que tiene este tipo de topología.

Ventajas:

- Es posible obtener un alto nivel de seguridad.
- Es posible conectar terminales tontas
- Es perfecta para cuando tenemos configuraciones en donde se conectaran muchas estaciones a una.
- Las velocidades de transmisión pueden diferir.
- Se facilita la detección y localización de fallas.

- La transmisión de los mensajes esta controlada por el nodo central.

desventajas :

- Las velocidades en comparación de las topologías de bus y anillo son inferiores debido a la actividad de la estación central.
- Altos costos por la complejidad tecnológica.
- Y por ultimo la susceptibilidad de fallas en la estación central.

1.6.2.6 Configuración en anillo

La red en anillo esta formada por un circulo de conexiones punto a punto de estaciones contiguas. Esta red se organiza de tal manera que los datos pasan de una estación a otra a través de repetidores conectados entre sí secuencialmente, por medio de cualquier medio físico de transmisión (coaxial, par trenzado, fibra óptica). Una variante de este tipo de topología y que actualmente es más utilizado es el anillo doble redundante que permite tener más seguridad, velocidad y la hace bidireccional.

A continuación describiremos brevemente los factores de evaluación de una topología en anillo.

- Aplicación. La aplicación de una red en anillo es interesante en situaciones en las que se tiene que asignar la capacidad de la red en forma equitativa, o si se tienen que conectar estaciones en un número reducido para funcionar a velocidades muy altas en cortas distancias.
- Complejidad. El hardware requerido por este tipo de topologías es relativamente complicado. Pero dado que el mensaje solamente se mueve en una sola

dirección el desvío de mensajes se hace sencillo ya que la estación emisora solo necesita saber la dirección del receptor.

- Respuesta. A pesar de que el tiempo de espera aumenta más lentamente que en otras topologías y que con tráfico la respuesta del sistema es muy estable, el tiempo de espera medio es muy alto aun cuando la carga sea muy baja.
- Vulnerabilidad. En este renglón esta topología es fácilmente susceptible a fallas, ya que si una estación falla provoca que todo el sistema deje de funcionar debido a su interdependencia. Además es difícil localizar alguna falla y para tratar de mantener la red en funciones es necesario una inversión muy alta.
- Expansión. En este renglón es sencillo crecer la red si esta fue equipada con centros conectores adecuados para poder anexar estaciones sin necesidad de suspender o interrumpir el sistema, todo lo anterior a costos no muy altos.

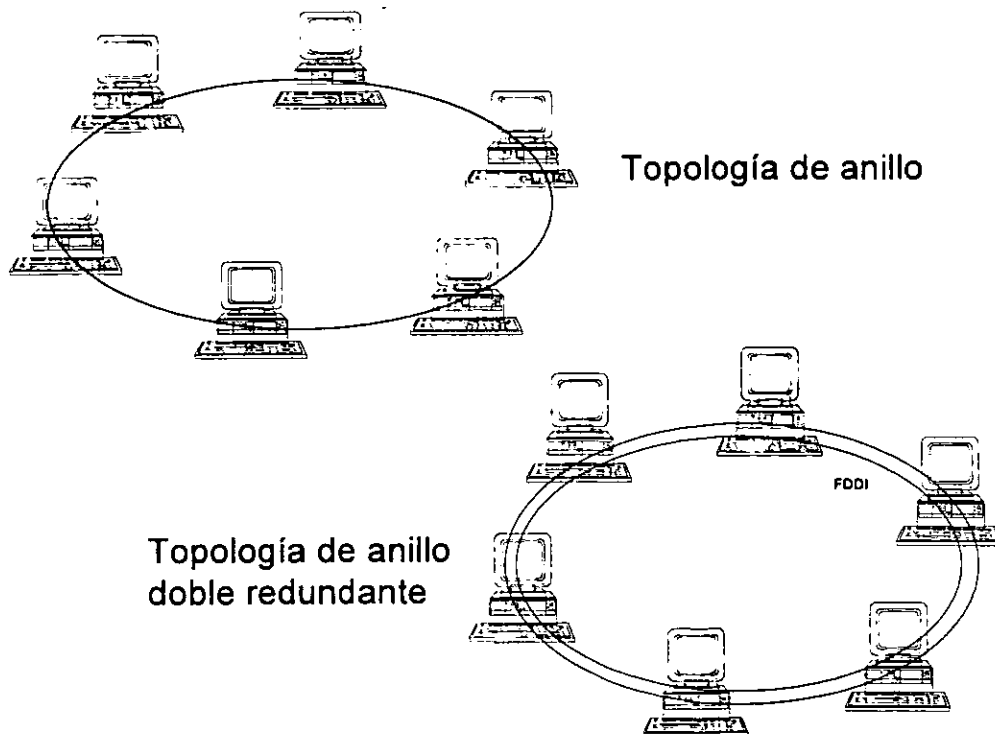


Figura 1.6.2.6.1 Topología en anillo y anillo doble

En la figura anterior se muestra la configuración en anillo. Como en las demás topologías se tienen ventajas y desventajas y es importante conocerlas ya que pueden influir en la decisión de que topología utilizar.

Ventajas:

- Todos los usuarios o estaciones se reparten la capacidad de transmisión.
- La red no depende de un nodo central y permite utilizar distintos medios de transmisión.
- La incidencia de errores es muy pequeña, fácilmente se pueden comprobar los errores de transmisión y se facilita la localización de los nodos y enlaces que los originan.
- Es relativamente fácil enviar mensajes comunes y la distribución de los mismos.
- Se consiguen velocidades de transmisión altas y el tiempo de acceso es moderado, incluso con tráfico.

Desventajas:

- En los repetidores se concentra la confiabilidad de la red, la instalación es muy complicada y se necesita un monitor.
- Si la red no está equipada adecuadamente con centros de conexión, puede ser difícil integrar nuevas estaciones sin interrumpir el sistema.
- El integrar una estación nueva puede resultar muy caro.

1.6.2.7 Configuración de bus

Esta topología se caracteriza por tener un canal único de comunicación a donde se conectan todas las estaciones sin que una dependa de otra y donde la señal o datos viajan en ambas direcciones. Para este tipo de topología se tienen longitudes de segmentos de cable más reducidos o bien colocando repetidores o incluso utilizando un número reducido de estaciones, con el propósito de que la señal mantenga su intensidad. La figura 1.6.2.7.1. nos muestra la configuración de bus.

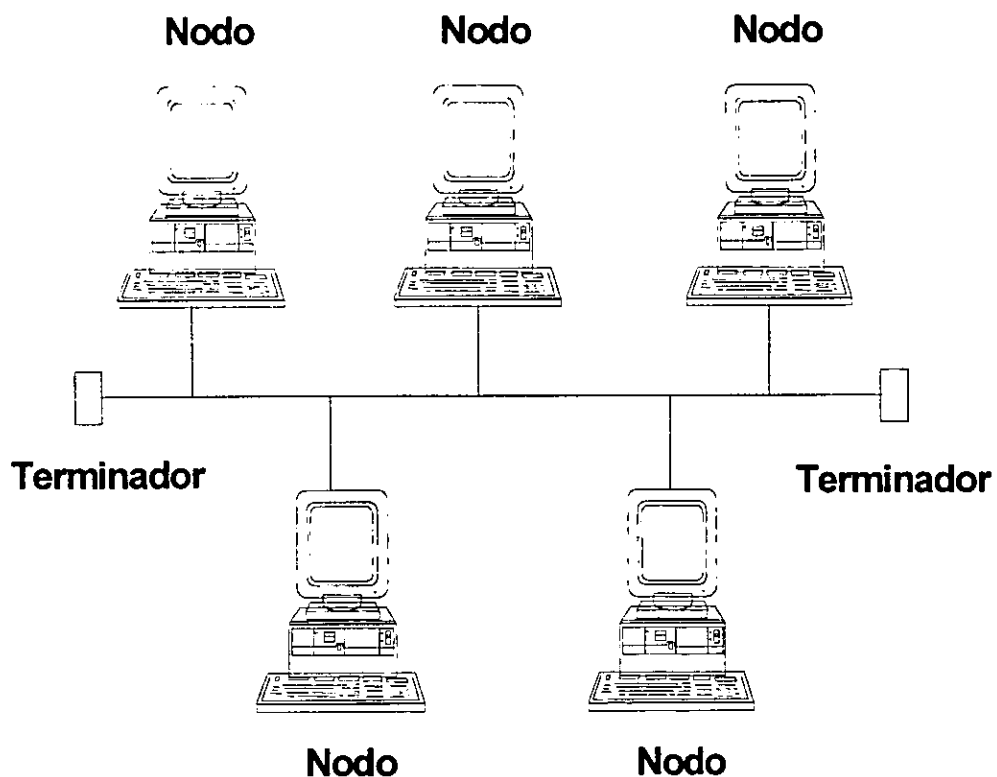


Figura 1.6.2.7.1 Configuración de bus

Como en las demás topologías tenemos factores que nos permiten evaluar este tipo de configuración.

- Aplicación. Este tipo de configuración es utilizado normalmente en redes pequeñas y con poco tráfico.

- Complejidad. En este caso resultan ser relativamente sencillas.
- Respuesta. Esta depende de la cantidad de tráfico existente ya que a mayor tráfico menor es el tiempo de respuesta.
- Vulnerabilidad. El principal problema de esta topología es que se dañe el canal principal ya que además de suspenderse la transmisión, es difícil localizar la falla.
- Expansión. El crecimiento es sencillo y rápido sin que esto afecte el desempeño de la red en gran medida.

Las ventajas y desventajas que se pueden observar en estas configuraciones son las siguientes.

Ventajas:

- Es posible utilizar toda la capacidad de transmisión disponible.
- El medio de transmisión es totalmente pasivo y fácil de instalar.
- Es sencillo conectar nuevos dispositivos y es adecuada para tráfico muy alto.

Desventajas:

- Las longitudes de los medios de transmisión son muy pequeñas, además los recursos del sistema no se reparte equitativamente.
- La interfaz con el medio de transmisión debe ser inteligente si las estaciones o dispositivos no lo son.
- La red es fácilmente accesada o intervenida sin interrumpir su funcionamiento.

1.6.2.8 Configuración en malla

En los últimos años se ha venido utilizando más este tipo de topología, esto por su relativa inmunidad a los problemas de tráfico y averías. Gracias a la multiplicidad de caminos que ofrece a través de los distintos Equipos terminales de datos (ETD) y los equipos de conmutación de datos (ECD), estos equipos hacen posible orientar el tráfico a través de trayectorias alternativas en caso de que algún nodo este averiado u ocupado.

A pesar de que la realización de este método es compleja y cara (para proporcionar estas funciones especiales, la lógica de control de los protocolos de una red en malla puede llegar a ser sumamente complicada, así como los equipos de transmisión utilizados), A pesar de eso mucho usuarios prefieren este tipo de red por su confiabilidad. En la figura 1.6.2.8.1 se muestra un ejemplo de este tipo de configuración.

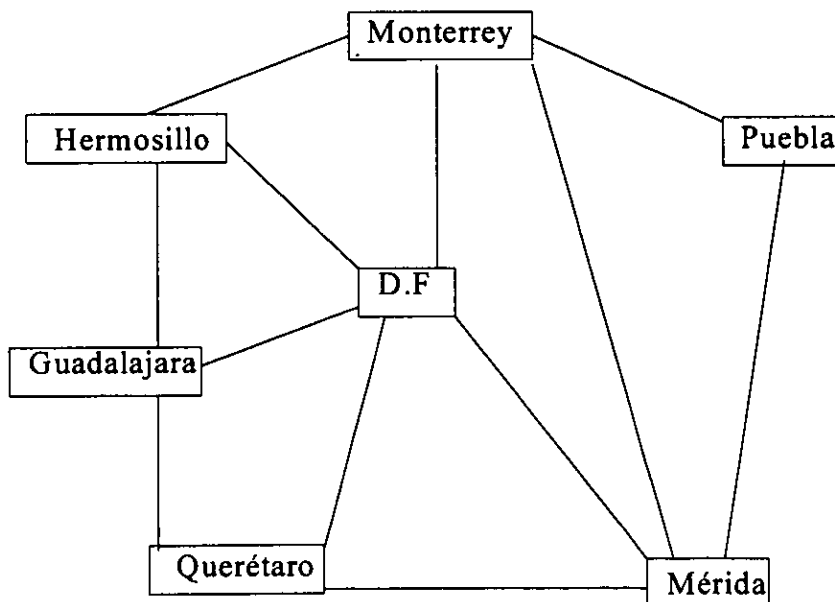


Figura 1.6.2.8.1 Configuración de red en malla

1.6.2.9 Dispositivos de conectividad

Además de instalar el medio de transmisión, se deben conectar las estaciones de trabajo al medio. Los dispositivos e interfaces que se utilizan para conectar los dispositivos de cómputo y los medios de transmisión son llamados hardware de conectividad. En esta sección se presentan dos grupos de hardware de conectividad, que son los siguientes:

- De red
- Inter-red

El término "de red" se refiere a una red sencilla independiente aislada, en la que dispositivos individuales se conectan entre sí a través de ella.

El segundo término, inter-red, hace referencia a un grupo de redes independientes que se conectan entre ellas.

Hardware de conectividad de red

Para empezar a construir una red de cómputo, se necesita cierto número de dispositivos de hardware para conectar cada computadora al medio de comunicación. Estos dispositivos incluyen:

- Conectores
- Adaptadores de red

También se pueden conectar varios segmentos separados de medios de transmisión para formar una red grande. Para este propósito, se utilizan los siguientes dispositivos de red:

- Módems

- Repetidores
- Concentradores
- Puentes
- Multiplexores

Enseguida se describe cada uno de los dispositivos de conectividad que ya se han listado.

Conectores

Estos son dispositivos mecánicos que se utilizan para conectar el medio de transmisión y la tarjeta de red de la PC. Cada medio tiene uno o más conectores físicos con los cuales se pueden enlazar varios dispositivos. En la figura 1.6.2.9.1 se muestran algunos conectores.

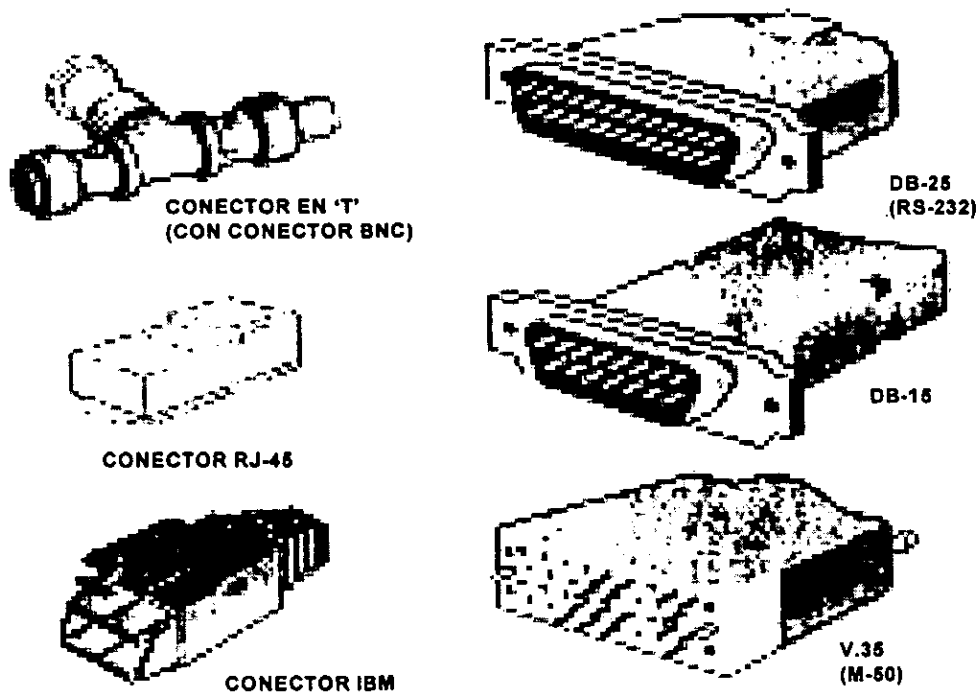


Figura 1.6.2.9.1 Ejemplos de conectores para medios de transmisión.

Adaptadores de red

El siguiente dispositivo de conectividad necesario es un adaptador de red. Técnicamente, un adaptador de red incluye todas las conexiones físicas y lógicas entre la computadora, u otro dispositivo, y el medio de transmisión. Típicamente, este es una tarjeta lógica que se instala en una computadora para enlazarla al conector del cable.

Sin embargo, los adaptadores de red pueden ser una tarjeta de interface, una porción de una tarjeta lógica de dispositivo, software y un puerto genérico, o un número de dispositivos externos (figura 1.6.2.9.2).

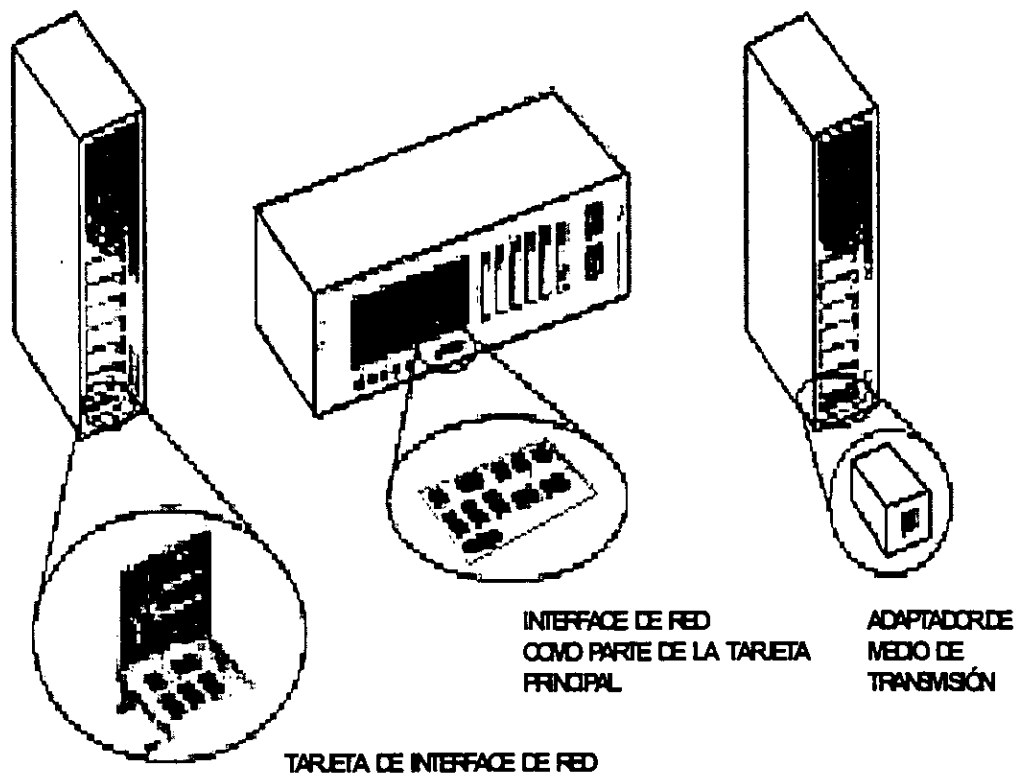


Figura 1.6.2.9.2 Adaptadores de red.

A continuación describiremos los adaptadores de red con otros términos, que también se utilizan normalmente para estos dispositivos

- **Transceptor.** Todas las tarjetas de interfaces de red incluyen algún tipo de transceptor. Los transceptores son dispositivos que pueden transmitir así como recibir señales eléctricas o electromagnéticas en el medio de transmisión. Vienen en diferentes tipos pero todos ellos emiten ondas eléctricas, luminosas o electromagnéticas. Cuando se utiliza cable como medio, los transceptores a menudo son acoplados con conectores, que son del género opuesto a los que están directamente conectados al medio (dichos conectores son descritos como machos o hembras). Cuando se utiliza un medio no basado en cable, los transceptores son sólo dispositivos de transmisión o recepción debido a que no se requieren conectores mecánicos.
- **Tarjeta de interface de red.** También conocidas como NIC (Network Interface Card). Cuando el dispositivo de usuario final no proporciona un puerto compatible o un circuito de interface a la red, se utilizan tarjetas de circuito impreso llamadas tarjetas de interface de red. Estas tarjetas incluyen la circuitería y las conexiones mecánicas para convertir las señales eléctricas de la computadora a señales eléctricas o electromagnéticas utilizadas en el medio. Una tarjeta generalmente incluye un transceptor sencillo pero puede proveer uno o más tipos de conectores.
- **Adaptador de medio de transmisión.** Cuando un circuito de interface de red utiliza un conector, que es diferente al que ya está conectado al medio de transmisión, se utiliza un adaptador de medio de transmisión. El propósito de este adaptador es recibir señales de un tipo de conector y convertirlas para ser usadas por otro tipo.

Módems

Los módems (MOdulador/DEModulador) convierten las señales digitales de la computadora a una señal de transmisión analógica para ser usada por líneas telefónicas o transceptores de microonda. Los módems son necesarios por que los

PSTN's (red telefónica publica conmutada) y las microondas usan ondas electromagnéticas, y la computadora utiliza pulsos eléctricos. La figura 1.6.2.9.3 muestra este dispositivo.

Los módems también son útiles cuando la señal de un transceptor no es lo suficientemente potente para atravesar una distancia requerida sin una pérdida significativa de los datos; los módems también pueden ser usados para amplificar señales. También se pueden utilizar los módems cuando más de una comunicación está ocurriendo en el mismo medio. En este caso, los módems pueden ser configurados para usar diferentes bandas de frecuencia electromagnética.

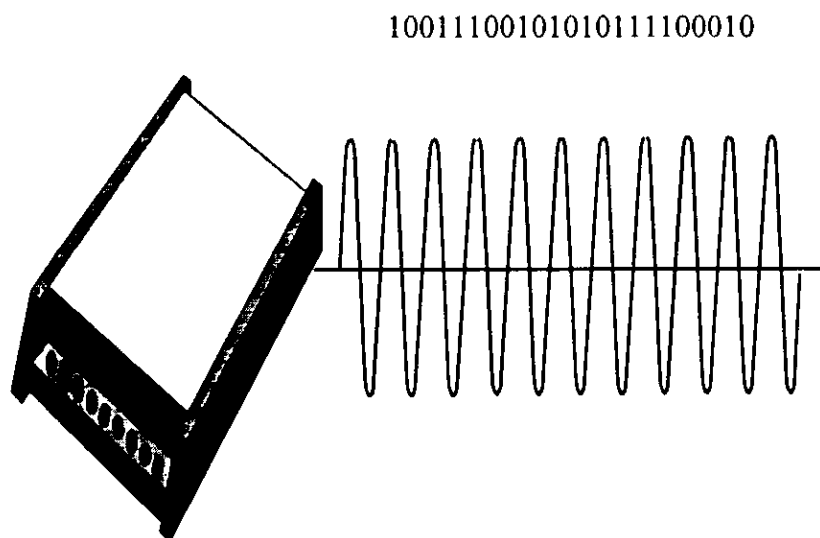


Figura 1.6.2.9.3 Módem.

Repetidores

Como se ha mencionado anteriormente, las ondas electromagnéticas se debilitan (atenúan) cuando pasan a través de un medio de transmisión. Cada medio de transmisión puede ser utilizado para una cierta distancia. Sin embargo, se puede

exceder la distancia efectiva máxima del medio físico utilizando un dispositivo de amplificación llamado repetidor, como se muestra en la figura 1.6.2.9.4.

Un tipo de repetidor amplifica todas las señales electromagnéticas que llegan incluyendo el ruido. Otro tipo, conocido como repetidor regenerador de señal, saca los datos de la señal de transmisión. Este reconstruye la señal y la retransmite hacia el otro segmento del medio. La nueva señal es una copia exacta de la original, aumentada a su potencia original.

Concentradores

Algunas redes requieren de un punto central de conexión entre segmentos del medio de transmisión. Estos puntos centrales son conocidos como hubs, repetidores multipuerto o concentradores. Los concentradores son repetidores especiales que superan las limitaciones electromecánicas de un medio de vía sencilla. En la figura 1.6.2.9.5 se muestra un ejemplo de una instalación de un concentrador.

El concentrador organiza los cables y transmite las señales que llegan hacia otros segmentos. Enseguida se muestran tres tipos de concentradores:

- **Concentrador Pasivo.** Un concentrador pasivo conecta entre sí varios segmentos, nada más. No se realiza regeneración de señal, así cada segmento está limitado a extenderse sólo a la mitad de su distancia efectiva máxima. Además, cada computadora recibe las señales enviadas por todas las demás computadoras.
- **Concentrador activo.** Un concentrador activo es como un concentrador pasivo pero este sí regenera o amplifica la señal. Los concentradores activos son por lo tanto participantes activos en la generación de señales, las cuales se extienden a la máxima longitud del cable. La principal desventaja es que algunos concentradores activos además de amplificar la señal también

amplifican el ruido del cable. Todas las computadoras conectadas a concentradores activos reciben las señales de todas las demás computadoras.

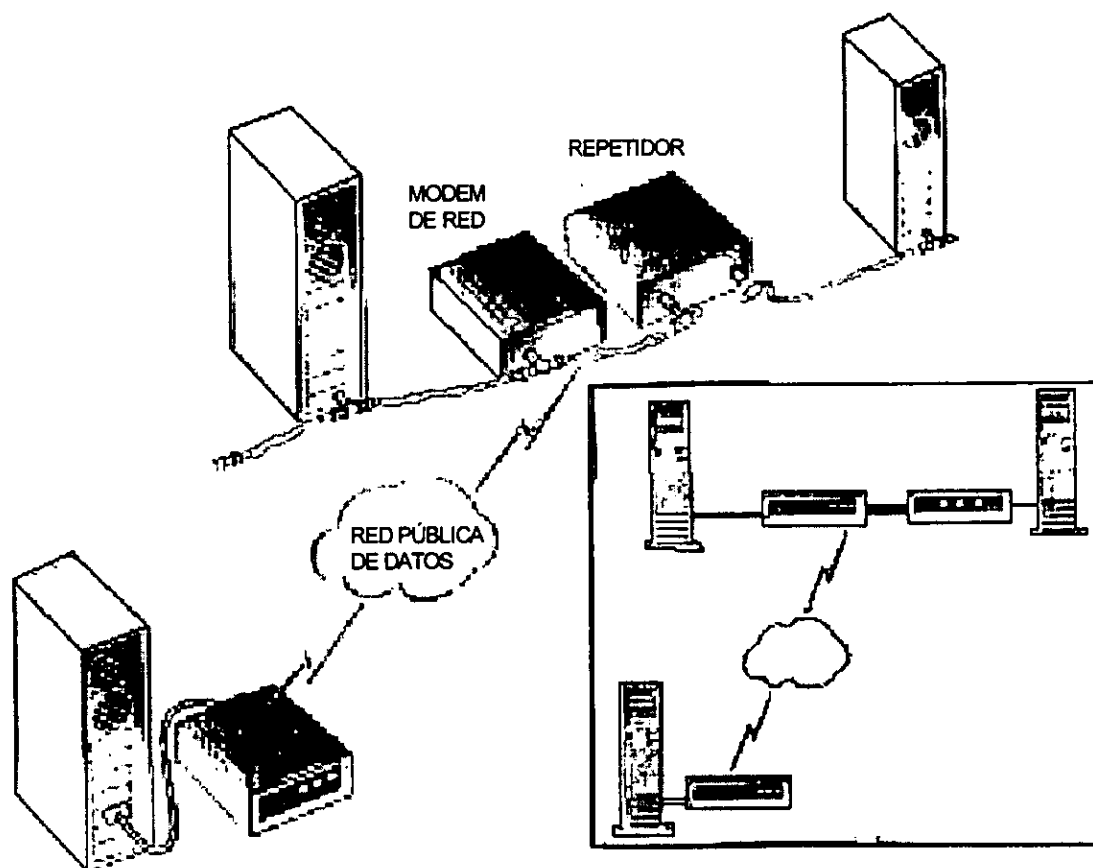


Figura 1.6.2.9.4 Instalación de un repetidor.

- **Concentrador inteligente.** Los concentradores inteligentes (incluyendo los concentradores de switcheo), además de regenerar la señal y manejar la red, también realizan actividades tales como la selección inteligente de la ruta, y pueden escoger diferentes rutas para la entrega de la señal, esto es que también hacen funciones básicas de ruteo. La capacidad de switcheo significa que se puede establecer una red donde todos los segmentos de medios de transmisión están permanentemente conectados, pero cada segmento es utilizado sólo cuando una señal está direccionada a una computadora en ese segmento.

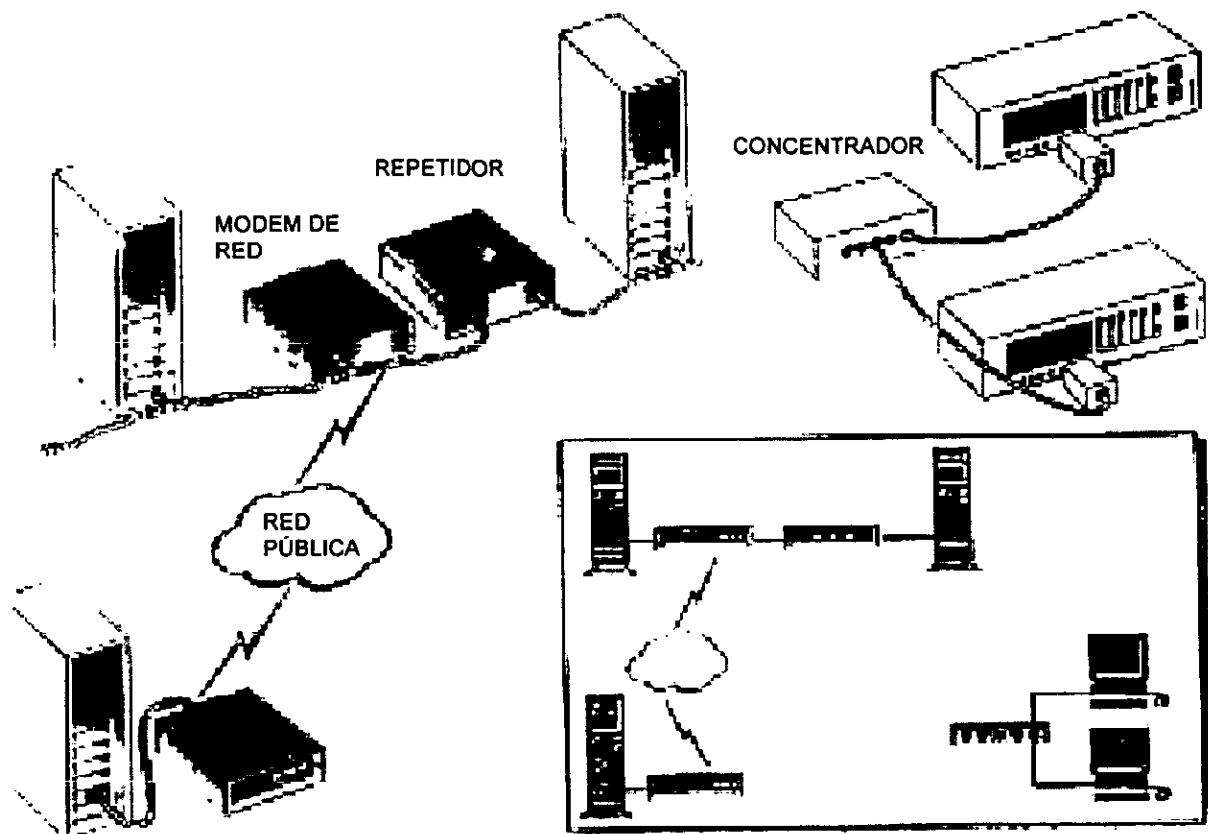


Figura 1.6.2.9.5 Instalación de un concentrador.

Puentes

Un puente extiende a la máxima distancia una red mediante la conexión de segmentos de red separados (los cuales son longitudes o ramas de medios de transmisión). Los puentes pasan señales selectivamente de un segmento de red a otro, como se puede observar en la figura 1.6.2.9.6. Enseguida se muestran las acciones que debe realizar un puente:

- Recibir todas las señales enviadas de un segmento de red.
- Descartar las señales que están direccionadas a otros nodos en ese mismo segmento de red (filtración).

- Retransmitir todas las demás señales a través del puerto apropiado.
- Realizar las mismas funciones con los datos de otros segmentos conectados.

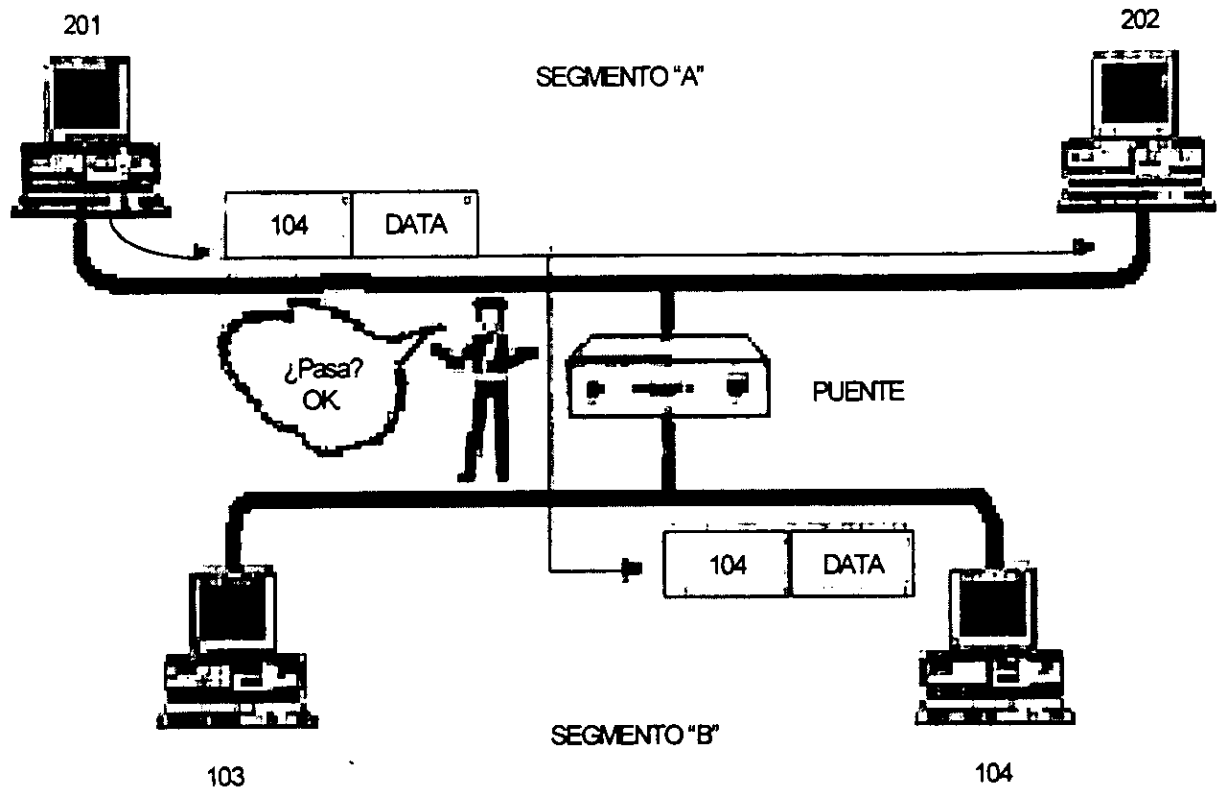


Figura 1.6.2.9.6 Operación de un puente.

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

Debido a que los puentes pueden filtrar señales mediante la dirección, se pueden utilizar para separar una red sobrecargada en segmentos separados. El puente previene el tráfico intersegmento que llega de otros segmentos. Mientras el tráfico intersegmento no sea muy pesado, esta estrategia efectivamente reduce el tráfico de la red.

Multiplexores

Ocasionalmente se utilizarán medios de transmisión que proporcionen más capacidad de la que puede ocupar una señal sencilla. Para utilizar eficientemente el ancho de banda completo del medio de transmisión, se pueden instalar multiplexores. Un multiplexor combina dos o más señales separadas en un segmento de medio de transmisión.

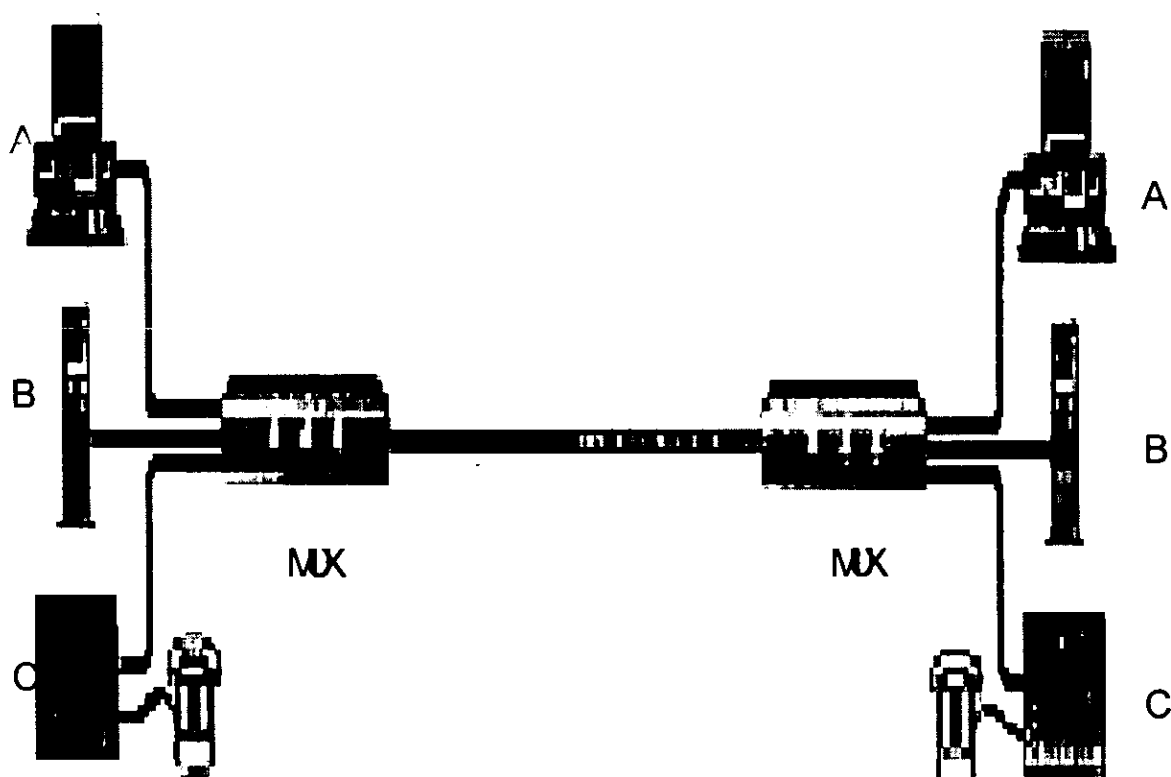


Figura 1.6.2.9.7 Operación de un multiplexor (MUX).

Hardware de conectividad inter-red.

En ocasiones, se necesita conectar dos redes individuales sin que pierdan sus identidades propias. Para esto se requieren dispositivos de conectividad inter-red. Los siguientes dispositivos conectan distintas redes al tiempo que protegen su individualidad:

- Routers
- Brouters
- CSU/DSU

Routers

Los routers, o ruteadores conectan dos o más redes separadas lógicamente. Las subdivisiones lógicas de una red a menudo son llamadas subredes. Una subred puede o no mapear directamente a un segmento físico sencillo, pero siempre representa una red discreta (y por lo tanto lógicamente separada).

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

Como se puede observar, a excepción de mantener la identidad de la red intacta, los ruteadores realizan una función muy similar a la de un puente. Típicamente, los ruteadores tienen un procesamiento más intenso que los puentes. Como resultado, sus velocidades de procesamiento (generalmente medidos en paquetes - o bloques de datos - enviados por segundo) no son tan altas. Por otro lado, los ruteadores tienen una capacidad mucho más sofisticada de selección de rutas. La decisión de adquirir un puente o un ruteador depende de los requerimientos específicos y el ambiente específico de la red. En la figura 1.6.2.9.8 se muestra un ejemplo del funcionamiento de este dispositivo.

Brouters

Muchos ruteadores en realidad son brouters. Los brouters son esencialmente ruteadores que también pueden puentear. Un brouter checa primero si puede soportar el protocolo que está siendo utilizado por el paquete. De lo contrario, en vez de

simplemente suprimir el paquete, el paquete es puentado usando información de direccionamiento físico.

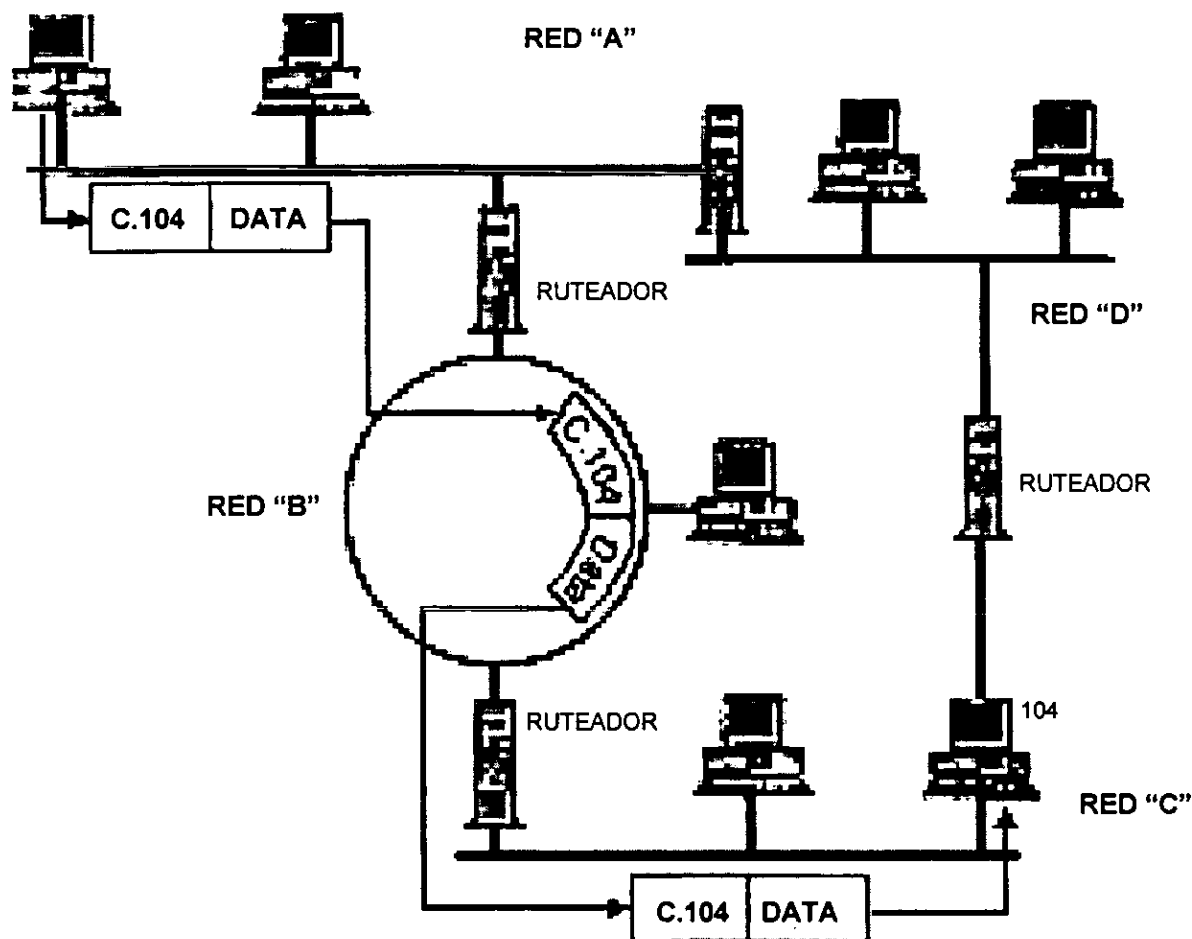


Figura 1.6.2.9.8 Operación de un ruteador.

Unidad de Servicio de Canal/Unidad de Servicio Digital

En ocasiones se tienen muchas ventajas al utilizar las redes públicas. Para conectarse a estas redes se requiere utilizar unidades de servicio de canal y unidades de servicio digitales (CSU/DSU - Channel Service Unit/Digital Service Unit). Los CSU/DSU son dispositivos que preparan los pulsos eléctricos para transmisión en medios WAN. Estos dispositivos aseguran que la señal utiliza la potencia y formato correctos. Estas

unidades protegen tanto al que transmite como a otros usuarios de la red pública de ruidos eléctricos o voltajes peligrosos. Además, preparan los datos para ser transmitidos de acuerdo a las reglas especificadas por la red.

Switch Digital

Una estrategia para construir redes de área amplia es utilizar un proveedor de servicios de interconexión o crear una infraestructura de conmutación de paquetes (Switches), con tecnologías como X.25, FrameRelay y actualmente con ATM (Asynchronous Transfer Mode; Modo de Transferencia Asíncrona) de conmutación de celdas de longitud fija de 53 bytes.

En este esquema todos los ruteadores funcionan como equipos de acceso a una red conmutada conocida como nube, figura; con este esquema no es necesario tener un enlace de cada una de las localidades hacia un punto central, en este caso todos los sitios se conectan al proveedor de servicio. En este bosquejo se tiene la ventaja de que se puede utilizar la infraestructura de telecomunicaciones para transmitir otro tipo de información dentro de la misma red.

Las LANs en las organizaciones de tamaño medio-grande están construidas alrededor de dos tipos de plataformas de red diferenciadas. Los conmutadores de capa 2 proporcionan ancho de banda a bajo costo. Los routers (de capa 3) implementan control de red, pero a un costo mucho mayor.

El siguiente paso en la evolución de las redes LAN reemplaza el router tradicional con una plataforma de gran ancho de banda soportando funciones avanzadas de control de red, es el conmutador (Switch) de capa 3.

La conmutación de capa 3 consiste esencialmente en introducir características propias hasta hoy de los routers en los conmutadores o switches. Efectúa lo que hasta ahora hacía un router, pero de forma más rápida, a menor costo, y sin llegar a perder las

funcionalidades de control asociadas a él. En definitiva, aunque se consigue la misma velocidad que con la conmutación de nivel 2, se añade un mayor control al incorporar al conmutador las prestaciones de los routers.

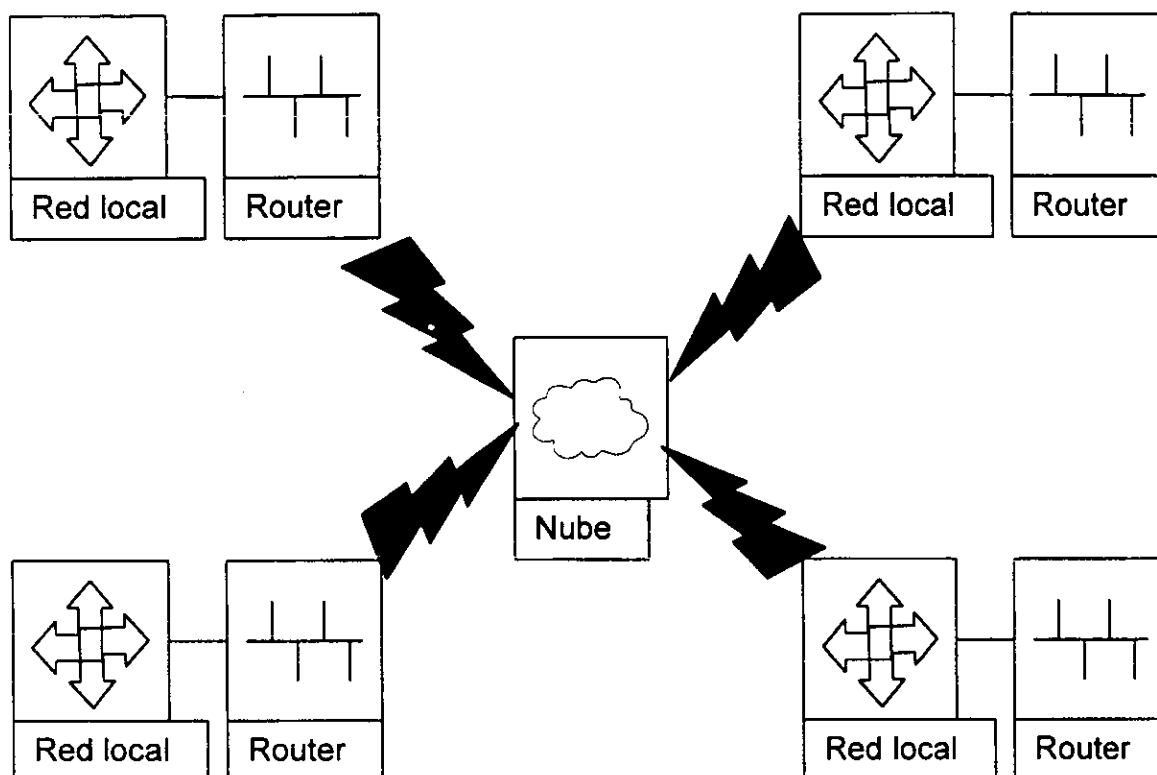


Figura 1.6.2.9.9 Esquema de conexión conmutada (switch)

Hacia un ecosistema conmutado

Existen varios tipos de plataformas de conmutación que soportan diferente grado de control de red y que tienen, por lo tanto, un costo asociado diferente:

- **Plataforma de Capa 2:** En esta plataforma los conmutadores toman decisiones de envío basándose en la dirección física o MAC de destino contenida en cada paquete. Al contrario que el bridge (puente), el switch puede reenviar los datos con periodos de latencia muy bajos. Con esta tecnología conseguimos

incrementar el ancho de banda tanto en segmentos de LAN compartidas como dedicados, ya que segmenta la red en dominios de colisión más pequeños proporcionando mayor porcentaje de ancho de banda para cada estación, y elimina los cuellos de botella entre varias LAN. El uso de ASIC (Circuitos Integrados Específicos de Aplicaciones) permite a los conmutadores enviar paquetes de forma simultánea a través de todos los puertos, proporcionando así un nivel de prestaciones muy superior al de los bridges tradicionales. Existen dos tipos de conmutadores de capa 2 según el modo de reenvío: (1) Cut-Thorough: los conmutadores inician el proceso de reenvío antes de haber recibido la trama completada (no comprueba la integridad de la trama) y (2) Store-and-forward: que leen y validan el paquete entero, incrementando la latencia proporcionalmente al tamaño del paquete, pero ganando control de flujo. En definitiva, los entornos más beneficiados son aquellas redes LAN complejas que en la actualidad utilizan routers para su segmentación. En este caso el conmutador sustituye con ventaja al router por su bajo tiempo de latencia y su inferior costo por puerto.

- **Plataforma de Capa 2/3:** El diseño de red se basa en la creación de redes LANs virtuales (VLANs), para minimizar la necesidad del routing. En esta plataforma conviven conmutadores y routers. Una red virtual no es tanto una tecnología desde el punto de vista físico, como un método o una fórmula de concepción de las redes que las hace más seguras, más fáciles de gestionar y más aptas para el suministro de aplicaciones. Las redes virtuales ocultan al usuario la complejidad de la red y permiten a su administrador centrarse en las aplicaciones y los servicios que proporciona, y no en el cableado y los equipos.
- **Plataforma de Capa 3 Cut-Through:** Combina conmutación y routing en un solo equipo y separa el envío de paquetes y el proceso de ruteo. La operación comienza cuando el primer paquete de una transmisión es enviado al router para la autenticación. El conmutador aprende la dirección de ese primer paquete y, gracias a las nuevas capacidades que se incorporan en él, puede controlar que

los siguientes paquetes de esa transmisión lleguen a su punto de destino. En esta plataforma Cisco propone NetFlow Switching (basado en CiscoFusion) y 3Com propone Fast IP basado en Dynamic Access. En esta plataforma también pueden crearse VLANs.

- **Plataforma de Conmutación de capa 3:** En esta plataforma el conmutador lee la dirección de destino de la capa 3 (en redes TCP/IP la dirección de red o dirección IP). Para conseguir estas prestaciones se utilizan Circuitos Integrados Específicos de Aplicaciones o ASICs, en los que se integra la tecnología de ruteo. Junto con el control y la velocidad, las principales ventajas de la conmutación de capa 3 son su mayor rendimiento, simplicidad, seguridad, flexibilidad y su menor costo. Además se complementan con otro tipo de funcionalidades, por el momento en proceso de estandarización: VLANs, calidad de servicio con la que cuentan ya las redes de celdas. En un futuro cercano el tráfico multimedia podrá ser discriminado por los conmutadores asegurando la calidad de servicio; en la actualidad existen soluciones propietarias de servicios basados en reglas, como la tecnología PACE que soporta FIRE de 3Com y que da prioridad en las colas a los paquetes designados con latencia baja o de alta prioridad.

Cuando sólo existían los conmutadores de capa 2, los campos de acción estaban claros: los conmutadores parecían diseñados especialmente para proporcionar más ancho de banda y los routers proporcionaban seguridad y funciones de gestión de red, permitiendo aislar el tráfico para facilitar el diagnóstico de problemas, efectuaban reenvío inteligente de paquetes ya que operaba en la capa 3 y proporcionaban acceso a redes WAN a un costo limitado.

Con la llegada de los conmutadores de capa 3 la única ventaja del router frente al conmutador es el acceso a redes WAN, ventaja que sin duda desaparecerá en breve haciendo posible la conectividad WAN mediante conmutación. Realmente la aparición de los conmutadores de capa 3 hay que entenderlo como un paso evolutivo del router

hacia la conmutación, es decir, hacia la alta velocidad manteniendo todas las características de gestión y control del router.

La conmutación de nivel (layer 3 switching en inglés) consiste esencialmente en introducir características propias hasta hoy de los routers en los conmutadores o switches. El switch de nivel 3 realiza lo que hasta ahora hacía un router, pero de forma más rápida, a menor costo y no llega a perder las funcionalidades de control asociadas a él. Por tanto, se consigue la misma velocidad que en la conmutación de nivel 2, y se añade un mayor control al incorporar las funciones del router al switch.

El switch lee la dirección de las tramas y solo reenvía los paquetes a los nodos correspondientes (a diferencia del hub, que lo retransmite a todo el mundo) liberando por tanto a la red de este tráfico. Por tanto, si el switch sólo lee la dirección física es de nivel 2, en cambio si además es capaz de interpretar la dirección lógica, estaremos ante un switch de nivel 3.

Arquitectura de los switches

Los switches de nivel 3 son posibles gracias al diseño de los ASICs, cuyo desarrollo ha evolucionado bastante para que se pueda integrar en ellos la tecnología de enrutamiento. Por tanto actualmente todos los fabricantes tienden a utilizar ASICs, como se podrá observar en las soluciones aportadas por algunas de las más importantes marcas como pueden ser:

3Com: Cuya estrategia llama TranscendWare. 3Com para permitir escalar el rendimiento, ha diseñado un nuevo ASIC, llamado FIRE (Flexible Intelligent Routing Engine), el cual realiza las funciones de conmutación de nivel 3, y a su vez puede realizar conmutación a nivel 2, y posee funciones de calidad y servicio, VLANs o Fast IP, características integradas en el Corebuilder 3500, y en el posterior Corebuilder 9000, que combina la velocidad de los switches con el control e inteligencia de los routers.

Cisco: En el caso de Cisco su estrategia es llamada Netflow Switching, la cual combina la conmutación y el enrutamiento en una sola plataforma. Tiene origen en la arquitectura CiscoFusion que separa el envío de paquetes y el proceso de enrutamiento. Esta arquitectura permite que cuando el primer paquete de una transmisión es enviado al router para su autenticación, el switch pueda aprender la dirección de ese primer paquete y pueda controlar que los siguientes paquetes de la transmisión lleguen a su destino.

Conmutación de Nivel 3 vs Router

Además del control y la velocidad, las principales ventajas de la conmutación son su mayor rendimiento, simplicidad, seguridad, flexibilidad y menor costo, por tanto las ventajas son varias, ya que el switch de nivel 3 siempre trabajará como un switch de nivel 2, y si se necesita realizará funciones de router de una LAN a otra.

El mayor inconveniente de la solución de conmutación de nivel 3 es la latencia, en este caso, el switch, tardará más tiempo en determinar donde enviar cada paquete ya que ahora deben entender el protocolo que se está utilizando.

Por tanto se prevé que en un futuro no muy lejano sustituirán a los routers de backbone en entornos LAN, pero esta tecnología en el campo WAN tiene todavía un acceso limitado, pero no se puede descartar que en un futuro se utilice para el campo WAN. Por tanto los entornos más beneficiados serán aquellas redes complejas que en la actualidad utilizan routers para su segmentación, ya que en este caso el switch sustituye con ventaja al router por su bajo tiempo de latencia (hasta 10 veces menor en algunos casos) y con muy inferior costo por puerto.

Otro entorno beneficiado será el de las redes con tráfico multimedia, ya que los nuevos switches serán capaces de diferenciar el tipo de tráfico multicast y filtrarlo. Como se ha indicado anteriormente, la conmutación de nivel 3, realiza un routing a la misma velocidad que llega la información a todos los puertos, esto posibilita por tanto que la

máquina pueda controlar todo ya que trabaja a nivel 3, al tiempo que va tan rápido como una de nivel 2. Además de estas características, el switch de nivel 3 se complementa con otro tipo de funcionalidades, que están en proceso de estandarización, como VLANs o calidad de servicio con las que cuentan ya las redes de celdas. Y por tanto, es aquí donde se ve realmente la solución presentada por la conmutación de nivel 3 respecto al tráfico multimedia, realizando una discriminación de tráfico en la estación de trabajo, y indicando la prioridad.

Por tanto la conmutación de nivel 3, no se sitúa frente a la conmutación de nivel 2 sino que se la compara con el típico router de backbone, que corresponde a un dispositivo (según los fabricantes) caro, lento y insustituible, hasta este momento, ya que realiza labores de control. Por tanto, según esto, no es un final de la conmutación de nivel 2, sino a una extensión de ésta, ya que seguirá siendo utilizada en entornos en los cuales el control no es necesario.

Realmente, los switches de nivel 2 nacieron en un momento en el que el router era una máquina cara y lenta, y las redes estaban concentradas en él, y es en este momento donde, el switch de nivel 2 proporcionó una mayor velocidad al proceso. Pero actualmente, debido a las necesidades que surgen, se requiere que a esa velocidad proporcionada por el switch de nivel 2 se le añada control, y es por ello el desarrollo de los switches de nivel 3.

Como se ha visto anteriormente, las características propias de los routers son introducidas en los switches, y por tanto un switch de Nivel 3 realiza todas las características que hasta ahora realizaba un router, pero de una forma más rápida, teniendo un menor costo y con iguales funcionalidades de control.

1.6.3 Comunicaciones

En la actualidad las comunicaciones son de una importancia vital, puesto que es a través de los diferentes medios utilizados para comunicarse que se facilitan las tareas de interconexión, manipulación y acceso a sistemas de información. En este sentido las telecomunicaciones juegan un papel importante, ya que es a través de sus medios de transmisión que podemos tener acceso a equipos, dispositivos e información que se encuentran en diferentes puntos geográficamente distantes y con gran rapidez y confiabilidad.

1.6.3.1 Modelos de Interconexión

Podemos entender como modelos de interconexión a aquellos que tienen la capacidad de que diferentes sistemas de computadoras, redes, sistemas operativos y aplicaciones puedan trabajar juntos. Existen diferentes modelos de interconexión, y no por que dos sistemas se conecten y puedan compartir información, significa que se pueda acceder a esa información desde cualquier aplicación. Esto se da por el hecho de que existe una gran variedad de equipos, sistemas operativos, protocolos de comunicación, aplicaciones, medios de transmisión, etcétera; Y eso requiere que se tengan interfaces tanto físicas como lógicas para poder lograr que dichos sistemas puedan acceder a la información aun cuando procedan de entornos diferentes.

La utilización de sistemas abiertos han provocado que surjan diferentes modelos de interconexión para los mismos. A continuación describiremos brevemente algunos de los modelos más comunes.

- **Modelo de interconexión de sistemas abiertos (OSI).** Este modelo define una serie de niveles para un entorno de sistemas abiertos, donde un proceso que se ejecuta en una computadora puede comunicarse con un proceso similar en otra computadora, si tienen implementado el mismo protocolo de comunicación de niveles OSI.

- Modelo de objetos del sistema (SOM) de IBM. Esta es una técnica orientada a objetos que tiene como finalidad la mejora de proyectos de programación procedimental ya existentes con la inclusión de técnicas orientadas a objetos. Es además independiente del lenguaje, ya que permite el trabajo conjunto de clases escritas en diferentes lenguajes siempre y cuando cumplan ciertas ligas.
- Modelos de objetos en sistemas distribuidos (DSOM). Este modelo es una variante del SOM y permite a los objetos se puedan comunicar a través de plataformas en ambientes de informática distribuida.
- Modelo genérico de objetos (COM). Consiste en un conjunto de interfaces de red distribuidas diseñadas para permitir que las aplicaciones Windows tengan acceso a aplicaciones y objetos que se encuentran sobre otras plataformas.

1.6.3.2 Modelo OSI

En este punto tratamos más ampliamente el modelo OSI ya que es un modelo aceptado por la organización internacional de estandarización o normalización (ISO). Este modelo consta de siete niveles, los cuales pueden ser descritos como sistemas conectados y comunicados unos con otros. Es utilizado mundialmente como un punto de referencia para el diseño y la construcción de sistemas interoperables. El nivel inferior define los componentes físicos reales tales como conectores, cables y la transmisión eléctrica de los bits entre los sistemas. El siguiente nivel define los métodos de empaquetamiento y direccionamiento de datos. A continuación se encuentran los métodos para mantener activas las sesiones de comunicación y, finalmente los niveles superiores nos describen como las aplicaciones utilizan los sistemas de comunicaciones para interactuar con las aplicaciones de otros sistemas.

Cada nivel define los procedimientos y las reglas que los subsistemas de comunicación deben seguir para poder comunicarse con sus procesos pares de los otros sistemas. En una sesión de comunicación real, cada nivel de la pila de protocolos se comunica con

su nivel par en otro sistema; para lo que añade la información que necesita comunicar en un paquete que se pasa al siguiente nivel inferior del protocolo. Como se muestra en la figura 1.6.3.2.1.

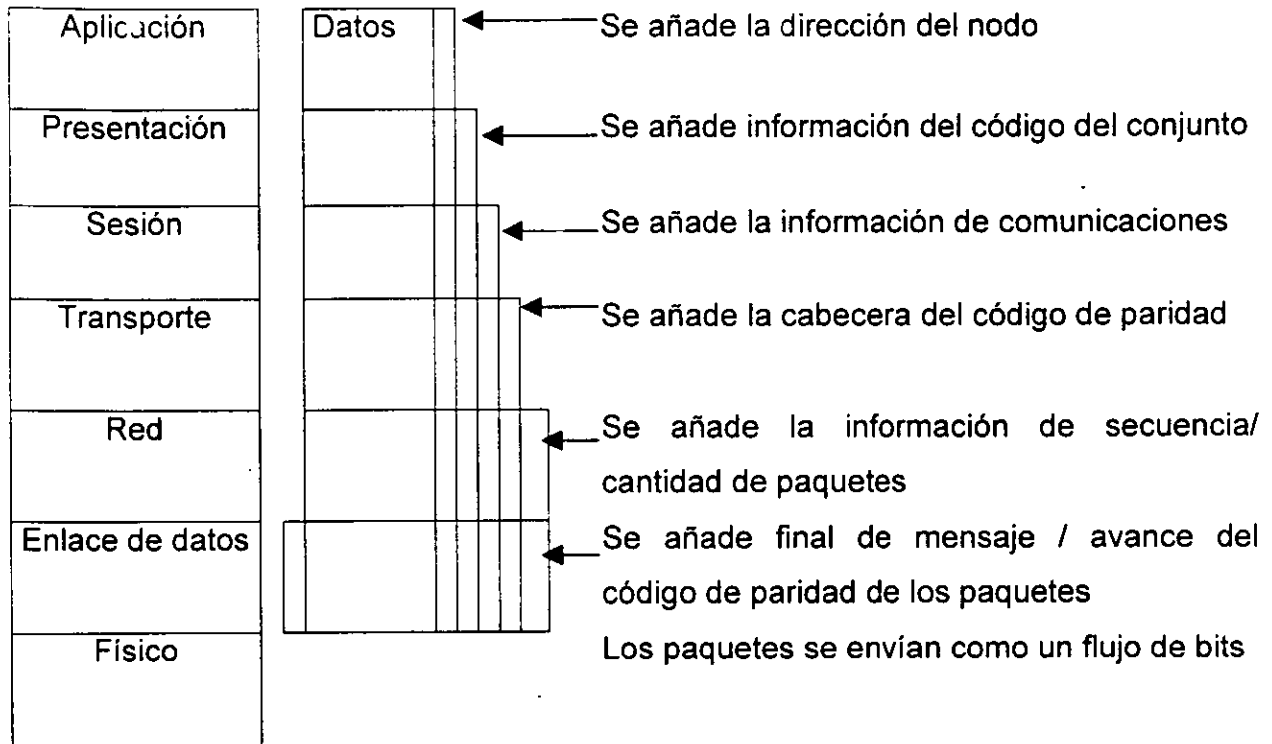


Figura 1.6.3.2.1 Capas o niveles del modelo OSI

A continuación se describe cada uno de los siete niveles o capas del modelo OSI. Así como algunos de los protocolos que utilizan.

Nivel físico. Este define las características físicas de la interfaz como pueden ser los componentes y conectores mecánicos, los aspectos eléctricos tales como los valores binarios que representan los niveles de tensión y los aspectos funcionales entre los que se incluyen el establecimiento, mantenimiento y liberación del enlace físico. Las interfaces del nivel físico más conocidas son las RS-232 y RS-449, los sistemas de red más conocidas son las LAN y ethernet y anillo así como la interfaz de datos distribuidos (FDDI).

Nivel de enlace de datos. Aquí se definen las reglas para el envío y recepción de la información a través de la conexión física entre dos sistemas. Además de ofrecer corrección y detección de errores, este nivel codifica y sitúa los datos en tramas para la transmisión. A pesar que aquí se tiene detección de errores, el rendimiento mejora si esta detección se realiza en niveles superiores. Es en este nivel de la pila de protocolos donde operan los puentes. Enseguida listamos los protocolos genéricos que se colocan en el nivel de enlace de datos.

- Control de enlace de datos de alto nivel (HDLC).
- Controladores de LAN y métodos de acceso como ethernet y anillo con testigo.
- Redes de área extensa de paquetes rápidos como Frame Relay y modo de transferencia asíncrono (ATM).
- Especificación de la interfaz del controlador de la red (NDIS) de Microsoft.
- Interfaz abierta de enlace de datos (ODI) de Novell.

Nivel de red. Este nivel define los protocolos para abrir y mantener un camino sobre la red entre los sistemas. Esta relacionado con los procedimientos de la transmisión y conmutación de datos, ocultando dichos procedimientos a los niveles superiores. Los Ruteadores operan en este nivel ya que toma las direcciones del paquete para determinar los métodos de ruteo. Si se direcciona un paquete a una estación de la red local, este es enviado directamente ahí. Si es direccionado a una red con un segmento diferente, el paquete es enviado al dispositivo de ruteo para que lo envíe a la red correspondiente. Los protocolos que ocupan este nivel son:

- Protocolo Internet (IP).
- Protocolo X.25.

- Intercambio de paquetes entre redes (IPX).
- Protocolo internet vines (VIP) de Banyan.

Nivel de transporte. El nivel de transporte proporciona un alto nivel de control para trasladar la información entre sistemas, por lo que se incluyen las utilidades más sofisticadas de gestión de errores, prioridades y seguridad. Se ofrecen servicios de calidad y distribución segura mediante el uso de servicios orientados a la conexión entre los dos sistemas finales. Controla secuencia de paquetes, regulando el flujo de tráfico y reconociendo la duplicidad de paquetes. El nivel de transporte asigna a la información empaquetada un número de seguimiento que se controla en el destino, esto es, si el dato desaparece del paquete, el protocolo del nivel de transporte del sistema receptor acuerda con el del transmisor que tiene paquetes a retransmitir. Con lo que aquí se asegura que se reciben todos los datos y en el orden adecuado. Existen algunos protocolos que proporcionan servicios orientados a la conexión y no son OSI, aquí se listan algunos.

- Protocolo de control de transmisión (TCP) de Internet.
- Protocolo de datagramas de usuario (UDP) de Internet.
- Intercambio secuencial de paquetes (SPX) de Novell.
- Protocolo de comunicación entre procesos VINES (VICP) de Banyan
- NetBIOS/netBEUI de Microsoft.

Nivel de sesión. Este nivel coordina el intercambio de información a través de técnicas de conversación o diálogos. Estos últimos no son muy necesarios, pero algunas aplicaciones necesitarán saber donde reiniciarán una transmisión de datos si se perdió

temporalmente una conexión, o para saber donde se termina un conjunto de datos e inicia uno nuevo.

Nivel de presentación. Estos protocolos de este nivel forman parte del sistema operativo y de la aplicación que el usuario ejecuta en su estación de trabajo. Es aquí donde se le da formato a la información para visualizarla o imprimirla. Los códigos son interpretados dentro de los datos, por ejemplo las tabulaciones o gráficos especiales. De igual forma se gestiona el cifrado de datos y la traducción de otro conjunto de caracteres.

Nivel de aplicación. El nivel de aplicación es utilizado para definir una serie de aplicaciones que gestionan transferencia de archivos, sesiones de terminales e intercambio de mensajes. Los protocolos en este nivel son los siguientes:

- Terminal Virtual.
- Acceso y gestión de la transferencia de archivos (FTAM).
- Procesamiento de transferencia distribuida(DTP).
- Sistema de gestión de mensajes (X.400).
- Servicios de directorio (X.500).

En la figura 1.6.3.2.2 se muestra como fluyen los datos a través de una pila de protocolos y sobre el medio desde un sistema a otro. Los datos comienzan en los niveles de aplicación y presentación, donde un usuario trabaja con una aplicación de red, como el correo electrónico¹ por ejemplo. Las peticiones de servicio del nivel de aplicación se pasan al nivel de sesión, el cual comienza el proceso de empaquetado de la información. Entre ambos sistemas se abre una sesión de comunicación orientada a conexión para ofrecer transmisiones fiables. Una vez que se establece la sesión, los niveles de protocolos comienzan el intercambio oportuno de información.

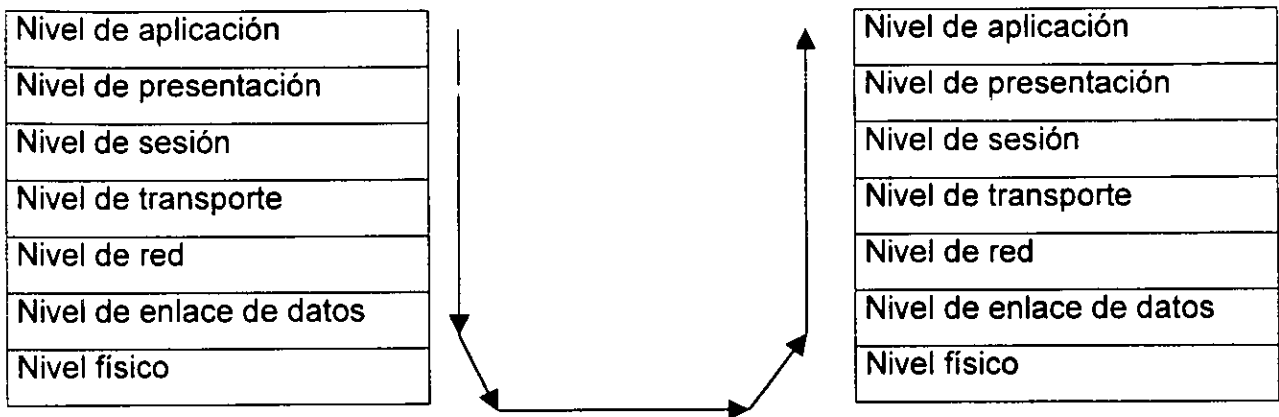
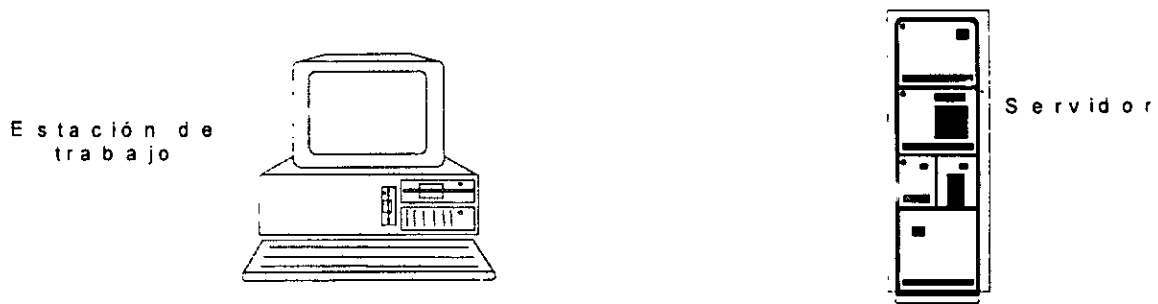


Figura 1.6.3.2.2 Flujo de datos de un sistema a otro en el modelo OSI.

1.6.3.3. Arquitectura TCP/IP (Protocolo de control de transmisión / internet protocolo)

A la arquitectura de TCP/IP es común que se le confunda con la arquitectura de internet, esto debido a la estrecha relación existente entre ambas. A pesar de que tanto TCP/IP y el modelo OSI se desarrollaron de forma paralela y que no tienen ningún tipo de conflicto ya que ambas se ayudaron mutuamente. Si existen diferencias de importancia que se originan en los requerimientos básicos de TCP/IP, estos requerimientos son los siguientes:

- Un conjunto común de aplicaciones.
- Enrutamiento dinámico.

- Protocolos sin conexión en nivel de red.
- Conectividad universal.
- Conmutación de paquetes.

Las diferencias entre la arquitectura OSI y la de TCP/IP están relacionadas con las capas que se encuentran arriba del nivel de transporte y nivel de red. TCP/IP agrupa en la capa de aplicación a lo que equivale a la capa de sesión y presentación en la arquitectura OSI. Al tener un protocolo sin conexión hizo que TCP/IP combinara los niveles físico y de vinculo de datos de OSI en un nivel de red. En la figura 1.6.3.3.1. se muestra un comparativo de las arquitecturas por niveles o capas tanto del TCP/IP como de OSI.

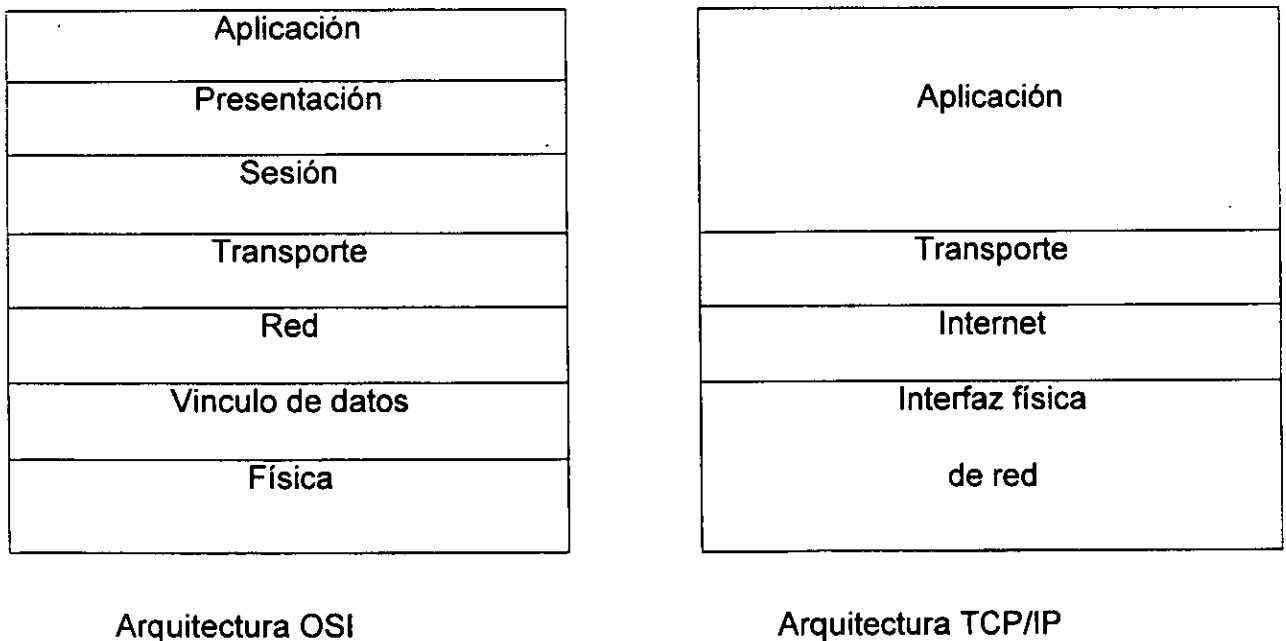


Figura 1.6.3.3.1 Comparativo de las Arquitecturas OSI y TCP/IP.

1.6.3.4 Pasado y presente

El nacimiento de TCP/IP y su arquitectura fueron propuestas en 1974. En donde se describía un sistema que incluía un protocolo de aplicación estandarizado, que también utilizaba confirmación de extremo a extremo. Esto no fue novedoso, lo novedoso aquí, fue la propuesta, de que dicho protocolo fuera independiente de la red y del hardware de computadora.

De igual forma se proponía conectividad universal a través de la red, lo que permitió que cualquier plataforma participara en la red. Para 1981 TCP/IP ya había sustituido al Programa de control de la red (NCP) como protocolo dominante. Y como se observo en el punto de arquitectura de TCP/IP, donde se realizo la comparación con el modelo OSI, al parecer TCP/IP puede ganarle la batalla al modelo OSI, pero eso no lo sabremos sino con el tiempo.

A continuación se listaran una serie de situaciones por las que TCP/IP esta siendo tan popular actualmente.

- TCP/IP está vivo y en funcionamiento y tiene un récord.
- TCP/IP tiene un cuerpo establecido y funcionando.
- Existen miles de aplicaciones que actualmente utilizan TCP/IP, así como sus interfaces de programación de aplicaciones. (API)
- Este es la base de casi todos los sistemas UNIX, que están ocupando gran parte del mercado de los sistemas operativos de red.
- Y algo muy importante TCP/IP es independiente del fabricante, ya que puede ser utilizado por cualquier plataforma

1.6.3.5 Servicios en modo conexión y en modo sin conexión

Las capas pueden ofrecer dos tipos diferentes de servicios a las capas que se encuentran sobre ellas: uno orientado a conexión y otro sin conexión. En este punto veremos ambos tipos y sus diferencias.

El servicio orientado a la conexión es muy similar a un sistema telefónico, es decir, el usuario del servicio establece una conexión, la utiliza y posteriormente termina la conexión. La principal característica es que así como se envían los datos en ese mismo orden se reciben.

De otra forma el servicio sin conexión, es similar al del sistema postal, es decir cada mensaje lleva consigo una dirección y cada uno de ellos se dirige independientemente a su destino. Puede ocurrir que se envíen mensajes al mismo destino, sin embargo en este caso, el primero en ser enviado es el primero en ser recibido, siempre y cuando no haya retrasos.

Cada servicio se caracteriza por la calidad del servicio. La fiabilidad es una característica importante, pero esto depende de que pierdan o no información que transportan. Para esto el servicio debe notificar al emisor que recibió el mensaje para que el emisor este seguro de que el mensaje llego al destino. Este proceso puede ocasionar mucho trafico y retardo, que no siempre es conveniente.

Un ejemplo típico de un servicio orientado a conexión confiable, es el de transferencia de archivos ya que el usuario desea estar seguro de que sus archivos fueron enviados completamente y en orden.

El servicio orientado a conexión tiene dos variantes: secuencia de mensajes y flujo de octetos. En la primera de ellas, se mantienen los límites de mensaje. Es decir, si se envían dos mensajes de 1k se reciben como dos mensajes de 1k y no como uno solo de 2k. En la segunda variante, la conexión es solo un grupo de octetos sin límite de

mensaje. Cuando llegan al receptor 2k de octetos, no es posible saber si fueron 1, 2 o 2048 mensajes de un octeto.

Un servicio sin conexión no es fiable, es conocido como servicio datagrama, y este no proporciona un acuse de recibo de la señal recibida.

En ocasiones no es conveniente establecer una conexión para enviar un mensaje pequeño, pero si es fundamental que el proceso sea fiable. Para este tipo de aplicaciones se proporcionaría el servicio de datagrama con asentimiento de información. Otro servicio alternativo que sea fiable es el servicio de pregunta / respuesta. En la figura 1.6.3 5.1 se muestran los tipos de servicio descritos.

	Servicio	Ejemplo
Orientado a conexión	Flujo de mensaje fiable	Secuencia de paginas
	Flujo de octeto fiable	Conexión remota
Sin conexión	Conexión no fiable	Voz digitalizada
	Datagrama no fiable	Correo electrónico basura
	Datagrama con asentimiento	Correo certificado
	Pregunta - respuesta	Interrogación de base de datos

Figura 1.6.3.5.1 Diferentes tipos de servicio con y sin conexión

1.6.3.6 Direccionamiento

En una red sola, son necesarios varias porciones de la información para asegurar la correcta entrega de los datos. Los principales componentes son la dirección física y la dirección de vínculo de datos.

Cada dispositivo de una red que se comunica con otros tiene una dirección física única, la cual le sirve para ser identificada, esta es conocida como dirección de hardware. Esta dirección es única y normalmente esta cifrada en la tarjeta de interfaz de red y se establece ya sea por interrupciones o por software. En el modelo OSI la dirección se ubica en la capa o nivel físico.

En la capa física se realiza el análisis de cada datagrama que recibe. Si la dirección del receptor coincide con la dirección física del dispositivo el datagrama pasa a las capas superiores, si no se ignora el datagrama. Al mantener este análisis en la capa más inferior se evitan retrasos. La longitud de la dirección física varia dependiendo del sistema de red.

Ethernet y algunos más utilizan direcciones de 48bits. El IEEE es el encargado de asignar direcciones físicas universales para las subredes. Esta dirección se conoce como el identificador único de organización (OUI) y es de 24 bits, lo que permite que los otros 24 bits los utilice la organización para sus direcciones.

En la figura 1.6.3.6.1 se muestra el formato de OUI en donde el primer bit es el de dirección individual o de grupo, el segundo el bit que nos indica si es local o universal y los 22 restantes forman la dirección física de la unidad. Lo anterior nos da 24 que combinados con los 24 de la organización nos da una dirección de 48 bits que se conoce como dirección de control de acceso a medios (MAC). En la misma figura se muestra un ejemplo de una MAC de HP (Hewlett Packard).

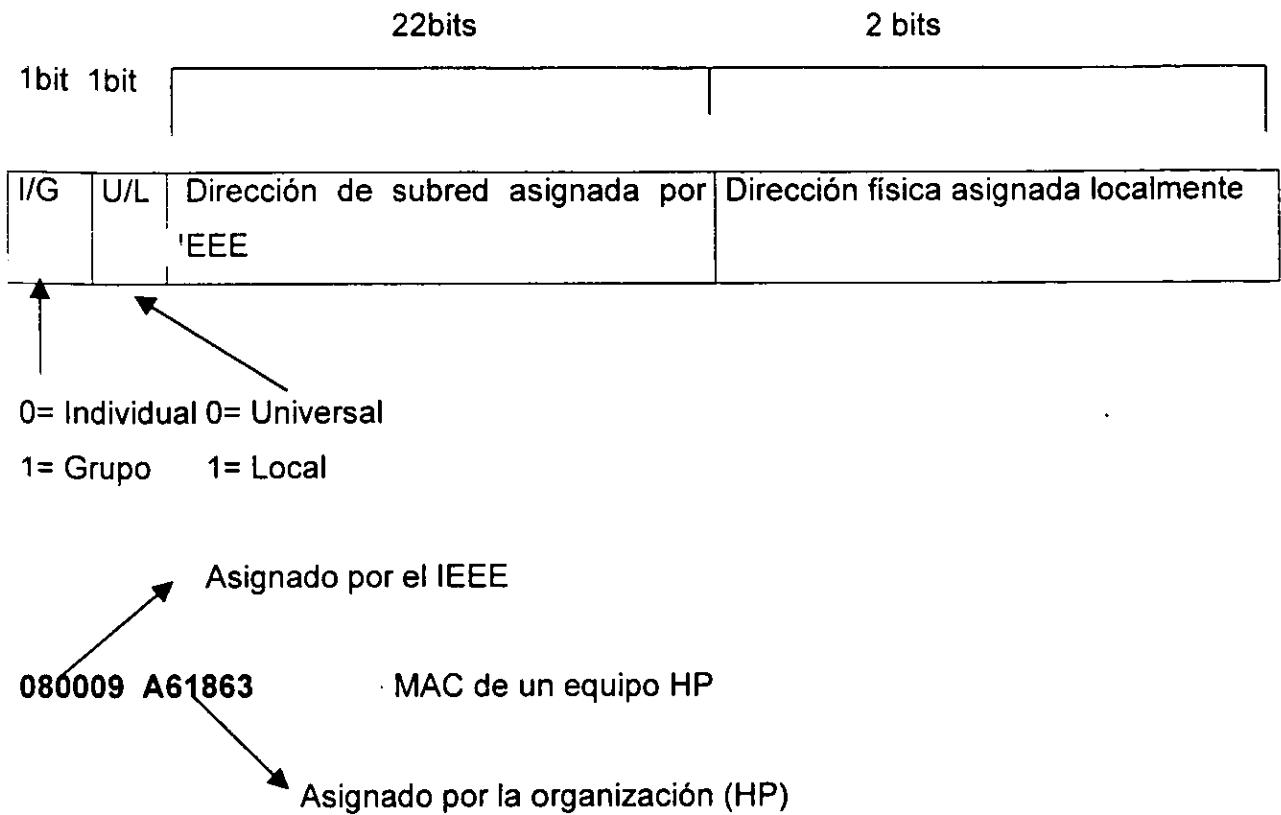


Figura 1.6.3.6.1 Diseño de OUI (identificador único de organización)

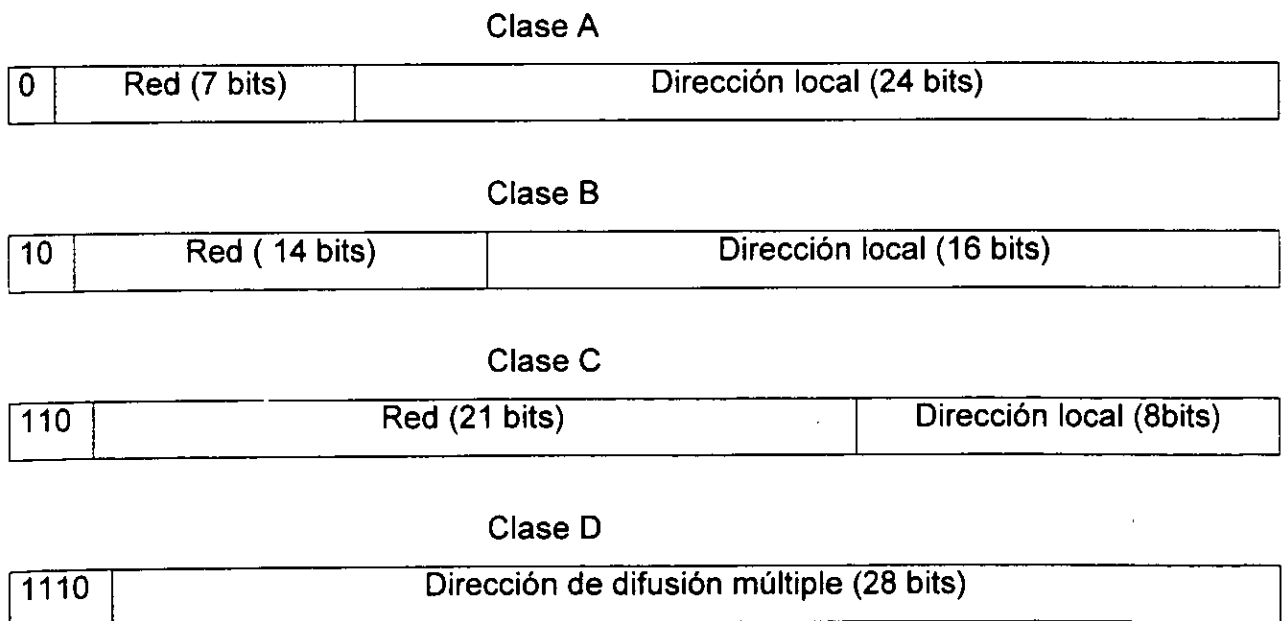


Figura 1.6.3.6 Las cuatro estructuras de dirección IP.

Direcciones IP

TCP/IP utiliza direcciones de 32 bits para identificar una maquina y la red a la cual esta conectada. Las direcciones IP identifican la conexión de la maquina a la red, no a la maquina misma para eso esta la MAC. Entonces siempre que se modifique la localización de la maquina la dirección IP cambia..

Existen cuatro formatos o clases para las direcciones IP y esto depende de su tamaño. Los cuatro formatos son: Clase A, clase B, clase C y clase D y se muestran en la siguiente figura 1.6.3.6.1.

- Clase A. Dirección de local de 24 bits y dirección de red de 7 bits, lo que limita el número de redes identificables.
- Clase B. Dirección local de 16 bits y dirección de red de 14 bits, lo que incrementa el número de redes y decrementa el número de dispositivos por red.
- Clase C. Dirección local de 8bits y 21 bits para la de red, lo que limita el número de dispositivos a 256 por red.
- Clase D. Se usan con fines de multidifusión, cuando se requiere una difusión general a más de un dispositivo.

1.6.3.7 Ruteo

El enrutamiento o ruteo se refiere a la transmisión de un paquete de información de una maquina a través de otra.

Ya se ha expuesto anteriormente la forma en que los datagramas pasan de un ordenador de la red a otro mediante el protocolo IP, sin embargo en esta sección se

comenta con más detalle el proceso que permite que la información llegue hasta su destino final. Esto se conoce con el nombre de **routing (ruteo)**.

Las tareas de routing son implementadas por el protocolo IP sin que los protocolos de un nivel superior tales como TCP o UDP tengan constancia de ello. Cuando se quiere enviar información por Internet o por la red a un ordenador, el protocolo IP comprueba si el ordenador de destino se encuentra en la misma red local que el ordenador origen. Si es así, se enviará el correspondiente datagrama de forma directa: la cabecera IP contendrá el valor de la dirección Internet del ordenador destino, y la cabecera Ethernet contendrá el valor de la dirección de la red Ethernet que corresponde a este mismo ordenador.

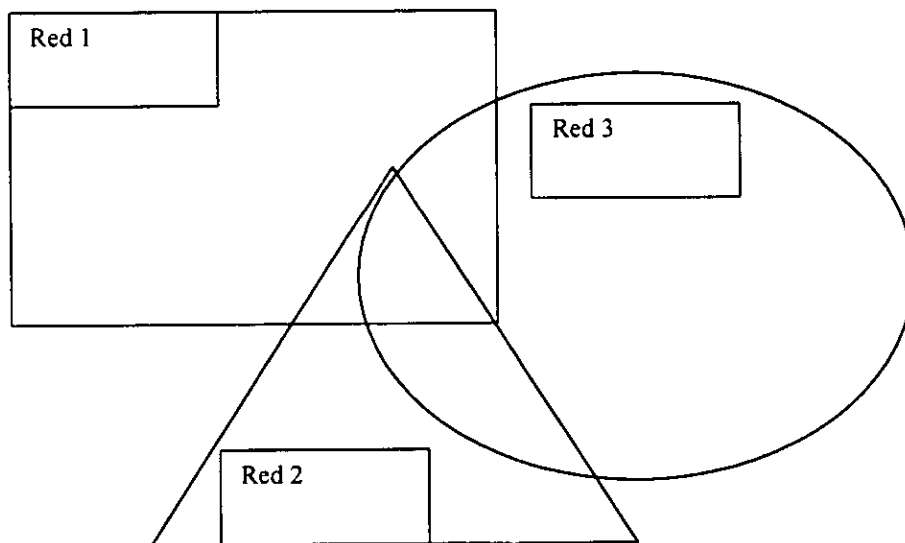


Figura 1.6.3.7.1 Ejemplo de cómo pueden entrelazarse dos o más redes

Cuando se pretende enviar información a un ordenador remoto que está situado en una red local diferente al ordenador de origen, el proceso resulta más complicado. Esto se conoce como routing indirecto, y es el caso que se presenta más frecuentemente cuando se envía información en Internet. La figura 1.6.3.7.1 muestra un ejemplo en el que dos redes locales que utilizan la tecnología de Internet se enlazan para

intercambiar información, creando una red lógica de mayor tamaño gracias a la funcionalidad del protocolo IP.

En Internet existen un elevado número de redes independientes conectadas entre sí mediante el uso de los **routers (ruteadores o encaminadores)**. Un ordenador puede actuar como un router si se conecta a varias redes al mismo tiempo, disponiendo por lo tanto de más de una interfaz de red así como de varias direcciones IP y Ethernet (tantas como redes a las que se encuentre conectado). El router, por supuesto, puede enviar y recibir información de los **hosts (anfitriones, equipo principal)** de todas las redes a las que está conectado, y siempre será de forma directa. Continuando con el ejemplo anterior, el host A puede comunicarse de forma directa con el host B, así como los hosts A y B pueden enviar o recibir información del router. En ambos casos se trata de routing directo, pues el ordenador que actúa como router está conectado a la red 'alfa' de la misma manera que los ordenadores A y B, teniendo una dirección IP propia asignada que lo identifica dentro de esta misma red. La situación es la misma para la red 'omega' donde el router es identificado a través de una segunda dirección IP que corresponde con esta red.

Si sólo fuésemos a enviar información de manera directa dentro de una misma red no sería necesario el uso del protocolo TCP/IP, siendo el mismo especialmente indicado cuando se desea una comunicación con otras redes. En este caso los datagramas tendrán que ser encaminados a través del router para llegar a su destino. La forma de hacer esto es a través del protocolo IP, el cual decide si la información puede enviarse directamente o si por el contrario debe utilizarse el método indirecto a través de un router. Tomamos el ejemplo de la figura 1.6.2.9.8. De la pagina 146.

Suponemos que el host B de la red 'A' necesita comunicarse con el host X situado en la red 'C'. Una vez que se ha determinado que el destino no se encuentra en la misma red, envía el datagrama IP hacia el router correspondiente. Como este router y el ordenador que envía la información se encuentran conectados a la misma red, se trata por tanto de routing directo, ya comentado anteriormente, y por consiguiente sólo será necesario

determinar la dirección Ethernet del router mediante empleo del protocolo ARP (protocolo de resolución de direcciones). El paquete enviado incluirá la dirección del router como dirección Ethernet de destino, pero sin embargo, la dirección de destino IP corresponderá al ordenador final al que va dirigido el paquete, el host (c.104) en el ejemplo. El router recibe el paquete y a través del protocolo IP comprueba que la dirección de Internet de destino no corresponde con ninguna de las asignadas como suyas, procediendo entonces a determinar la localización de la 'omega', en la que se entrega el paquete al ordenador de destino.

Hasta este punto se ha supuesto que sólo existe un único router, pero es bastante probable que una red con conexión a Internet posea múltiples enlaces con otras redes, y por lo tanto más de un router. Entonces... ¿cómo determina el protocolo IP el sistema correcto al que debe dirigirse? Para resolver este problema cada ordenador utiliza una tabla donde se relaciona cada una de las redes existentes con el router que debe usarse para tener acceso.

Debe tenerse en cuenta que los routers indicados en estas tablas pueden no estar conectados directamente a las redes con las que están relacionados, sino que lo que se indica es el mejor camino para acceder a cada una de ellas. Por esta razón, cuando un router recibe un paquete que debe ser encaminado, busca en su propia tabla de redes la entrada correspondiente a la red para, una vez encontrada, entregarlo al ordenador de destino. Es importante notar que en el caso de que el router no tenga conexión directa a la misma red que el ordenador de destino, la búsqueda en su tabla de redes dará como resultado la dirección de un nuevo router al que dirigir el paquete, y así continuará el proceso sucesivamente hasta encontrar el destino final.

A causa de la extensión de Internet, es normal que un paquete atravesase numerosas redes (pueden ser decenas) hasta llegar a su destino. La ruta que tiene que recorrer un paquete en su viaje a través de la red no está determinada inicialmente, sino que es el resultado de la consulta en las tablas de direcciones individuales de los ordenadores intermedios.

Ya se ha mencionado anteriormente que todos los host de Internet necesitan disponer de una tabla de routing con la información de otras redes, pero esto supondría algunos inconvenientes adicionales (como el tamaño y la necesidad de mantenimiento). Con la finalidad de reducir los inconvenientes se utilizan los routers (o gateways) por defecto. De esta manera cuando un host no posee información del camino correcto para un determinado paquete, éste es enviado al router que tiene asignado por defecto. Si este router es el único del que dispone la red no habrá ningún inconveniente y el paquete continuará su camino. Sin embargo, cuando existen varios routers para la misma red puede ocurrir que el utilizado por defecto no sea el más apropiado para el paquete que se quiere enviar, por lo que se necesita algún procedimiento para notificar el error al host que envió el paquete. El protocolo ICMP (Protocolo internet de mensajes de control) es el utilizado para enviar estos mensajes de notificación que informan al host de la ruta correcta, y que en muchos casos éste utiliza para actualizar su propia tabla de routing y que los próximos paquetes con el mismo destino sean dirigidos de forma correcta.

La creación y mantenimiento de la tabla de redes para routing es un proceso complejo que debe ser realizado por el administrador de la red. Aquí hay que tener en cuenta que la enorme extensión de Internet supone una gran dificultad para conseguir que sean correctas todas las entradas de la tabla, además de que esta tabla puede llegar a tener un tamaño considerable.

La utilización de routers por defecto mejora la situación al permitir que sean estos los que guarden el registro de la red sin que los ordenadores individuales tengan que ocuparse en ello, pero estos routers sí que deberían tener una tabla completa. Para facilitar el mantenimiento de la tabla existen algunos protocolos para routing que permiten que un router o gateway cualquiera pueda encontrar por sí mismo la localización de otros routers o gateways y guardar la información acerca del mejor camino para acceder a cada red.

Lógicamente el proceso real de routing sobre Internet suele ser mucho más complejo que el expuesto aquí, principalmente por el uso de redes y tecnologías muy distintas e incompatibles. Esto obliga a que se realicen conversiones en el formato de los paquetes para que puedan pasar a través de medios diferentes, pero en cualquier caso el protocolo IP proporciona una transmisión transparente para los protocolos de nivel superior y las aplicaciones de red.

1.6.3.8 Servicio de nombres

El sistema de nombres comúnmente denominado DNS (Domain Name System) es una forma alternativa de identificar a una máquina conectada a la red. La dirección IP resulta difícil de memorizar, siendo su uso más adecuado para los ordenadores. El sistema de nombres por dominio es el utilizado normalmente por las personas para referirse a un ordenador en la red, ya que además puede proporcionar una idea del propósito o la localización del mismo.

El nombre por dominio de un ordenador se representa de forma jerárquica con varios nombres separados por puntos (generalmente 3 ó 4, aunque no hay límite). Típicamente el nombre situado a la izquierda identifica al host, el siguiente es el subdominio al que pertenece este host, y a la derecha estará el dominio de mayor nivel que contiene a los otros subdominios:

nombre_ordenador.subdominio.dominio_principal

Aunque esta situación es la más común, el nombre por dominio es bastante flexible, permitiendo no sólo la identificación de hosts sino que también puede utilizarse para referirse a determinados servicios proporcionados por un ordenador o para identificar a un usuario dentro del mismo sistema. Es el caso de la dirección de correo electrónico, donde el nombre por dominio adquiere gran importancia puesto que el número IP no es suficiente para identificar al usuario dentro de un ordenador.

Para que una máquina pueda establecer conexión con otra es necesario que conozca su número IP, por lo tanto, el nombre por dominio debe ser convertido a su correspondiente dirección a través de la correspondiente base de datos. En los inicios de Internet esta base de datos era pequeña de manera que cada sistema podía tener su propia lista con los nombres y las direcciones de los otros ordenadores de la red, pero actualmente esto sería impensable. Con esta finalidad se utilizan los servidores de nombres por dominio (DNS servers).

Los servidores de nombres por dominio son sistemas que contienen bases de datos con el nombre y la dirección de otros sistemas en la red de una forma encadenada o jerárquica. Para comprender mejor el proceso supongamos que un usuario suministra el nombre por dominio de un sistema en la red a su ordenador local, realizándose el siguiente proceso:

- El ordenador local entra en contacto con el servidor de nombres que tiene asignado, esperando obtener la dirección que corresponde al nombre que ha suministrado el usuario.
- El servidor de nombres local puede conocer la dirección que se está solicitando, entregándosela al ordenador que realizó la petición.
- Si el servidor de nombres local no conoce la dirección, ésta se solicitará al servidor de nombres que esté en el dominio más apropiado. Si éste tampoco tiene la dirección, llamará al siguiente servidor DNS, y así sucesivamente.
- Cuando el servidor DNS local ha conseguido la dirección, ésta se entrega al ordenador que realizó la petición.
- Si el nombre por dominio no se ha podido obtener, se enviará de regreso el correspondiente mensaje de error.

1.6.3.9 Suite del protocolo TCP/IP

Con el propósito de comprender mejor el papel que juegan los muchos componentes de la familia del protocolo TCP/IP, resulta útil que sepa lo que puede realizar en una red el mismo protocolo TCP/IP. De tal forma que una vez entendidas las aplicaciones es más fácil comprender los protocolos.

A continuación mencionamos las aplicaciones principales para cada una de las partes que componen el protocolo de TCP/IP y que son de utilidad para el usuario las cuales se muestran en la figura 1.6.3.9.1 y se comparan con las capas o niveles del modelo OSI.

Telnet.- Registro de entrada remoto	Nfs.-Servidor de archivos de red
FTP.-Protocolo de transferencia de archivos	RPC.-Llamada de procedimiento remoto
SMTP.-Protocolo simple de transferencia de correo	TFTP.-Protocolo trivial de tranferencia de archivos
X.- Sistema X windows	TCP.- Protocolo de control de transmisión
Kerberos.- Seguridad	Protocolo de datagramas de usuario
DNS.-Sistema de nombre de dominio	IP.- Protocolo internet.
ASN.-Notación de sintaxis abstracta	ICMP.- Protocolo internet de control de mensajes
SNMP.- Protocolo simple de administración de red.	

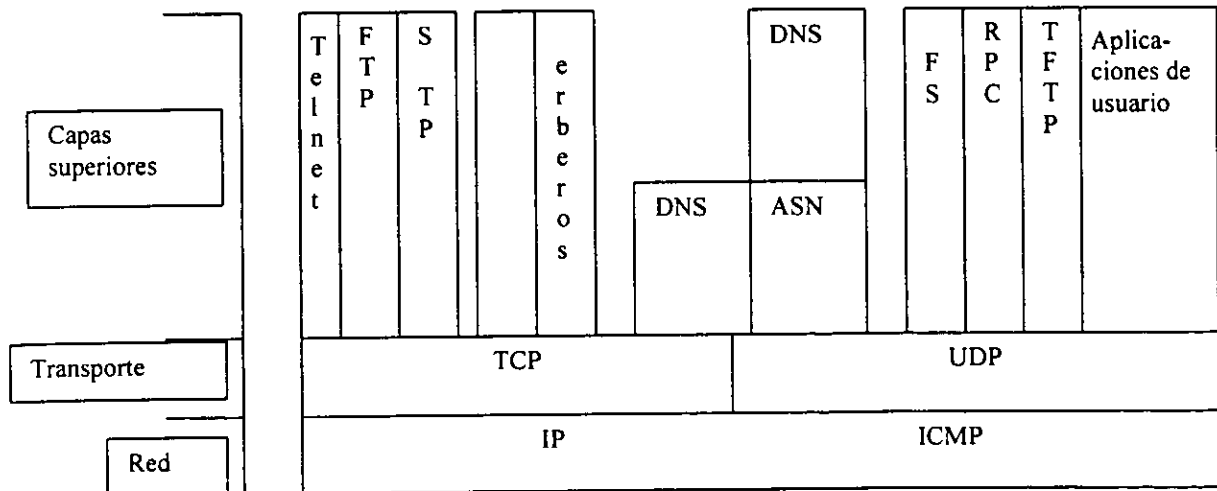


Figura 1.6.3.9.1 Grupo TCP/IP y capas OSI

1.6.3.10 Aplicaciones y utilidades

- **Telnet.** El programa telnet proporciona capacidad de registro de entrada remoto. Esto permite a un usuario de una maquina registrarse en otra maquina, y trabajar como si estuviera directamente en la maquina en la que entro. La conexión se puede realizar en cualquier parte de la red, siempre y cuando el usuario tenga permisos para acceder.
- **Protocolo de transferencia de archivos.** Permite copiar archivos de un sistema a otro de igual manera se requiere de tener permisos. Una vez establecida la conexión se pueden depositar o extraer archivos sin afectar el sistema original.
- **Protocolo simple de transferencia de archivos.** Es utilizado principalmente para la transferencia de correo electrónico, esto lo hace de una manera similar al FTP
- **Kerberos.** Es un protocolo de seguridad de amplio soporte, que utiliza un dispositivo especial conocido como servidor de autenticación. Este revalida contraseñas y esquemas de cifrado (o encriptación).
- **Servidor de nombre de dominio.** El servidor de nombre de dominio habilita un dispositivo con un nombre común para que sea convertido a una dirección especial de red. Por ejemplo, no se puede tener acceso a un sistema llamado juan_pc desde una red del otro lado del país, a menos que este disponible algún método de verificación de los nombres de las maquinas locales. DNS proporciona la conversión del nombre común local a la dirección física única de la conexión de red del dispositivo.
- **Notación sintáctica abstracta.** Es un lenguaje utilizado tanto por OSI como por TCP/IP para definir estándares. Es decir permite definiciones sin ambigüedad para los términos que se utilizan en dichos estándares.

- **Protocolo simple de administración de red.** Este protocolo utiliza como mecanismo de transporte el protocolo de datagrama de usuario (UDP). Emplea términos diferentes de TCP/IP, como administradores y agentes en vez de clientes y servidores. Un agente proporciona información sobre un dispositivo, en tanto que el administrador se comunica a través de red.
- **Servidor de archivo de red.** Es un conjunto de protocolos desarrollado por Sun Microsystems para habilitar en forma transparente a varias máquinas el acceso a los directorios de las demás máquinas. Estas consiguen lo anterior utilizando un esquema de sistema de archivo distribuido. Los sistemas NFS son comunes en grandes entornos corporativos que utilizan estaciones de trabajo UNIX.
- **Llamadas de procedimiento remoto.** Son funciones que permiten que una aplicación se comunique con otra máquina (servidor). Proporcionan funciones de programación, códigos de regreso y variables predefinidas para soportar computación distribuida.
- **Protocolo trivial de transferencia de archivo.** Este es un protocolo de transferencia de archivos muy sencillo, sin complicaciones que carece totalmente de seguridad. Utiliza el UDP como transporte.
- **Protocolo de control de transmisión.** Es un protocolo de comunicaciones, que proporciona transferencia confiable de datos. Es responsable de ensamblar datos, pasados desde aplicaciones de capas superiores a paquetes estándar y asegurarse que los datos se transfieren correctamente.
- **Protocolo de datagramas de usuario.** Es un protocolo orientado a comunicaciones sin conexión, lo que significa que no tiene mecanismo para la retransmisión de datagramas (a diferencia de TCP, que es orientado a conexión). UDP no es muy confiable, pero si tiene fines particulares. Si las

aplicaciones que utilizan UDP tienen su propia verificación de confiabilidad, los inconvenientes de UDP se pueden superar.

- **Protocolo Internet.** Este es responsable de mover a través de las redes paquetes de datos ensamblados, ya sea por TCP o UDP. A fin de determinar enrutamientos y destinos, utiliza un conjunto de direcciones únicas para cada dispositivo en la red.
- **Protocolo internet de control de mensajes.** Este es responsable de generar y verificar mensajes sobre el estado de los dispositivos de una red. Puede utilizarse para informar a otros dispositivos de la falla de una máquina en particular, generalmente funciona de manera conjunta con IP.

1.6.3.11 Protocolo de control de transferencia

El protocolo de control de transmisión (TCP) pertenece al nivel de transporte, siendo el encargado de dividir el mensaje original en datagramas de menor tamaño, y por lo tanto, mucho más manejables. Los datagramas serán dirigidos a través del protocolo IP de forma individual. El protocolo TCP se encarga además de añadir cierta información necesaria a cada uno de los datagramas. Esta información se añade al inicio de los datos que componen el datagrama en forma de cabecera.

La cabecera de un datagrama contiene al menos 160 bits que se encuentran repartidos en varios campos con diferente significado. Cuando la información se divide en datagramas para ser enviados, el orden en que éstos lleguen a su destino no tiene que ser el correcto. Cada uno de ellos puede llegar en cualquier momento y con cualquier orden, e incluso puede que algunos no lleguen a su destino o lleguen con información errónea. Para evitar todos estos problemas el TCP numera los datagramas antes de ser enviados, de manera que sea posible volver a unirlos en el orden adecuado. Esto permite también solicitar de nuevo el envío de los datagramas individuales que no hayan llegado o que contengan errores, sin que sea necesario volver a enviar el mensaje completo.

A continuación de la cabecera, figura 1.6.3.11.1, puede existir información opcional. En cualquier caso el tamaño de la cabecera debe ser múltiplo de 32 bits, por lo que puede ser necesario añadir un campo de tamaño variable y que contenga ceros al final para conseguir este objetivo cuando se incluyen algunas opciones. El campo de tamaño contiene la longitud total de la cabecera TCP expresada en el número de palabras de 32 bits que ocupa. Esto permite determinar el lugar donde comienzan los datos.

Formato de la cabecera TCP.			
Puerto origen		Puerto destino	
Número de secuencia			
Señales de confirmación			
Tamaño	Reservado	Bits de control	Window
Checksum		Puntero a datos urgentes	

Figura 1.6.3.11.1 Formato de cabecera de protocolo TCP

Dos campos incluidos en la cabecera y que son de especial importancia son los números de puerto de origen y puerto de destino. Los puertos proporcionan una manera de distinguir entre las distintas transferencias, ya que un mismo ordenador puede estar utilizando varios servicios o transferencias simultáneamente, e incluso puede que por medio de usuarios distintos. El puerto de origen contendrá un número cualquiera que sirva para realizar esta distinción. Además, el programa cliente que realiza la petición también se debe conocer el número de puerto en el que se encuentra el servidor adecuado. Mientras que el programa del usuario utiliza números prácticamente

aleatorios, el servidor debe tener asignado un número estándar para que pueda ser utilizado por el cliente. (Por ejemplo, en el caso de la transferencia de ficheros FTP el número oficial es el 21). Cuando es el servidor el que envía los datos, los números de puertos de origen y destino se intercambian.

En la transmisión de datos a través del protocolo TCP la fiabilidad es un factor muy importante. Para poder detectar los errores y pérdida de información en los datagramas, es necesario que el cliente envíe de nuevo al servidor unas señales de confirmación una vez que se ha recibido y comprobado la información satisfactoriamente. Estas señales se incluyen en el campo apropiado de la cabecera del datagrama, que tiene un tamaño de 32 bits. Si el servidor no obtiene la señal de confirmación adecuada transcurrido un período de tiempo razonable, el datagrama completo se volverá a enviar. Por razones de eficiencia los datagramas se envían continuamente sin esperar la confirmación, haciéndose necesaria la numeración de los mismos para que puedan ser ensamblados en el orden correcto.

También puede ocurrir que la información del datagrama llegue con errores a su destino. Para poder detectar cuando sucede esto se incluye en la cabecera un campo de 16 bits, el cual contiene un valor calculado a partir de la información del datagrama completo (checksum). En el otro extremo el receptor vuelve a calcular este valor, comprobando que es el mismo que el suministrado en la cabecera. Si el valor es distinto significaría que el datagrama es incorrecto, ya que en la cabecera o en la parte de datos del mismo hay algún error.

La forma en que TCP numera los datagramas es contando los bytes de datos que contiene cada uno de ellos y añadiendo esta información al campo correspondiente de la cabecera del datagrama siguiente. De esta manera el primero empezará por cero, el segundo contendrá un número que será igual al tamaño en bytes de la parte de datos del datagrama anterior, el tercero con la suma de los dos anteriores, y así sucesivamente. Por ejemplo, para un tamaño fijo de 500 bytes de datos en cada

datagrama, la numeración sería la siguiente: 0 para el primero, 500 para el segundo, 1000 para el tercero, etc.

Existe otro factor más a tener en cuenta durante la transmisión de información, y es la potencia y velocidad con que cada uno de los ordenadores puede procesar los datos que le son enviados. Si esto no se tuviera en cuenta, el ordenador de más potencia podría enviar la información demasiado rápido al receptor, de manera que éste no pueda procesarla. Este inconveniente se soluciona mediante un campo de 16 bit (Window) en la cabecera TCP, en el cual se introduce un valor indicando la cantidad de información que el receptor está preparado para procesar. Si el valor llega a cero será necesario que el emisor se detenga. A medida que la información es procesada este valor aumenta indicando disponibilidad para continuar la recepción de datos.

1.6.3.12 Protocolo de transferencia de archivo.

Este protocolo permite establecer "depósitos" de datos donde los usuarios conectados pueden recibir o enviar archivos y programas.

Un servidor FTP funciona como un gran disco duro con directorios o divisiones en el cual, si tenemos permiso asignado podemos enviar nuestros archivos o copiar los que allí existen.

Al usar un programa de FTP, elegimos primero la máquina que deseamos acceder y nos identificamos con un nombre y contraseña. Una vez conectados, se nos presenta una lista de los archivos y directorios disponibles en dicho sitio.

Numerosos sitios FTP mantienen una parte abierta al público que podemos acceder usando el nombre "anonymous" como usuarios anónimos y nuestra dirección de correo como contraseña. Pero, para evitar el flujo excesivo de información, algunos restringen la cantidad de usuarios anónimos conectados simultáneamente a 100 o un número específico. Así que si intenta acceder un servidor FTP y es rechazado, intente más tarde, preferiblemente en una hora de menor flujo. Los servidores FTP son una forma

conveniente de hacer disponibles al público informaciones, resúmenes de discusiones, investigaciones, programas y actualizaciones de software.

Por medio de los sitios FTP los fabricantes de software y hardware distribuyen documentos sobre los problemas confrontados por otros usuarios y cómo solucionarlos, incluyendo "parches" y actualizaciones para sus productos.

Por ejemplo, miles de usuarios en todo el mundo, al instalar Windows 95 en sus PC's el año pasado, recurrieron a los sitios FTP buscando la última versión del software manejador ("driver") de sus tarjetas de vídeo, sonido y otros dispositivos para lograr configurarlos bajo el nuevo sistema operativo.

1.6.3.13 Protocolos TCP/IP

El TCP/IP es un conjunto de protocolos 'a capas' o, si se prefiere, 'a niveles'. Para entender qué significa todo lo anterior pongamos un ejemplo sencillo. Imaginemos que de tiene que enviar correo a través de Internet. Lo primero que se necesita es definir un protocolo específico para el correo, o sea, un conjunto de reglas unívocamente reconocidas por todos los ordenadores conectados en red. Dicho protocolo tendrá la tarea de coger la carta que hay que enviar, añadirle el emisor y el destinatario y enviarla a quien corresponda. Esto último es la tarea del protocolo específico de gestión del correo, que podría ser comparado al de una persona a la que un amigo muy ocupado le deja una carta y ella se encarga de ponerla en el sobre, escribir los datos de expedición y depositarla en el correo para poder enviarla.

Evidentemente, si sólo existiese esta figura la carta se quedaría eternamente en el buzón sin que nadie se preocupase de hacerla llegar a su destino. Sin embargo, nuestro amigo muy ocupado tendría suerte ya que existe una camioneta del servicio de correos que dos veces al día vacía el buzón y transporta las cartas que allí encuentra a un lugar donde serán clasificadas y diferenciadas; allí su importante carta será cuidada y protegida hasta que llegue al buzón del destinatario.

Para continuar con el paralelismo del ejemplo, diremos que el TCP/IP representa el sistema de transporte de Internet. En particular, el TCP se preocupa de 'empaquetar' bien todos los datos que le son suministrados por los protocolos de nivel superior; es posible que los subdivide en más partes si resultasen demasiado largos para un solo envío en red; asimismo recuerda lo que ha sido enviado, se acuerda de volver a enviarlo en el caso en que se hubiera perdido y controla que todo se realice de forma transparente para el usuario.

Ya que este tipo de operaciones es de uso general y es necesario tanto para enviar correo como para enviar ficheros u otras cosas, se ha pensado en hacer un protocolo propio, que pueda ser utilizado por muchos otros. Es precisamente por este motivo por lo que hemos definido protocolo de bajo nivel.

El TCP, sin embargo, no es el protocolo de nivel más bajo desde el momento en que éste utiliza el IP para realizar determinadas acciones. De hecho, a pesar de que el TCP sea muy utilizado, existen protocolos que prefieren no usarlo y que para funcionar sólo necesitan las funciones que puede ofrecer incluso el más humilde IP. Este tipo de organización 'a capas' permite una gran eficiencia y un menor gasto de recursos.

Para terminar con un ejemplo, el envío de un mensaje de correo electrónico a través de Internet utiliza un sistema compuesto por cuatro capas:

- Un protocolo de alto nivel específico para el correo
- El protocolo TCP que es utilizado también por otros protocolos de alto nivel
- El protocolo IP que se ocupa de la tarea específica de tomar los paquetes y enviarles a su destino
- El protocolo del hardware específico, que se utiliza para la transmisión y la recepción de los datos

En este punto nos aparece claro el motivo por el que el conjunto de los protocolos de Internet es llamado genéricamente TCP/IP. De hecho, estos son los protocolos más utilizados y de los que sólo pueden prescindir muy pocos protocolos de un nivel más alto.

Antes de terminar esta exposición general sobre el funcionamiento del TCP/IP es necesario introducir el concepto de datagrama (datagram), que representa cada uno de los paquetes de informaciones que es enviado a través de la red. Como ya hemos dicho antes, un conjunto de informaciones demasiado largo que es subdividido en paquetes más pequeños, precisamente llamados datagrama, que viajan individualmente en la red. Esto significa que si un fichero que se debe enviar es subdividido en 10 datagramas secuenciales, no está dicho que el cuarto llegue antes que el séptimo, desde el momento en que éste puede perderse o tomar un camino equivocado. Será una tarea de los diversos protocolos el hacer que dicho paquete sea enviado nuevamente y colocado en el orden correcto secuencial a su llegada a destino.

Y ahora, para evitar los ataques de los "puristas" diremos que a pesar de que los términos datagrama y paquete son muy a menudo utilizados como sinónimos, en realidad existe una diferencia. Mientras el datagrama es específico del TCP/IP y representa la mínima unidad lógica utilizable por los diversos protocolos, el paquete es una entidad física bien presente para quien administra una red de tipo Ethernet. En el caso, por lo demás muy frecuente, que en un paquete viaje un solo datagrama, la diferencia es sólo teórica pero existen también configuraciones específicas de hardware de red que utilizan paquetes de dimensión menor respecto a la del datagrama individual. Entonces sucede que un datagrama se descompone en más paquetes durante el envío a la red específica y que sea recompuesto a su llegada, de forma absolutamente transparente respecto al mismo datagrama que... 'no se da cuenta' de haber sido descompuesto y luego recompuesto. Es evidente cómo en dicha situación los términos paquete y datagrama no coinciden. Es una buena medida, por tanto, acostumbrarse a utilizar el término datagrama cuando se habla del TCP/IP.

1.6.3.14 Protocolo básico de gestión de red

Este es en la actualidad el protocolo de gestión más utilizado. Opera sobre el protocolo de datagrama de usuario (UDP). Que a su vez forma parte de TCP/IP. El SNMP es un protocolo sencillo de comunicación que permite recoger información de gestión de los dispositivos conectados a la red. Ciertos agentes localizados en los dispositivos recogen la información y la registran en una base de información de gestión (MIB). Esta información se refiere a las características del dispositivo, la capacidad de procesamiento de datos, las sobrecargas en el tráfico y los errores. Los MIB tienen un formato común, de forma que las utilidades de gestión SNMP, aun siendo de fabricantes distintos, pueden recabar información y presentársela al administrador del sistema en la consola de gestión.

La clave de la popularidad de SNMP radica en su simplicidad. Consta de un reducido conjunto de ordenes que permite recoger información de casi cualquier dispositivo de red. En un entorno SNMP, la mayor parte de trabajo recae sobre el sistema de gestión de la red. Los dispositivos gestionados no se van sobrecargados con un procesamiento adicional que podría perjudicar a su eficiencia. En la figura 1.6.3.14.1 se muestra un entorno SNMP típico. A continuación se describen los componentes del entorno:

- Dispositivos gestionados. Estos contienen agentes que se encargan de recoger información y transmitirla a la consola de gestión por medio del protocolo SNMP.
- Consola de gestión. Esta recoge en una base de información de gestión (MIB) la información enviada por los agentes situados en los dispositivos de red.
- SNMP. Este protocolo consulta a los agentes de los dispositivos gestionados y envía la información a la consola de gestión
- Agente proxy. Este agente es capaz de ejecutar los agentes de dispositivos que no son capaces de hacerlo, proporcionándoles las funciones SNMP.

SNMP utiliza un proceso de solicitud y respuesta donde la consola de gestión solicita información a los agentes de los dispositivos gestionados. SNMP es básicamente un protocolo de transporte no orientado a la conexión, con un sencillo conjunto de ordenes que accede a los agentes para consultar valores y cambiar el estado de los elementos de la red. La mayoría de estos sistemas son capaces de descubrir automáticamente la topología de la red y visualizarla gráficamente. Aquí tenemos algunas de sus funciones

- Obtención del mapa de la topología.
- Detección de sucesos con alarmas.
- Control de trafico.
- Funciones de diagnostico de la red.
- Generadores de informes.
- Gestión de un registro histórico y análisis de tendencias.

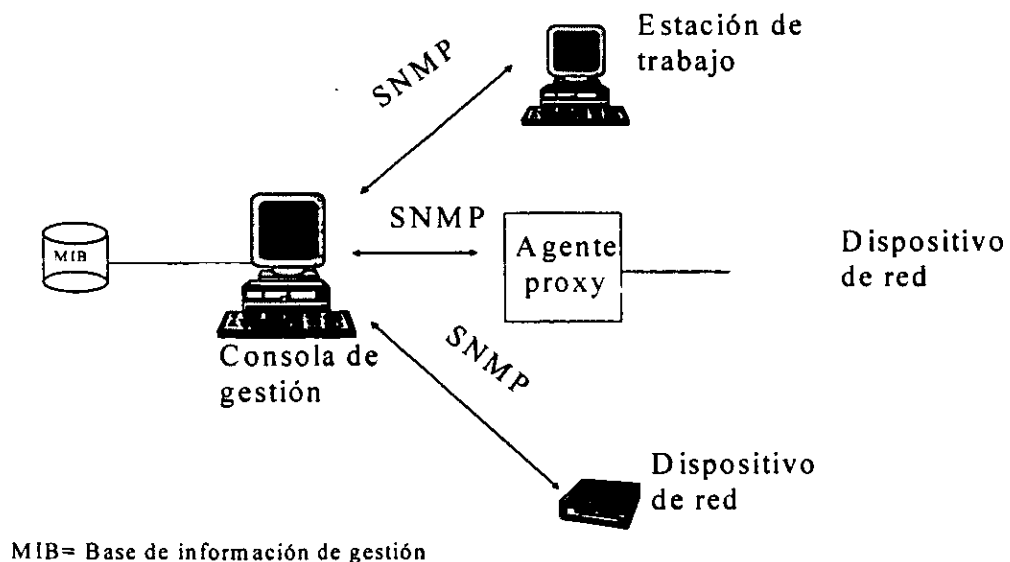


Figura 1.6.3.14.1 Entorno de gestión SNMP.

1.7 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE VISUAL FOX PRO 5.0

Introducción

Los lenguajes de consulta constituyen una funcionalidad importante de los sistemas de gestión de bases de datos orientadas a objetos. El usuario puede recuperar los datos especificando simplemente las condiciones que estos datos deben cumplir. De manera general, estos lenguajes deber ser:

- De alto nivel, de modo que se puedan formular fácilmente operaciones complejas.
- Declarativos, esto es, que el foco de atención esté dirigido hacia QUÉ se quiere y no hacia CÓMO se obtiene.
- Eficientes: las construcciones deben permitir optimizaciones.
- Independientes de la aplicación, de modo que se puedan usar sobre cualquier esquema de base de datos.

1.7.1 Ventajas

La ventajas más significativa de los sistemas de gestión de bases de datos orientadas a objetos es la flexibilidad y soporte para el manejo de tipos de datos complejos.

Es decir una base de datos orientada a objetos puede añadir una subclase de la clase para manejar las modificaciones que representan los clientes por referencia.

La subclase hereda todos los tributos, características de la clase original, además se especializara en especificar los nuevos campos que se requieren así como los métodos para manipular solamente estos campos. Obviamente se generan los campos para almacenar dicha información. Una ventaja adicional es que una base de datos orientada a objetos puede ajustarse a usar siempre el espacio de los caminos que son necesarios, eliminando espacio desperdiciado en registros con campos que nunca se usan.

Otra ventaja que podemos nombrar es que una base de datos orientada a objetos, es que manipula datos complejos en forma rápida y ágilmente. La estructura de la base de datos está dada por referencias o apuntadores lógicos entre objetos. No se requieren búsquedas en tablas o uniones para crear relaciones. Esta capacidad resulta atractiva en aplicaciones de la ingeniería, donde las relaciones entre componentes dependen de factores diversos.

Por ejemplo considérese una aplicación en el diseño de sillas. El fabricante que quiere determinar una lista de partes necesarias para una silla, para un modelo en particular requiere de diferentes decisiones subsecuentes para elaborarla, si el modelo es automático o estándar, se necesita de una diseño en particular así como el material con el que va a ser fabricada. Escoger un tipo en particular obliga a decidir sobre otras partes requeridas, todo esto hasta el nivel de componentes y piezas individuales.

Armar esta lista de componentes resulta más ágil en una base de datos orientada a objetos, que en una base de datos relacional. En una base de datos relacional las tablas deben ser barridas, buscadas cada vez que se indica una condición, resultando posiblemente en miles de direccionamientos.

1.7.2 Desventajas

La inmadurez en el mercado de la tecnología orientada a objetos y en las mismas bases de datos orientadas a objetos, constituye una posible fuente de problemas, por lo que se debe analizar con detalle la presencia en el mercado del proveedor para adoptar su producto en una línea de producción sustantiva.

La falta de estándares en la industria orientada a objetos puede ser otra desventaja, sin embargo existe una organización internacional de proveedores de sistemas de información y usuarios dedicada a promover estándares para el desarrollo de aplicaciones y sistemas orientados a objetos en ambiente de computo en red.

El paradigma de los usuarios a utilizar una nueva tecnología es muy difícil de romper. Aquellos que esperan resultados a corto plazo y con un costo reducido quedarán desilusionados. Sin embargo, para aquellos usuarios que planean a un futuro intermedio con una visión tecnológica de avanzada, el uso de la misma, así como el uso de tecnología orientada a objetos, paulatinamente compensará todos los riesgos.

La tecnología de bases de datos orientada a objetos está en su desarrollo, sin embargo, establece amplios signos de madurez. Ante la disyuntiva de tomar una decisión estratégica, recalcamos que las empresas e industrias que desean conformar un liderazgo tecnológico están en la posibilidad de explorar los productos comercialmente disponibles a los centros de investigación para iniciar la experimentación y desarrollo de proyectos pilotos.

1.7.3 Microsoft Visual FoxPro 5.0

Introducción

El entorno de desarrollo de base de datos orientadas a objetos en Microsoft Visual FoxPro 5.0 continúa la tradición de liderazgo adquirido con la versión 3.0. Se caracteriza por la eficaz extensibilidad de ActiveX, un rendimiento y características de conectividad mejoradas, y un entorno de desarrollo ampliado.

Microsoft Visual FoxPro 5.0 Es la versión mas reciente del entorno de desarrollo de aplicaciones de bases de datos orientadas a objetos de Microsoft. Visual Fox Pro 5.0 ofrece a los programadores las herramientas necesarias para administrar datos (para organizar tablas de información o realizar consultas, para crear sistemas de administración de bases de datos relacional o para programar una aplicación completa de gestión de datos para usuarios finales).

Ventajas

Construcción rápida de aplicaciones de bases de datos con las últimas técnicas de la programación orientadas a objetos. Utiliza los controles ActiveX y las bibliotecas de clases de Visual FoxPro para ensamblar aplicaciones con componentes ya probados y para construir rápidamente aplicaciones nuevas.

Creación de aplicaciones más rápidas en su ejecución. Las aplicaciones se ejecutan hasta un 40% más rápido como resultado de la mejora del motor de bases de datos y de la interfaz de usuario. Los formularios se cargan y actualizan mas rápidamente y los datos se recuperan con prontitud tanto en local como en un entorno

cliente-servidor.

Construye y depura aplicaciones mas rápidamente con un entorno de desarrollo mejor. Un editor mejorado, con un depurador totalmente nuevo y la integración de Visual SourceSafe, ayuda al programador que realiza su trabajo solo o en equipo a terminar su tarea en muy poco tiempo.

Características

- Creación de servidores de automatización ActiveX. Las aplicaciones Visual FoxPro ahora pueden distribuirse como servidores de automatización ActiveX. Estos servidores pueden ser consultados por una gran variedad de aplicaciones para el usuario en local, dentro de una red local, de una intranet o desde internet.
- Creación de aplicaciones de múltiples niveles distribuidas. Las aplicaciones de múltiples niveles pueden centralizar las reglas de negocios y optimizar los recursos de la red. Con Visual FoxPro 5.0, es posible crear servidores de reglas de negocios de nivel medio y disponer de aplicaciones de usuario hasta servidores de nivel medio creados con otras herramientas.
- Integración de aplicaciones con un servidor Web. Visual FoxPro 5.0 incluye herramientas para integración de aplicaciones Visual FoxPro con Microsoft Internet Information Server. Es posible crear aplicaciones basadas en Web para que los usuarios consulten y actualicen la información de las bases de datos y compartan datos con otros usuarios de la red local.
- Asistente de búsqueda de WWW. Cree rápidamente búsquedas en

la Web con Internet Search Wizard. El asistente de búsqueda de WWW crea una página de búsqueda donde los usuarios pueden formular una petición de búsqueda transmitida a una base de datos Visual FoxPro. Los resultados de consultas se convierten automáticamente a paginas HTML.

- Soporte para usuarios móviles y distantes con vistas sin conexión. Cuando hay usuarios que se encuentran en distintas ubicaciones o desplazados y necesitan compartir información, utiliza la nueva función Vista sin conexión para petición y a continuación sincroniza de nuevo las múltiples modificaciones de las bases de datos.
- Permite volver a una versión anterior de su código o de la aplicación. Se puede programar con confianza y flexibilidad mediante el programa opcional Visual SourceSafe. Guarda las versiones de cada componente y de la aplicación en un mismo conjunto. Después, regresa cuando se requiera a una versión anterior. Es una gran solución para programadores, tanto si trabajan solos, como si lo hacen en equipo.
- Puede coordine un equipo de programación con Visual SourceSafe. Muchos trabajadores pueden trabajar sobre un mismo proyecto empleando las características avanzadas de protección, desprotección, combinación, bifurcación y diferenciación de Visual SourceSafe. Visual SourceSafe, está ahora integrado con Visual FoxPro 5.0.
- Un depurador totalmente nuevo y más eficaz. El nuevo depurador incluye características avanzadas, como la capacidad de guardar y cargar mas secciones de depuración, una vista en árbol para inspeccionar todos los objetos de la aplicación y un soporte

complejo para puntos de interrupción. La función arrastra y coloca esta completamente activada en todas las componentes de depuración.

- Crea pantallas más rápidas con la asignación de campos personalizados. Están definidas dentro de una base de datos, las asignaciones de campos permiten al programador personalizar todo, desde un título de campo hasta una clase personalizada empleada para visualizar el mismo campo en cualquier formulario de la aplicación. Esto acelera tanto la creación de formularios como su mantenimiento.

Las nuevas uniones externas ahorran tiempos de programación SQL. Las uniones externas ya son compatibles mediante SQL en ANSI estándar. Así se ahorra código lo que se traduce en consultas más rápidas. El diseñador de consultas de Visual FoxPro utiliza automáticamente la nueva sintaxis.

Características Mejoradas

- Un rendimiento mejorado de los formularios. Con las prestaciones de objetos mejorados, los usuarios pueden presentar y restaurar con mayor rapidez los formularios. Al reducir los requisitos de memoria de los formularios más usados, Visual FoxPro consigue acelerar en un 40% su ejecución.

- Mejoras de rendimiento. Con Visual FoxPro 5.0, ahora los programadores pueden optar por el enlace de datos diferido. Esto es útil en el caso de formularios de múltiples páginas, ya que el enlace diferido implica que los formularios se presentan y restauran con más rapidez, por que hay menos datos a los que se tiene acceso a

medida que los usuarios se desplazan hacia el registro que necesitan.

- Rendimiento mejorado de motor de base de datos. Visual FoxPro 5.0 mantiene su liderazgo en cuanto a prestaciones con una recuperaciones aún más rápidas en bases locales y un acceso mejorado a datos en bases en servidores como Microsoft SQL Server.

Posibilidad de elegir entre un mayor número de controles ActiveX. Visual FoxPro 5.0 incluye 26 controles ActiveX y ahora es compatible con más controles de otros fabricantes, incluyendo los que usted mismo crea con las herramientas disponibles en Visual Studio 97. Los controles ActiveX son componentes reutilizables que funcionan con una gran variedad de herramientas y aplicaciones.

- Un editor más eficaz y productivo. La lectura del código es más cómoda por que las instrucciones están ahora codificadas en color siguiendo su esquema de colores preferido. Arrastre y coloque segmentos de código tan fácilmente como lo hace con su procesador de textos. Y salte de forma instantánea a un subprocedimiento gracias al explorador mejorado.

La programación ha sufrido en los últimos años numerosas modificaciones, uno de esos cambios ha sido determinado por los entornos de programación visuales que exigen mayor rapidez en el desarrollo de aplicaciones. Lenguajes que resultaban áridos en su concepción se han hecho mucho más accesibles para los programadores. Visual FoxPro se una de las herramientas que más acorde se encuentra con estos requerimientos y por ello es elegida cada vez más como solución corporativa de desarrollo.

Visual FoxPro incorpora la programación orientada a objetos, que es considerada como una de las técnicas de programación más reconocidas. La cada vez mayor implantación de entornos Windows también exige que el lenguaje, además de funcionar sobre este, permita realizar aplicaciones con su estilo particular.

Visual FoxPro es uno de los más rápidos del mercado en acceso a datos y esta es la razón principal por la que tiene más aceptación en grandes corporaciones. Es un lenguaje que facilita el trabajo. De ahí ha surgido el término para definir a estos lenguajes que permiten realizar aplicaciones en poco tiempo R.D.A. (Rapid Application Development).

Visual FoxPro incorpora capacidades como son la programación en internet y la arquitectura cliente-servidor.

Una de las desventajas de Visual FoxPro es que no se dispone de gran bibliografía, cursos ni artículos en revistas, sin embargo esta desventaja disminuye en gran medida por su ayuda en línea y soluciones de programación basadas en aplicaciones de ejemplos que son incluidas con el software. El uso de la tecnología Rushmore hace que Visual FoxPro sea una de las herramientas más rápidas del mercado. Rushmore es una técnica de búsqueda interna que aprovecha los índices de la tabla para acceder a grupos de registros. Comprende diversas técnicas, entre ellas están la forma de indexación, compresión de datos, optimización de consultas, almacenamiento de bitmaps, gestión de entrada/salida de datos y gestión de memoria.

Requisitos del Sistema

- Equipo/Procesador: PC con procesador PENTIUM o superior.

- Memoria se recomiendan 16MB para Windows 95 y 32MB para Windows NT Workstation.
- Disco duro Instalación mínima 30MB o instalación completa 190MB.
- Unidad de CD-ROM y monitor VGA o de resolución superior.
- Sistema operativo Microsoft Windows 95, NT o Workstation versión 3.51, o superior.

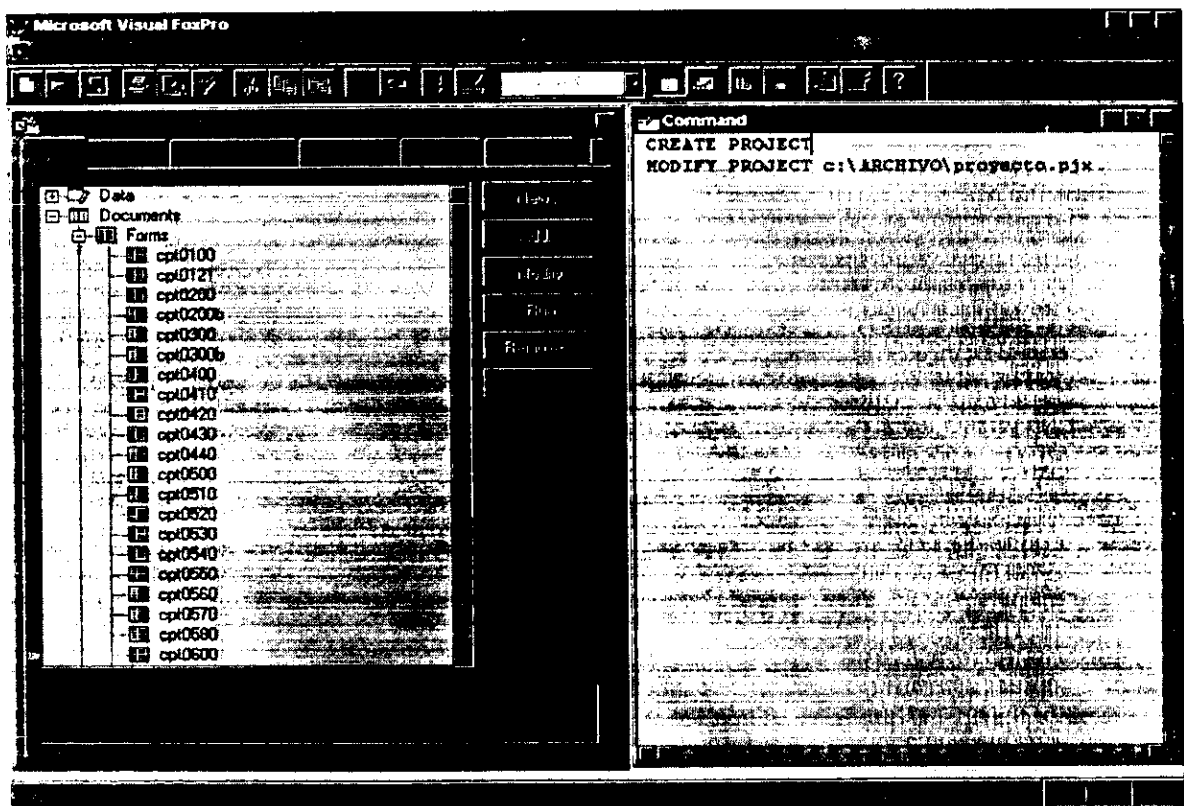


Figura 1.7.3.1 Pantalla principal de Visual FoxPro

En Visual FoxPro, si creamos una base de datos, deberemos incorporar tablas sobre las cuales podemos utilizar los siguientes aspectos.

- Tablas
- Relaciones
- Integridad referencia
- Procedimientos almacenados
- Consultas
- Vistas locales y vistas remotas

En Visual FoxPro cuando abrimos una tabla estamos accediendo a su información. Ahora bien, la acción de abrir supone que esa tabla se carga en memoria, en concreto en un área determinada. Nosotros podemos abrir tantas tablas como áreas tenga VFP, y en este caso son 255; salvo que el comando **Files** de Config.sys tenga un valor menor, lo cual limitaría este número máximo.

Una vez que tenemos las tablas en diferentes áreas, son accesibles paralelamente. Es fácil de controlar qué tablas hay en memoria utilizando la ventana Ver que se abre mediante la opción correspondiente del menú Ventana. En la Figura 1.7.3.3 se muestra.

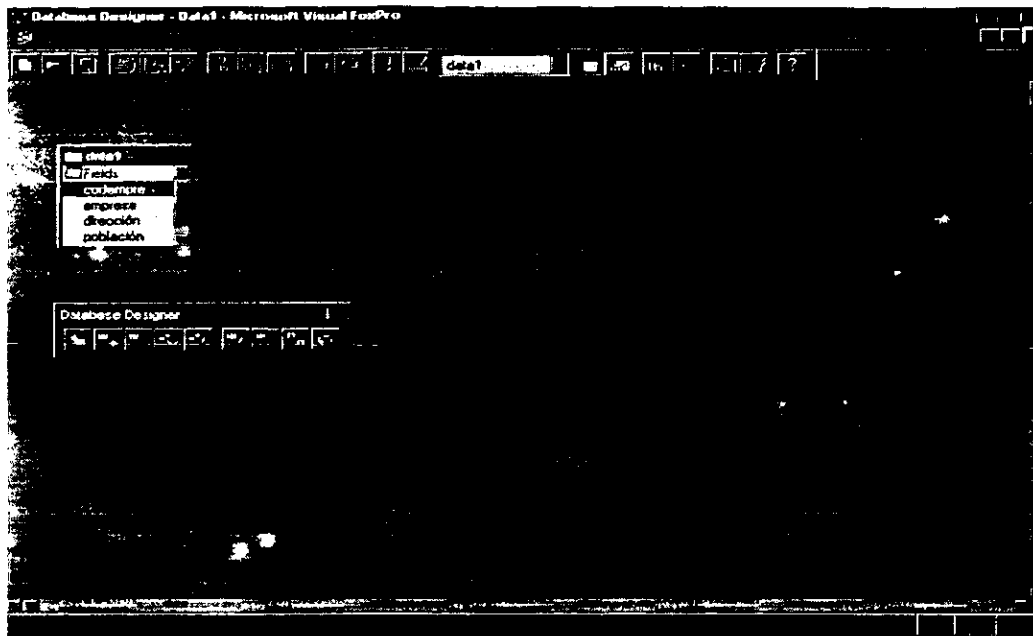


Figura 1.7.3.2 La barra de herramientas nos permite crear, agregar, modificar, examinar y eliminar tablas, así como crear y modificar vistas locales y remotas.

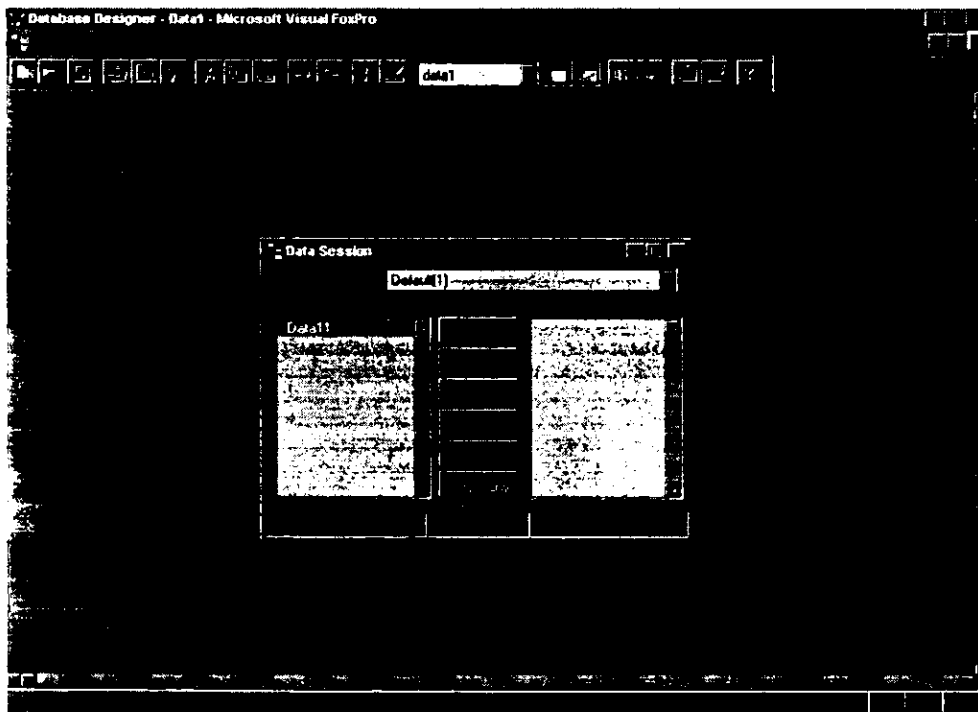


Figura 1.7.3.3 Sesión de datos

Cuando se crea una tabla en Visual FoxPro nos aparece la siguiente ventana, en donde se muestran las diferentes carpetas que contiene así como los diferentes valores y atributos que se pueden asignar a los diferentes campos:

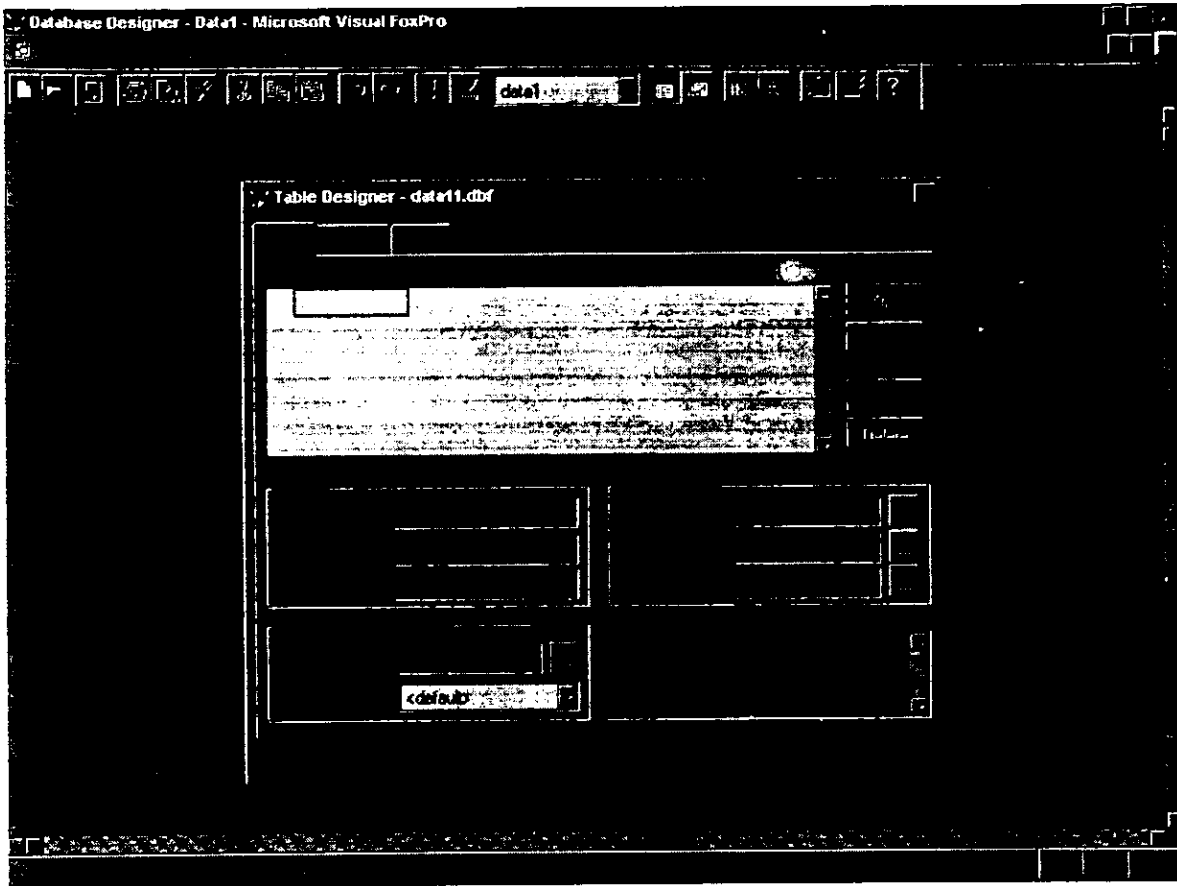


Figura 1.7.3.4 Diseñador de tablas

La diferencia respecto a una tabla libre es notoria en cuanto nos fijamos en las propiedades de campo y tabla que aparecen en la ventana. En la figura 1.7.3.5 se muestra. Sin embargo, no todo son ventajas en las tablas dependientes, pues podemos tener problemas si cambiamos la tabla de sitio, ya que la BD guarda el directorio, lo que nos llevará a tener que añadir de nuevo la tabla, puesto que VFP no la busca

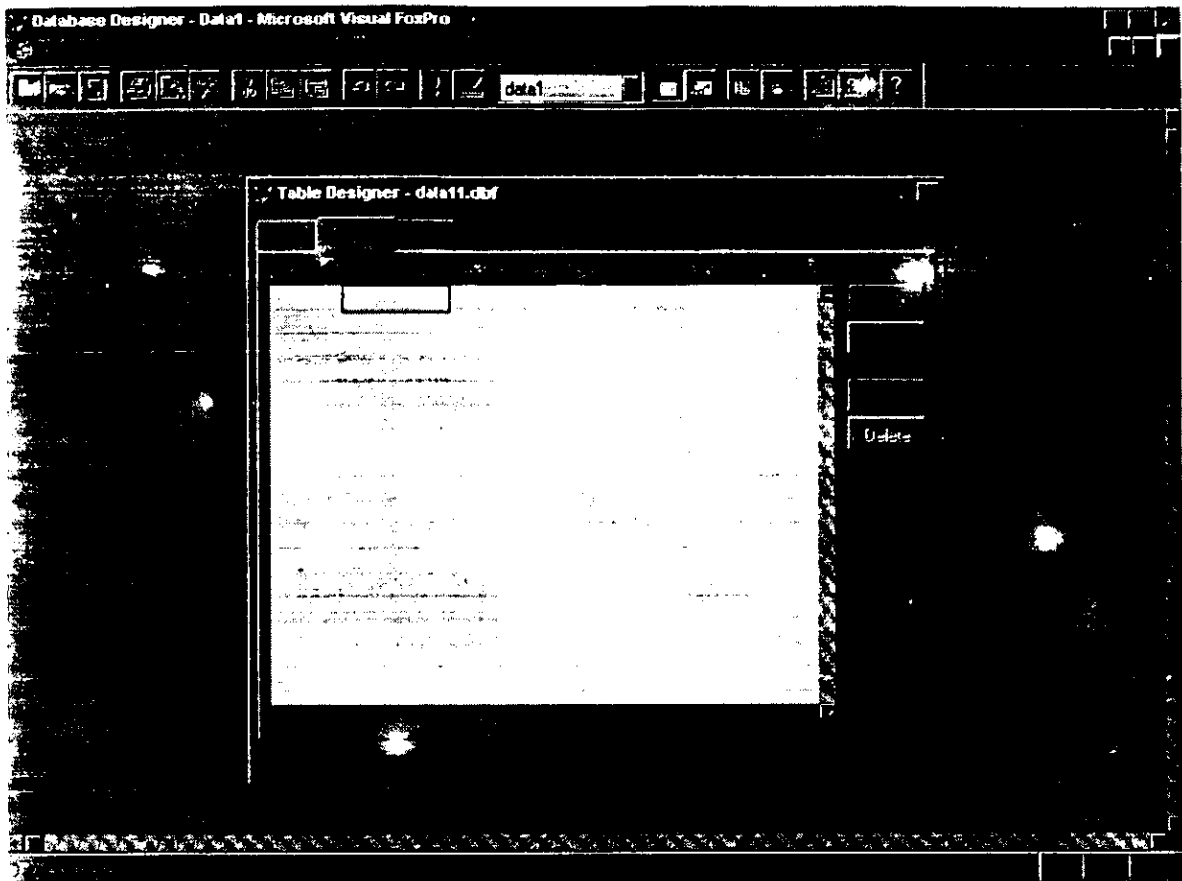


Figura 1.7.3.5 Tabla vacía

Asimismo, una tabla que pertenece a una BD no puede serlo de otra al mismo tiempo. En aplicaciones en las que el directorio varía, por ejemplo en las que existan varias empresas como en una contabilidad, lo mejor es utilizar las tablas libres.

Tipos de campos

Visual FoxPro, incorpora mayor número de campos de los que poseían las versiones anteriores. Aquí vuelve a tomar prestaciones de Access.

Estas son todas las posibilidades:

Tipo	Descripción	Ancho	Ejemplo
Numérico	Enteros o fracciones	10	1200
Moneda	Importes monetarios. Incluye siempre el valor con cuatro decimales. Para que una variable tome valor moneda debe ir precedida del símbolo \$.	8	1200,0000
Doble	Número de signo flotante de doble precisión. Ocupa menos memoria que el Numérico. Normalmente con este tipo es suficiente.	8	1200
Flotante	Igual que Numérico cuando incluyen decimales.	10	1200
Entero	Valores enteros. Si estamos seguros de que el campo no va tener nunca decimales.	4	1200
Fecha/ Hora	Datos de la Fecha y Hora	8	31/12/99 15:07:00
Lógico	Valor verdadero o Falso	1	.T.
Carácter	Cualquier texto	10	"Texto"
Memo	Texto con longitud indefinida	4	"Texto"
General	Guarda un objeto OLE	4	
Carácter (Binario)	Información guardada tal cual, sin cambiar de tabla de códigos.	10	
Memo (Binario)	Información con longitud ilimitada sin cambiar de tabla de códigos.	10	

Realmente se puede seguir trabajando con los tipos clásicos sin ningún problema. Esto es con: carácter, número, lógico, memo, fecha y general.

Toda esta variedad de tipos de campo nos lleva a tener sumo cuidado en la programación a la hora de concatenar variables, de cargar su información en otras variables ya definidas, o cuando se hacen comparaciones. La siguiente tabla ilustra las funciones que permiten la conversión entre distintos tipos de datos.

TABLA DE CONVERSIONES		
TIPO ORIGEN	FUNCIÓN	TIPO DESTINO
Carácter o Memo	VAL()	Numérico
Numérico, Flotante, Doble o Entero	STR()	Carácter
Numérico	NTOM()	Moneda
Moneda	MTON()	Numérico
Carácter o Memo	CTOD()	Fecha
Fecha	DTOC() ó DTOS()	Carácter
Carácter	CTOBIN()	Carácter binario
Carácter binario	BINTOC()	Carácter
Carácter o Memo	CTOT()	FechaHora
FechaHora	TTOC()	Carácter
Fecha	DTOT()	FechaHora
FechaHora	TTOD()	Fecha
Lógico	IIF()	Cualquiera

Propiedades de campo

Como también ocurre en Access podemos establecer una serie de propiedades para cada campo de una tabla dependiente. En VFP se han incorporado algunas nuevas, sobre todo pensando en la orientación a objetos. Esto se muestra en la figura 1.7.3.6

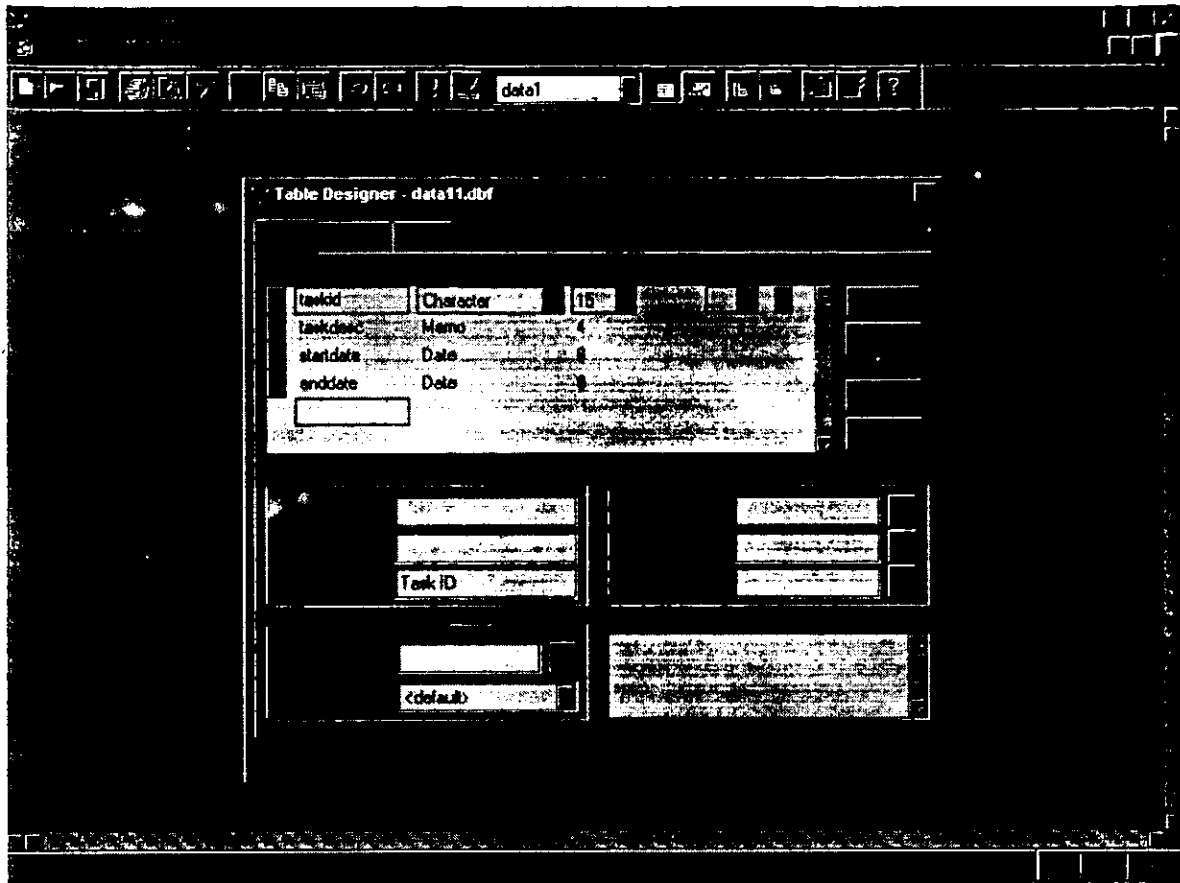


Figura 1.7.3.6 Propiedades de campo

CAPITULO II

PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2.1 SITUACIÓN ACTUAL

Actualmente la empresa cuenta con el apoyo de agentes externos, tales como bancos y tiendas comerciales (entre los que están: Gigante, Samborns, Sears, etcétera), para realizar el cobro del servicio telefónico, desafortunadamente no se cuenta con un soporte tecnológico adecuado o un sistema eficaz para explotar de una forma más sencilla la facilidad que estos agentes nos brindan como parte de un valor agregado para sus clientes.

Con el propósito de conocer la manera en que se realiza la cobranza a través de los agentes se presenta dos casos específicos ya que son los dos primeros agentes que comenzaron a trabajar en conjunto con la empresa para tratar de brindarle un mejor servicio a sus clientes.

Agente : Tiendas comerciales (Gigante).

Porcentaje de cobranza mensual: 10%

Situación: El agente procede a cobrar desde un punto determinado de la tienda, solo recibe pagos hasta 5 días antes de la fecha de corte del recibo posteriormente se recopilan los recibos y se elabora un reporte manual, los cuales son enviados vía mensajería al corporativo divisional de la tienda. Una vez que el corporativo recolecto la cobranza de sus tiendas correspondientes, lo cual le lleva tres días, y una vez clasificados todos los recibos y los reportes de cada tienda, son enviados nuevamente por mensajería a el área correspondiente dentro de la empresa , en este caso cobros especiales. Cuando se han recibido tanto los recibos como los reportes, el dinero correspondiente a esta cobranza es recibido por la empresa entre siete y quince días después a través de un cheque el cual es depositado un día después de su recepción por el área de cobros especiales que es el que lo recibe. El hecho de que exista un retraso o

defasamiento en la recepción de esta cobranza provoca que, en ocasiones, el cliente se vea afectado en su servicio, que puede llegar a la suspensión del mismo debido a que se retrasa el envío de la información por parte de los agentes, lo que causa que los pagos realizados por el cliente no se apliquen en tiempo. Además al cliente le genera desconfianza ya que no reconoce a los agentes como canal oficial de recolección de pagos, por lo que el cliente prefiere desplazarse a alguna oficina propia de la empresa.

Agente : Bancos.

Porcentaje de cobranza mensual: 31%.

Situación: En este caso el envío de la información (recibos) se realiza de la misma manera que la anterior, por mensajería, y el pago de la cobranza realizada por ellos se realiza de tres a cinco días después de que ellos la conciliaron. Por otra parte para aquellas personas que no son clientes de la institución que realiza la cobranza se les aplica un cargo de \$ 5.00 por el pago del recibo.

Como podemos ver se ha desaprovechado la gran ventaja que se tiene si consideramos que la mayoría de los clientes con los que cuenta la empresa acuden a las tiendas y centros comerciales en un promedio de dos veces al mes como mínimo, y que sí representa un beneficio, al poder contar con las mismas facilidades de pago que si lo hicieran en una de las propias oficinas receptoras de pago del servicio de la empresa.

También se puede tener un ahorro considerable en cuanto a infraestructura, tanto material como en recursos humanos, así como el poder contar con un sistema de información que pueda asegurar el control de la recolección de la cobranza realizada por terceros, mediante un sistema automatizado que permita hacer más rápidos los procesos de integración de la cobranza para ser enviados al sistema institucional con el

menor retraso posible (diferencia de un día si se hiciera en un centro de atención de la misma empresa) y con un mínimo porcentaje de errores.

Ahora bien, dentro de la empresa en el área receptora de la información también se siguen procesos lentos debido a que, precisamente, no se cuenta con un sistema que agilice esos procesos. A continuación describimos como se realizan esos procesos a través de las entidades de la empresa involucradas.

- Cobros especiales. Es el área que realiza la integración de la cobranza recibida vía mensajería, captura la cobranza en un archivo y con un formato definido para ser enviado al sistema institucional y elabora las formas contables correspondientes para el control de ingresos- egresos. Existen muchos problemas ocasionados por la forma en que se realiza este manejo manual de la información, tanto en su verificación y conciliación como en su almacenamiento, dicho manejo lo realizan a través de dos jefaturas de sección con el apoyo de 16 personas. Los errores más comunes son la recepción de talones o recibos defectuosos (rotos, rayados, etcétera), errores en las relaciones de cobranza de los agentes lo que obliga a revisar uno por uno cada recibo dado que en ocasiones se incluyen recibos de otro tipo de servicios que no corresponden al que la empresa proporciona.
- Control de ingresos. Esta área se encarga de la verificación y aclaraciones de los pagos realizados por el cliente en sucursales foráneas o a través de terceros. Cuando se tiene que realizar la verificación o alguna aclaración de algún pago, el proceso de seguir y encontrar en que parte del proceso seguido para el cobro y aplicación del pago se ha tenido algún error es lento y susceptible de más errores ya que se realizan muchos movimientos de manera manual. Otro gran problema es que casi diario se tiene que verificar cobranza duplicada, pagos mal aplicados, lo que acarrea consecuencias tanto en el servicio del usuario, ya que no contaría con un servicio al 100%, como problemas contables dado que se tienen que hacer muchos ajustes en las cuentas contables de las áreas correspondientes involucradas (contabilidad, comercial y finanzas) y en sí de toda la empresa.

2.2 REQUERIMIENTOS DEL USUARIO

Con lo descrito en el punto anterior y a petición de las dos áreas involucradas en el control de la cobranza realizada a través de terceros, existe la necesidad primordial de contar con un sistema que asegure el control adecuado de esta información, haciéndolo más rápido, eficiente y accesible con lo que se evitarían muchos errores. Así como la integración de los pagos de terceros en una cuenta bancaria única a nivel nacional, con lo que se evitarían más retrasos en la recuperación de la cobranza y en el depósito de la misma que harían los agentes externos (terceros) a la empresa.

Los requerimientos que el usuario tiene para poder realizar de una manera eficiente, tanto su trabajo como el manejo de la información, además de una adecuada y más rápida aplicación de los cobros realizados por los agentes externos (terceros), requiere de un sistema que le permita, realizar lo siguiente:

- Contar con una interface y una base de datos confiable y de fácil acceso de la información.
- Obtener una reducción en los tiempos para realizar una integración rápida y efectiva de la cobranza diaria de los agentes.
- Obtener una disminución en los tiempos y procesamiento de la información al automatizarlos.
- Obtener de los terceros la información mediante archivos con un formato seguro y que minimice los errores de captura.
- La realización de respaldos de la información periódicamente
- La obtención de los reportes necesarios para las áreas de cobros especiales y control de ingresos así como los que requiera los agentes.

- La transmisión del archivo con la cobranza integrada de todos los agentes al sistema institucional.
- Obtención de la información de la cobranza realizada por los terceros en tiempo y forma, así como la obtención del pago de los mismos mas rápidamente.
- Debe mantener la integridad y constancia de la información, con el propósito de que sea información confiable.
- Debe existir control de seguridad en los accesos a la información.
- Debe existir una bitácora de todas las transferencias de archivos realizadas con éxito así como las acciones o procesos realizados en el sistema.
- Debe contener un archivo histórico de movimientos
- Debe tener un modulo de depuración de archivos
- Debe usar un menú gráfico en todas las opciones que describan la acción a realizar
- Debe de ser amigable, y debe contener una ayuda en línea durante la navegación en el sistema y debe ser autónomo.
- Debe ser seguro y mantener la integridad del sistema, así como seguridad en el acceso a la información
- Que se minimicen los costos de realización utilizando Hardware y Software que posea la empresa.
- Que sea transportable a cualquier equipo Pc que cumpla requerimientos mínimos que permitan ejecutarlo sin problemas.

2.3 RECOPIACIÓN Y ANÁLISIS DE LA INFORMACIÓN

En el punto 2.1 vimos la situación actual y el procedimiento que se sigue para realizar la recopilación de la información de la cobranza realizada a través de agentes externos, por lo que en base a eso y a las necesidades que tiene el usuario obtendremos los documentos necesarios para basar la realización del sistema de cobranza por terceros. En esta etapa se conocen los requerimientos funcionales y no funcionales de las áreas usuarias, las características fundamentales con las que deberá contar el sistema con el fin de obtener el conocimiento preciso de todos los aspectos que involucren el sistema. Adicionalmente se genera una participación de las áreas usuarias en todo el proceso.

A continuación se enlistan los datos o perfiles de usuarios para el sistema con los que trabajara el sistema.

Número de Usuarios que usaran el sistema

Sistema de Cobranza por Terceros	
Supervisor del Sistema:	1
Gerente de Cobranza:	1
Cobros especiales	1
Control de ingresos	1
Coordinador de Terceros:	(Depende del no. de agentes dados de alta mínimo: 1 máximo: indefinido)
TOTAL:	Mínimo 5

Tamaño de la Base de Datos

Entidad: Cobros.
Volumen 160,000 registros por 45 días tomando en consideración 2 agentes
Observaciones Incremento diario de 3,000 registros promedio diarios
Depuración de 30 días al mes, transfiriéndolos al archivo histórico.

Entidad: Histórico de cobros.
Volumen: 720,000 registros
Observaciones: Incremento mensual de 110,000 registros promedio mensuales
Depuración de 30 días cada mes, respaldándolos en
unidad de cinta.

Nota: Cabe hacer mención que estas cifras son tomadas con dos agentes,
Esta misma información puede variar dependiendo del N° de agentes
dados de alta.

A continuación se muestran algunas formas contables internas que son llenadas con base en la cobranza realizada en general y que incluye, por supuesto, la realizada a través de los agentes externos (terceros).

En estas formas es ingresada la información obtenida de los reportes recibidos de la cobranza realizada por los agentes externos y que resulta en ocasiones muy elaborado ya que toda esa información recibida se contabiliza normalmente de manera manual ya que no se cuenta con la información en algún sistema o base de datos a la que se puedan realizar consultas rápidas y obtener esta información de manera rápida y más segura, ya que se evitarían errores al momento de hacer la concentración de la información

En la figura 2.3.1 se muestra la forma en donde se concentran los datos de los ingresos por caja (también llamada vía de cobro) y que para los agentes externos les corresponde un número de caja en particular, la cual es asignada por el área de finanzas. Es necesario que del sistema de cobranza se pueda obtener la información necesaria para esta forma, y de ser posible que emita un reporte con esas características y formato, de tal forma que se pueda anexar a los documentos oficiales para su revisión y envío al departamento de contabilidad.

**REPORTE DE COBRANZA DIARIA DE CAJA
SUSCRIPTOR GENERAL**

Lugar										Mes			Día			Año															
Caja	Reportado					diferencia					Caja	Reportado					Diferencia														
	Importe					P.D.						(+ -)					Importe					P.D.					(+ -)				
301											345																				
302											346																				
:											:																				
:											:																				
:											:																				
:											:																				
344											388																				

Figura 2.3.1 Reporte de cobranza diaria de cajas.

En la figura 2.3.2 observamos un ejemplo del informe diario de ingresos y aquí son registrados, entre otros datos, los depósitos que realizan los agentes externos de la cobranza que ellos recaudaron.

Las formas antes ilustradas son algunas de las que se tiene que ingresar datos obtenidos de la cobranza realizada a través de terceros y que por la manera de llevarse actualmente es susceptible de muchos errores ya que casi todo se realiza de manera manual. El sistema nos deberá proporcionar según requerimientos del usuario por lo menos el reporte diario de caja, el cual es requerido por el área de cobros especiales. Con lo que este reporte ya no deberá ser hecho manualmente.

Los agentes externos que actualmente proporcionan un reporte y un paquete con los recibos cobrados para que estos sean ingresados manualmente al sistema institucional, deberán ahora entregar archivos integrados los cuales tendrán que traer la siguiente información, formato o layout. Como se muestra en la figura 2.3.3. De igual manera en

TELEFONOS DE MEXICO S.A DE C.V Dirección de finanzas y administración Contraloría- Contabilidad de ingresos						INFORME DIARIO DE INGRESOS			
						Fecha :		Día	Mes
CONCEPTO (cargos)	Catalogo		Catalogo Nuevo			IMPORTE			
	Cta.scta.ss cta	UN	Ubic. Geo	Niv. afe c	Cuenta sub-cta	Este día	Acumula do	Para uso Fccin	
			A	P					
1									
2									
3									
4									
:									
:									
18									
Total Cargos									
Depósitos en:									
19									
20									
21									
22									
:									
:									
35									
Total Depósitos									
Total									
Observaciones:									
Formulo			Reviso			Aprobó			
Nombre y Firma			Nombre y Firma			Nombre y Firma			

F-1021-A

Figura 2.3.2 Formato del informe diario de ingresos.

ARCHIVOS DE COBRANZA (TERCEROS) LONGITUD DE ARCHIVO: 30

NUM	POS.	IDENTIFICACION	NO.	FORMATO	OBSERVACIONES
1	1-10	NUMERO DE TELEFONO	10	99	NUMERO TELEFONICO CON CLABE LADA
2	11-18	FECHA DE PAGO	8	X(8)	AA/MM/DD FECHA QUE PAGA EL CLIENTE
3	19-20	ESTACION DE COBRO	2	99	BANCO O COMERCIO: 01=BANAMEX; 02=BANCO LADATEL; 03=SERFIN; 04=BANPAIS; 05=BANCO BILBAO VIZCAYA; 06=CONFA; 07=BANORTE; 08=BANCRECER; 09=BANEJERCITO; 10=BANCOMER; 11=BITAL; 12=SANTANDER; 13=GIGANTE; 18=BANSUR; 19=ATLANTICA; 21=INBURSA; 22=BANCO UNION; 25=PROMEX; 26=BODEGA GIGANTE; 44=INVERLAT; 45=CITIBANK 46=SANBORNS
4	20-21	ESTACION DE COBRO	2	99	SUCURSALES 1-99
5	22-30	FECHA DE PROCESO	8	X(8)	AA/MM/DD FECHA DE ENVIO A TELMEX

Figura 2.3.3 Layout de archivo de agentes externos

ARCHIVO DE ENVIO A PISA

LONGITUD DE ARCHIVO: 47

NUM CAMP O	POS. DE - A	IDENTIFICACION	NO. BYTES	FORMATO	OBSERVACIONES
1	1-10	LADA - TELEFONO	10	9	LADA - TELEFONO DE PAGO
2	11-20	FECHA DE COBRO	10	X(10)	FECHA DE COBRO
3	21-31	IMPORTE	11	X(11)	IMPORTE DE PAGO
4	32-33	CONSTANTE	2	X(2)	"CE" CONSTANTE
5	34-36	VIA DE COBRO	3	9(3)	VIA DE COBRO
6	36-37	CONSTANTE	1	9(1)	"4"
7	38-47	FECHA DE PROCESO	10	X(10)	FECHA DE PROCESO O EVIO A PISA

Figura 2.3.4 Layout del archivo que se enviara al sistema institucional.

la figura 2.3.4 se muestra el formato o layout que deberá de tener el archivo que se generará para mandarlo al sistema institucional.

Básicamente es la información que maneja tanto el área de cobros especiales como el área de control de ingresos las cuales a su vez le reportan a contabilidad entregándoles copias de los reportes obtenidos con información de la cobranza realizada por los agentes externos, y es con base en estas necesidades básicas que se desarrollará el sistema. Otro punto importante es que las áreas de cobros especiales y de control de ingresos minimizarán sus problemas, dado que se evitarán muchos de los procesos manuales que actualmente se realizan, como son la captura de los recibos enviados por los Agentes externos.

2.4 PLANTEAMIENTO DEL PROBLEMA

La necesidad de tener un sistema como el que se plantea, se debe a que la empresa requiere realizar los procesos de cobranza más rápido, con el propósito de poder brindarle un servicio más eficiente al cliente y que sus pagos se apliquen en tiempo y forma, con el fin de evitarle problemas posteriores de pagos no aplicados y una posible suspensión de su servicio. Además de tener la posibilidad de ampliar la gama de opciones que pueda tener para realizar el pago de su servicio sin que tenga la necesidad de desplazarse a un centro de atención de la empresa. Lo anterior nos permite contar con horarios de cobro del servicio, más amplios y cómodos para el cliente, así como el de poder captar de una manera más rápida los pagos realizados en los establecimientos de los terceros.

El sistema de Cobranza por Terceros se deriva de la necesidad de proporcionar al cliente, mas centros donde pagar su recibo telefónico, evitando así la molestia de trasladarse a una oficina comercial de TELMEX para realizar dicho pago.

Con este proyecto se pretende ampliar las alianzas estratégicas con los centros comerciales mas concurridos por los clientes, los cuales aparte de realizar sus compras

normales, también puedan pagar su recibo telefónico teniendo la ventaja que incluso en días no hábiles puedan contar con el servicio, evitando perdida de su tiempo. Los objetivos primordiales son:

Otra finalidad que persigue la implementación de un sistema como el que se plantea es el hecho de que la empresa pueda recibir de una manera casi inmediata los depósitos de la cobranza realizada por los agentes externos y así evitar las demoras en este aspecto que actualmente son de hasta quince días, desde que se recibe los reportes y los recibos hasta que se realiza el deposito de dinero.

- Mejorar la calidad en la respuesta de los procesos.
- Eficientar de tal forma el proceso que se pueda atender a todas las Entidades Externas de manera rápida, precisa y confiable.
- Mejorar el performance de los procesos para obtener la información oportuna y confiable.
- Mejorar la atención al cliente mediante un registro o aplicación adecuada y oportuna de sus pagos.
- Obtener de manera rápida el pago de los servicios cobrados por los agentes externos.

2.5 DESCOMPOSICIÓN FUNCIONAL

Introducción

Dentro de este capítulo se mostrará el detalle de todos los procesos involucrados dentro del área de Cobros Especiales y de Control de Ingresos de la empresa, así como la

frecuencia en la que se realizan estos procesos. Esta información nos servirá como punto de partida el desarrollo del Sistema de Cobranza por Terceros.

Alcance Funcional del Sistema

Transmisión de archivos

Se realizan todas las funciones de transmisión de archivos de cobranza; para integrarlos a la base de datos, y posteriormente generar el archivo que será enviado a PISA (Proyecto de integración de Servicios de Atención a usuarios), que es el sistema Institucional. También se realizan tareas de respaldo de archivos para el envío a PISA y la restauración de archivos al servidor.

- Transmisión de archivos por PISA

Realizan la transmisión del archivo que contiene toda la cobranza del día a PISA

Procesos extraordinarios

Son procesos que realizan ciertas actividades extraordinarias y de ayuda para el control de los archivos como lo es la regeneración de los archivos enviados por PISA y algunas tareas de integración de cobranza y de archivos faltantes.

- Regeneración de archivo PISA

Se genera nuevamente el archivo que se envía a PISA con la información de la base de datos.

- Integración de cobranza (Total)

Realiza la integración de la cobranza de los terceros de un día en específico a la base de datos.

- Integración de archivos de terceros faltantes (Parcial).

Una vez recuperados los archivos de los terceros por la opción extracción de archivos de terceros a servidor se procede a integrar la información a la base de datos

Creación de Reportes

Reportes que se generan en las áreas de la empresa involucradas en la cobranza de los pagos por terceros.

- **Control de Pagos.**
Refleja el total cobrado por semana y el total pagado por los terceros, también proporciona los datos por estación de cobro, teléfono e importe por un día en específico
- **Reporte Diario de Caja.**
Indica el total cobrado por estación de cobro, el número total de recibos cobrados así como el total por vía de cobro.
- **Reporte Mensual.**
Proporciona: la fecha de cobranza, el número de recibos y el importe por día de una vía de cobro (agente) de todo un mes
- **Reporte semanal por vía de Cobro**
Proporciona los siguientes datos: población, estación de cobro, día y fecha de la semana consultada y total por semana y por vía

Bitácora

La información de todos los procesos es guardada diariamente (respaldo) y es utilizada como referencia para posteriores consultas de los procesos, acciones y transmisiones realizadas.

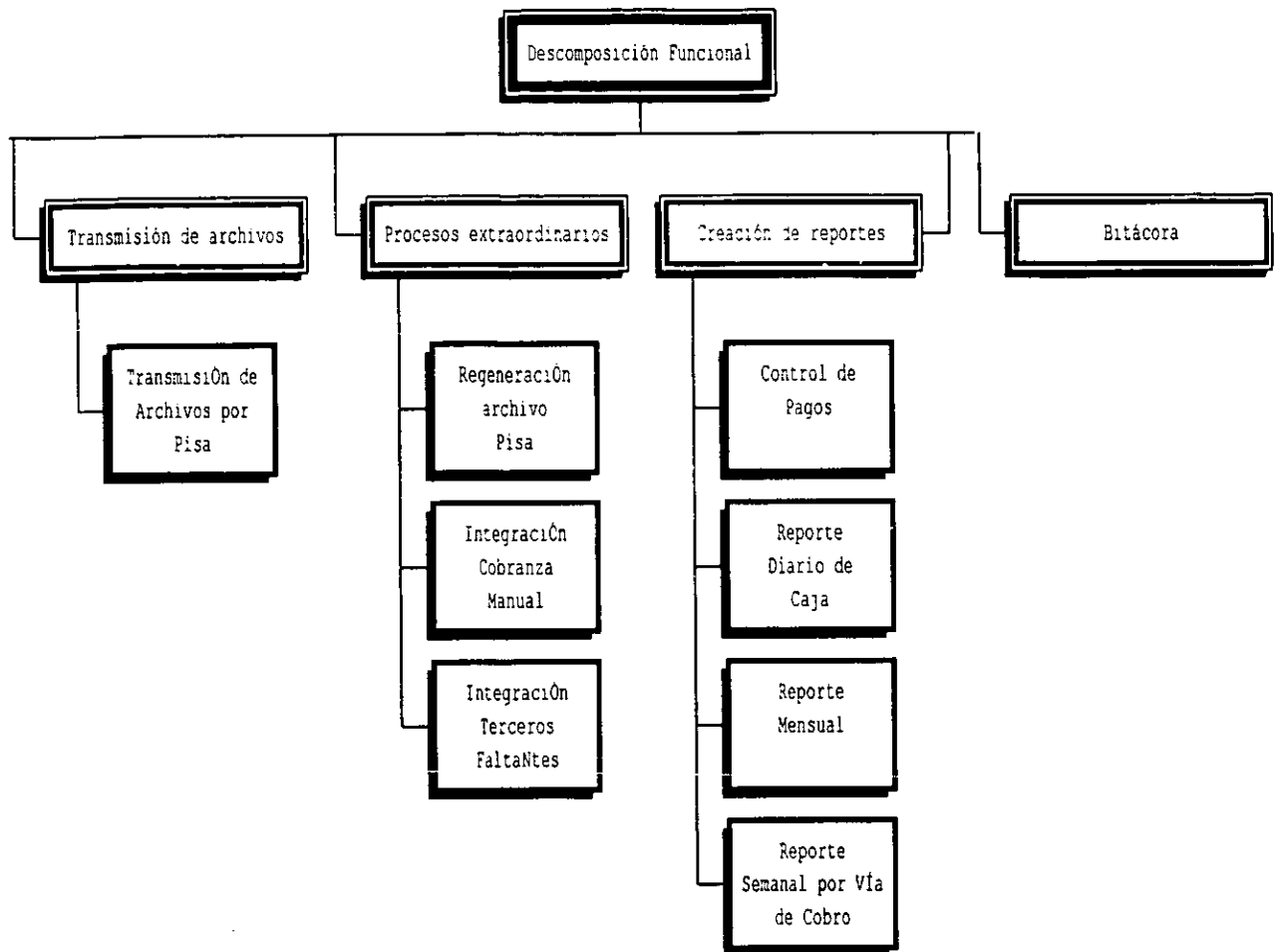


Figura 2.1.1 Descomposición funcional

Frecuencia de ejecución de procesos

Proceso: Transmisión de archivo PISA

Frecuencia: Diario

Proceso: Respaldo Archivo PISA para su envío

Frecuencia: Eventual

Proceso: Restaura archivo de cobranza de terceros para su integración a la base de datos

Frecuencia: Eventual

Proceso: Regeneración de archivo PISA para su envío
Frecuencia: Eventual

Proceso: Integración de cobranza (Total)
Frecuencia: Eventual

Proceso: Integración de archivos de terceros faltantes (Parcial)
Frecuencia: Eventual

Proceso: Ajuste tabla de cobros o histórico de cobros
Frecuencia: Eventual

Proceso: Ajuste a la tabla de pagos
Frecuencia: Eventual

Proceso: Consulta de Pagos
Frecuencia: Semanal

Proceso: Diario de Agente
Frecuencia: Diario

Proceso: Cobranza
Frecuencia: Eventual

Proceso: Reporte Diario de Caja.
Frecuencia: Diario

Proceso: Estadístico
Frecuencia: Mensual

Proceso: Reporte semanal por vía de Cobro

Frecuencia: Semanal

Proceso: Respaldo Semanal de tablas, catálogos y archivos de cobranza.

Frecuencia Semanal

Proceso: Restauración de Respaldos.

Frecuencia: Eventual

Proceso: Depuración de archivo histórico

Frecuencia: Trimestral

Proceso: Bitácora de Procesos

Frecuencia: Diario

2.6 OPCIONES DE SOLUCIÓN

Introducción

En este capítulo se analizarán las diferentes opciones de solución (lenguajes de programación y bases de datos) que pueden ser tomadas en cuenta para realizar el desarrollo e implementación del Sistema de Cobranza por Terceros.

Como seleccionar un manejador de bases de datos orientada a objetos

La tecnología orientada a objetos nos permite seleccionar un manejador de base de datos a aquellos que cumplan con las siguientes características:

- **Reutilización.** Las clases estas diseñadas para que se reutilicen en muchos sistemas. Para maximizar la reutilización, las clases se construyen de modo que se puedan adaptar. Un depósito debe estar poblado de una creciente colección de clases reutilizables. Es probable que las bibliotecas de clases crezcan rápidamente. Un objetivo fundamental de las técnicas orientadas a objetos es lograr la reutilización masiva al construir software.
- **Estabilidad.** Las clases diseñadas para una reutilización repetida se vuelven estables, de la misma manera que los microprocesadores y otros componentes se hacen estables. Las aplicaciones se construyen a partir de componentes de software cuando sea posible.
- **El diseñador piensa en términos del comportamiento del objeto y no en detalles de bajo nivel.** El encapsulamiento oculta los detalles y hace que las clases complejas sean fáciles de utilizar. Las clases son como cajas negras; el investigador utiliza la caja negra y no ve hacia el interior de esta. Solo debe entender el comportamiento de la caja negra y como comunicarse con ella.

- Se construyen clases cada vez más complejas. A partir de una clase se construyen otras clases, las cuales a su vez se integran entre sí. Así como los bienes manufacturados se fabrican a partir de una serie de materiales de partes y subpartes ya existentes, también el software se crea mediante una serie de materiales de clases ya existentes y probadas. Esto permite construir componentes complejos de software, que a su vez se convierten en bloques de construcción de software más rápido.

Opciones de solución:

A continuación se presentan algunas opciones de software para el desarrollo del Sistema de Cobranza por Terceros, así como algunas de sus principales características y un análisis comparativo entre ellos.

Front End	Back End
Visual Foxpro	Visual Foxpro
Visual Basic	Visual C++
Clipper	Clipper
Oracle	Oracle

Visual FoxPro 5.0 Microsoft Corp.

La programación ha sufrido en los últimos años numerosas modificaciones, uno de esos cambios a sido determinado por los entornos de programación visuales que exigen mayor rapidez en el desarrollo de aplicaciones. Lenguajes que resultaban áridos en su concepción se han hecho mucho más asequibles para los programadores. Visual Foxpro se sitúa como una de las herramientas que más acorde se encuentra con estos

requerimientos y por ello es elegida cada vez más como solución corporativa en desarrollo.

Visual Foxpro incorpora la programación orientada a objetos, que es considerada como una de las técnicas de programación más reconocidas. La cada vez mayor implantación de entornos Windows también exige que el lenguaje, además de funcionar sobre este, permita realizar aplicaciones con su estilo particular.

Visual Foxpro es uno de los más rápidos del mercado en acceso de datos y esta es la razón principal por la que tiene más aceptación en grandes corporaciones que a nivel de programador "casero".

Es un lenguaje que facilita el trabajo, no lo complica. De ahí a surgido un término para definir a estos lenguajes que permiten realizar aplicaciones en poco tiempo R.A.D. (Rapid Application Development).

Visual Foxpro incorpora capacidades como son la programación en internet y la arquitectura cliente-servidor.

Una de las desventajas de visual Foxpro es que no se dispone de gran bibliografía, cursos ni artículos en revistas, sin embargo esta desventaja disminuye en gran medida por su ayuda en línea y soluciones de programación basadas en aplicaciones de ejemplos que son incluidas con el software.

El uso de la tecnología Rushmore hace que Visual Foxpro sea una de las herramientas más rápidas del mercado. Rushmore es una técnica de búsqueda interna que aprovecha los índices de la tabla para acceder a grupos de registros. Comprende diversas técnicas, entre ellas están la forma de indexación, compresión de datos, optimización de consultas, almacenamiento de bitmaps, gestión de entrada/salida de datos y gestión de memoria.

Microsoft esta empleando ya esta característica en otros productos como Access y Visual Basic.

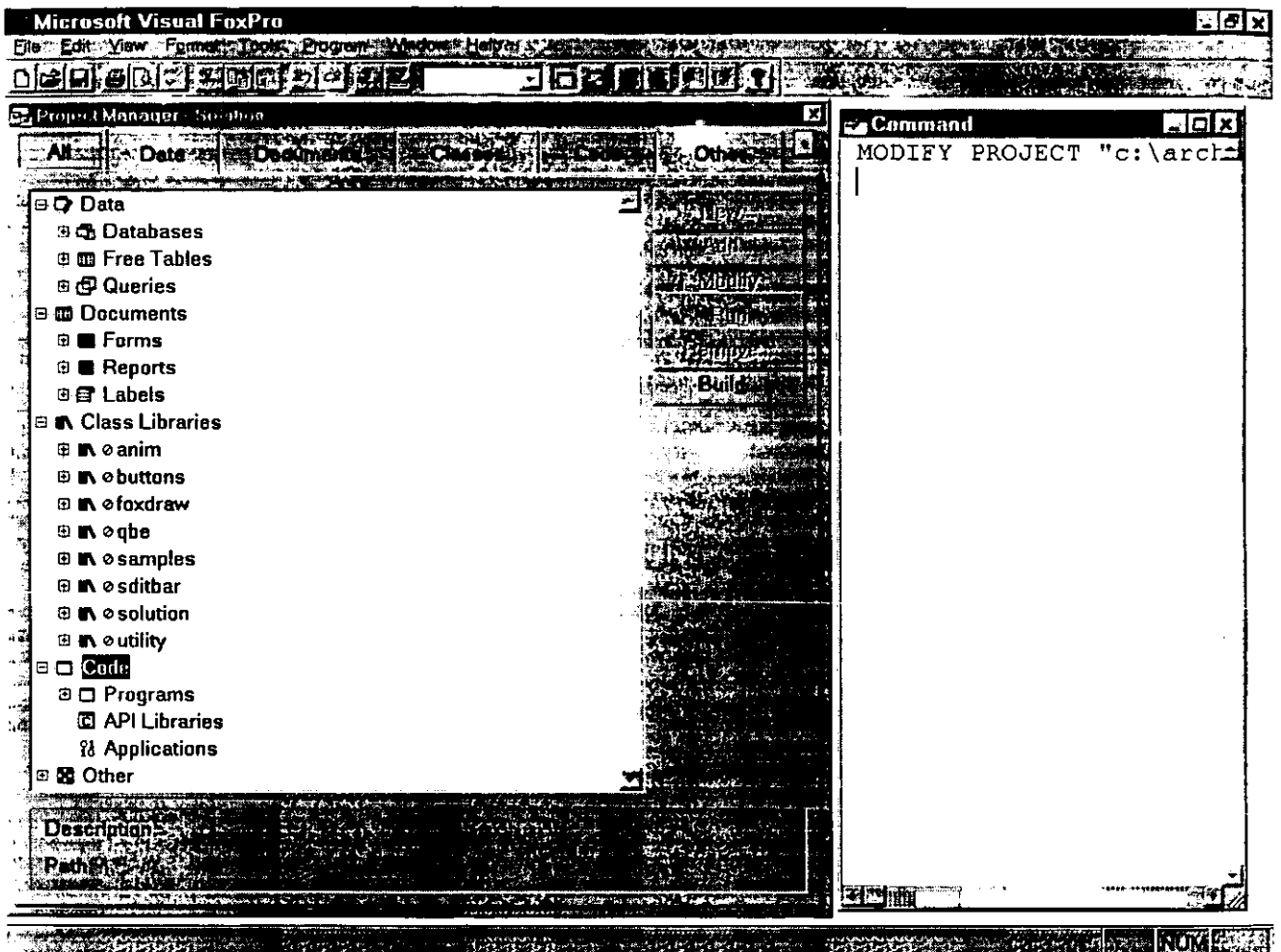


Figura 2.6.1 Pantalla principal de Visual Foxpro

Visual Basic 5.0 Microsoft Corp.

Visual Basic es un lenguaje de programación de entorno visual que ha sido muy aceptado, ya que tiene como principales características poder, flexibilidad y velocidad. Es un lenguaje interactivo que permite que su aprendizaje sea rápido y por lo mismo el desarrollo de aplicaciones Windows también

Utiliza varias formas para sus presentaciones visuales de sistemas, una "forma" en Visual Basic es una ventana asociada a controles, iconos, gráficos y código. La aplicación "colorbar" permite colorear gráficos.

Puede utilizar dibujos, bases de datos y aplicaciones basadas en Windows ya existentes para crear soluciones personalizadas.

Mediante Visual Basic se pueden generar Cambios de Datos Dinámicos (DDE) para aplicaciones basadas en ventanas; trabaja con bases de datos en una interacción y Objetos Ligados y Empotrados (OLE).

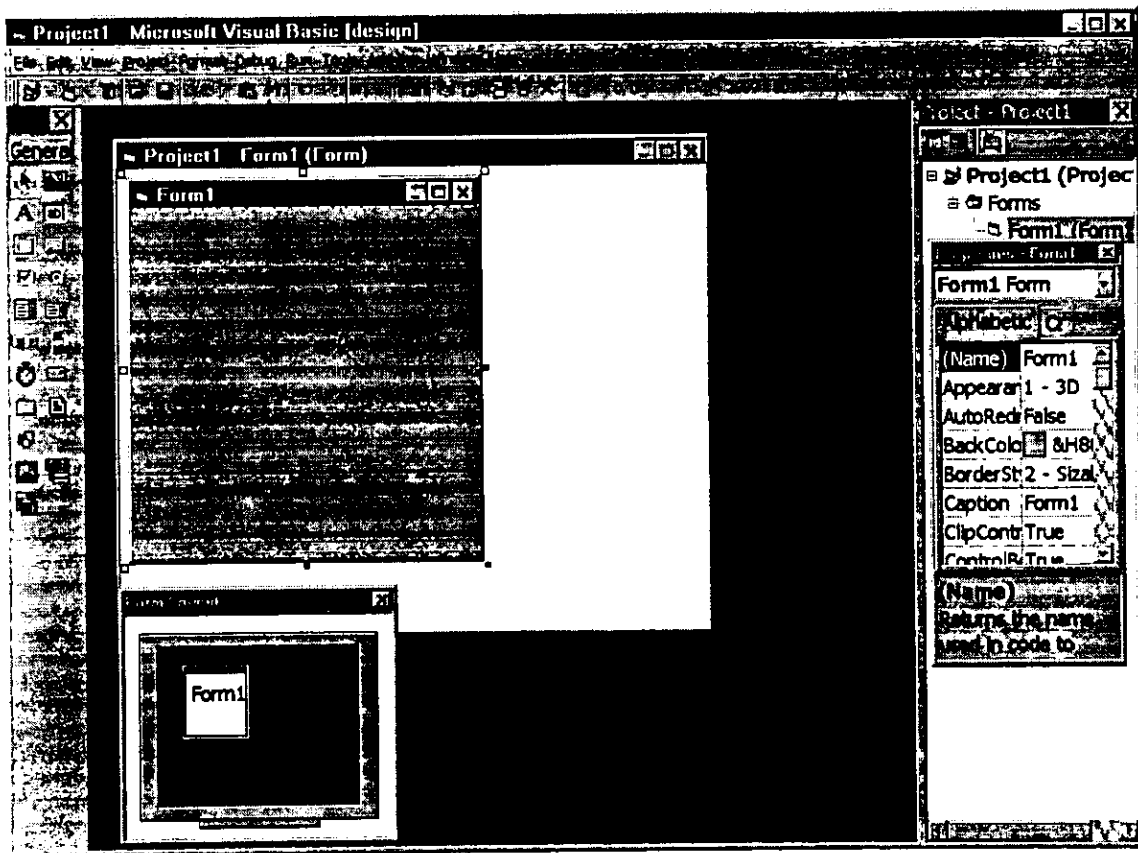


Figura 2.6.2 Pantalla principal de Visual Basic

Visual C++ Edición Profesional Microsoft Corp.

Visual C++ esta familiarizado con el lenguaje de programación C++ y el Desarrollo de Software para Windows (SDK) para lenguaje C.

Las librerías de C++ son parte importante en este lenguaje, ofrece librerías de aplicación como son la Interface de Documentos Múltiples para Windows (MIDI), la interface de programación de Conectividad de Bases de Datos Abiertas de Microsoft (ODBC) y Librerías de Ligados Dinámicos (DLL), así como también Objetos Ligados y Empotrados (OLE).

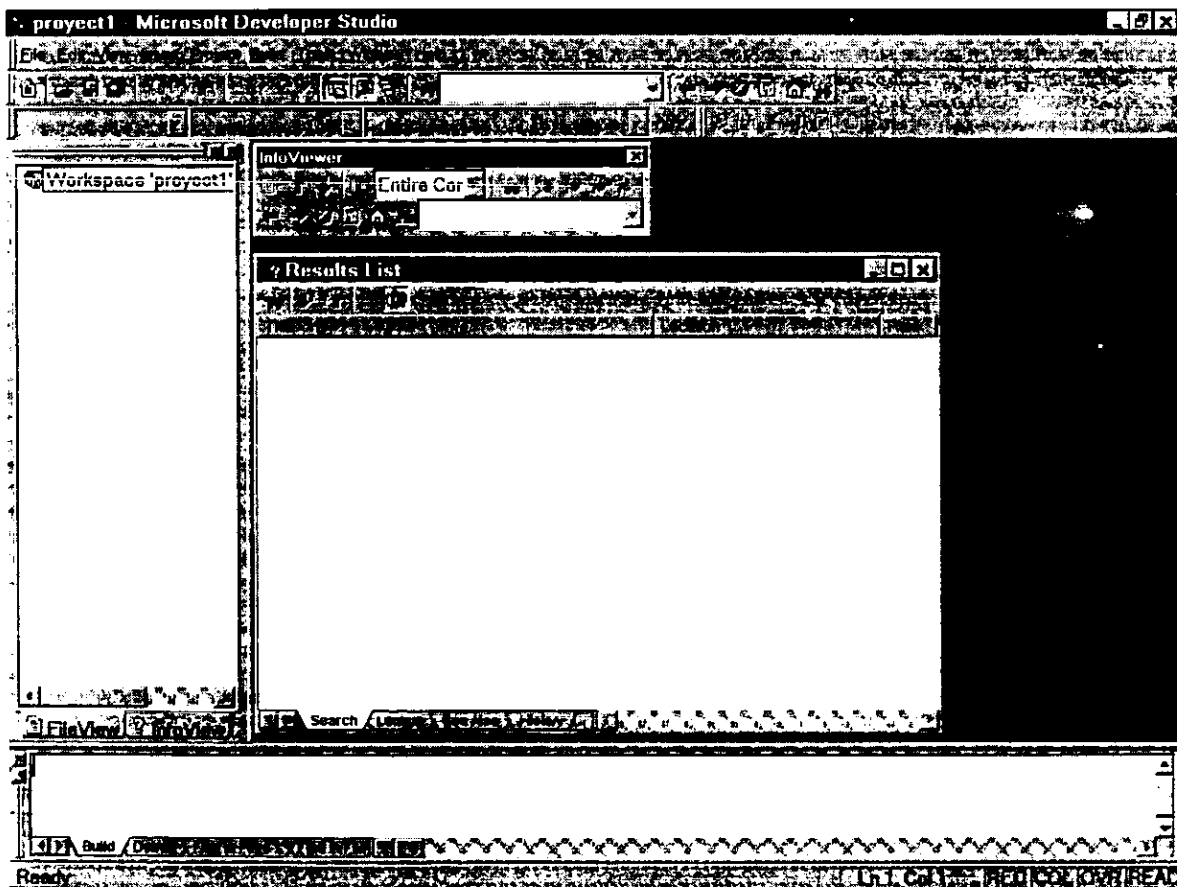


Figura 2.6.3 Pantalla principal de Visual C++

Presenta una tecnología mejorada del compilador, genera aplicaciones y componentes más rápidos y de menor tamaño, al volver a compilar el código existente se reduce su tamaño en un 10 % y aumenta la velocidad en otro 10 %.

C++ proporciona gran velocidad y capacidad de manejo de memoria, el desarrollo de programas se puede realizar en un ambiente gráfico parecido a Windows, con iconos y ventanas para producir sistemas amigables, seguros, confiables, de fácil manejo para el usuario y con calidad en los productos.

Clipper 5.3 Nantucket

Este software para el desarrollo de bases de datos a sido uno de los más utilizados por los programadores. Dos de sus capacidades más importantes son el generador de reportes (RL) y una función (DBU) para la creación y manejo de los archivos de la base de datos, escritos en el propio lenguaje de programación Clipper, y de los cuales se incluye el código fuente, que puede utilizarse como referencia o modificarse para agregarlo a las aplicaciones de la base de datos.

El compilador del programa es muy rápido, soporta los llamados bloques de código, pedazos de código ejecutable que se pueden almacenar como variables o pasar a otros programas como argumentos para ejecutar un bloque de código mediante la función EVAL().

Se puede mandar llamar la función MiMake para realizar operaciones de compilación, enlace y mantener los archivos sincronizados. Soporta las funciones para leer archivos binarios de DOS y escribir en ellos. El depurador de Clipper permite analizar el funcionamiento de código del programa, ejecutar comandos y revisar el estado de las variables, nombres de campos o expresiones en particular.

Oracle 8.0 Oracle Corp.

El servidor de datos Oracle es fácil de usar, provee una muy buena relación/rendimiento y buenas capacidades, incluye un conjunto integrado de herramientas de administración, distribución total, reaplicación y funciones de WEB.

Entre sus principales características se encuentra el control de concurrencia y la obtención de resultados confiables, lo cual logra mediante el bloqueo a nivel de registro sin restricciones, consultas libres de contención y ejecución de consultas sin bloqueo, multiversión y consistencia

El acceso a los datos de transacción puede realizarse de las siguientes formas; simple y columnas concatenadas, tablas en cluster, en función específica de la aplicación, basándose en los resultados de consultas directamente de la búsqueda de índices y los índices bitmap integrados.

Oracle tiene una alta disponibilidad mediante respaldo en línea por archivo, tablespace o base de datos, recuperación en línea, archivos de bitácora en espejo y multiseguimiento, redimensionamiento dinámico y automático de los archivos de la base de datos.

Una de sus capacidades es la de realizar consultas y transacciones distribuidas transparentes, joins distribuidos y transparencia de localización y red.

La reaplicación de datos puede realizarse de forma múltiple "snapshot" de solo lectura, de forma total (consistencia e integridad de datos), abarcando tablas completas y por subconjuntos y la actualización es basada en eventos por demanda.

A continuación se presentan cuadros comparativos entre el software analizado para el manejo de bases de datos orientadas a objetos, basándose en los siguientes puntos:

- Facilidad de uso
- Facilidad de programación
- Tiempo de procesamiento en Batch
- Tiempo de importación y generación de índices
- Número máximo de archivos abiertos
- Seguridad
- Diversidad en tipos de datos
- Manejo de gráficos para formularios

Facilidad de uso	Bueno	Regular
Visual C++	✓	
Visual Basic	✓	
Visual Foxpro	✓	
Clipper		✓
Oracle		✓

Tabla 2.6.1 Evaluación de facilidad de uso

Facilidad de programación	Bueno	Regular
Visual C++	✓	
Visual Basic	✓	
Visual Foxpro	✓	
Clipper		✓
Oracle	✓	

Tabla 2.6.2 Evaluación de facilidad de programación

Tiempo de procesamiento en Batch	Bueno	Regular
Visual C++		✓
Visual Basic		✓
Visual Foxpro	✓	
Clipper		✓
Oracle		✓

Tabla 2.6.3 Evaluación de tiempo de procesamiento en batch

Tiempo de importación y generación de índices	Bueno	Regular
Visual C++		✓
Visual Basic		✓
Visual Foxpro	✓	
Clipper		✓
Oracle	✓	

Tabla 2.6.4 Evaluación de tiempo de importación y generación de índices

Número máximo de archivos abiertos	Bueno	Regular
Visual C++	✓	
Visual Basic		✓
Visual Foxpro		✓
Clipper		✓
Oracle	✓	

Tabla 2.6.5 Evaluación de número máximo de archivos abiertos

Seguridad	Bueno	Regular
Visual C++	✓	
Visual Basic	✓	
Visual Foxpro	✓	
Clipper		✓
Oracle	✓	

Tabla 2.6.6 Evaluación de seguridad

Diversidad en tipos de datos	Bueno	Regular
Visual C++		✓
Visual Basic		✓
Visual Foxpro		✓
Clipper		✓
Oracle	✓	

Tabla 2.6.7 Evaluación de diversidad en tipos de datos

Manejo de gráficos para formularios	Bueno	Regular
Visual C++		✓
Visual Basic		✓
Visual Foxpro		✓
Clipper		✓
Oracle		✓

Tabla 2.6.8 Evaluación de manejo de gráficos para formularios

Asignando las siguientes puntuaciones, obtenemos la tabla de resultados finales (Tabla 2.6.9)

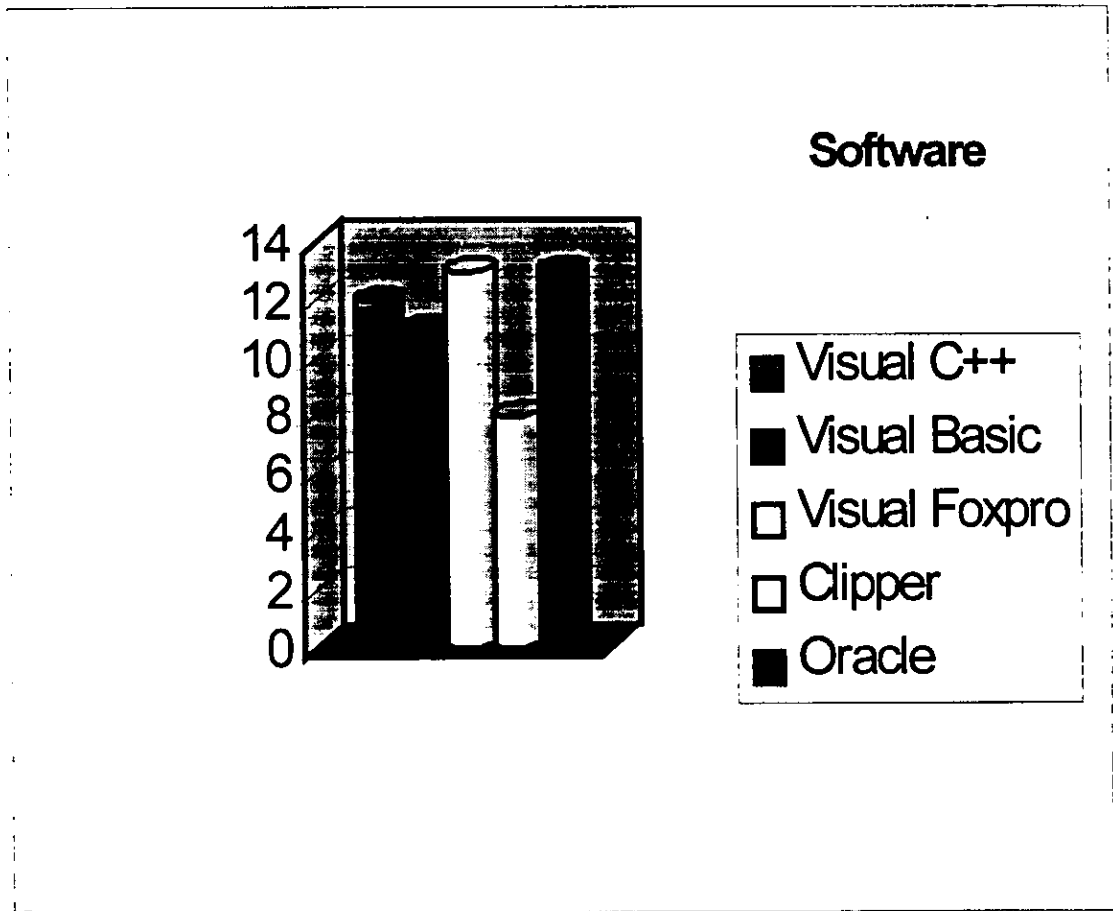
Bueno = 2 puntos

Regular = 1 punto

Características	Visual C++	Visual Basic	Visual Foxpro	Clipper	Oracle
Facilidad de uso	2	2	2	1	1
Facilidad de programación	2	2	2	1	2
Tiempo de procesamiento en Batch	1	1	2	1	1
Tiempo de importación y generación de índices	1	1	2	1	2

Número máximo de archivos abiertos	2	1	1	1	2
Seguridad	2	2	2	1	2
Diversidad en tipos de datos	1	1	1	1	2
Manejo de gráficos para formularios	1	1	1	1	1
TOTAL	12	11	13	8	13

Tabla 2.6.8 Resultados finales



Gráfica 2.6.1 Resultados Finales

2.7 ELECCIÓN DE SOLUCIÓN ÓPTIMA

Introducción

En la presente sección se determinan el software y hardware a utilizar para la implementación del sistema de cobranza por terceros, basándonos en las necesidades, requerimientos, hardware y software disponibles del área de cobranza de la compañía, además de las características de los lenguajes de programación y software de base de datos analizadas anteriormente.

Software seleccionado

Se ha determinado utilizar tanto para el Front End como para el Back End el lenguaje de programación Visual Foxpro ya que ofrece muy buenas características para el desarrollo de aplicaciones de gestión y también con la finalidad de tener integrado en el mismo ambiente de trabajo, el manejo de la base de datos, la interfaz de usuario y la obtención de reportes. Además es una de las herramientas homologadas de la empresa por lo que el área de cobranza cuenta con el software y licencias para utilizarlo.

Como sistema operativo se utilizará Microsoft Windows NT Server 4.0 para el servidor y los equipos de los usuarios cuentan con Windows 95.

Microsoft Visual Foxpro versión 5.0 principales características:

- Entorno de programación visual
- Programación orientada a objetos
- El más rápido acceso a datos del mercado
Tecnología Rushmore

- Desarrollo de aplicaciones en poco tiempo
R.A.D (Rapid Application Development)
- Capacidad de programación en internet
- Arquitectura cliente-servidor

Actualmente en la empresa se están migrando los últimos servidores Novell Netware a Windows NT Server, por lo cual se determina utilizar este sistema operativo para el servidor de datos del sistema. *

Hardware seleccionado

Los clientes de trabajo serán los actuales equipos de los usuarios con S. O. Windows 95 y cuyas características mínimas son las siguientes:

- Procesador Pentium -100 MHz-
- Memoria RAM de 32 MB
- Disco duro de 1 GB
- 2 ranuras de expansión libres ISA
- 2 ranuras de expansión libres PCI
- Tarjeta de red PCI (Operación Dual 10BASET/100BASETX)
- Monitor color SVGA
- Mouse (bus)

- 2 puertos seriales y 1 puerto paralelo

El área de cobranza ya cuenta con un servidor Windows NT, pero será necesario implementar el Sistema de Cobranza por Terceros en un nuevo servidor. *

Características del servidor Windows NT recomendadas

- Servidor Dual Pentium II (350 Mhz)
- Memoria RAM ECC de 256 MB
- Unidad de CD-ROM 8X (SCSI) -interna- booteable
- Unidad de Respaldo 24 GB (SCSI) -interna-
- RAID con 3 discos duros Hot-Swap de 9 GB Hot-Swap
- Drive interno de 3.5"
- 3 bahías Hot-Swap disponibles para crecimiento futuro
- 3 ranuras de expansión libres PCI
- Tarjeta de red PCI (Operación Dual 10BASET/100BASETX)
- 2 puertos seriales y 1 puerto paralelo
- Mouse (bus)
- Monitor SVGA color 15"

- UPS con interface al servidor (1200 Watts)

Requerimientos para el Desarrollo del Sistema

- Acceso a las instalaciones físicas.
- Disponibilidad de equipos, ambientes de desarrollo, y autorización para utilizarlos de acuerdo a los lineamientos internos de la organización.
- Licencias de paquetes de comunicaciones, herramientas de desarrollo y software de base de datos.

Hardware:

- Servidor Windows NT Server 4.0 con las características antes mencionadas
- Ruteador CISCO 2509
- UPS SmartPro 3000
- PC Pentium 100 MHz., 32 MB en RAM, 1G en Disco Duro. Para realización de pruebas

Software:

- Comunicaciones para CISCO 2509
- Visual Foxpro versión 5.0
- Windows NT Server v4.0

- Windows NT Service Pack 3.0

Contribución del Sistema a la solución del Problema

Con el sistema propuesto, el Departamento de Cobranza de la empresa obtendrá los siguientes beneficios:

- Dispondrá de información correcta y oportuna referente a la cobranza por terceros
- Se tendrá un enlace de comunicación eficiente, lo cual permitirá tener la capacidad adecuada para atender a todos los centros de cobranza, así como tener una reducción del tiempo de transferencia de datos entre el sistema y todas las entidades externas
- Se podrán procesar el volumen de transacciones requerido de todos los centros de cobranza sin tener problemas con la capacidad de procesamiento de información y en un tiempo mucho menor al actual.
- La captura de la cobranza por terceros será de una forma práctica y generará archivos para transferencia sin error.
- La información de la cobranza por terceros estará en el tiempo establecido, por lo que el pago de los terceros podrá realizarse oportunamente.
- El cliente podrá pagar su recibo telefónico con los mismos beneficios que ofrece la oficina comercial, con la ventaja que aprovecha los horarios de los establecimientos comerciales y bancos.
- La compañía recibirá con oportunidad los pagos realizados para realizar las acciones necesarias, mejorando su imagen y calidad de servicio con sus clientes.

CAPITULO III

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA

3.1 BACK END PARA CADA MÓDULO

Alcance Funcional del Sistema

Descripción Detallada:

En esta parte se mostrará el detalle de todos los procesos involucrados dentro del desarrollo del sistema de cobranza por terceros, como se muestra en la figura 3.1.1

Transferencia de archivos (módulo donde se realizan todas las funciones de transmisión de archivos de cobranza tanto de terceros – servidor con servidor – y PISA).

- Transmisión y extracción de archivos (se realizan todas las funciones de extracción de archivos de cobranza de terceros y el envío del archivo de la cobranza del día al sistema institucional PISA).
- Extracción de archivos (en el caso que algún tercero no haya realizado su transmisión de la cobranza por cualquier causa, el responsable de la coordinación de los terceros deberá realizar la extracción del archivo manualmente, el sistema le solicitara los siguientes datos: - Vía de Cobro – Estación de Cobro – Nombre del Archivo (sin extensión)).
- Transmisión de archivos PISA (módulo donde se realiza la transmisión del archivo que contiene toda la cobranza del día a PISA).
- Operaciones con archivos vía diskette (módulo de contingencia que tiene como objetivo respaldar o restaurar los archivos de cobranza del servidor para su envío a PISA).

- Respaldo de archivos PISA (respalda archivo de la cobranza total para su envío a PISA).
- Restauración de archivos PISA (restaura archivo de cobranza al servidor para integrar información a la base de datos).

Procesos Especiales (Procesos especiales del sistema que realizan ciertas actividades extraordinarias y de ayuda para el usuario).

- Regeneración de Archivos PISA (genera nuevamente el archivo que se envía a PISA con la información de la base de datos).
- Integración de Cobranza (Realiza la integración de la cobranza de los terceros de un día en específico a la base de datos)
- Integración de Archivos de Terceros Faltantes (una vez recuperados los archivos de los terceros por la opción extracción de archivos de terceros al servidor se procede a integrar la información a la base de datos).
- Módulo de Ajustes (poder hacer altas, bajas, modificaciones y consultas a las tablas de cobros, histórico de cobros y pagos).
- Ajustes a la Tabla de Cobros (altas, bajas, cambios, consultas a las tablas de cobros).
- Ajuste a la Tabla de Pagos (poder hacer cambios a la tabla de pagos).

Generación de Reportes. (módulo generador de todos los reportes del sistema).

- Control de Pagos (refleja el total cobrado y el total pagado por los terceros así como la estación, teléfono el importe por un día en específico).

- **Captura de Pagos de Agentes (consulta en pantalla)** (refleja el total cobrado y el total pagado por los terceros).
- **Diario de Agente (Consulta en pantalla)** (indica la estación de cobro, el número telefónico y el importe pagado de un día en específico).
- **Verificación de Pagos de Clientes (Consulta en pantalla)** (solicitando la fecha, vía de cobro y el teléfono, podemos saber datos esenciales de un cliente).
- **Reporte Diario de Caja** (indica el total cobrado por estación de cobro, el número total de recibos cobrados así como el total por vía de cobro).
- **Reporte Estadístico** (proporciona la fecha de cobranza, el número de recibos y el importe por día de una vía de cobro (agente) de todo un mes).
- **Reporte por Vía de Cobro** (Proporciona los siguientes datos: Población, Estación de Cobro, Día y Fecha de la Consulta y Total por Vía de Cobro).

Mantenimiento de Catálogos (Módulo de mantenimiento a los catálogos del sistema).

- **Mantenimiento al Catalogo de Agentes** (altas, bajas, modificaciones y consultas al catalogo de agentes (vía de cobro)).
- **Mantenimiento al Catálogo de Estaciones de Cobro** (altas, bajas, modificaciones y consultas al catalogo de estaciones de cobro).
- **Mantenimiento al Catalogo de Usuarios** (altas, bajas, modificaciones y consultas al catalogo de usuarios).
- **Mantenimiento al Catálogo de Grupos** (altas, bajas, modificaciones y consultas al catalogo de grupos).

Utilerias (módulo de utilerias del sistema).

- Depuración (módulo de depuración de bitacoras y tablas del sistema).
- Depuración de Bitácora de Procesos (proceso que depura la tabla que contiene la bitácora de procesos para su optima utilización y minimizar el tiempo de consulta de los procesos).
- Depuración de Bitácora de Transmisiones (proceso que depura la tabla que contiene la bitácora de transmisiones para su optima utilización y minimizar los tiempos de consulta de transmisiones).
- Depuración de la Tabla de Pagos (proceso que depura la tabla de pagos cuando el número de registros contenidos en dicha tabla es muy grande).
- Depuración de la Tabla de Cobros (proceso que depura la tabla de histórico de cobros en el momento en que el volumen de registros es considerable).
- Respaldo de Archivos (realiza el respaldo de la semana tanto de los archivos de la cobranza de envío como de los terceros recibidos así como las tablas y catálogos).
- Restauración de Respaldos (realiza la restauración de los archivos de la cobranza así como de las tablas y catálogos).
- Cambio de Password (cambiar el password de los usuarios del sistema).

Bitácoras (módulo de consulta de bitácora de todos los procesos y acciones hechas en el sistema, así como de las estaciones de cobro recibidas en una fecha específica).

- Consulta Bitácora de Procesos (muestra todas las acciones registradas proporcionando usuario, fecha, hora y acción realizada).
- Consulta Bitácora de Transmisiones (contiene la información de las transferencias).

Frecuencia de Ejecución de Procesos

Sistema de Cobranzas por Terceros.

Procesos: Extracción de archivos de terceros a servidor

Frecuencia: Eventual

Proceso: Transmisión de archivo PISA

Frecuencia: Diario

Proceso: Respaldo archivo PISA

Frecuencia: Eventual

Proceso: Restaura archivo PISA

Frecuencia: Eventual

Proceso: Regeneración de archivo PISA para su envío

Frecuencia: Eventual

Proceso: Integración de la cobranza

Frecuencia: Eventual

Proceso: Integración de archivos de terceros faltantes

Frecuencia: Eventual

Proceso: Ajuste tabla de cobros

Frecuencia: Eventual

Proceso: Ajuste a la tabla de pagos

Frecuencia: Eventual

Proceso: Consulta de pagos (consulta en pantalla)

Frecuencia: Semanal

Proceso: Diario de agente (consulta en pantalla)

Frecuencia: Diario

Proceso: Cobranza (consulta en pantalla)

Frecuencia: Eventual

Proceso: Reporte diario de caja

Frecuencia: Diario

Proceso: Estadístico

Frecuencia: Mensual

Proceso: Por vía de cobro

Frecuencia: Eventual

Proceso: Mantenimiento al catálogo de agentes

Frecuencia: Eventual

Proceso: Mantenimiento a estaciones de cobro

Frecuencia: Eventual

Proceso: Mantenimiento al catálogo de usuario

Frecuencia: Eventual

Proceso: Respaldo de tablas, catálogos y archivos de cobranza

Frecuencia: Eventual

Proceso: Restauración de respaldos

Frecuencia: Eventual

Proceso: Depuración de la tabla de cobros

Frecuencia: Eventual

Proceso: Bitácora de procesos

Frecuencia: Diario

Proceso: Bitácora de terceros recibidos

Frecuencia: Diario

Proceso: Carga inicial de la base de datos

Frecuencia: Una vez en la historia de la aplicación

Descripción de Procesos:

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Extracción de Archivos

Objetivo:

(En el caso algun tercero no haya proporcionado su información de la cobranza por cualquier causa, a la hora especificada, el responsable de la coordinación de los terceros deberá realizar la petición nuevamente del archivo paa integrarlo a la

base de datos, el sistema le solicitara los siguientes datos: - Vía de Cobro – Estación de Cobro – Nombre del archivo (sin extensión)).

Frecuencia:	1-3 veces/semana
Usuario :	Cobros especiales
Responsable de operación:	Cobros especiales

Entradas

Archivo de Cobranza

Salidas

Archivo de Cobranza

Entidades

Bitácora de procesos

Estaciones de Cobro

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Transmisión de Archivos PISA

Objetivo:

Módulo donde se realiza la transmisión del archivo que contiene toda la cobranza del día a PISA.

Frecuencia:	1 vez/día
Usuario:	Cobros especiales
Responsable de operación:	Cobros especiales

Entradas

Archivo PISA

Salidas

Archivo de proceso PISA

Entidades

Bitácora procesos

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Respalda Archivos PISA

Objetivo: Respalda archivos de PISA

Frecuencia:	1 vez/mes (eventual)
Usuario:	Cobros especiales
Responsable de operación:	Cobros especiales

Entradas

Archivo PISA

Salidas

Archivo de proceso PISA

Entidades

Bitácora procesos

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Restauración de archivos PISA

Objetivo:

Integrar el archivo de la cobranza del tercero por Diskette para su integración a la base de datos

Frecuencia:	1 vez/mes (eventual)
Usuario:	Administrador del Sistema
Responsable de operación:	Administrador del Sistema

Entradas

Archivo de Cobranza

Salidas

Archivo de Cobranza

Entidades

Bitácora procesos

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Conversión de Archivo

Objetivo:

Integra la información de los archivos de cobranza a tablas temporales para realizar la validación de la información.

Frecuencia: 1 vez/día

Usuario: BATCH

Responsable de operación: BATCH

Entradas

Archivo de Cobranza

Salidas

Entidades

Temporal de Cobranza

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Inserción en Base de Datos

Objetivo:

Realiza la integración de la cobranza de los terceros de un día en específico a la base de datos

Frecuencia: 1 vez/día

Usuario: BATCH

Responsable de operación: BATCH

Entradas

Salidas

Entidades

Poblaciones	E
Estaciones de Cobro	E
Agentes	E
Divisiones	E
Temporal de Cobranza	E
Cobros	E/S
Bitácora Transmisiones	S
Bitácora de Procesos	E

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Generación de Archivo PISA

Objetivo:

Generar archivo de cobranza para ser procesado en PISA

Frecuencia: 1 vez/semana

Usuario: BATCH

Responsable de operación: BATCH

Entradas**Salidas**

Archivo de Proceso PISA

Entidades

Bitácora Procesos	S
Cobros	E

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Genera Consulta de Pagos

Objetivo:

En este módulo se va a poder identificar el importe cobrado de los terceros, así como registrar su pago en una fecha específica

Frecuencia:	1 vez/semana
Usuario:	Cobros Especiales y Contador Terceros
Responsable de operación:	Cobros Especiales y Contador Terceros

Entradas

Salidas

Consulta de pagos

Entidades:

Bitácora Procesos	S
Pagos	E/S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Diario de Agente

Objetivo:

Identificar cada uno de los pagos hechos en las diferentes estaciones de cobro en una determinada

Frecuencia:	2 veces/semana
Usuario:	Cobros Especiales y Contador Terceros
Responsable de Operación:	Cobros Especiales y Contador Terceros

Entradas

Salidas

Consulta de diario de agente

Entidades:

Cobros	E
Divisiones	E

Agentes	E
Bitácora Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Verificación de Pagos de Clientes

Objetivo:

Identificar cada uno de los pagos hechos en las diferentes estaciones de cobro en una fecha determinada.

Frecuencia:	2 veces/semana
Usuario:	Gerente de Cobranza
Responsable de Operación:	Gerente de Cobranza

Entradas

Salidas

Consulta de verificación de pagos

Entidades

Cobros	E
Bitácora Procesos	S
Agentes	E

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Reporte Diario de Caja

Objetivo:

Conocer el desglose de la cobranza diaria por estación de cobro así como el total de la cobranza del día y número de recibos.

Frecuencia:	1 vez/día
Usuario:	Gerente de Cobranza

Responsable de Operación: Gerente de Cobranza

Entradas

Salidas

Reporte diario de caja

Entidades

Cobros	E
Poblaciones	E
Agentes	E
Bitácora Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Reporte Estadístico

Objetivo:

Genera Reporte Estadístico Mensual

Frecuencia: 1 vez/mes

Usuario: Gerente de Cobranza

Responsable Operación: Gerente de Cobranza

Entradas

Salidas

Reporte estadístico

Entidades

Cobros	E
Agentes	E
Pagos	E
Bitácora Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Reporte por Vía de Cobro

Objetivo:

Genera Reporte por Vía de Cobro

Frecuencia: 1 vez/semana
Usuario: Gerente de Cobranza
Responsable Operación: Gerente de Cobranza

Entradas

Salidas

Reporte por Vía de Cobro

Entidades

Cobros	E
Agentes	E
Bitácora Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Regeneración de Archivos PISA

Objetivo:

Genera nuevamente el archivo que se envía a PISA con la información de la base de datos.

Frecuencia: 1-3 veces/mes (eventual)
Usuario: Cobros Especiales
Responsable Operación: Cobros Especiales

Entradas

Salidas

Archivos de proceso PISA

Entidades

Cobros	E
Bitácora de Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Integración de Cobranza

Objetivo:

Realiza la Integración de la cobranza de los terceros de un día en específico a la base de datos en el caso que sea necesario realizar el proceso manual y no automático (situación proceso batch)

Frecuencia: 1-3 veces/mes (eventual)
Usuario: Administrador del Sistema
Responsable Operación: Administrador del Sistema

Entradas

Salidas

Entidades

Cobros	E/S
Estaciones de Cobro	E
Agentes	E
Poblaciones	E
Bitácora Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Integración de Terceros Faltantes

Objetivo:

Una vez recuperados los archivos de los terceros por la opción extracción de archivo de terceros a servidor se procede a integrar la información a la base de datos

Frecuencia: 1 –3 veces/mes
Usuario: Administrador del Sistema

Responsable Operación: Administrador del Sistema

Entradas

Salidas

Entidades

Agentes	E
Estaciones de Cobro	E
Cobros	E/S
Bitácora de Procesos	S
Poblaciones	E

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Mantenimiento al Catálogo de Agentes

Objetivo:

Dar mantenimiento al catálogo de agentes

Frecuencia: 1vez/mes (eventual)

Usuario: Cobros Especiales

Responsable Operación: Cobros Especiales

Entradas

Salidas

Actualiza agentes	Consulta catálogo de agentes
-------------------	------------------------------

Entidades

Agentes	E/S
Bitácora de Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Mantenimiento al Catálogo de Grupos

Objetivo:

Dar mantenimiento al catálogo de Poblaciones

Frecuencia: 1 vez/mes (eventual)
Usuario: Cobros Especiales
Responsable Operación: Cobros Especiales

Entradas

Salidas

Archivos de proceso PISA

Entidades

Poblaciones	E/S
Bitácora de Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS
Proceso: Mantenimiento al Catálogo de Estaciones de Cobro

Objetivo:

Dar mantenimiento al catálogo de estaciones de cobro

Frecuencia: 1 vez/mes (eventual)
Usuario: Cobros Especiales
Responsable Operación: Cobros Especiales

Entradas

Salidas

Actualiza estaciones de cobro Consulta estaciones de cobro

Entidades

Estaciones Cobros	E/S
Bitácora de Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS
Proceso: Mantenimiento al Catálogo de Usuarios

Objetivo:

Dar mantenimiento al catálogo de usuarios

Frecuencia: 1 vez/mes (eventual)
Usuario: Administrador del Sistema
Responsable Operación: Administrador del Sistema

Entradas

Actualiza usuarios

Salidas

Consulta catálogo de usuarios

Entidades

Usuarios	E/S
Bitácora de Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Respaldo de Archivos

Objetivo:

Generar respaldo semanal de movimientos de cobranza

Frecuencia: 1 vez/semana
Usuario: Administrador del Sistema
Responsable Operación: Administrador del Sistema

Entradas

Salidas

Reporte semanal

Entidades

Cobros	E
Bitácora de Procesos	S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Restauración de Respaldos

Objetivo:

Restaurar respaldo semanal a la base de datos

Frecuencia: 1 vez/mes (eventual)

Usuario: Administrador del Sistema

Responsable Operación: Administrador del Sistema

Entradas

Respaldos

Salidas

Entidades

Cobros E/S

Bitácora de Procesos S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Consulta Bitácora de Procesos

Objetivo:

Consultar los procesos realizados en el sistema de una fecha específica

Frecuencia: 1 vez/semanal (eventual)

Usuario: General

Responsable Operación: General

Entradas

Salidas

Bitácora de procesos

Entidades

Bitácora de Procesos E/S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Consulta Bitácora de transmisiones

Objetivo:

Consultar transmisiones de archivos realizadas por el sistema.

Frecuencia: 1 vez/día

Usuario: General

Responsable Operación: General

Entradas

Salidas

Bitácora de Terceros

Entidades

Bitácora de Transmisiones E

Bitácora de Procesos S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Depuración de Bitácora de Procesos

Objetivo:

Depurar bitácora de procesos

Frecuencia: 1 vez/mes

Usuario: Administrador del Sistema

Responsable Operación: Administrador del Sistema

Entradas**Salidas**

Archivos de proceso PISA

Entidades

Bitácora de Procesos

E/S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Depuración de Bitácora de Transmisiones

Objetivo: Depurar bitácora de transmisiones

Frecuencia:

1 vez/mes

Usuario:

Administrador del Sistema

Responsable Operación:

Administrador del Sistema

Entradas**Salidas****Entidades**

Bitácora de Procesos

S

Bitácora de Transmisiones

E/S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Depuración de la Tabla de Pagos

Objetivo:

Depurar la tabla de pagos

Frecuencia:

1 vez/mes

Usuario:

Administrador del Sistema

Responsable Operación:

Administrador del Sistema

Entradas**Salidas****Entidades**

Bitácora de Procesos	S
Pagos	E/S

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Proceso: Depuración de la Tabla de Cobros

Objetivo:

Depurar el Histórico de Cobros

Frecuencia:	1 vez/mes
Usuario:	Administrador del Sistema
Responsable Operación:	Administrador del Sistema

Entidades

Bitácora de Procesos	S
Histórico de Cobros	E/S

Sistema: SISTEMA DE COBRANZA POR TERCEROS.

Entrada: Respaldo.

Objetivo:

Respaldo semanal a restaurar

Usuario: Admin. Sistema

Procesos involucrados Restauración de respaldos 1 vez/mes (Eventual)		Volumen		
Datos	Tipo	Longitud	Decimales	Repetición
Número Telefónico	Numérico	8	0	1
Dígito Verificador	Numérico	1	0	1
Fecha de Facturación	Numérico	3	0	1
Fecha de Cobranza	Alfanumérico	10	0	1
Vía Cobro	Numérico	3	0	1
Población	Alfanumérico	13	0	1
Población Agente	Alfanumérico	3	0	1
Número de Estación de Cobro	Numérico	3	0	1
Fecha de Recepción	Alfanumérico	10	0	1
Importe de Recibo	Numérico	11	0	1
División	Alfanumérico	3	0	1

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Entrada: Archivo PISA

Objetivo:

Archivo de transacciones de cobranza para ser procesado en PISA

Usuario:

Cobros Especiales.

Procesos involucrados: Transmisión de Archivos PISA 1 vez/día Respaldar Archivo PISA 1 vez/mes (Eventual)		Volumen		
Datos	Tipo	Longitud	Decimales	Repetición
Número Telefónico	Numérico	8	0	1

Importe	Numérico	11	0	1
Fecha de pago	Fecha	0	0	1
Vía de Cobro	Numérico	3	0	1
Dígito Verificador	Numérico	1	0	1

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Entrada: Archivo de Cobranza

Objetivo:

Archivo de transacciones de cobranza en las generadas en las entidades externas

Usuario:

Entidades Externas

Procesos involucrados:		Volumen		
Extracción de Archivos de Terceros 1-3 veces/semana				
Conversión de Archivos 1 vez/día				
Respaldar Archivo PISA 1 vez/mes (Eventual)				
Datos	Tipo	Longitud	Decimales	Repetición
Número Telefónico	Numérico	8	0	1
Código de población	Numérico	8	0	1
Turno	Numérico	1	0	1
Tipo	Numérico	1	0	1
Dígito Verificador	Numérico	1	0	1
Importe	Numérico	11	0	1
Fecha de Facturación	Numérico	3	0	1
Número de Lote	Numérico	3	0	1
Número de Secuencia	Numérico	4	0	1

Verificador de Importe	Numérico	1	0	1
Tipo de Captura	Alfanumérico	1	0	1
Lada	Numérico	4	0	1
Vía de Cobro	Numérico	3	0	1

Sistema: SISTEMA DE COBRANZA POR TERCEROS.

Entrada: Actualiza Agentes

Objetivo:

Actualización del Catálogo de Agentes

Usuario:

Cobros Especiales

Procesos involucrados:		Volumen		
Mtto. Catálogo de agentes 1 vez/mes (Eventual)				
Datos	Tipo	Longitud	Decimales	Repetición
Vía de Cobro	Numérico	3	0	1
Nombre del Agente	Alfanumérico	20	0	1
Nombre del Responsable	Alfanumérico	20	0	1
Teléfono del Responsable	Numérico	11	0	1

Sistema: SISTEMA DE COBRANZA POR TERCEROS.

Entrada: Actualiza Estaciones de Cobro

Objetivo:

Actualizar el Catálogo de Estaciones de Cobro

Usuario:

Cobros Especiales

Procesos involucrados: Mtto. Catálogo de Estaciones de Cobro 1 vez/mes (Eventual)	Volumen
--	---------

Datos	Tipo	Longitud	Decimales	Repetición
Número de Estación de Cobro	Numérico	3	0	1
Vía de Cobro	Alfanumérico	3	0	1
Nombre de Estación de Cobro	Alfanumérico	20	0	1
Dirección de Estación de Cobro	Alfanumérico	20	0	1
Población de Estación	Alfanumérico	3	0	1
Responsable de Estación de Cobro	Alfanumérico	20	0	1
Teléfono de Estación de Cobro	Alfanumérico	11	0	1
División	Alfanumérico	3	0	1
Teléfono del Responsable	Numérico	11	0	1

Sistema: SISTEMA DE COBRANZA POR TERCEROS.

Entrada: Actualiza Usuarios

Objetivo:

Actualización del Catálogo de Usuarios

Usuario:

Adm. Sistema

Procesos involucrados: Mtto. Catálogo de Usuarios 1 vez/mes (Eventual)	Volumen			
Datos	Tipo	Longitud	Decimales	Repetición
Clave del Usuario	Alfanumérico	7	0	1
Password	Alfanumérico	7	0	1

Grupo	Alfanumérico	3	0	1
Nombre del Usuario	Alfanumérico	20	0	1
Dirección	Alfanumérico	20	0	1
Teléfono del Usuario	Numérico	11	0	1
Siglas	Alfanumérico	3	0	1
Vía de Cobro	Numérico	3	0	1

Descripción de Agregados de Entrada

Sistema: SISTEMA DE COBRANZA POR TERCEROS.

Agregado de Entrada: Archivos de Cobranzas

Entradas:

Archivos de Cobranza

Archivo PISA

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Agregado de Entrada: Admón. De Catálogos

Entrada:

Actualiza poblaciones

Actualiza Agentes

Actualiza Estaciones de Cobro

Actualiza Usuarios

Descripción de Agregados de Salidas

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Agregado de Salida: Archivos Transferidos

Salidas:

Archivo de Cobranza

Archivo de Proceso PISA

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Agregado de Salida: Archivos de Procesamientos

Salidas:

Archivo de Proceso PISA

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Agregado de Salida: Reporte de Cobranza

Salidas:

Consulta de Pagos

Reporte Diario de Caja

Consulta de Diario de Agentes

Consulta de Verificación de Pagos

Reporte Estadístico

Reporte por Vía de Cobro

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Agregado de Salida: Consulta de Catálogos

Salidas:

Consulta al Catálogos de Agentes

Consulta al Catálogo de Población

Consulta de Estaciones de Cobro

Consulta al Catálogo de Usuarios

Descripción de Agregados de Información

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Agregado de Información: Información de Cobranza

Entidades:

Cobros

Poblaciones

Estaciones de Cobro

Agentes

Usuarios

Temporal de Cobranza

Bitácora de Procesos

Bitácora Transmisiones

Pagos

Histórico de Cobros

Sistema: SISTEMA DE COBRANZA POR TERCEROS

Agregado de Información: Catálogos

Entidades:

Poblaciones

Estaciones de Cobro

Agentes

Usuarios

3.1.1 Diagrama de contexto

Para poder realizar un desarrollo más rápido y sencillo necesitamos tener un panorama general de lo que se necesita hacer sin importar la manera en que se realice esto durante el análisis y posteriormente en el diseño se toman las decisiones de cómo se van a llevar a cabo las cosas. Todo en un orden jerárquico de un nivel alto generalizado y posteriormente de una manera más detallada.

Al seleccionar la propuesta básica para resolver el problema, la propuesta de sistema es dividida en módulos o subsistemas. Esta manera de descomposición nos ayuda a entender de una manera más rápida el problema y tomar decisiones que afecten al sistema entero y a su vez poder trabajar de manera independiente cada módulo o subsistema.

En la figura 3.1.1.1 podemos observar el diagrama de contexto del sistema de cobranza por terceros en donde se tienen identificados los siguientes objetos: Transferencia, Procesos especiales, Reportes, Mantenimiento, Utilerías y Bitácoras en el primer nivel de nuestro sistema.

La figura 3.1.1.2 nos muestra el diagrama de contexto de transmisión en donde se reciben archivos vía diskette y se envían archivos vía la red como se ve en las figuras 3.1.1.3 y 3.1.1.4, en esta última se nos muestra el diagrama de contexto de la transferencia por diskette en donde se realiza tanto una restauración de un archivo hacia el sistema como un respaldo de cierta información del sistema a un archivo.

En la siguiente figura 3.1.1.5 tenemos el diagrama de contexto de procesos especiales en donde se realizan Ajustes de pagos y cobros mal aplicados como se ve en su diagrama de contexto de la figura 3.1.1.6 además de ajustes tenemos también, integración de faltantes en donde se integra la cobranza de agentes de los cuales no se recibió la información a tiempo, de integración de cobranza que

se recibió de manera normal y se regeneran archivos de cobranza que se enviara al sistema institucional (PISA).

La figura 3.1.1.7 nos muestra el diagrama de contexto de generación de reportes en donde obtenemos los reportes necesarios para elaborar las formas contables correspondientes así como control de pagos en donde se realiza la captura de los pagos hechos por los agentes a la empresa por la cobranza realizada y donde se verifican dichos movimientos como lo muestra la figura 3.1.1.8 que es el diagrama de contexto de control de pagos.

En la figura 3.1.1.9 tenemos el diagrama de contexto de mantenimiento de catálogos en donde se realizaran los cambios necesarios para mantener el sistema en buen funcionamiento como son usuarios, agentes, grupos y estaciones de cobro.

En la figura 3.1.1.10 nos muestra el diagrama de contexto de utilerías el cual nos muestra los objetos de depuración, respaldo, mantenimiento y seguridad.

Y por ultimo el diagrama de contexto de bitácoras, figura 3.1.1.11, donde se registran los movimientos realizados por los accesos hechos al sistema como también las transmisiones con el fin de verificar si tuvieron éxito o no.

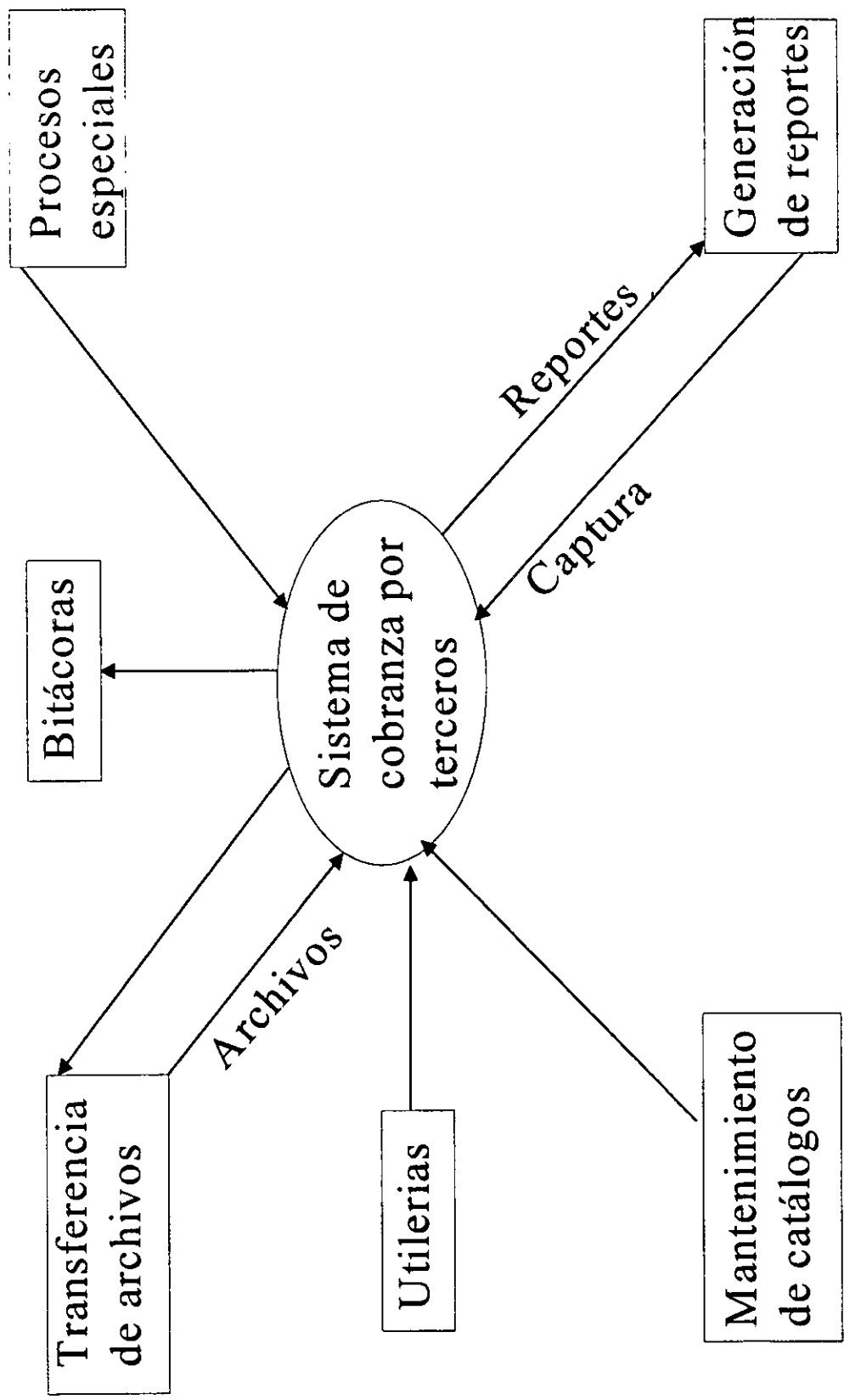


Figura 13.1.1. Diagrama de contexto del sistema de cobranza por terceros

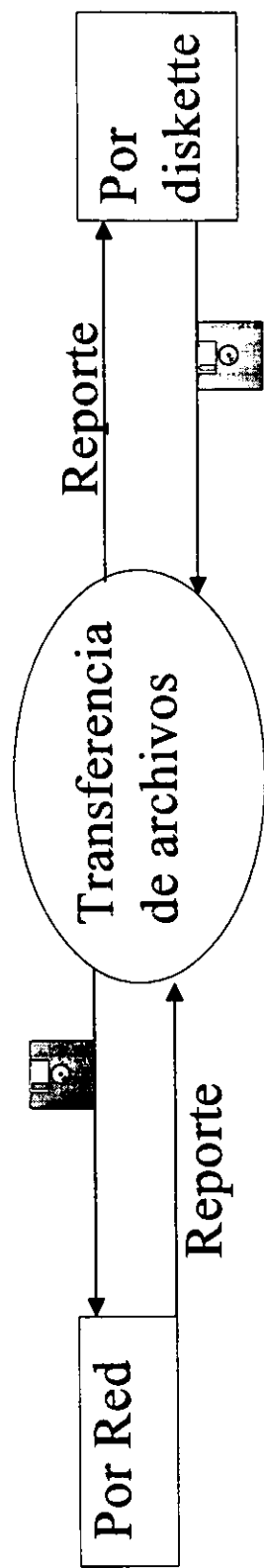


Figura 3.1.1.2. Diagrama de contexto de transferencia de archivos

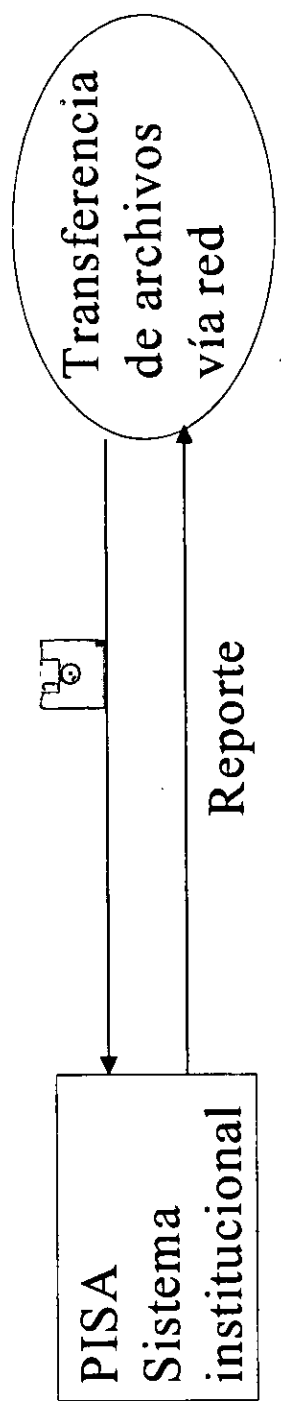


Figura 3.1.1.1.3. Diagrama de contexto de transferencia de archivos al sistema institucional

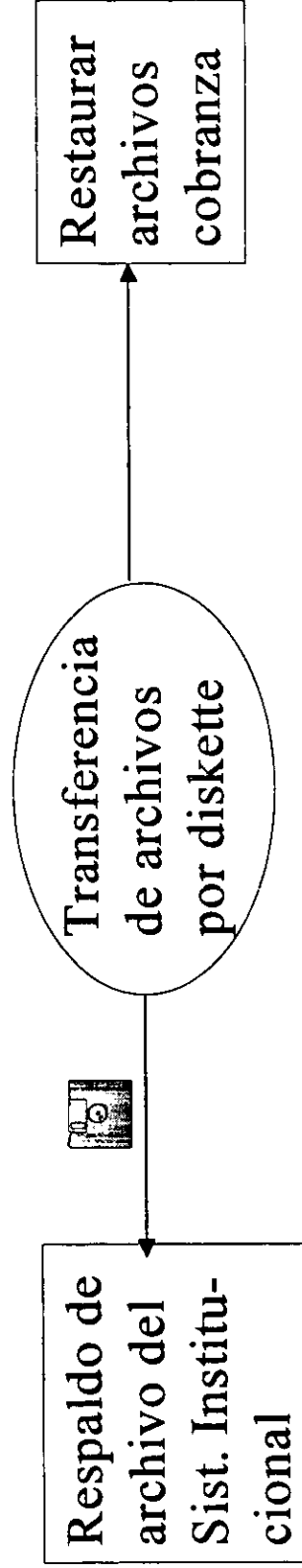


Figura 3.1.1.4. Diagrama de contexto de transferencia de archivos vía diskette al sistema de cobranza por terceros

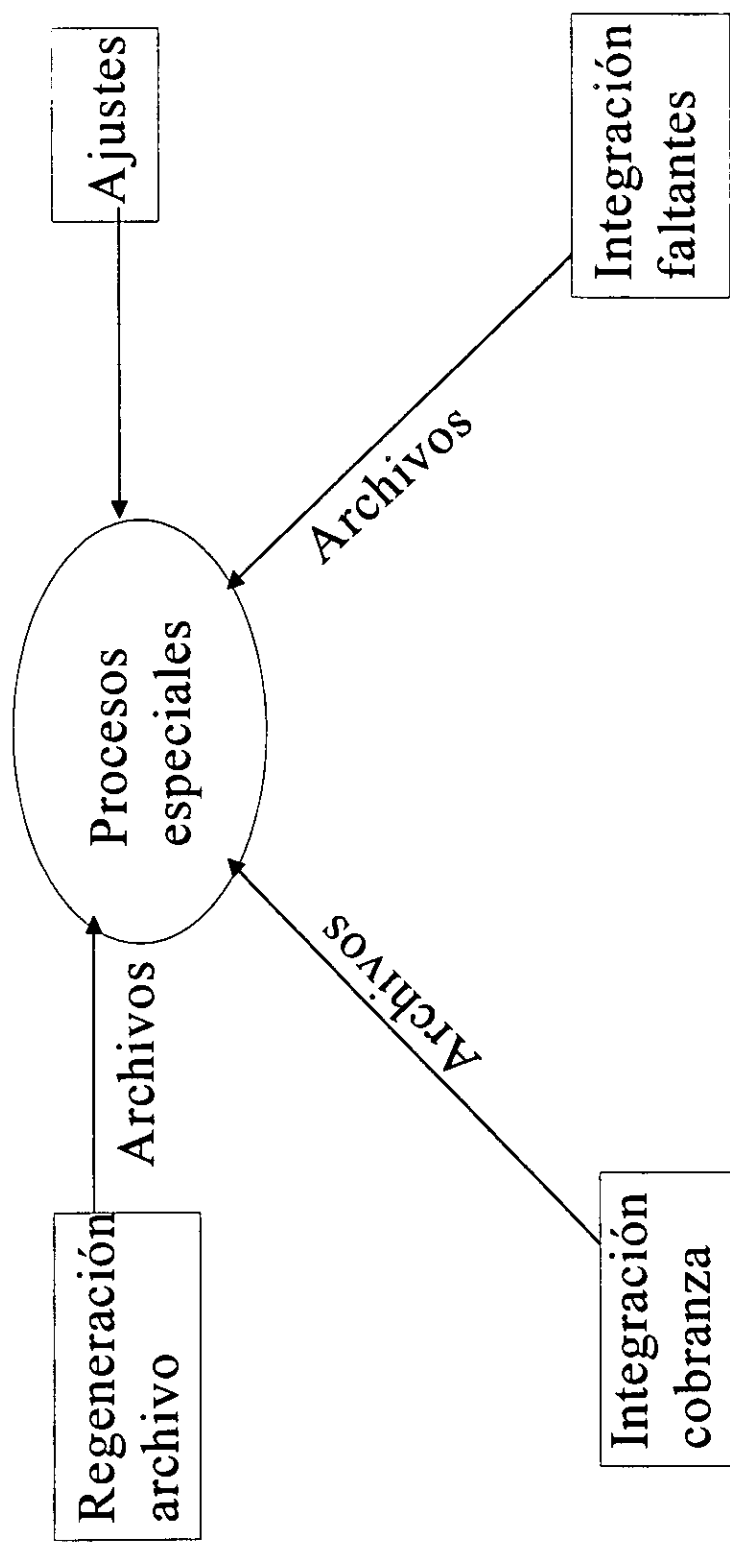


Figura 3.1.1.5. Diagrama de contexto de Procesos especiales

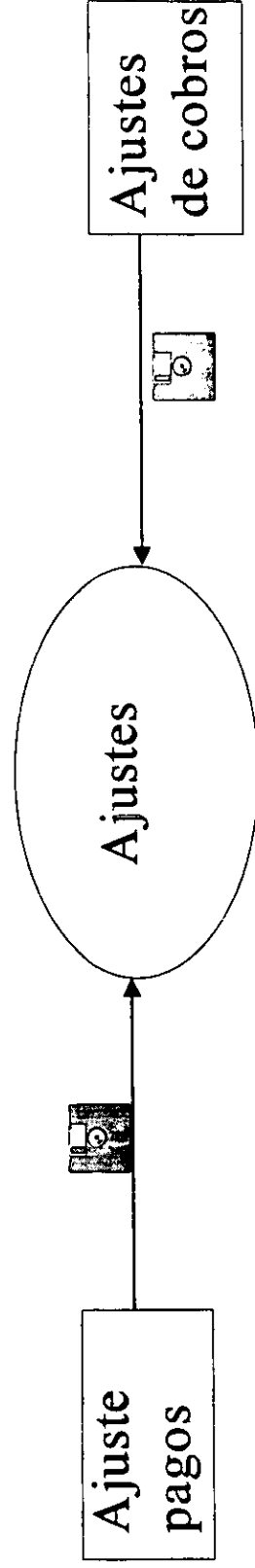


Figura 3.1.1.1.6. Diagrama de contexto de Ajustes

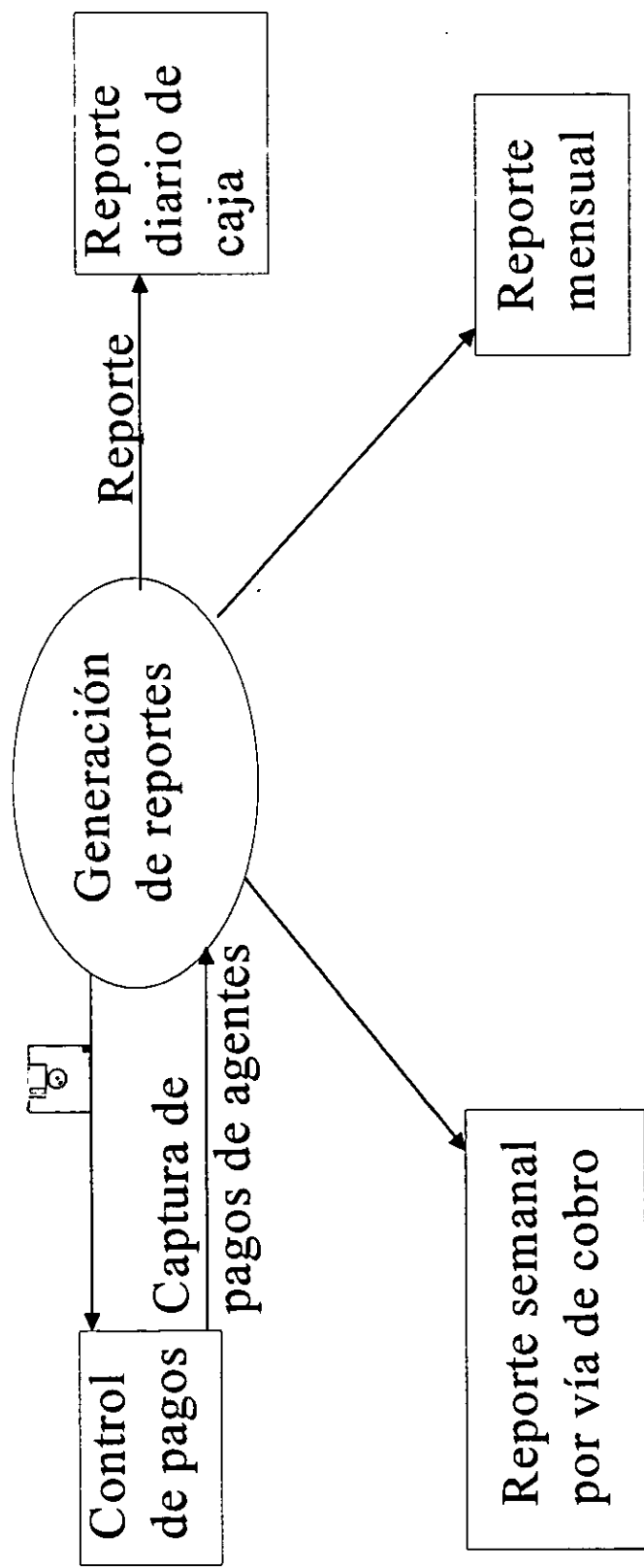


Figura 3.1.1.7. Diagrama de contexto de generación de reportes

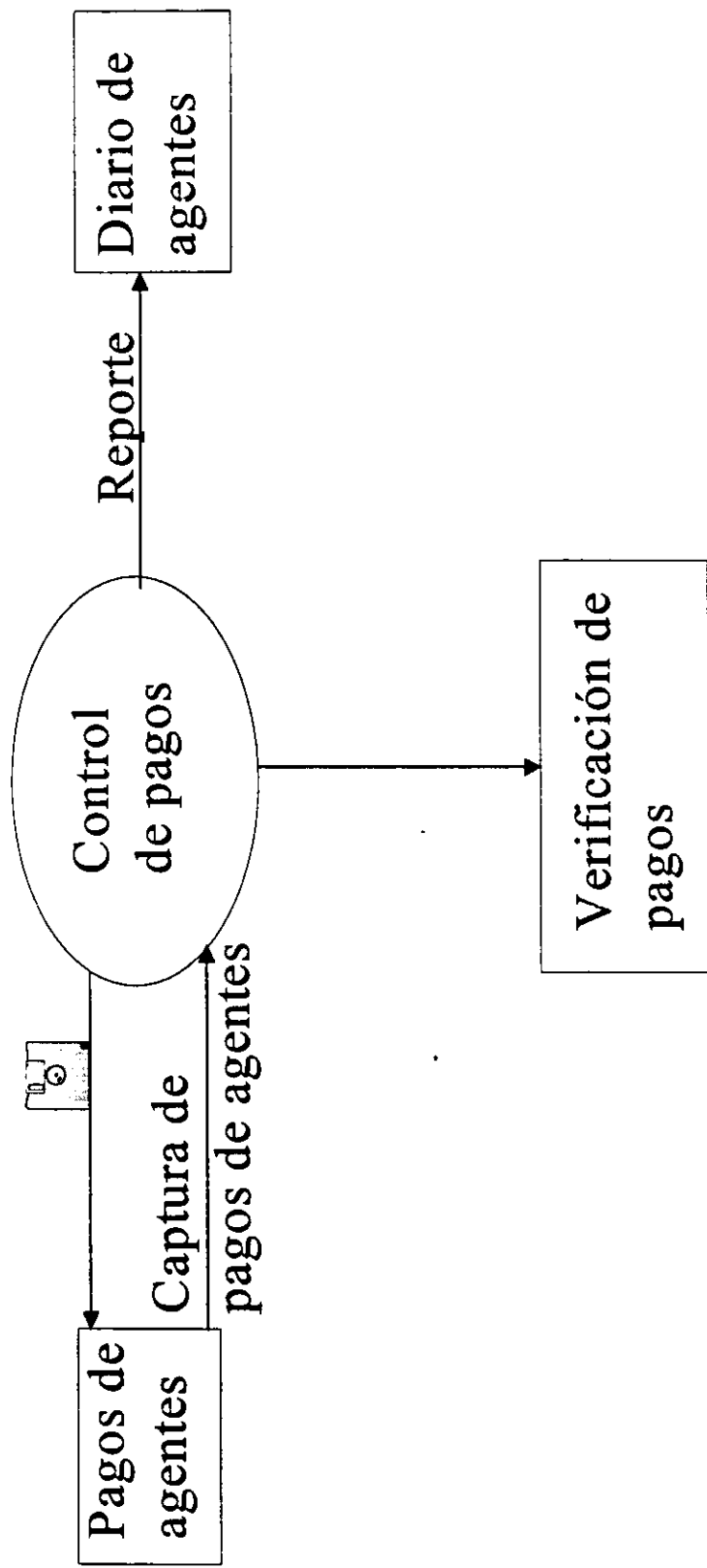


Figura 3.1.1.8. Diagrama de contexto de control de pagos

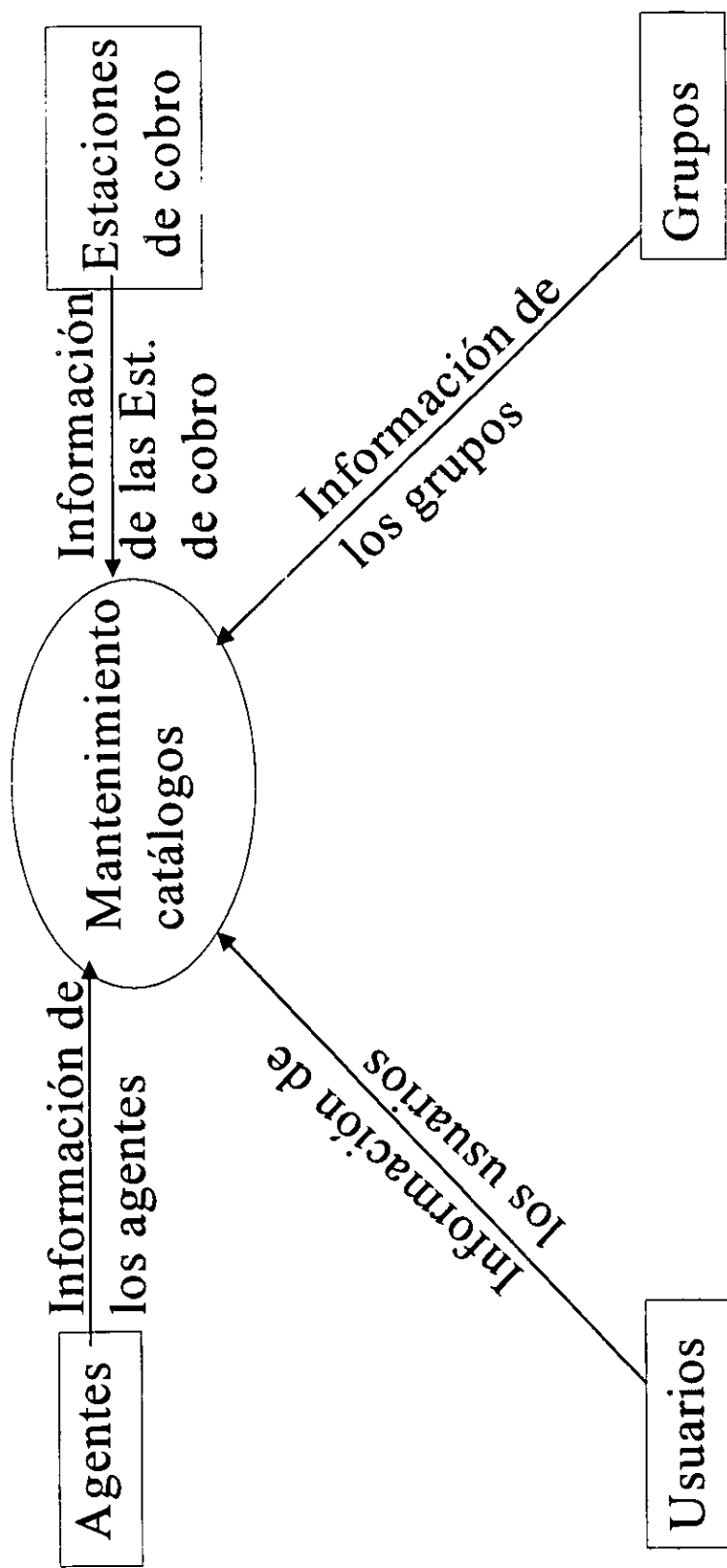


Figura 3.1.1.9. Diagrama de contexto de mantenimiento de catálogos

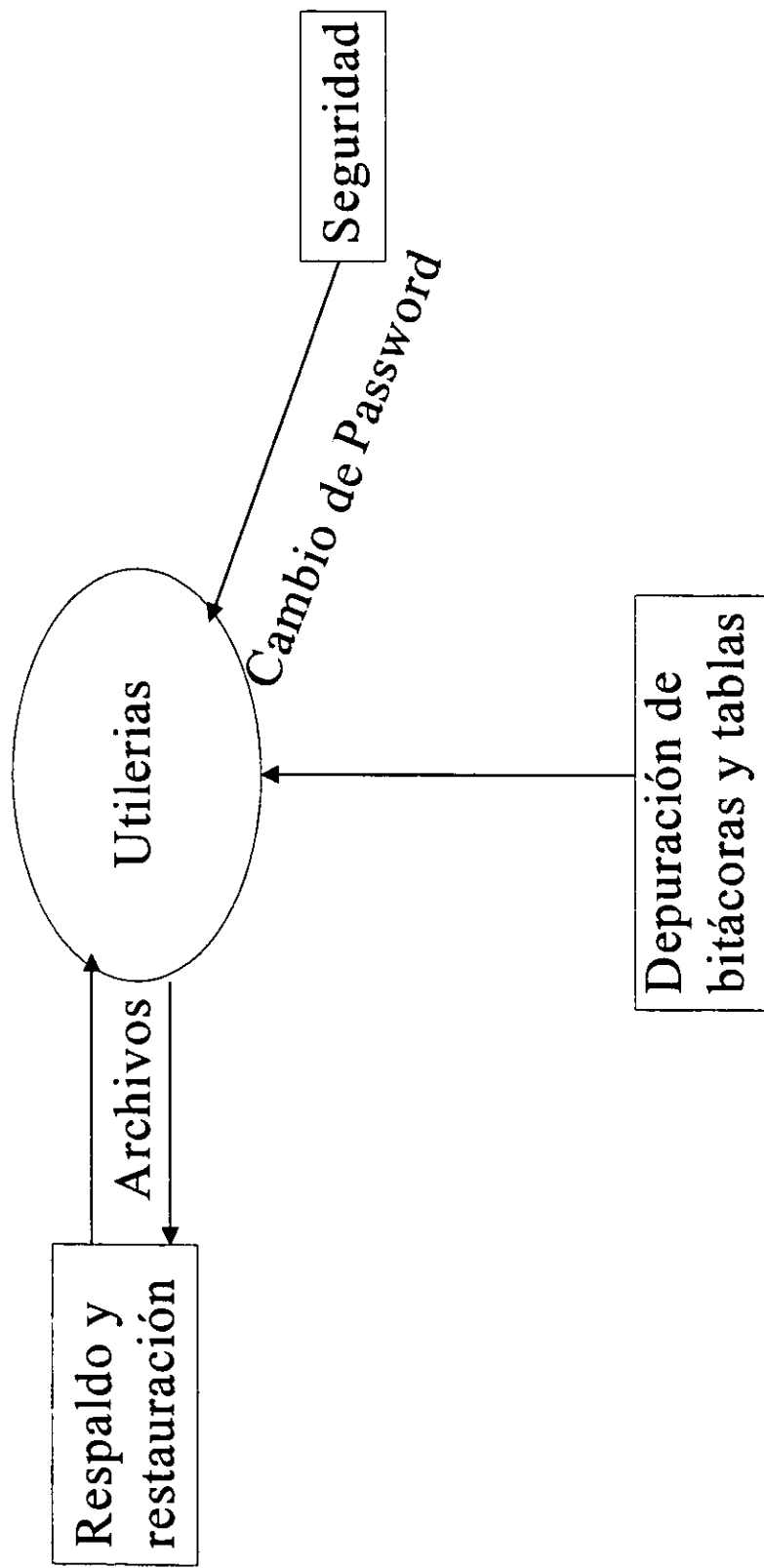
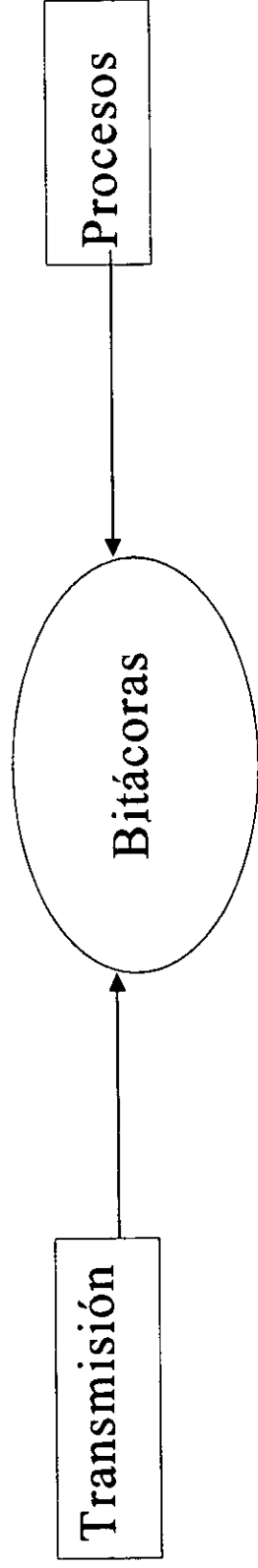


Figura 3.1.1.10. Diagrama de contexto utilerías



3.1.1.11. Diagrama de contexto de bitácoras

3.1.2 Diagrama de eventos

En el análisis orientado a objetos el mundo se describe en términos de los objetos y sus estados, así como de los eventos que modifican esos estados. Como ya se mencionó en el capítulo 1, un evento es un cambio en el estado de un objeto; sin los eventos, el mundo no cambiaría. En un mundo sin eventos podríamos construir y generalizar bases de datos sin preocuparnos por actualizarlas. Sin embargo, en la mayoría de las aplicaciones, si debe cambiar el contenido de las bases de datos, puesto que deseamos saber de esos cambios y reaccionar en forma adecuada ante ellos, deberemos entender y modelar los eventos.

Una operación hace que los eventos ocurran. Los eventos indican los puntos en el tiempo en que se da el cambio de estado de un objeto. Según el área que se modele, puede ocurrir más de un evento al terminar una operación, y cada uno de estos puede activar operaciones independientes.

Las operaciones externas también se representan y se utiliza un reloj cuando el tipo de evento lo provoca un reloj externo.

Una condición de control garantiza que un conjunto de eventos está completo antes de proceder con una operación. Antes de invocar de hecho a la operación, se verifica su condición de control. Si los resultados de la evaluación de la condición son verdaderos, se invoca a su operación. Si son falsos, no se invoca.

El diagrama de eventos para el sistema de cobranza por terceros se representa en la figura 3.1.2.1. En esa figura podemos observar de una forma muy general las principales acciones del sistema, así como los eventos que estas realizan en el mismo.

Figura 3.1.4.1. Diagrama de estado general
1 Menú principal

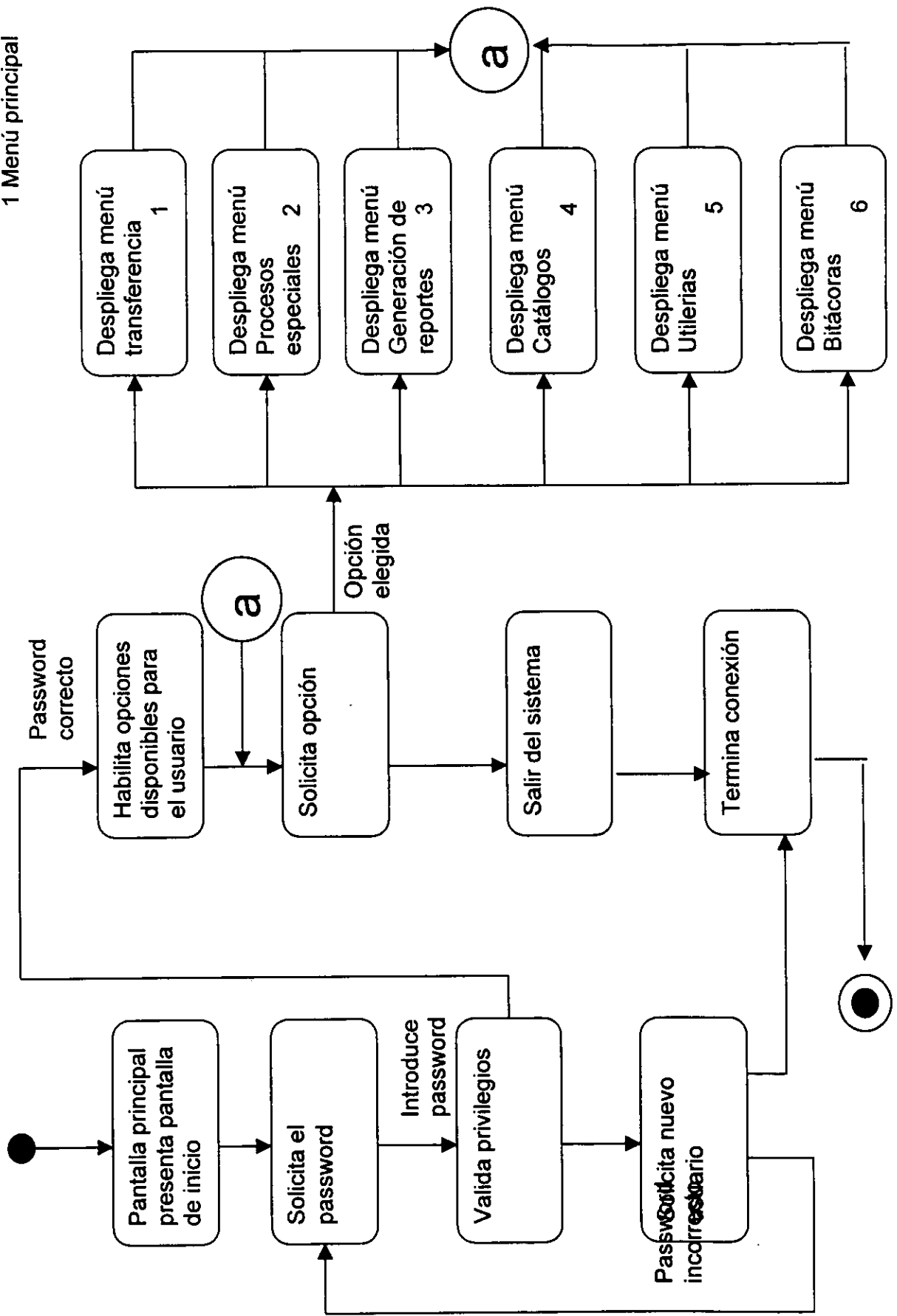


Diagrama de estado

1.1 Menú transferencia

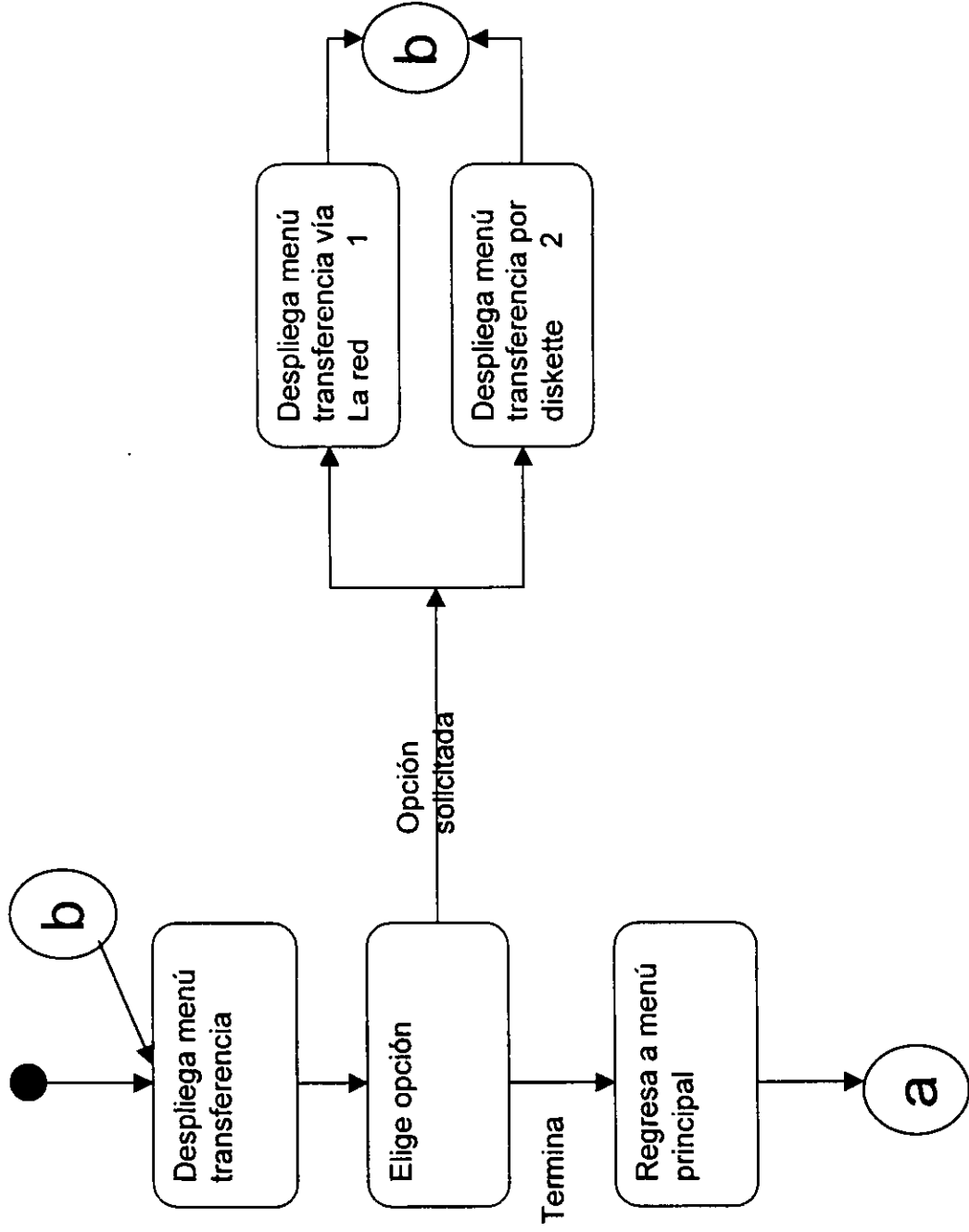


Diagrama de estado

1.1.1 Proceso Transferencia

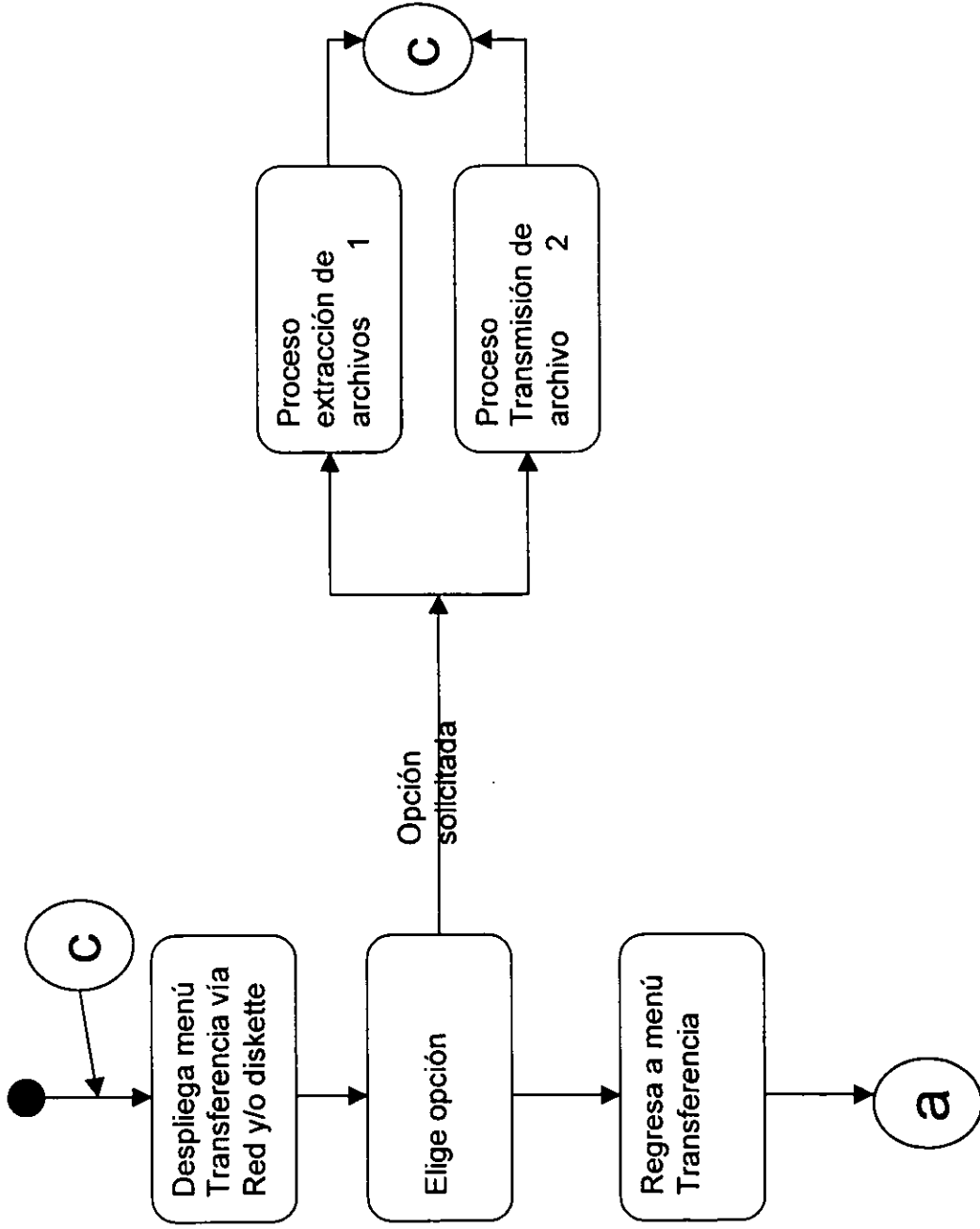


Diagrama de estado

1.1.1.1 Proceso extracción de archivo

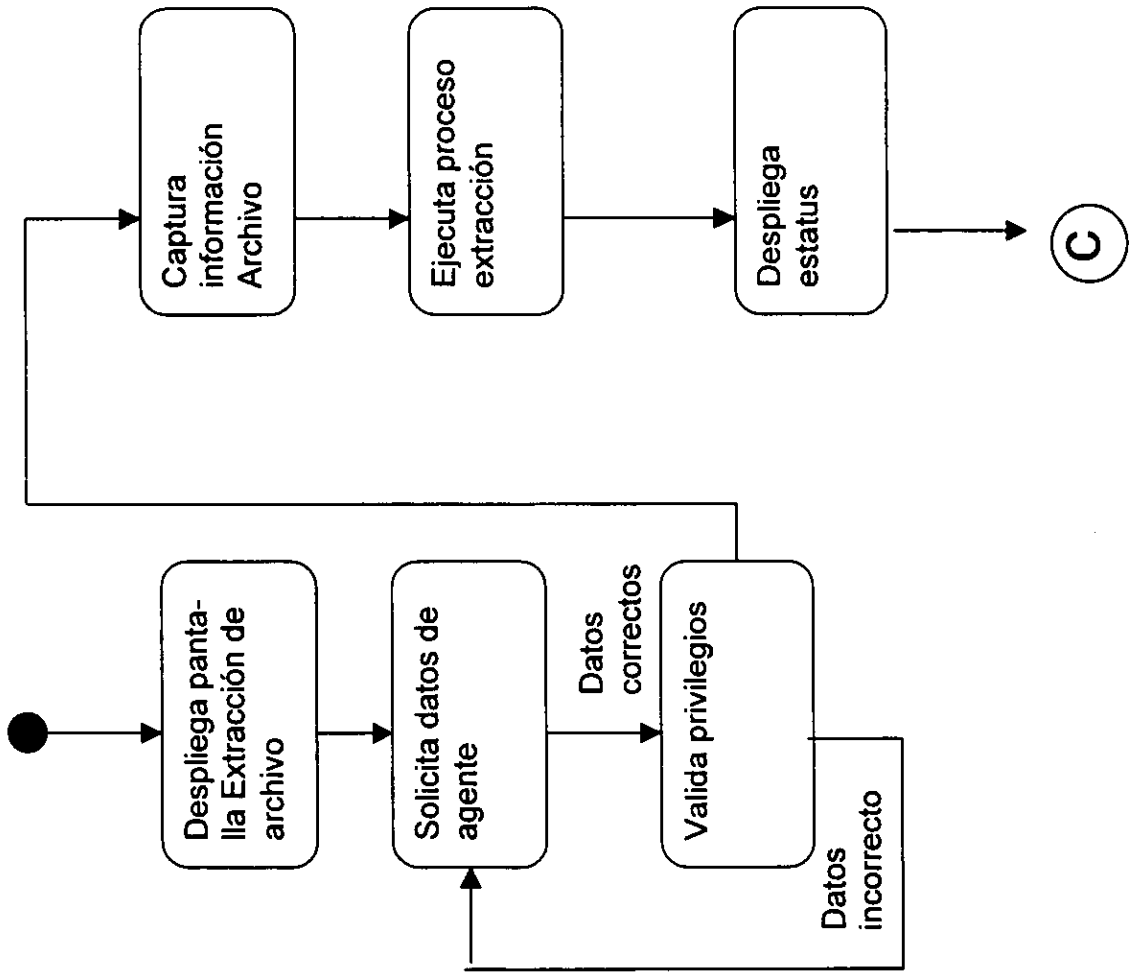


Diagrama de estado
1.1.1.2 Extracción

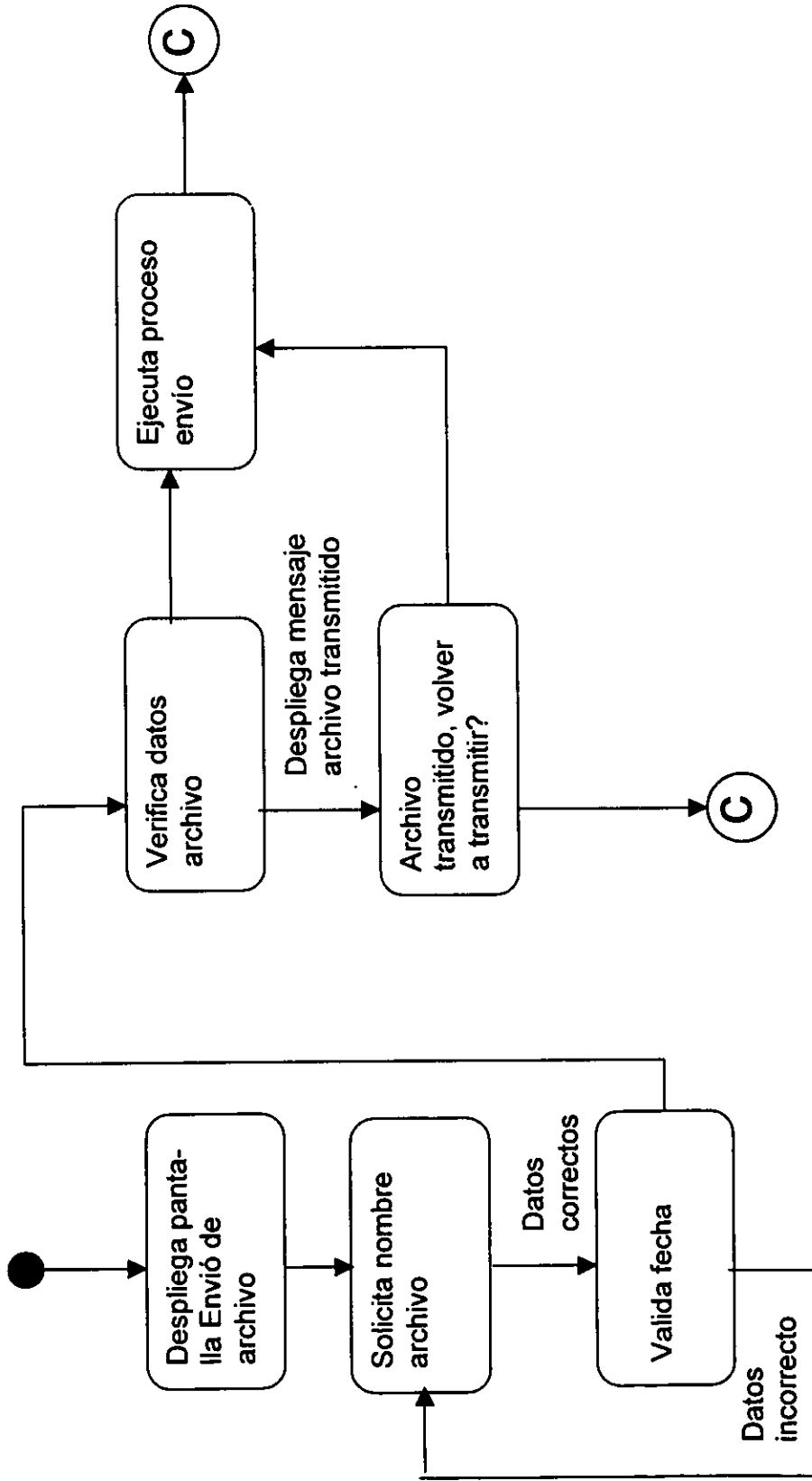


Diagrama de estado
1.1.2 Proceso transferencia

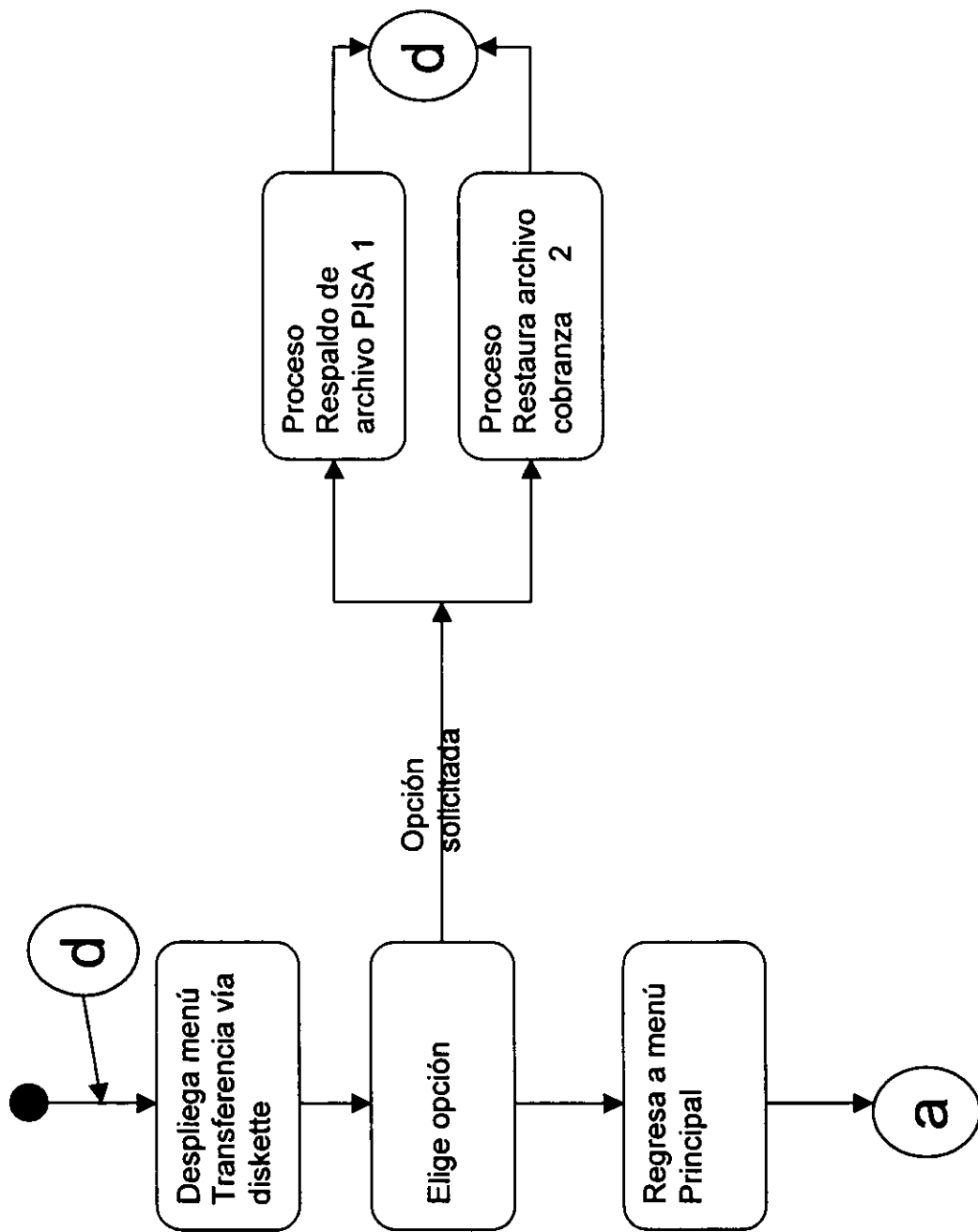


Diagrama de estado

1.1.2.1 Proceso respaldo archivo PISA

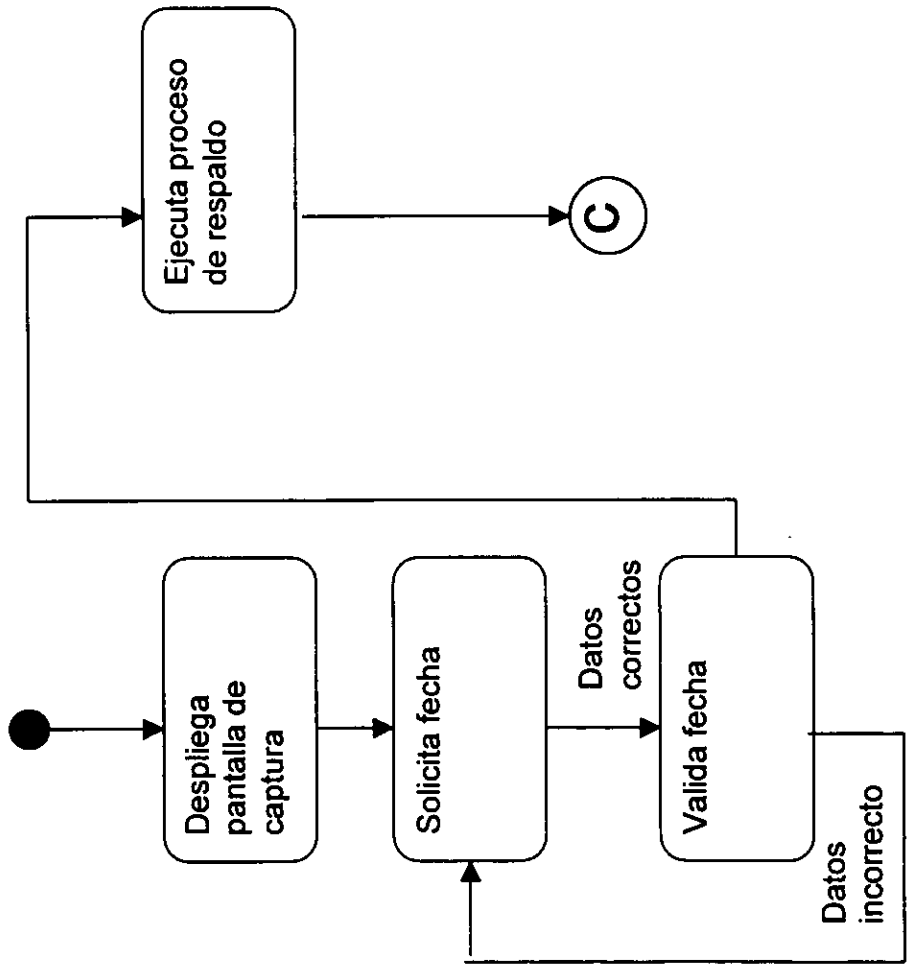


Diagrama de estado

1.1.2.2 Proceso restaurar archivo terceros PISA

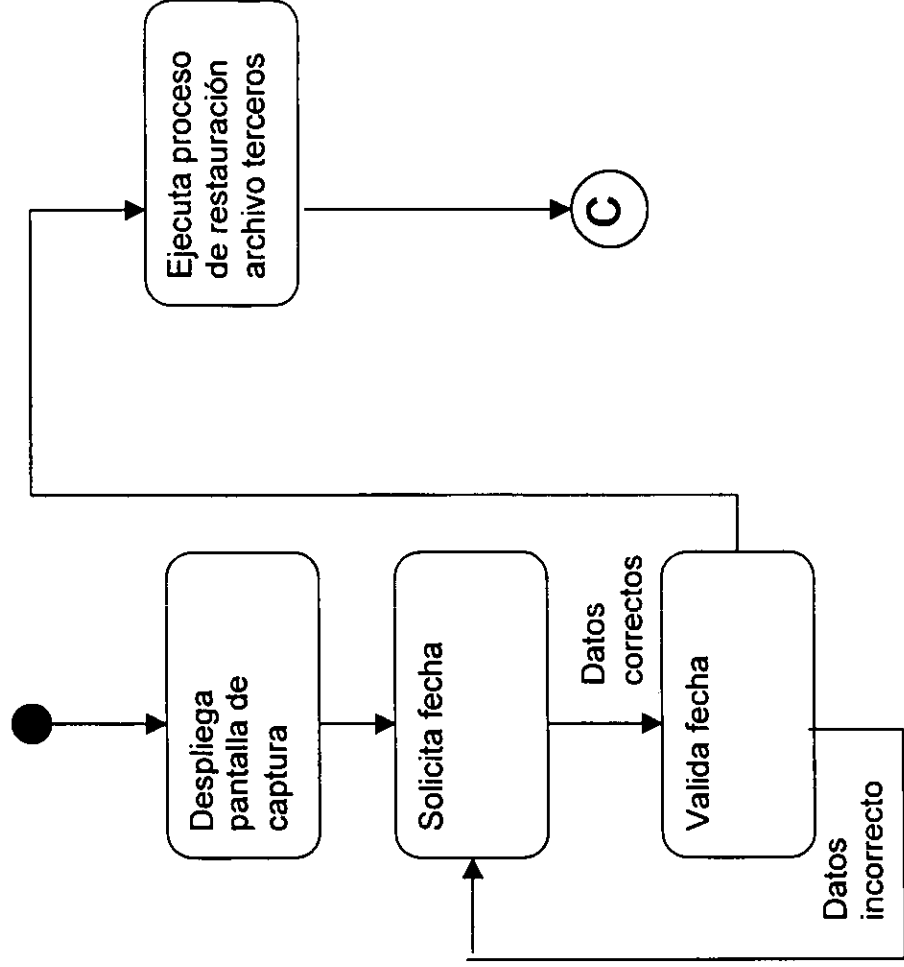


Diagrama de estado
1.2. Menú Procesos especiales

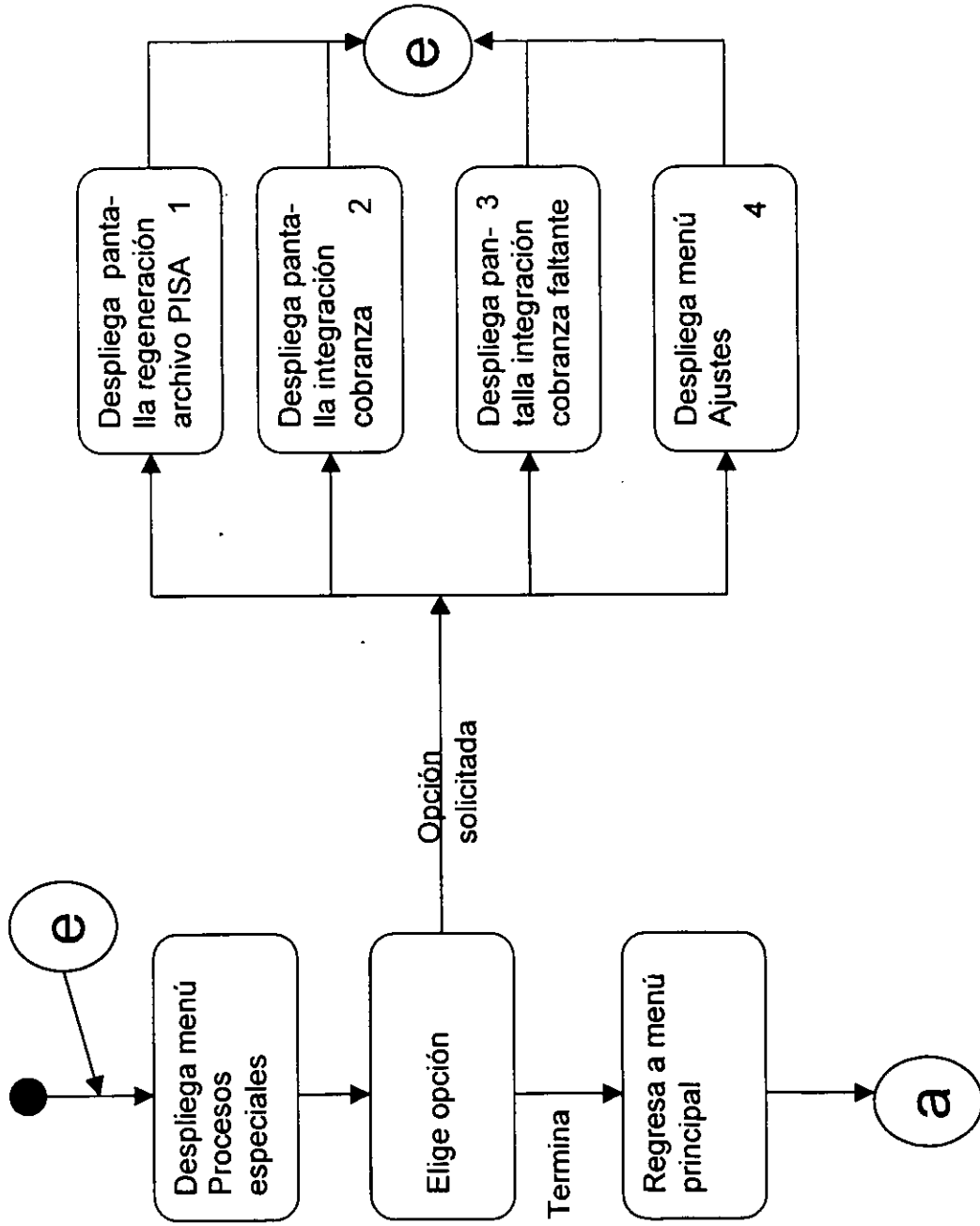


Diagrama de estado

1.2.1 Proceso Regenera archivo PISA.

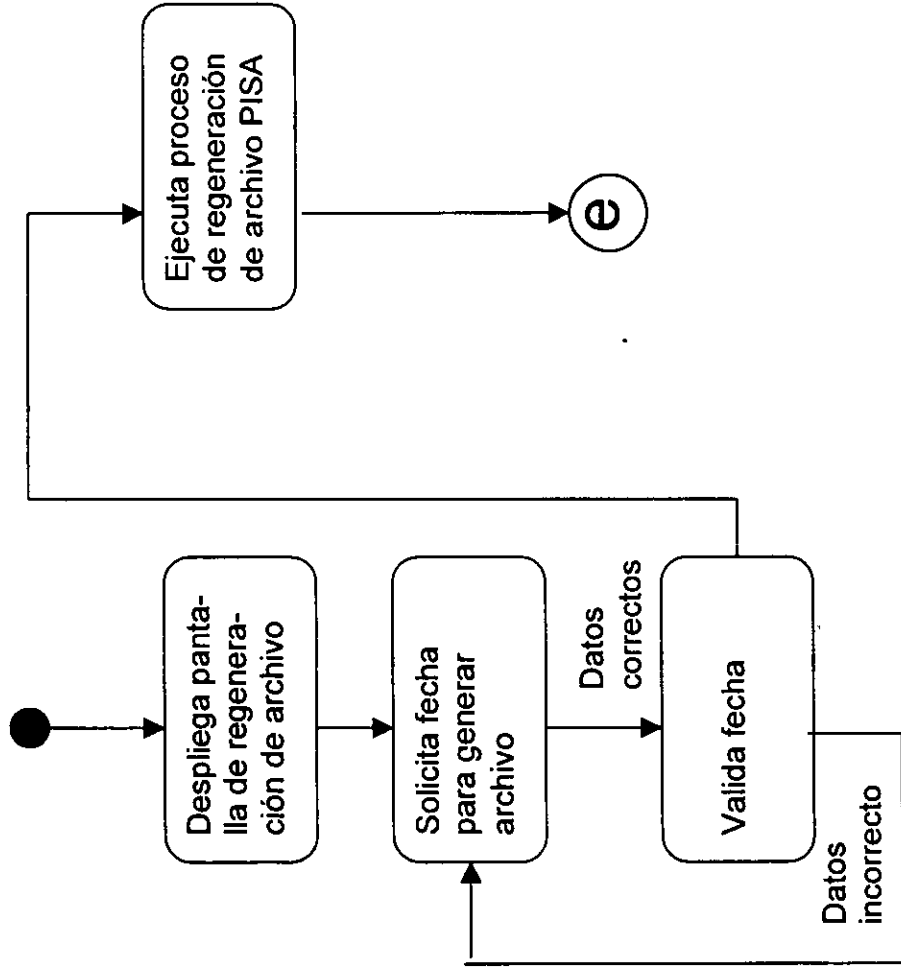
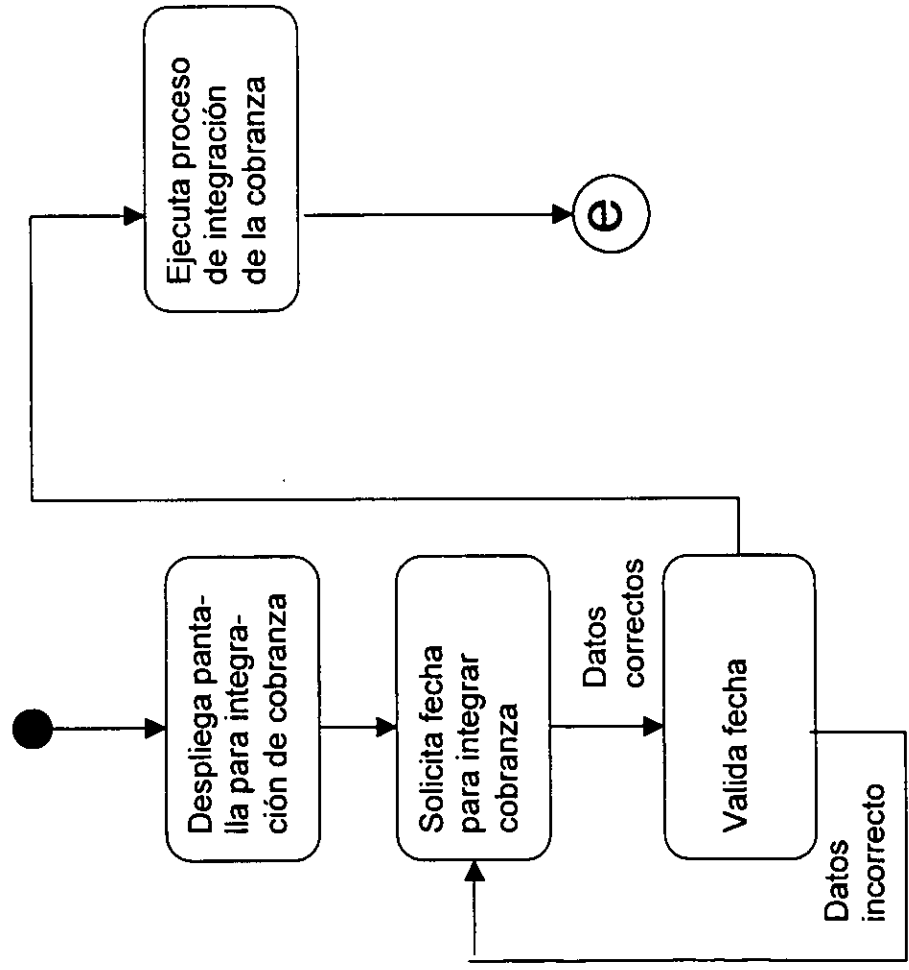


Diagrama de estado
1.2.2 Extracción



1.2.3 Integración terceros faltantes

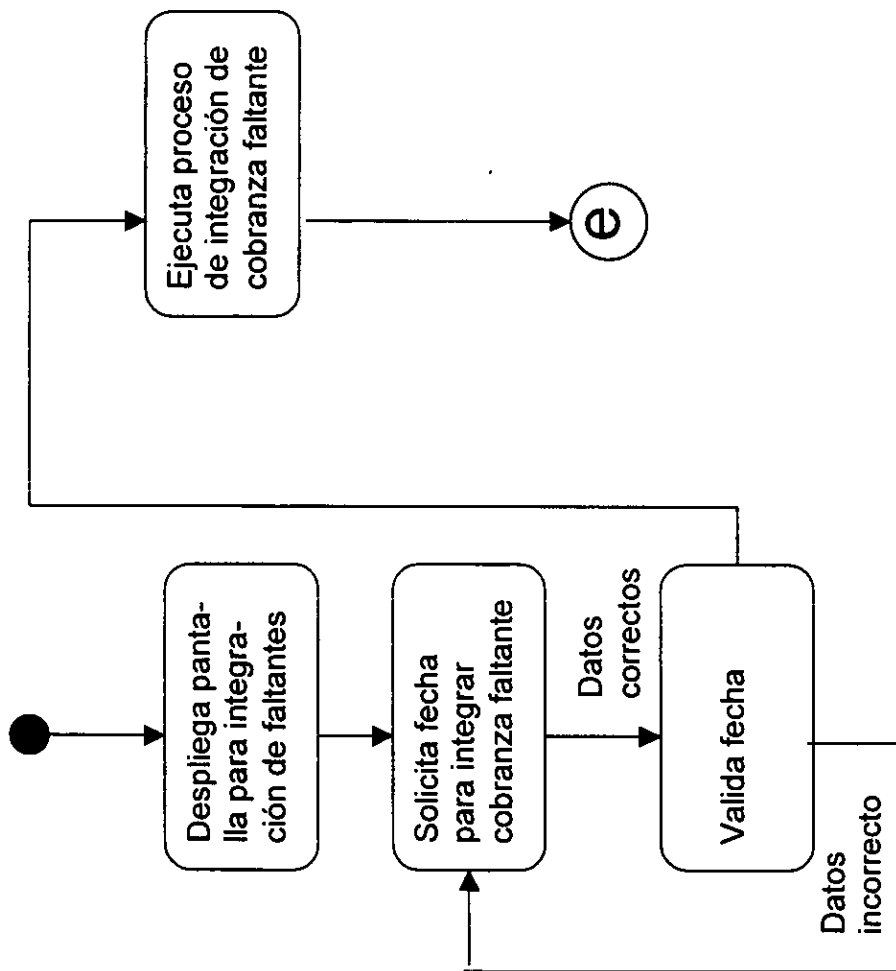


Diagrama de estado
1.2.4 Menú Ajustes a tablas de cobros y pagos

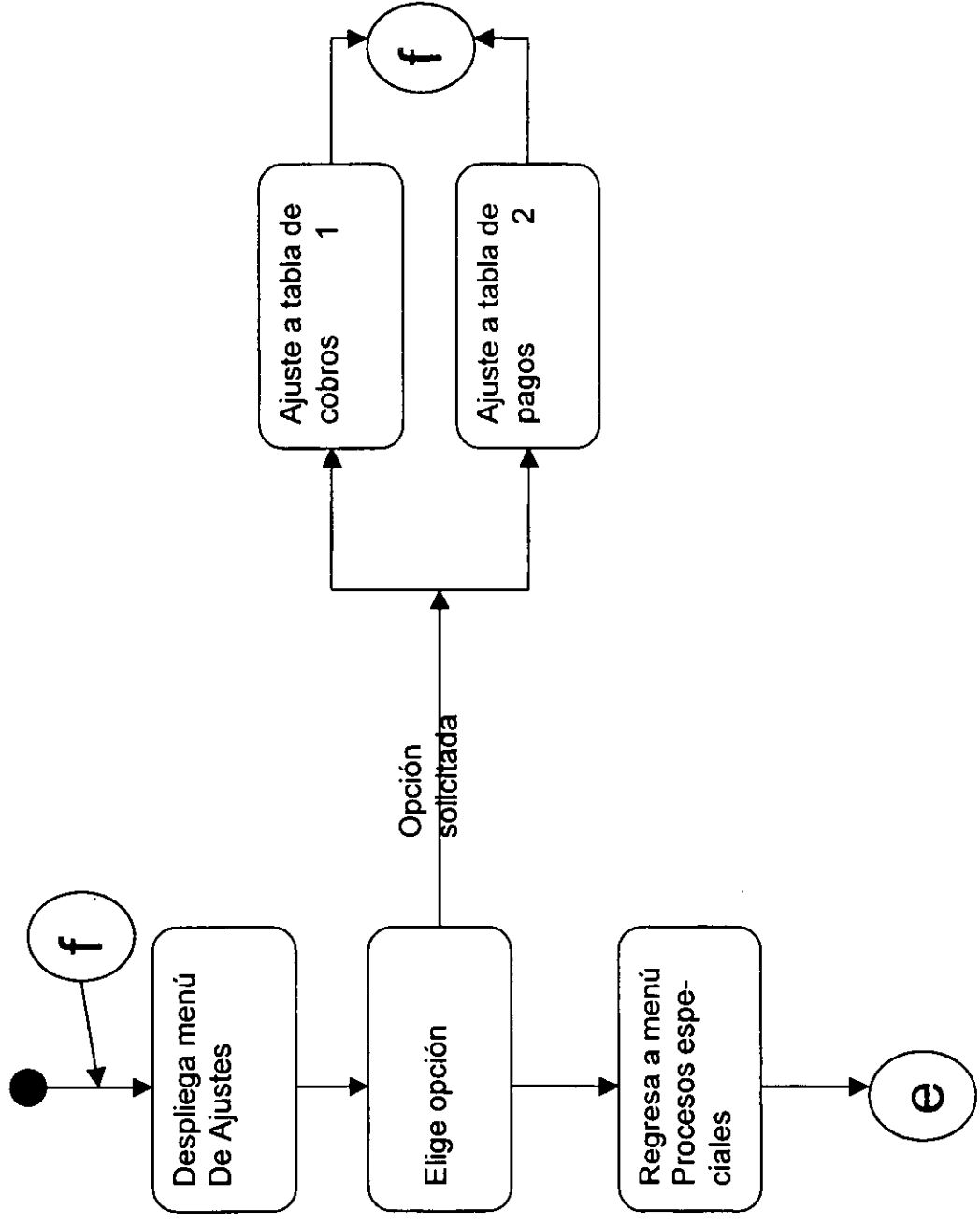


Diagrama de estado

1.2.4.1 Proceso ajuste de cobros

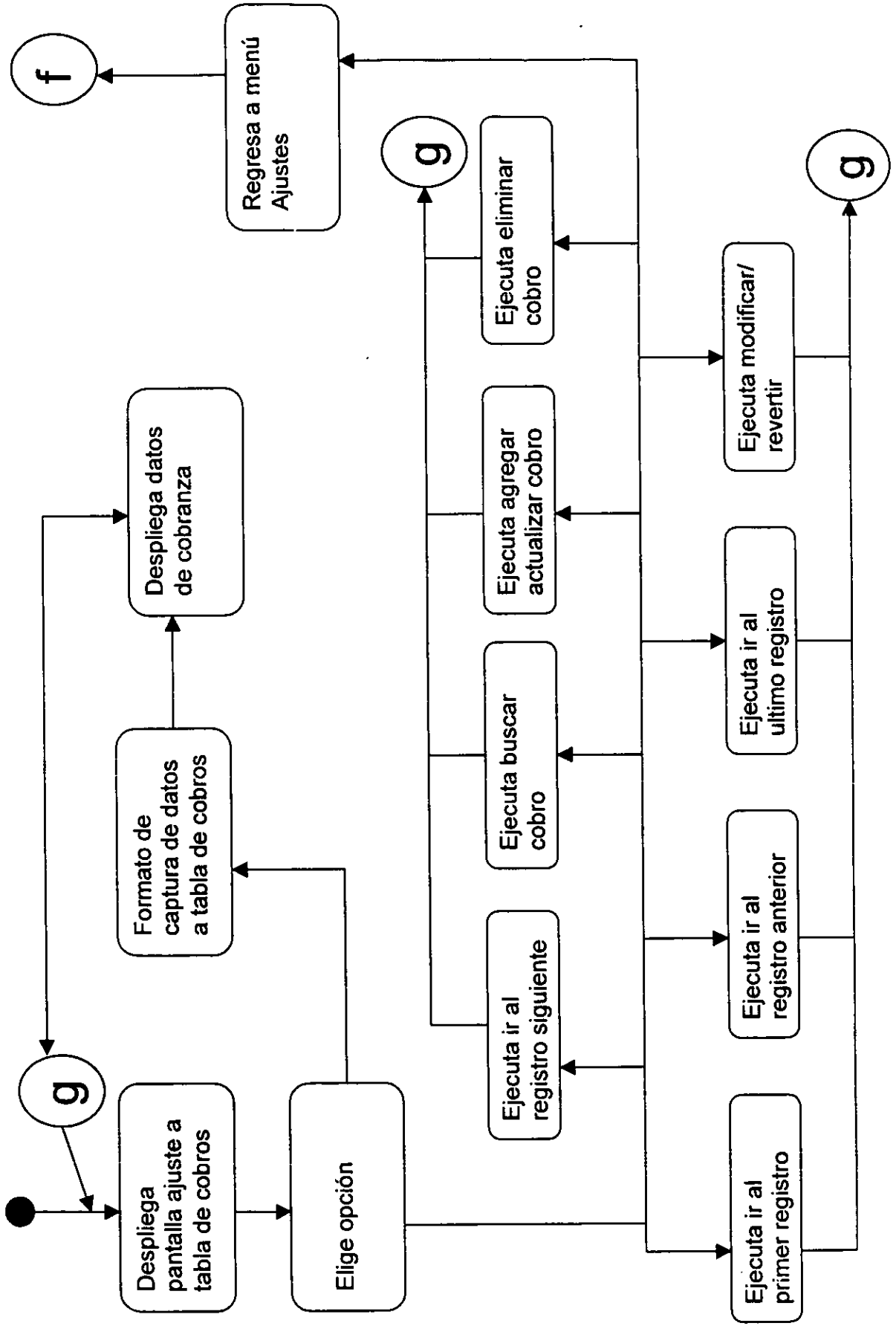


Diagrama de estado
 1.2.4.1.1 Proceso cobros agregar

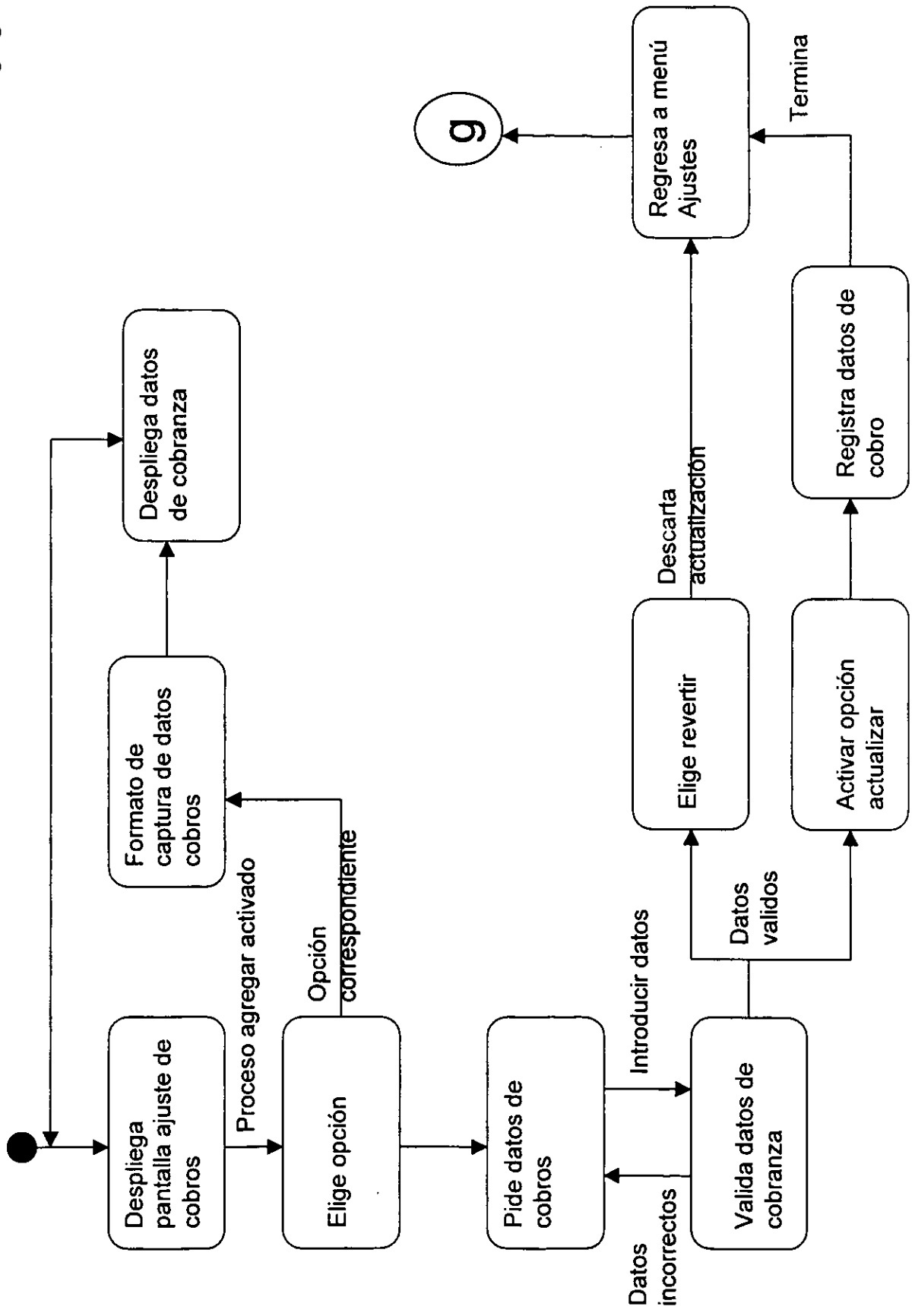


Diagrama de Estado
1.2.4.1.2 Proceso cobros seleccionar

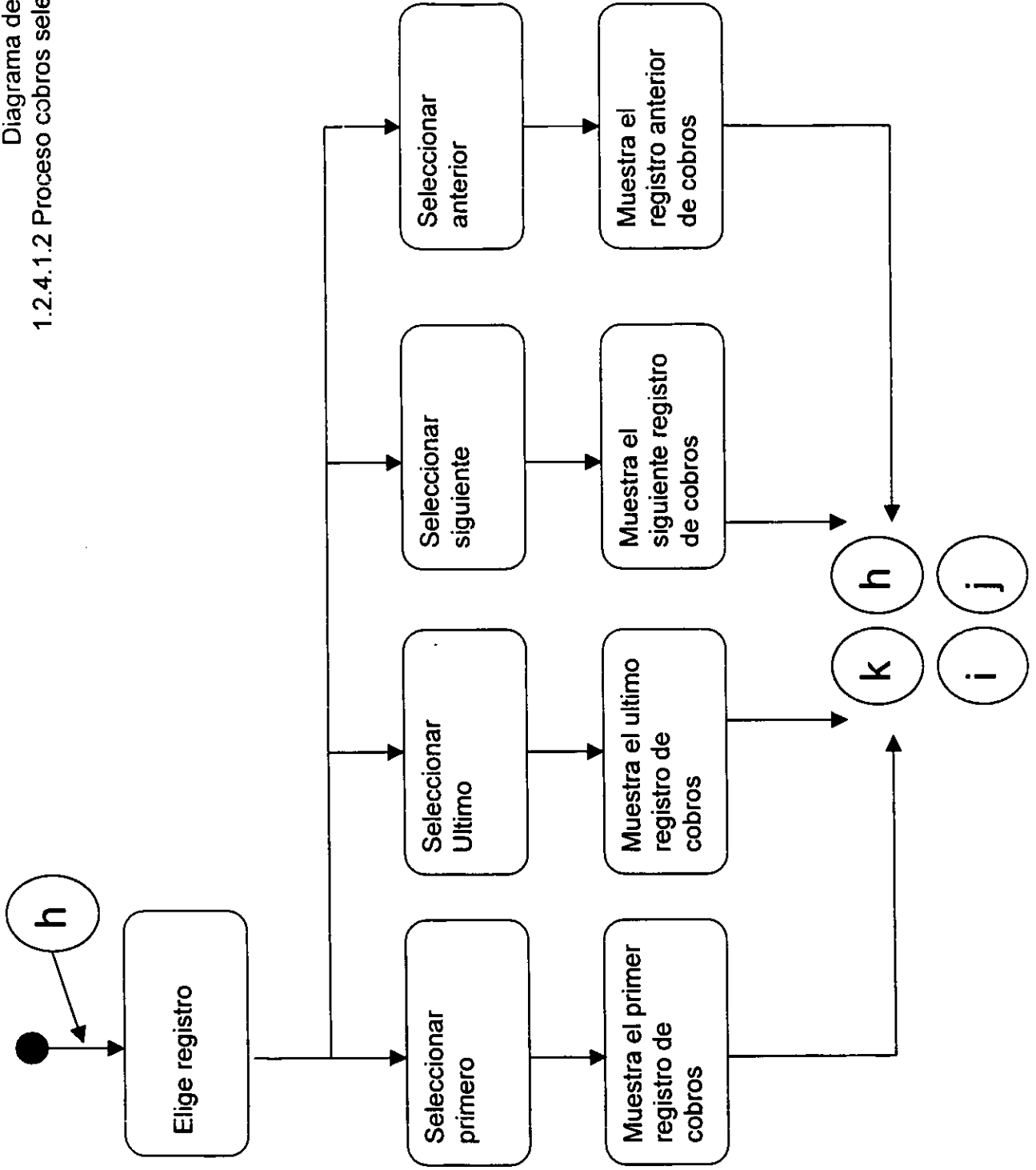


Diagrama de estado
1.2.4.1.3. Proceso Cobros modificar

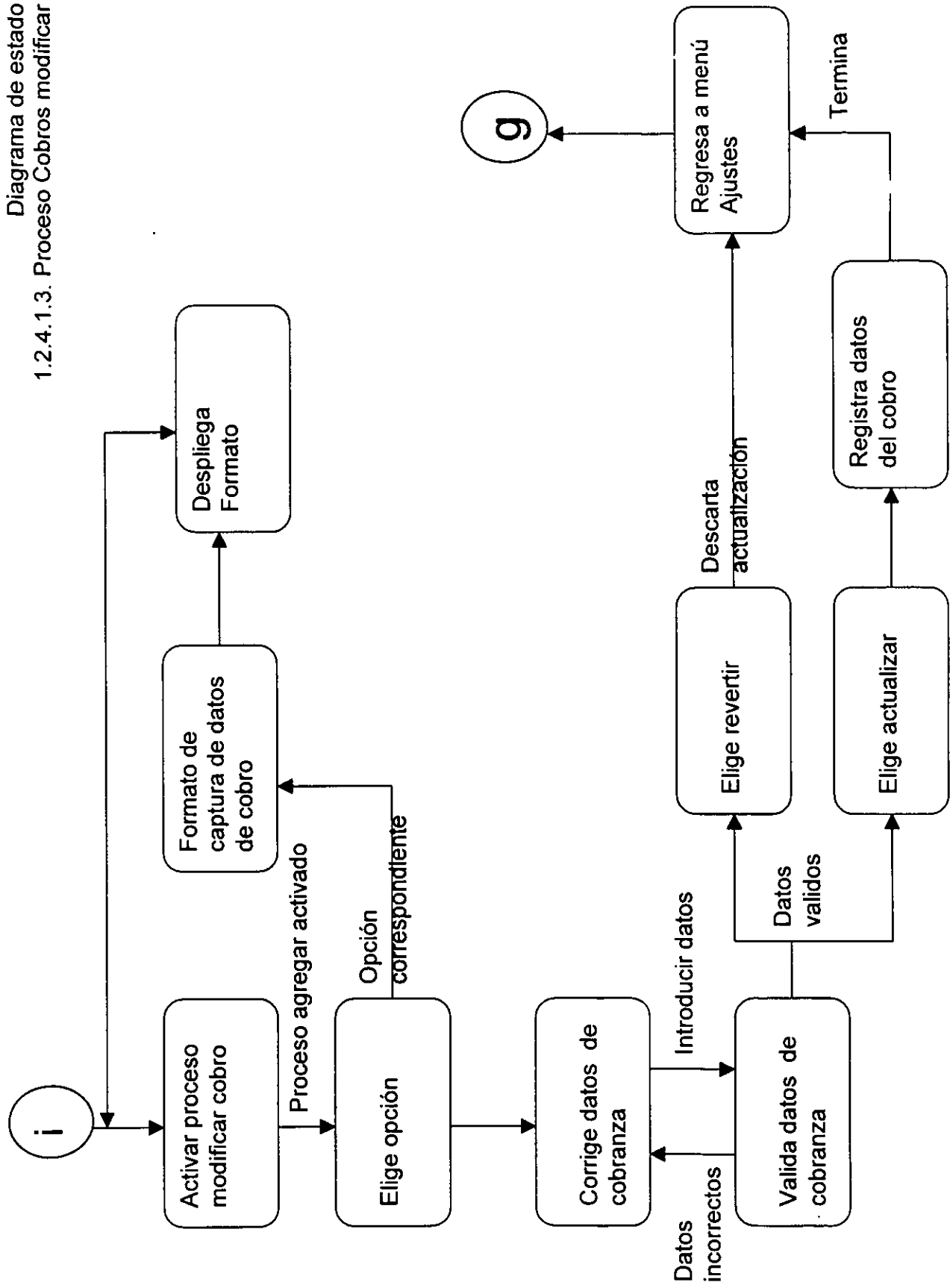


Diagrama de estado
1.2.4.1.4 Proceso eliminar Cobro

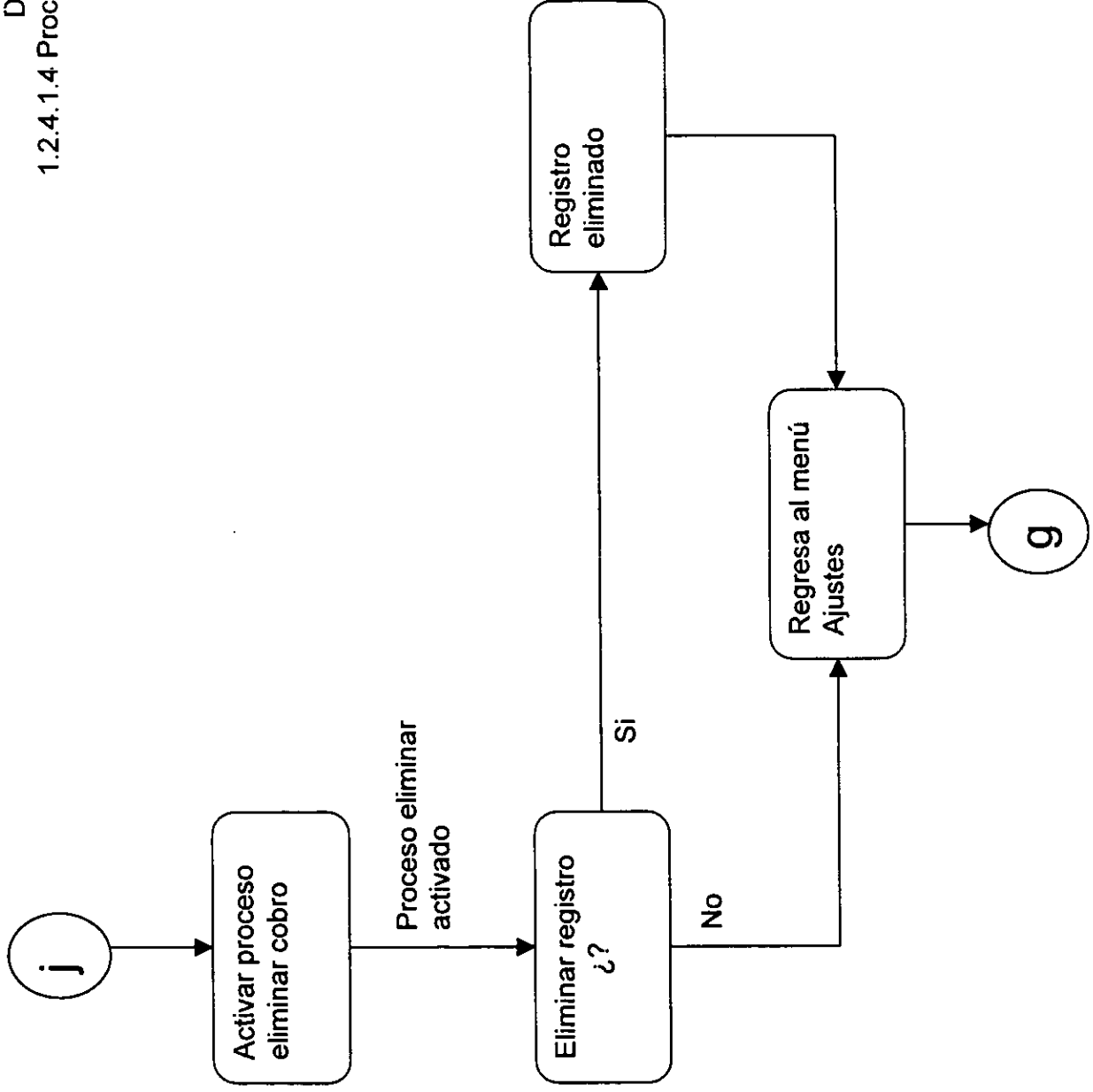


Diagrama de estado
1.2.4.1.5. Proceso cobros Imprimir

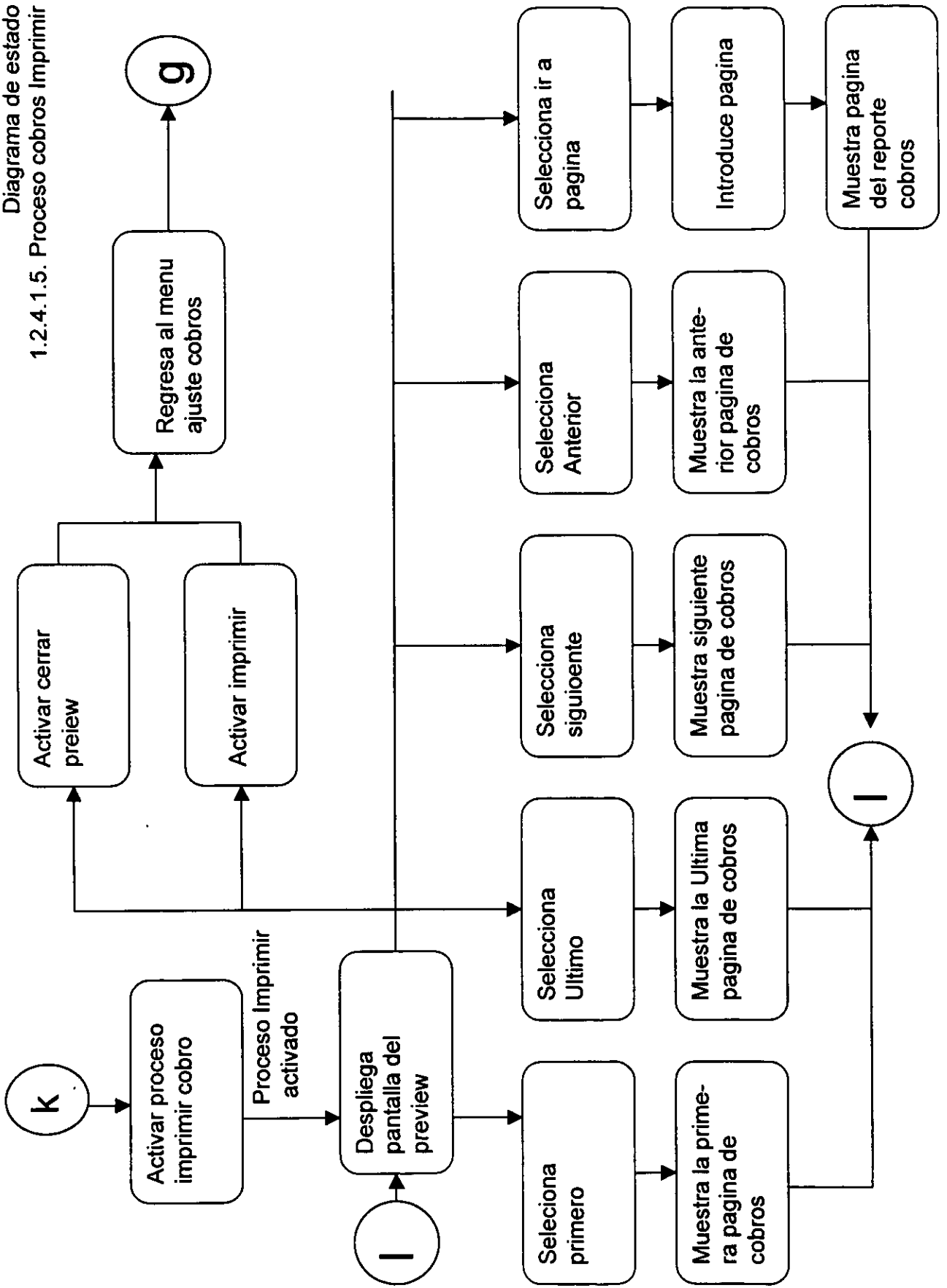


Diagrama de estado
 1.2.4.2 Proceso ajuste de pagos

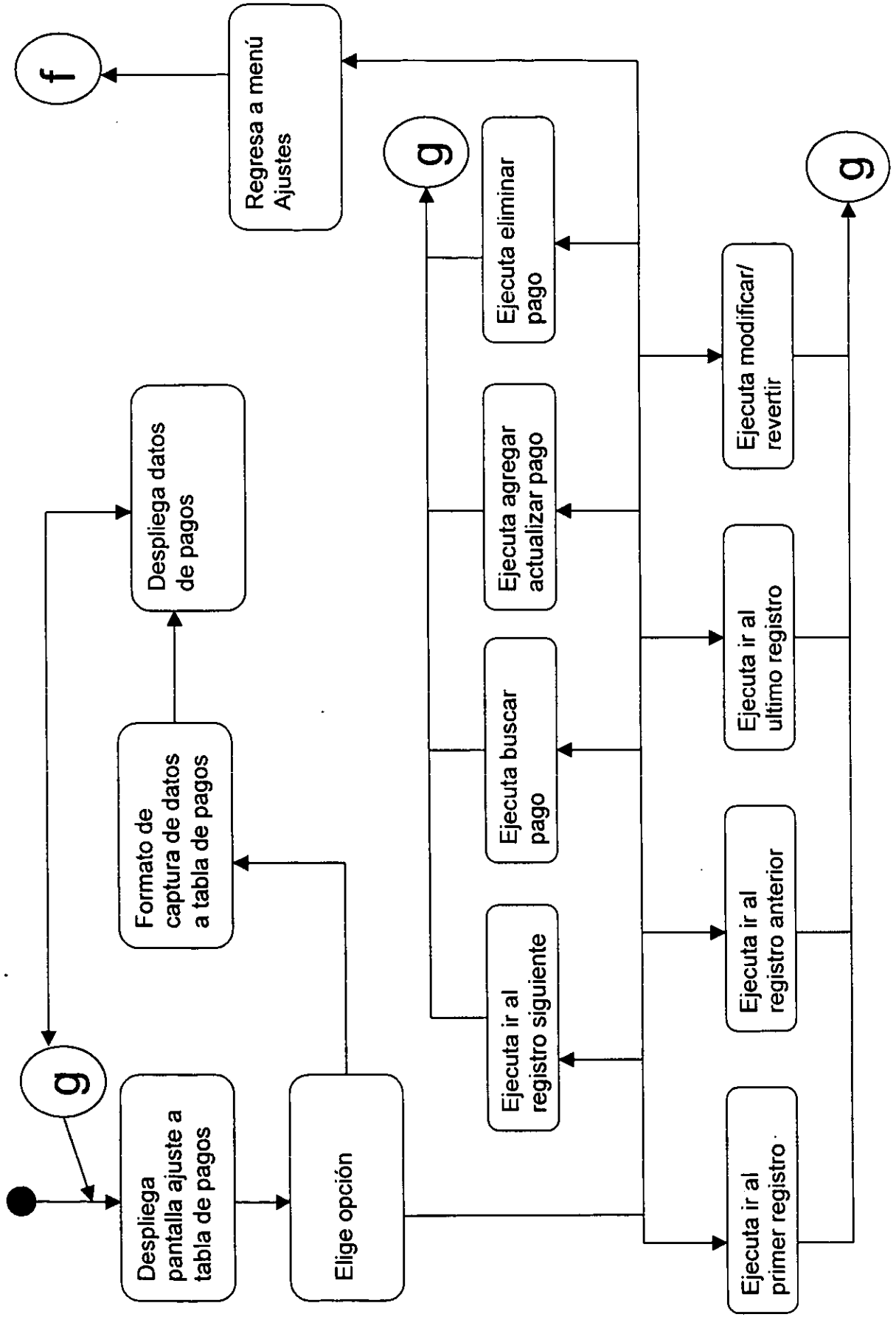


Diagrama de estado
1.2.4.2.1 Proceso pagos agregar

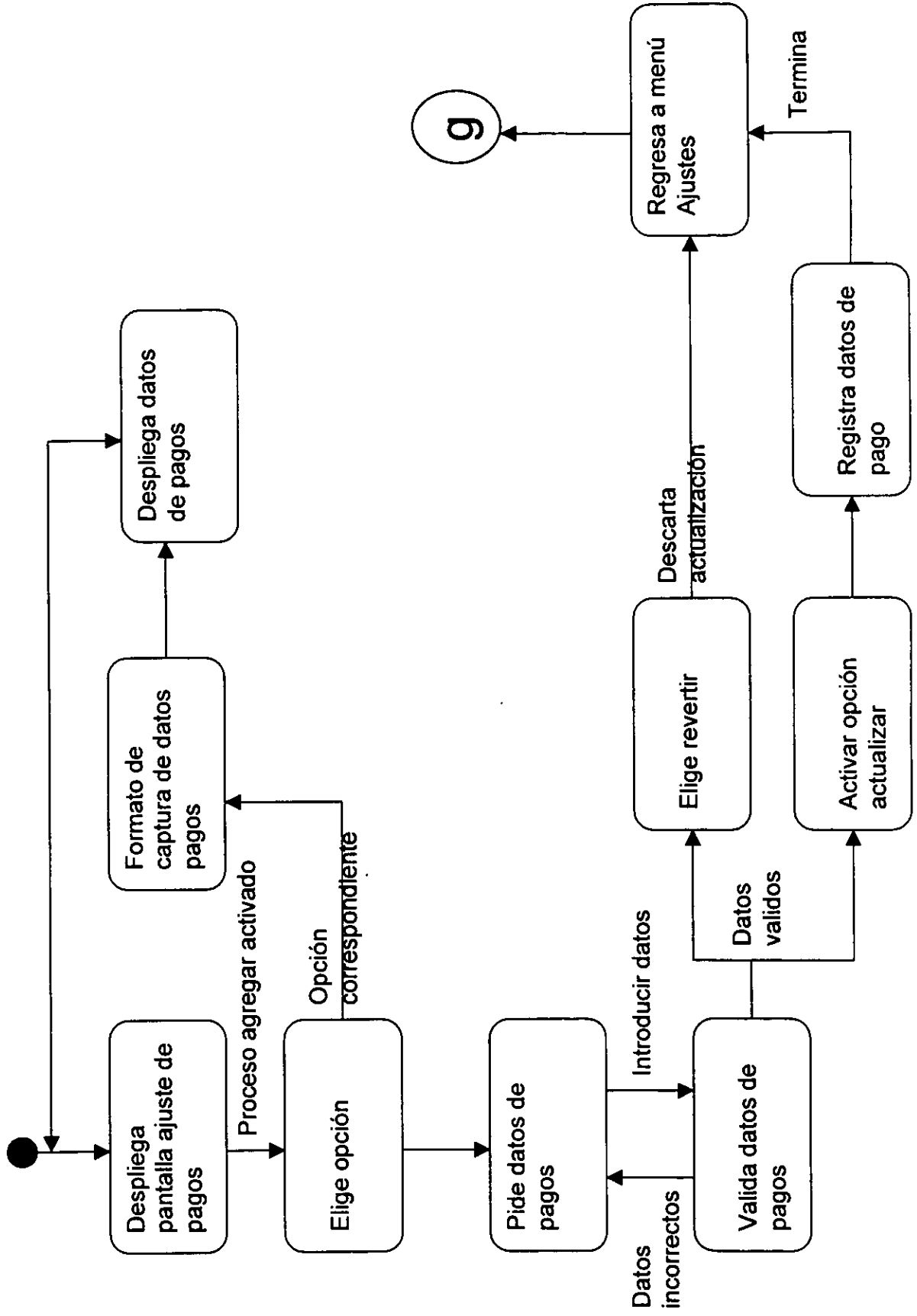


Diagrama de Estado
1.2.4.2.2 Proceso pagos seleccionar

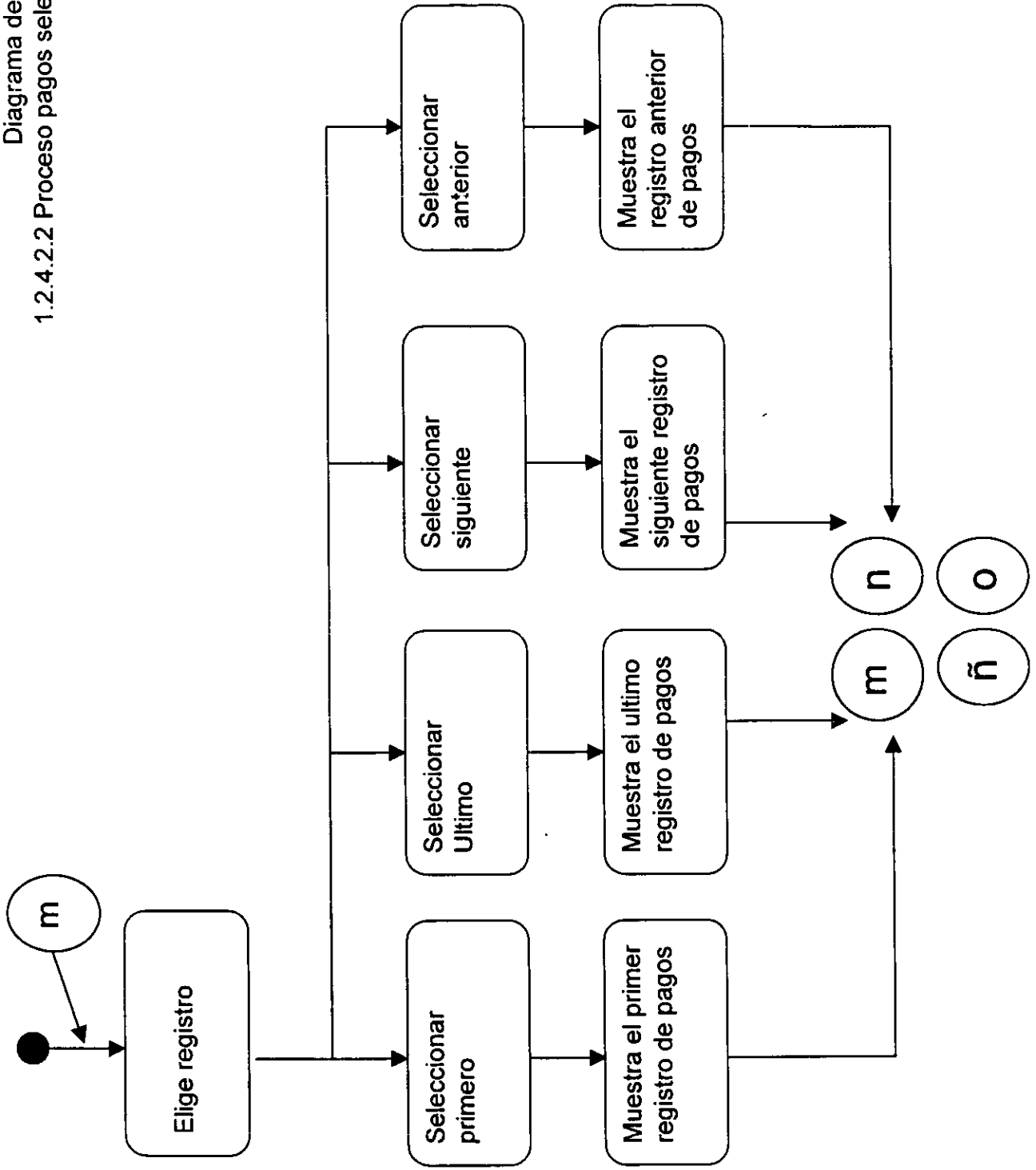


Diagrama de estado
 1.2.4.2.3. Proceso Pagos modificar

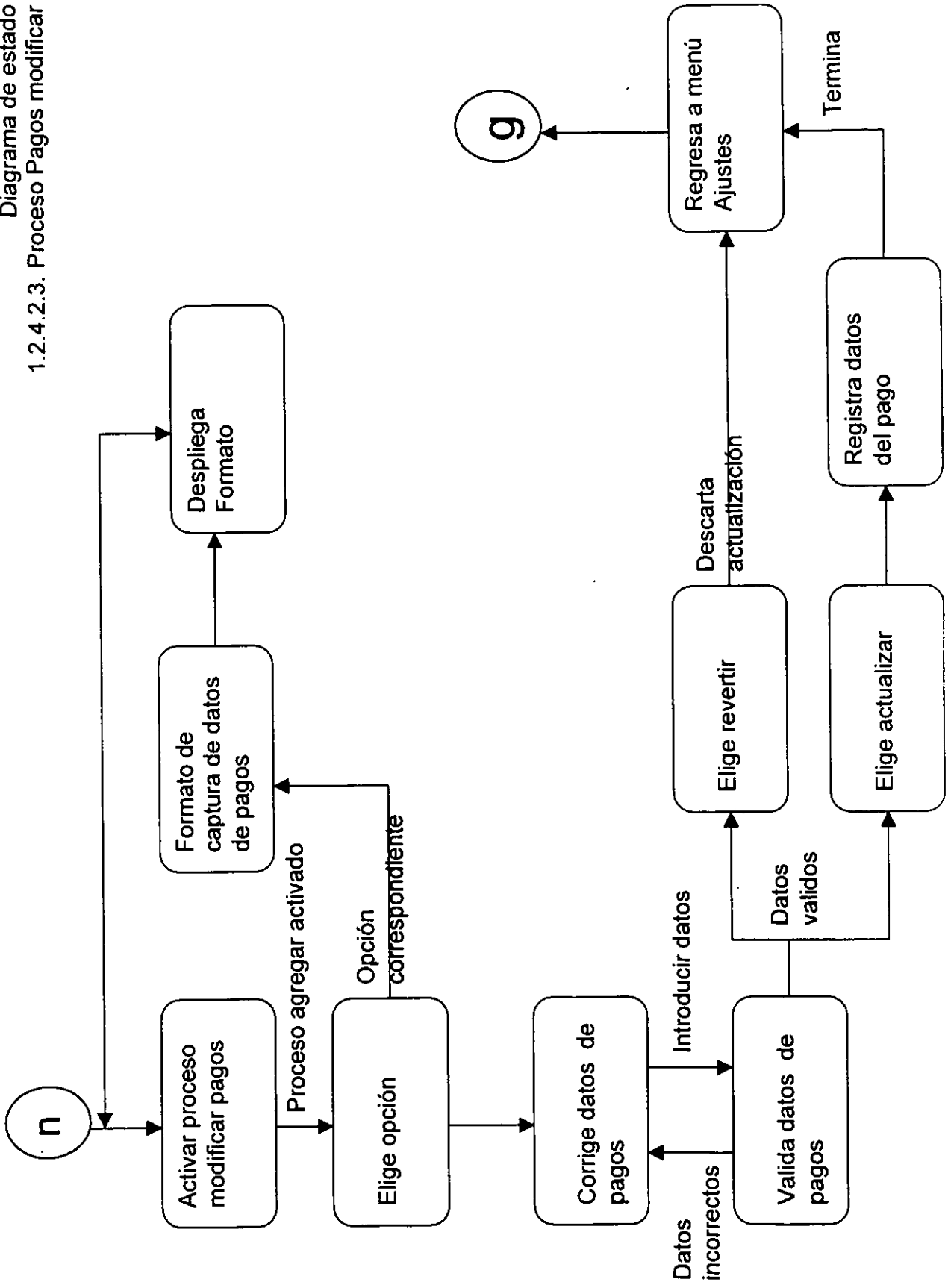


Diagrama de estado
1.2.4.2.4 Proceso eliminar Pagos

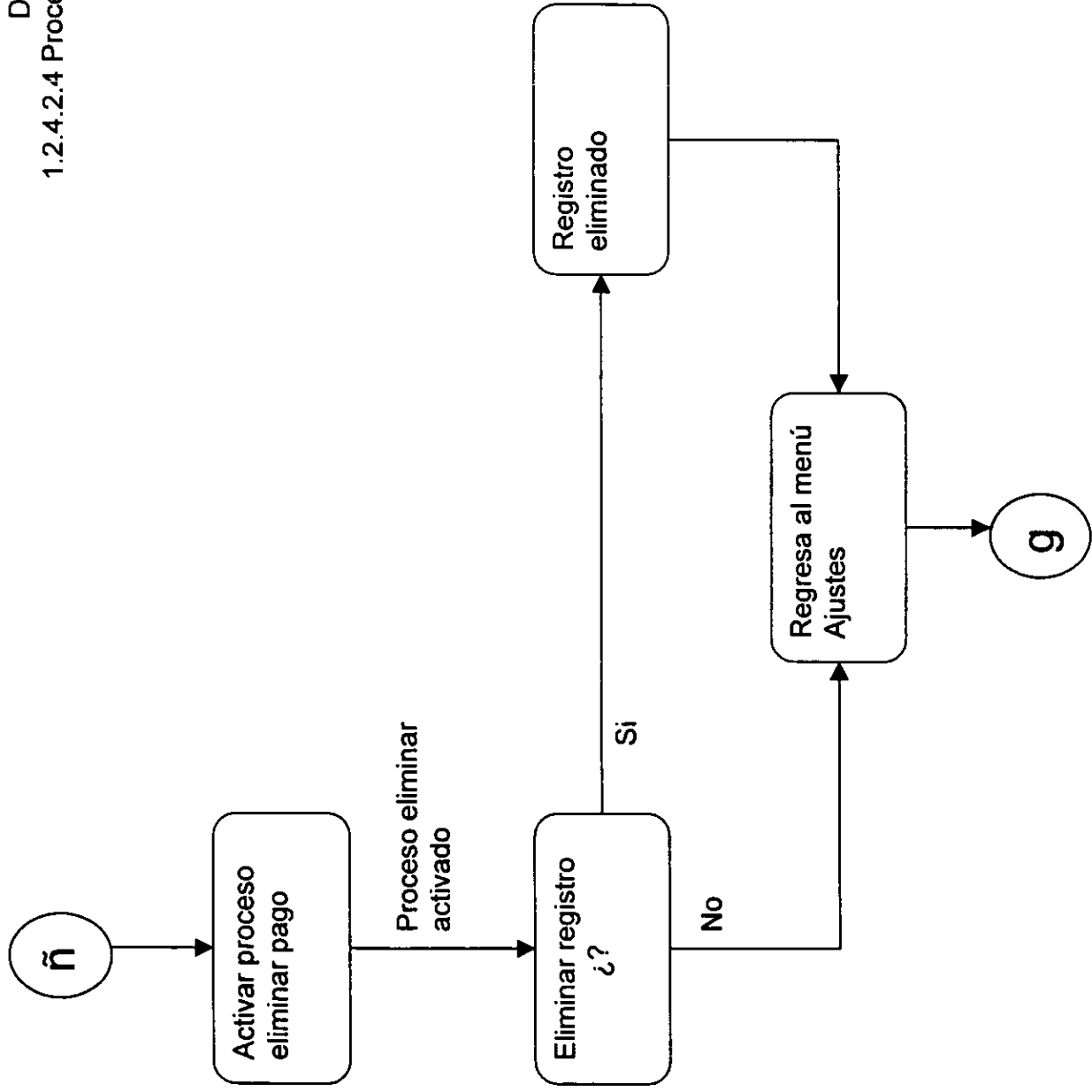


Diagrama de estado
1.2.4.2.5. Proceso pagos Imprimir

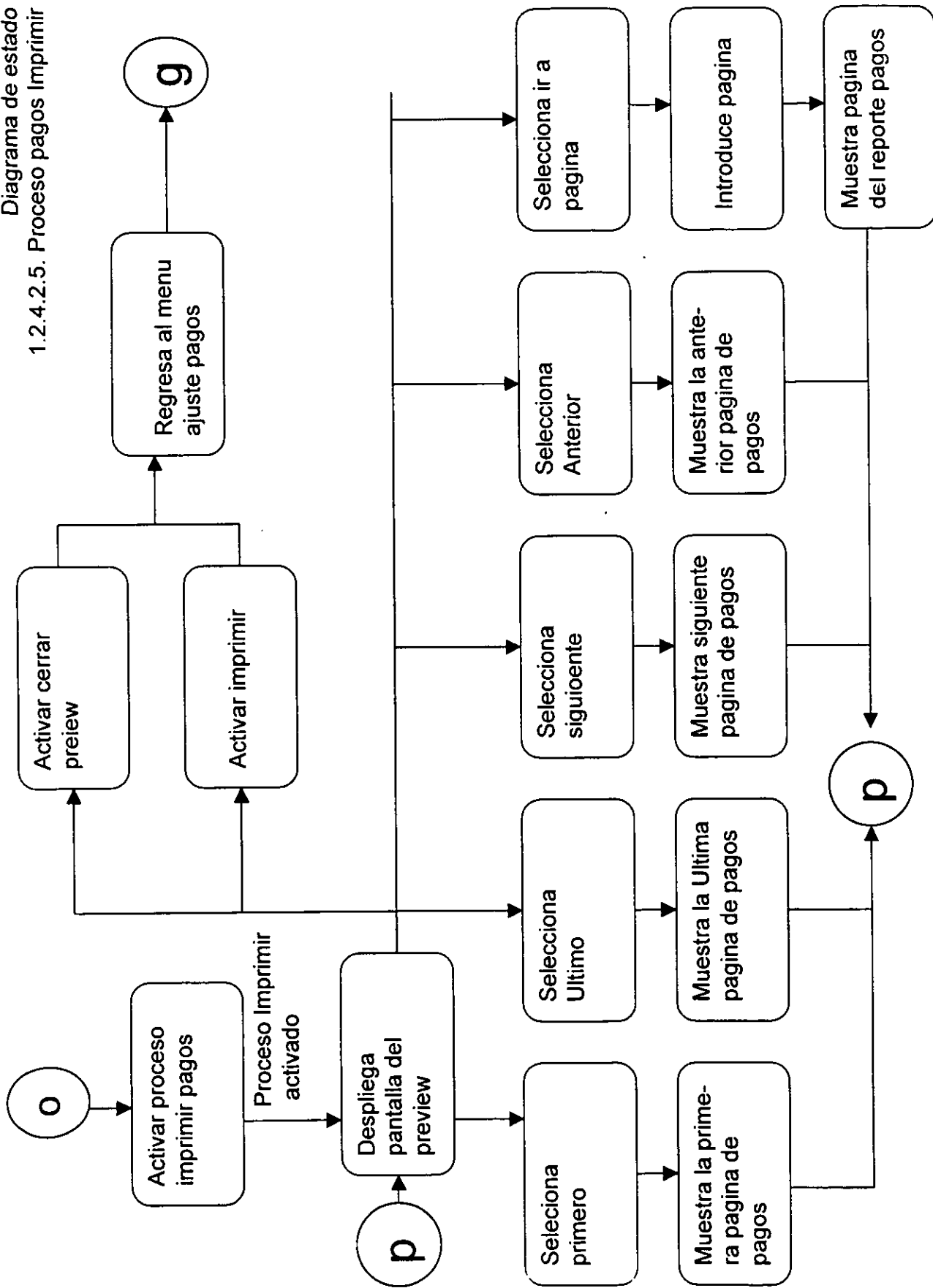


Diagrama de estado

1.3. Menú Generación de reportes

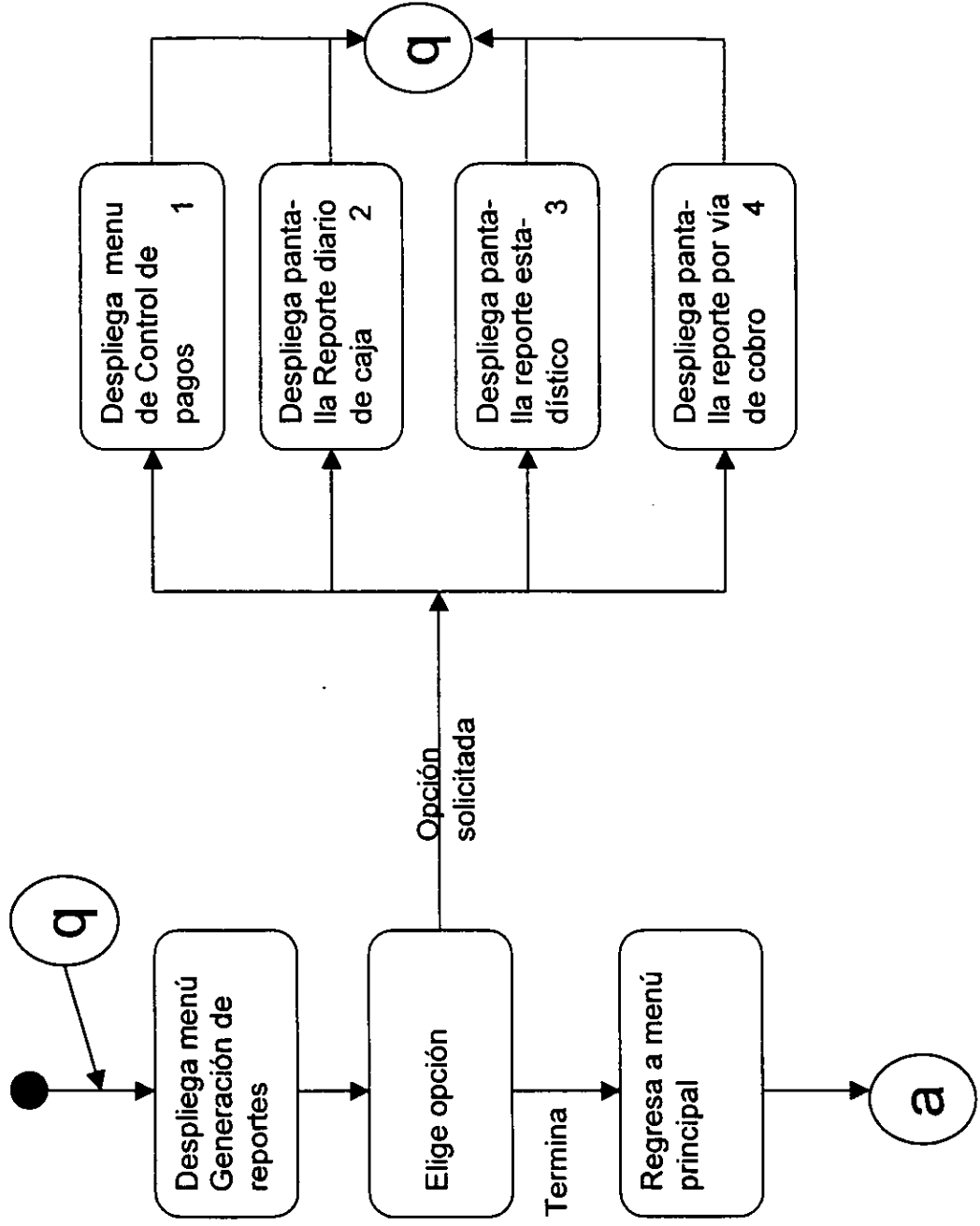


Diagrama de estado
1.3.1 Menú Control de pagos

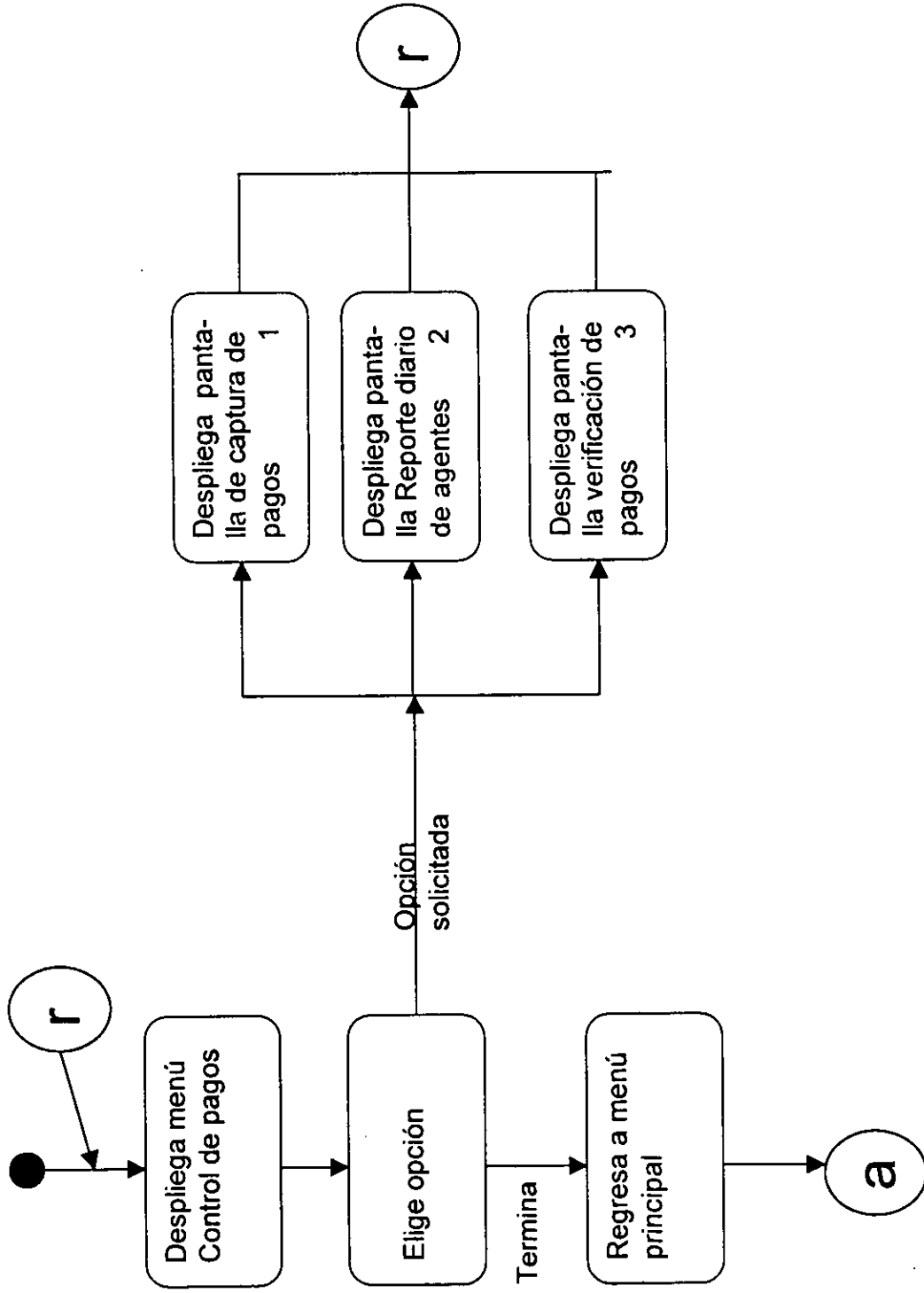


Diagrama de estado
 1.3.1.1 Proceso captura de pagos.

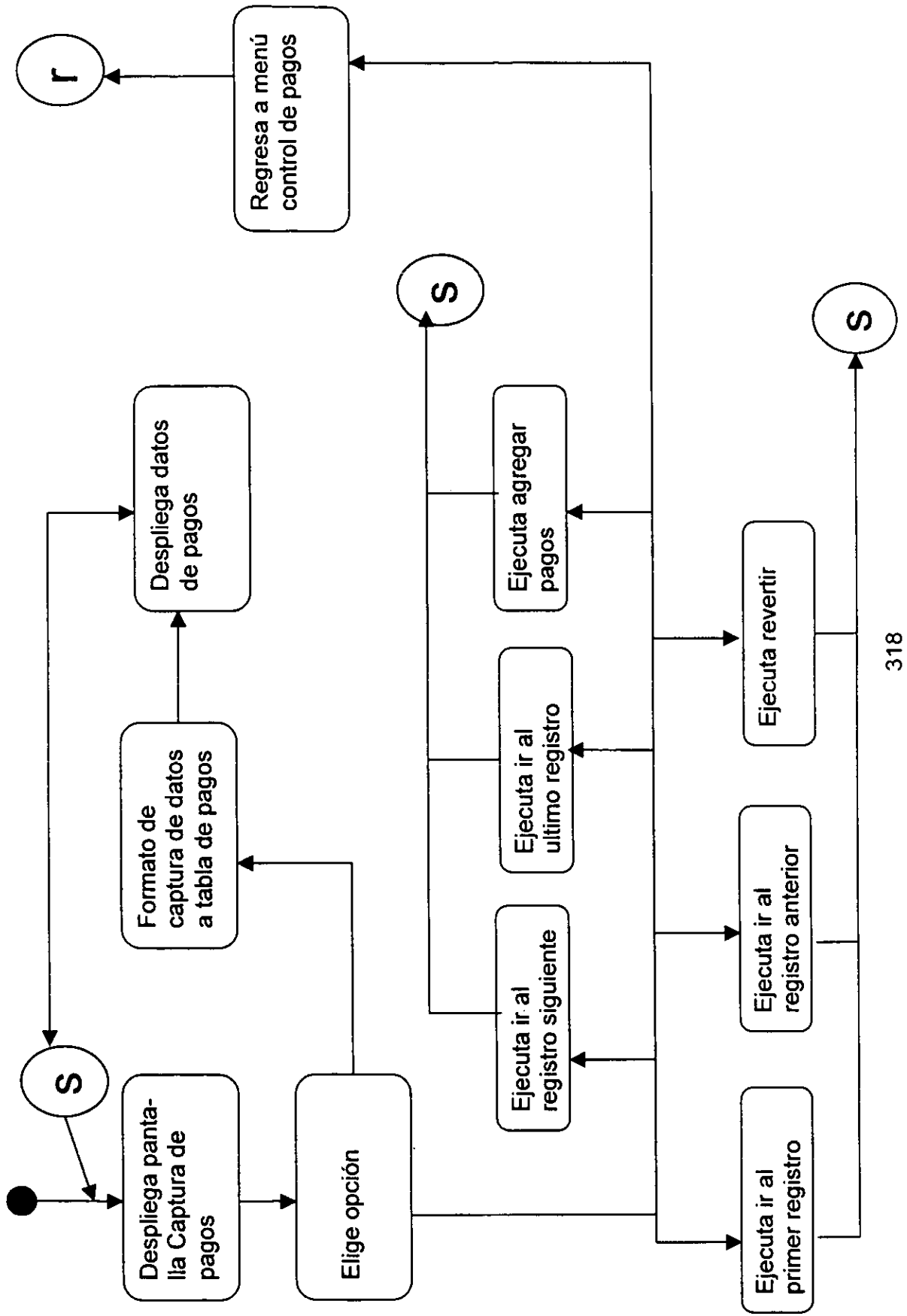


Diagrama de estado

1.3.1.1.1 Proceso agregar pago

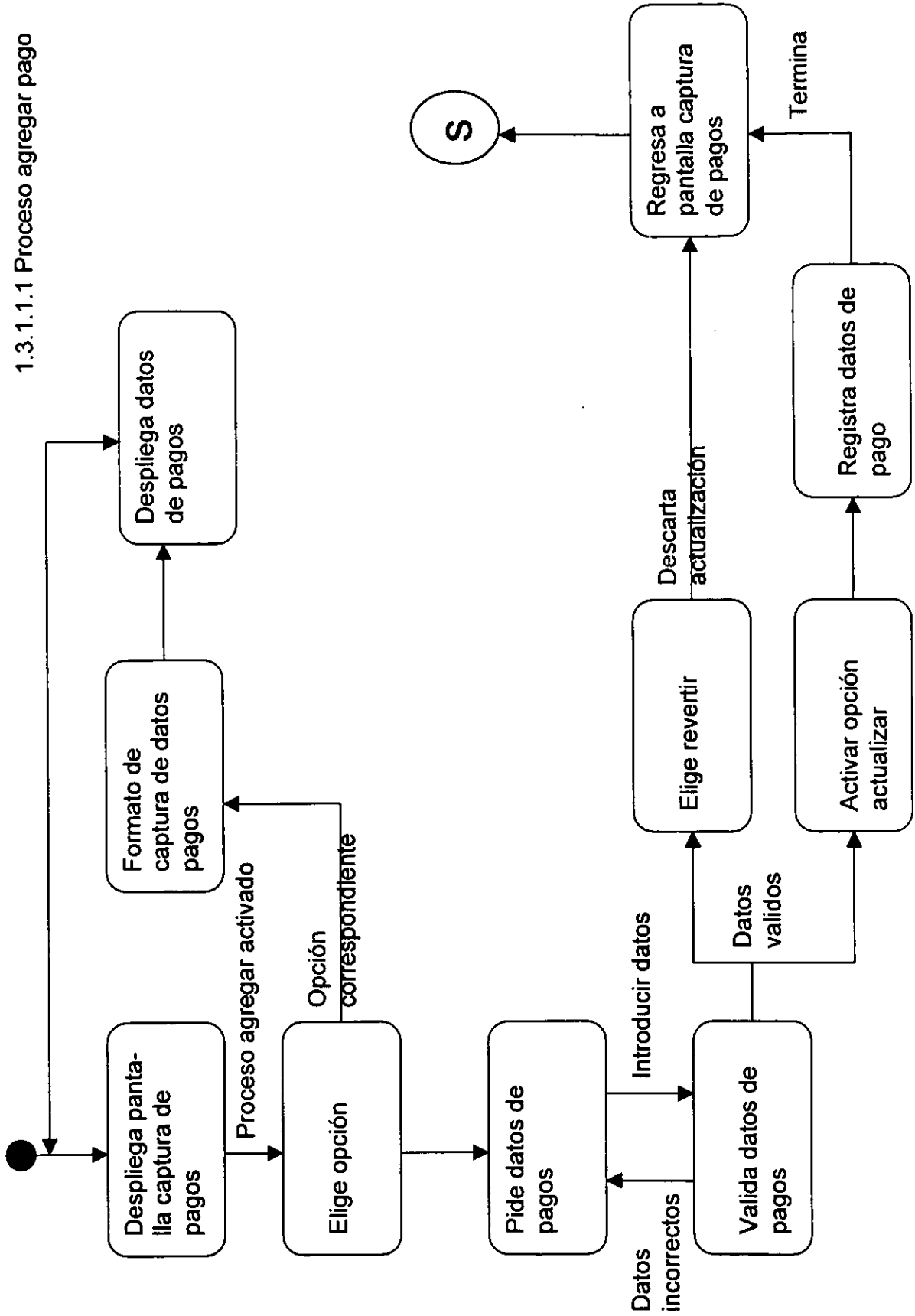


Diagrama de estado
1.3.1.2 Proceso Diario de agente

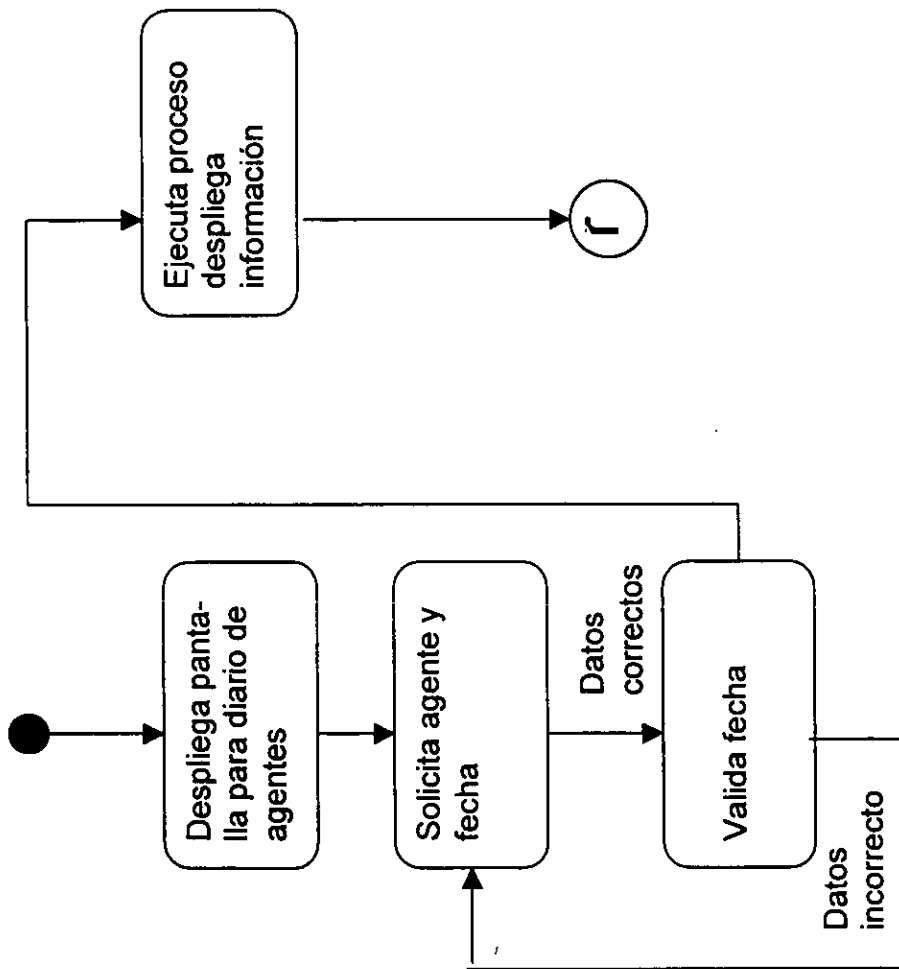


Diagrama de estado
1.3.1.3 Verifica pago de cliente

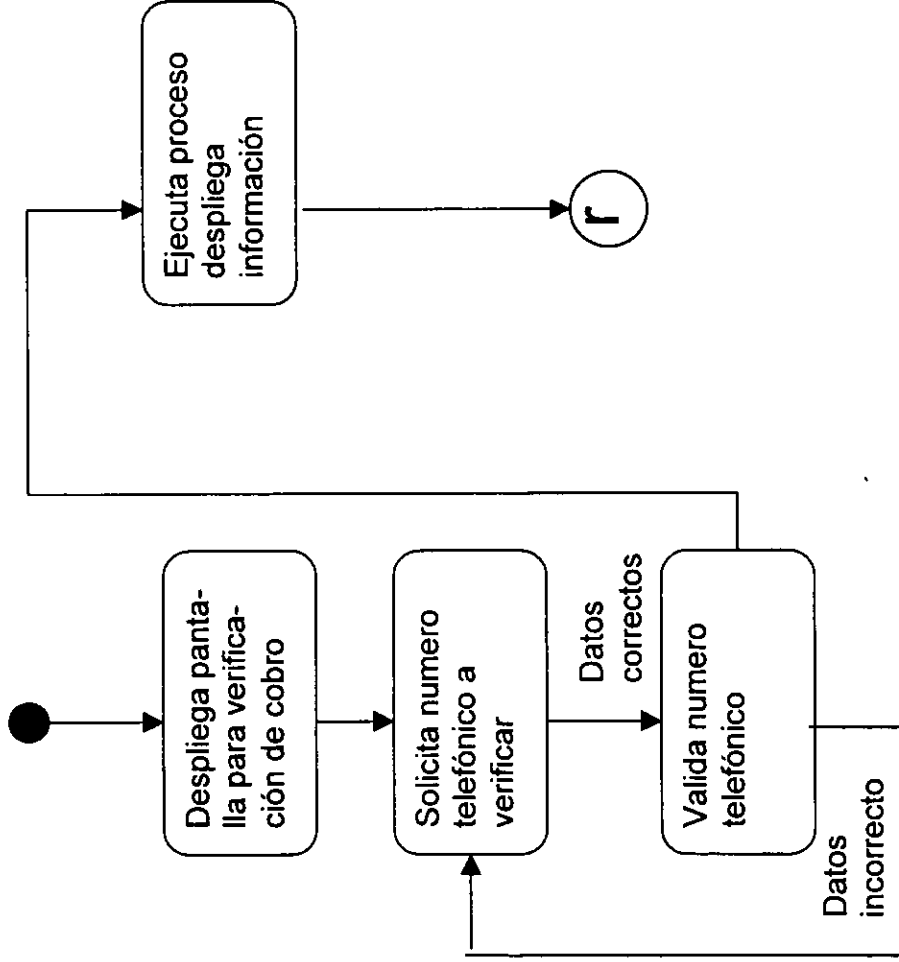


Diagrama de estado

1.3.2 Proceso diario de caja

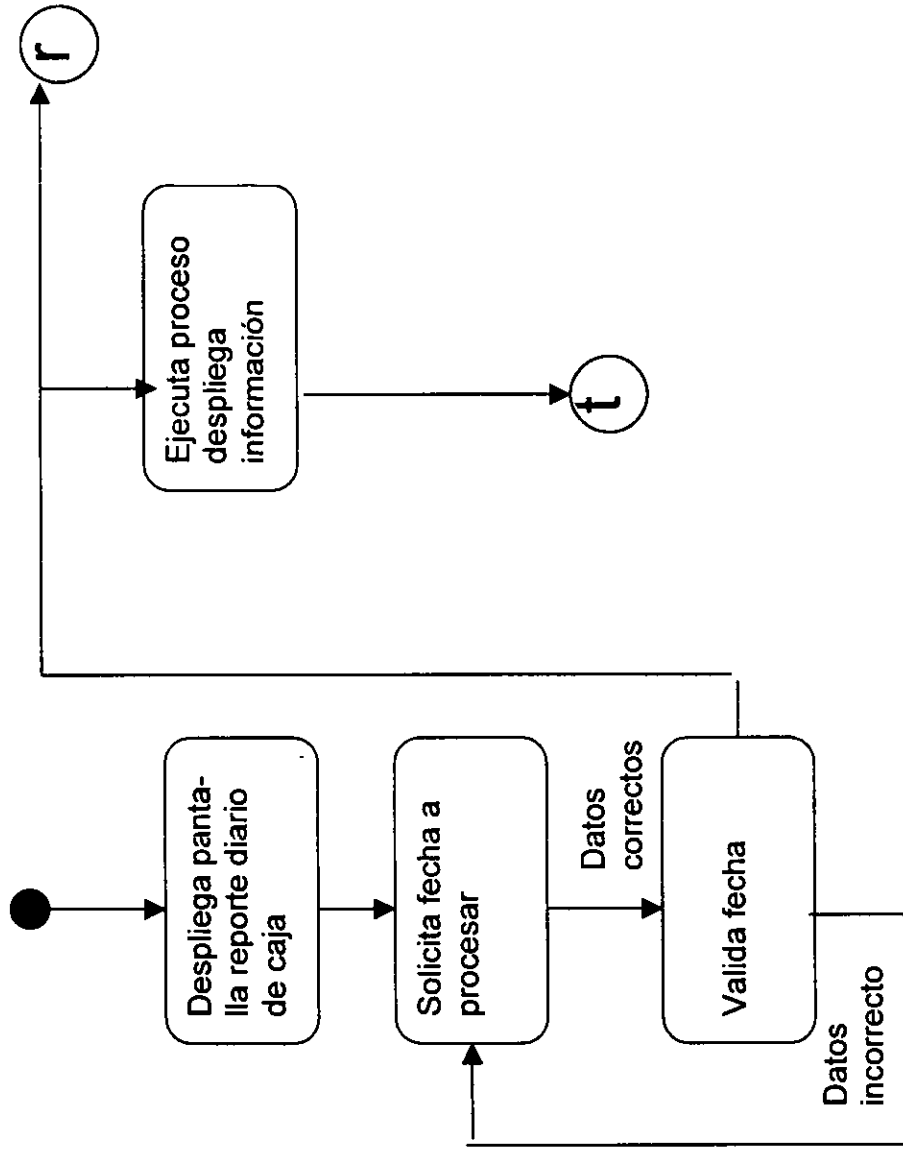


Diagrama de estado
1.3.2.1 Proceso imprimir diario de caja

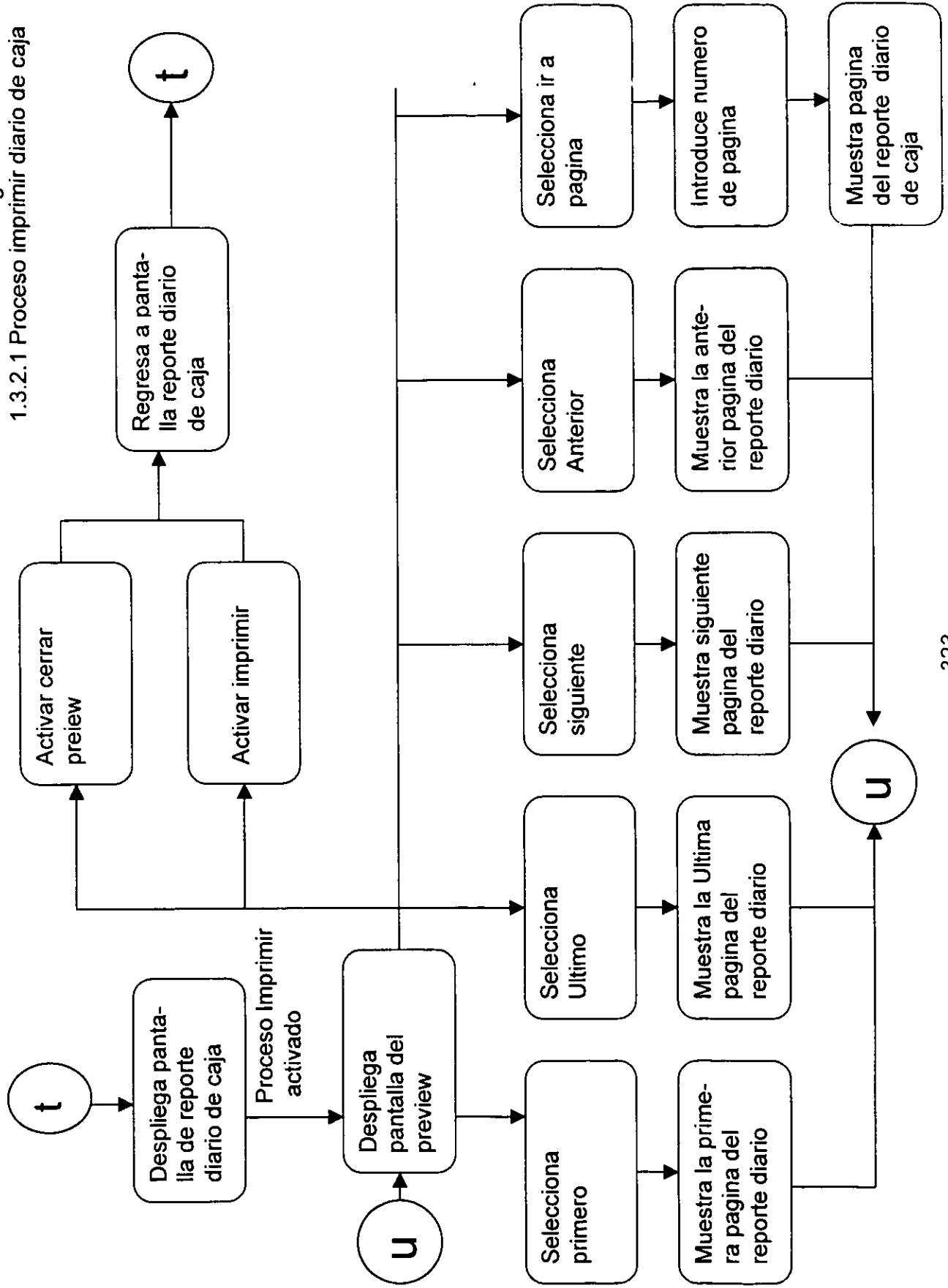


Diagrama de Estado
1.3.3 Proceso reporte estadístico

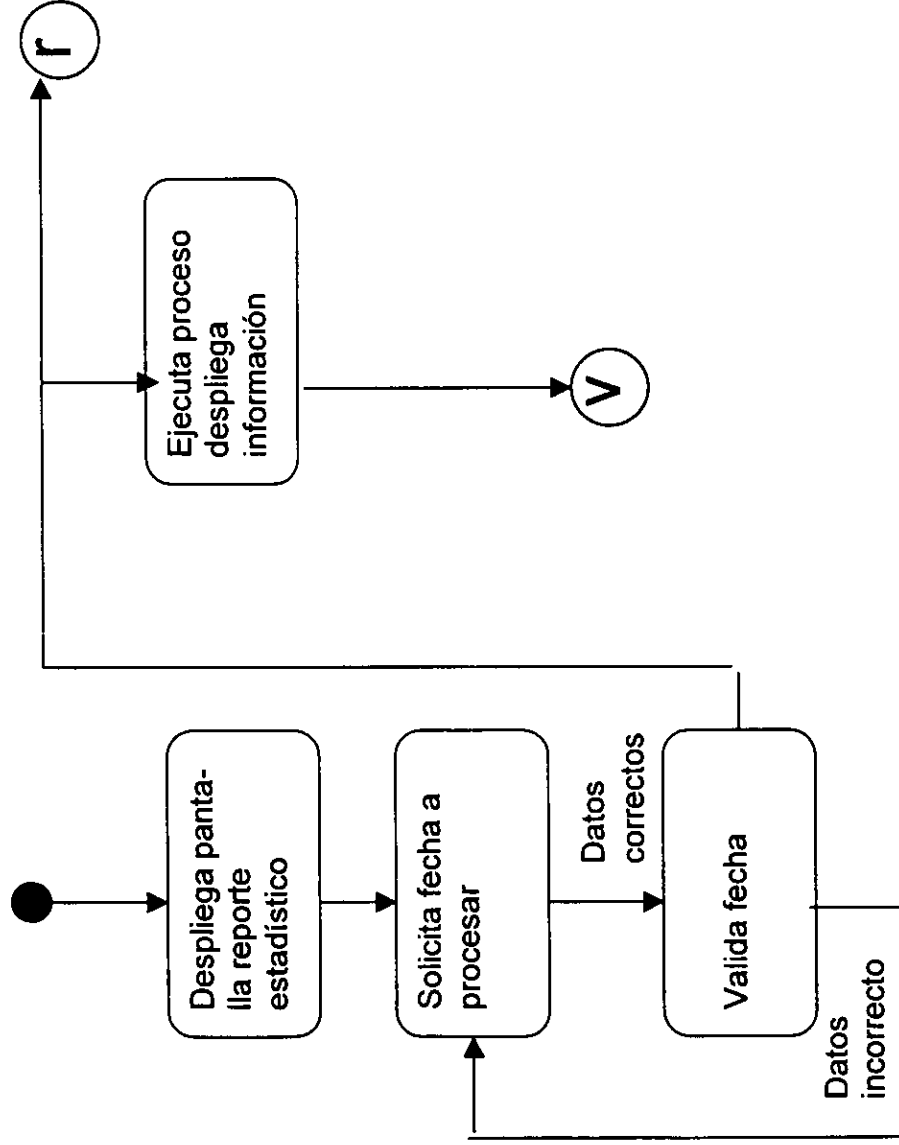


Diagrama de estado
1.3.3.1 Proceso imprimir reporte estadístico

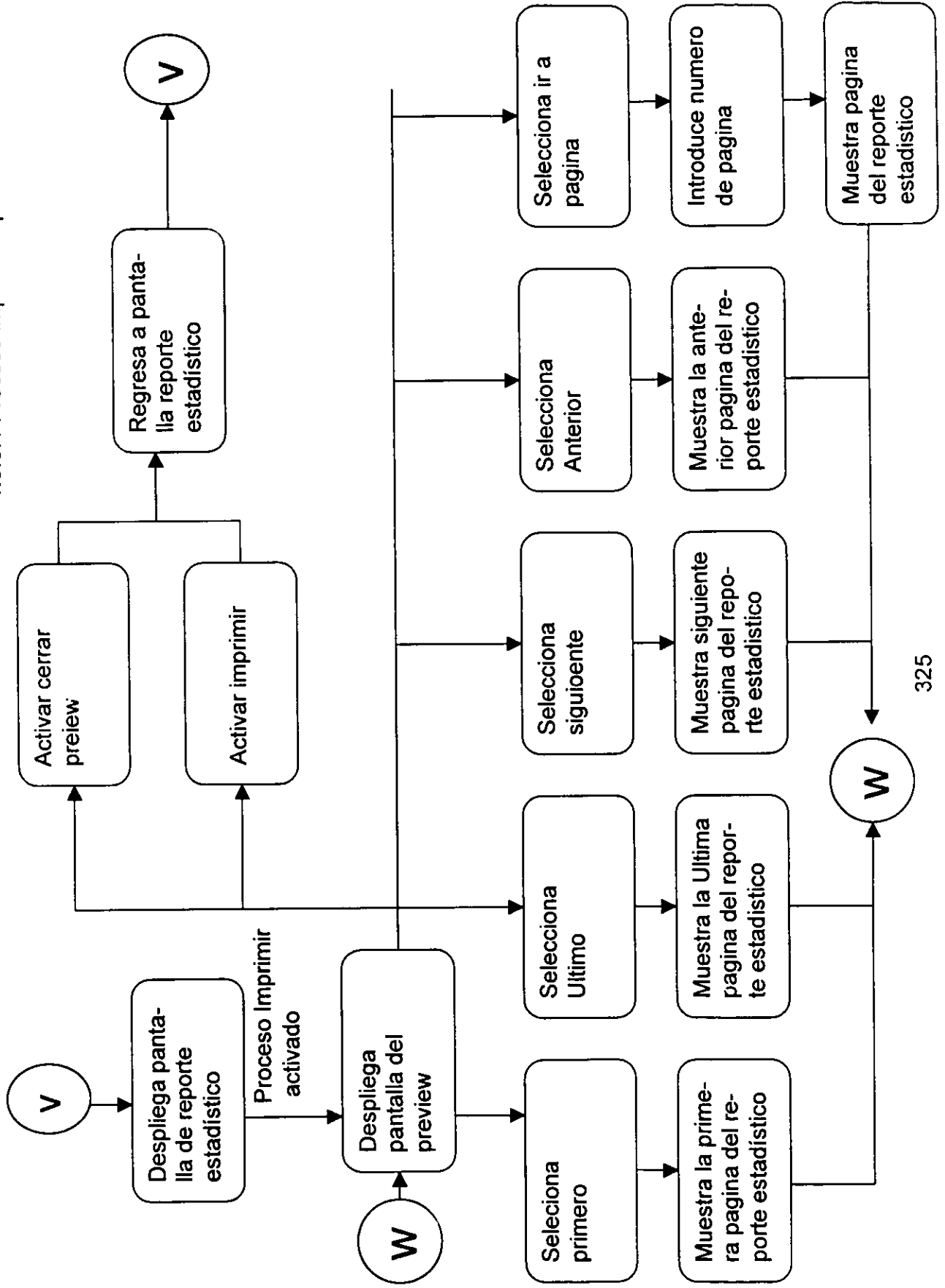


Diagrama de estado
1.3.4 Proceso reporte por via de cobro

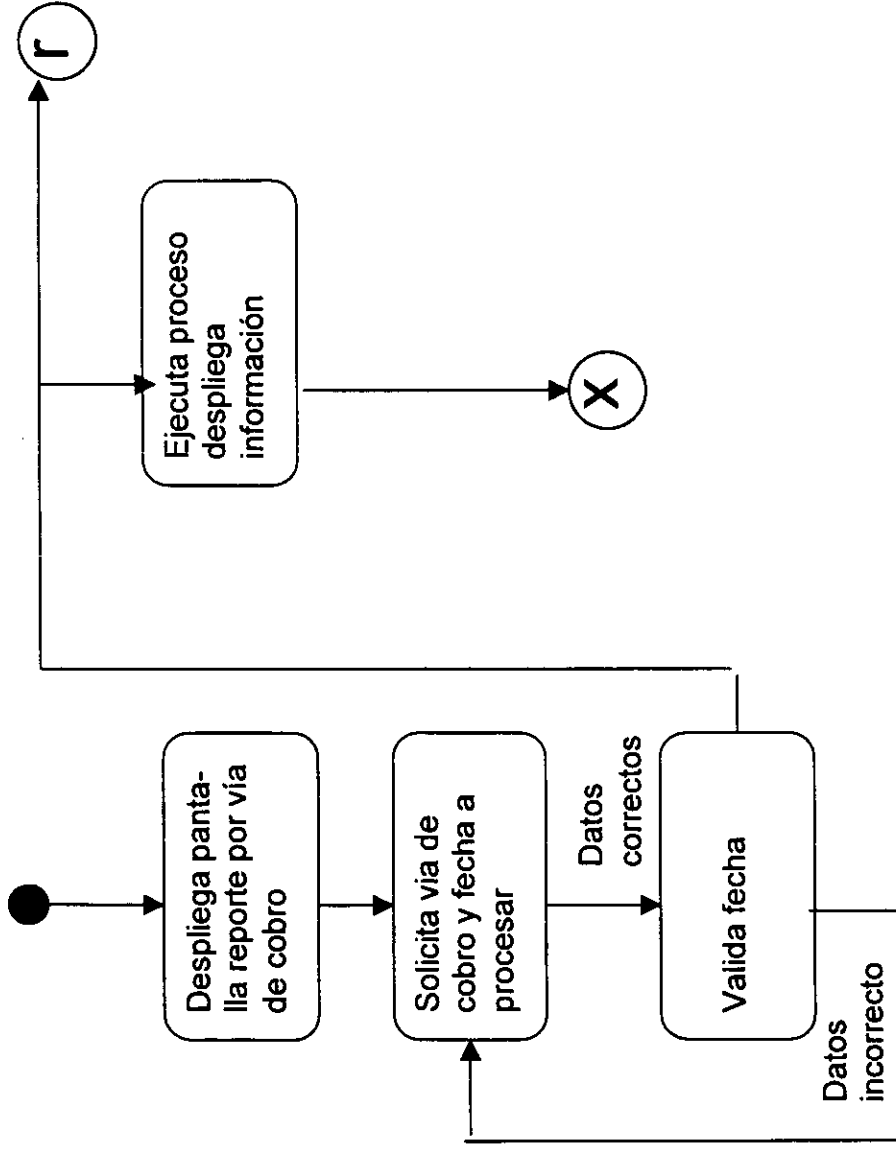


Diagrama de estado
1.3.4.1 Proceso cobros Imprimir

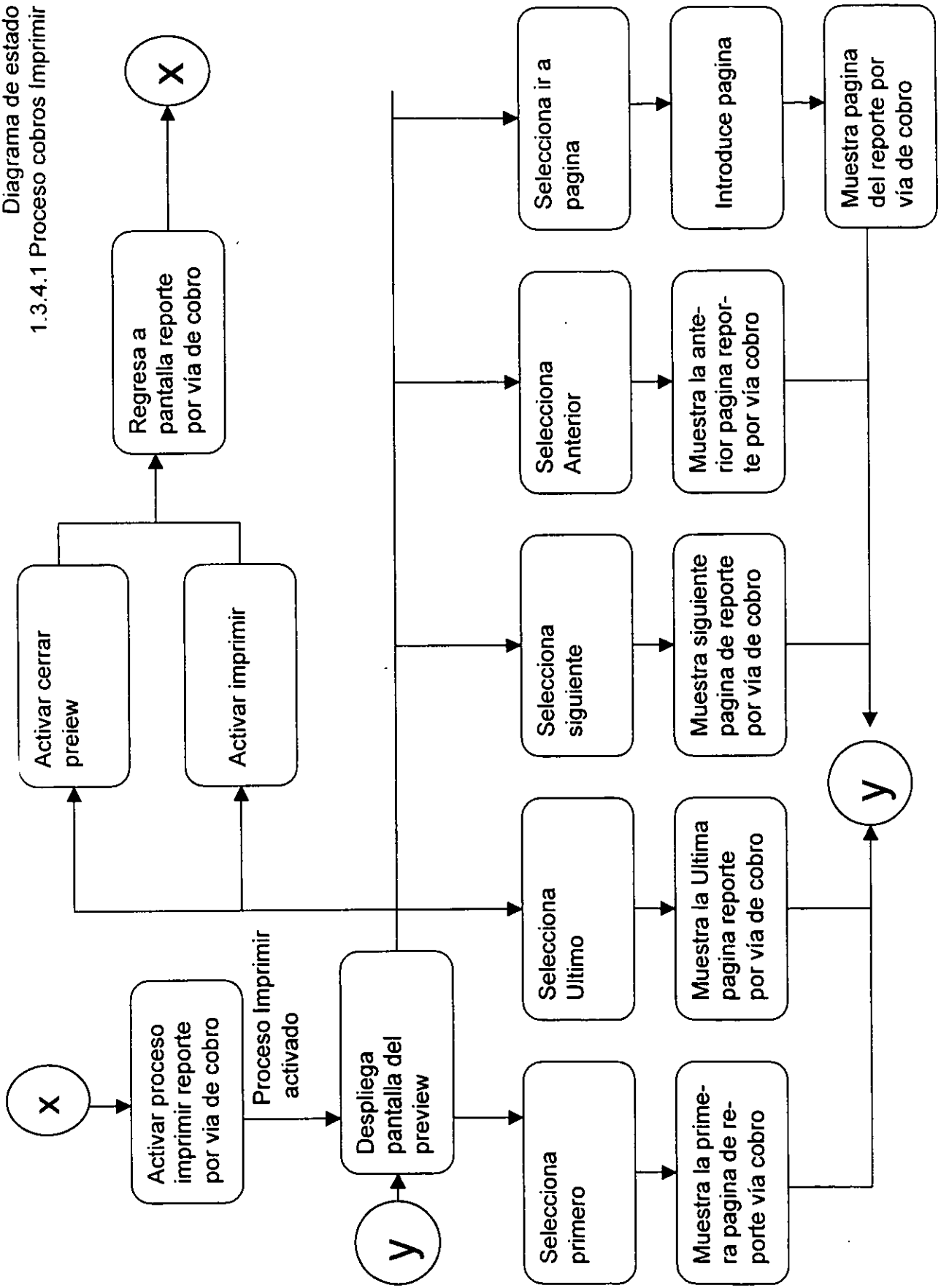


Diagrama de estado
4. Menú Mantenimiento de catálogos

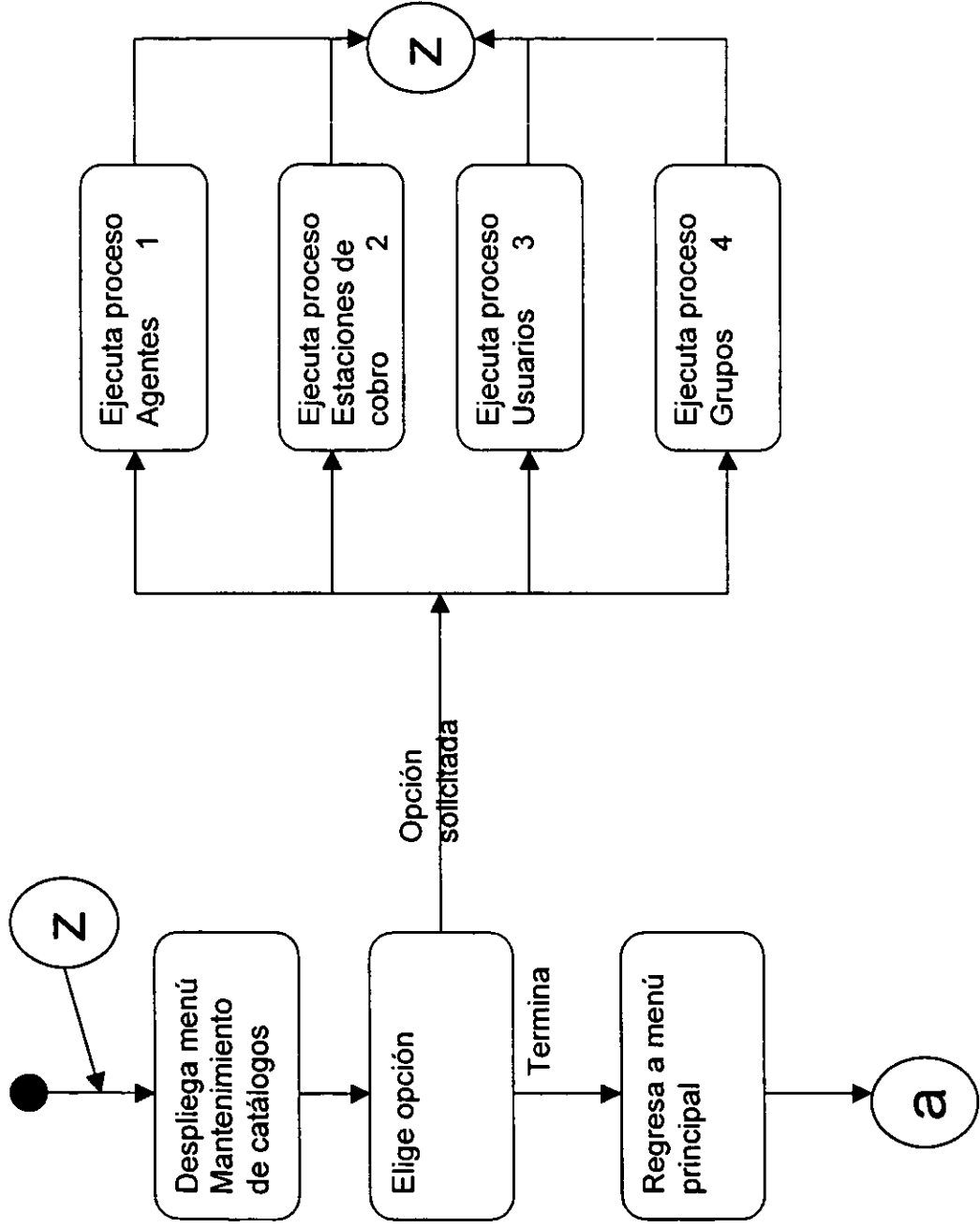


Diagrama de estado
4.1 Proceso Agentes

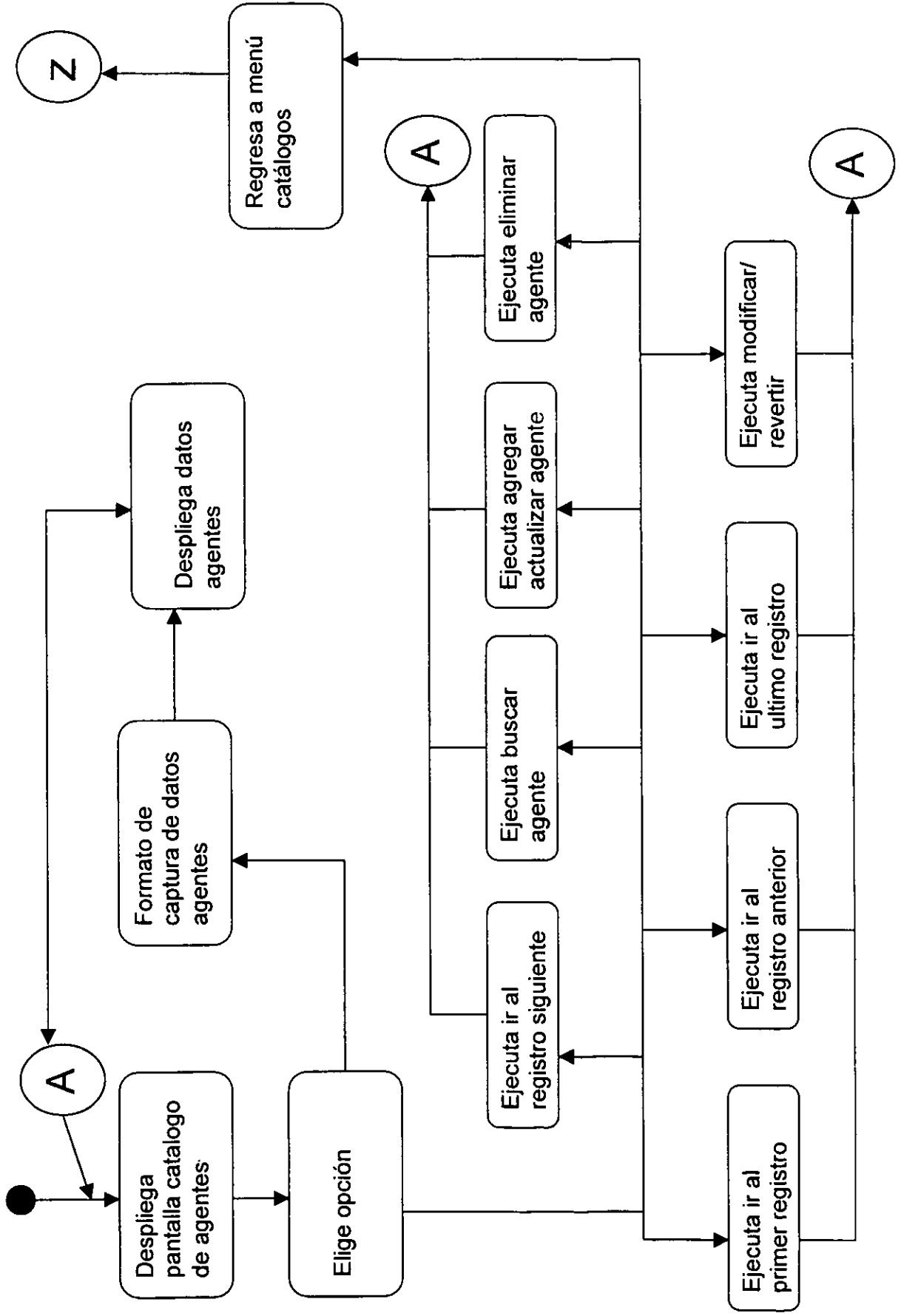


Diagrama de estado
4.1.1 Proceso agentes agregar

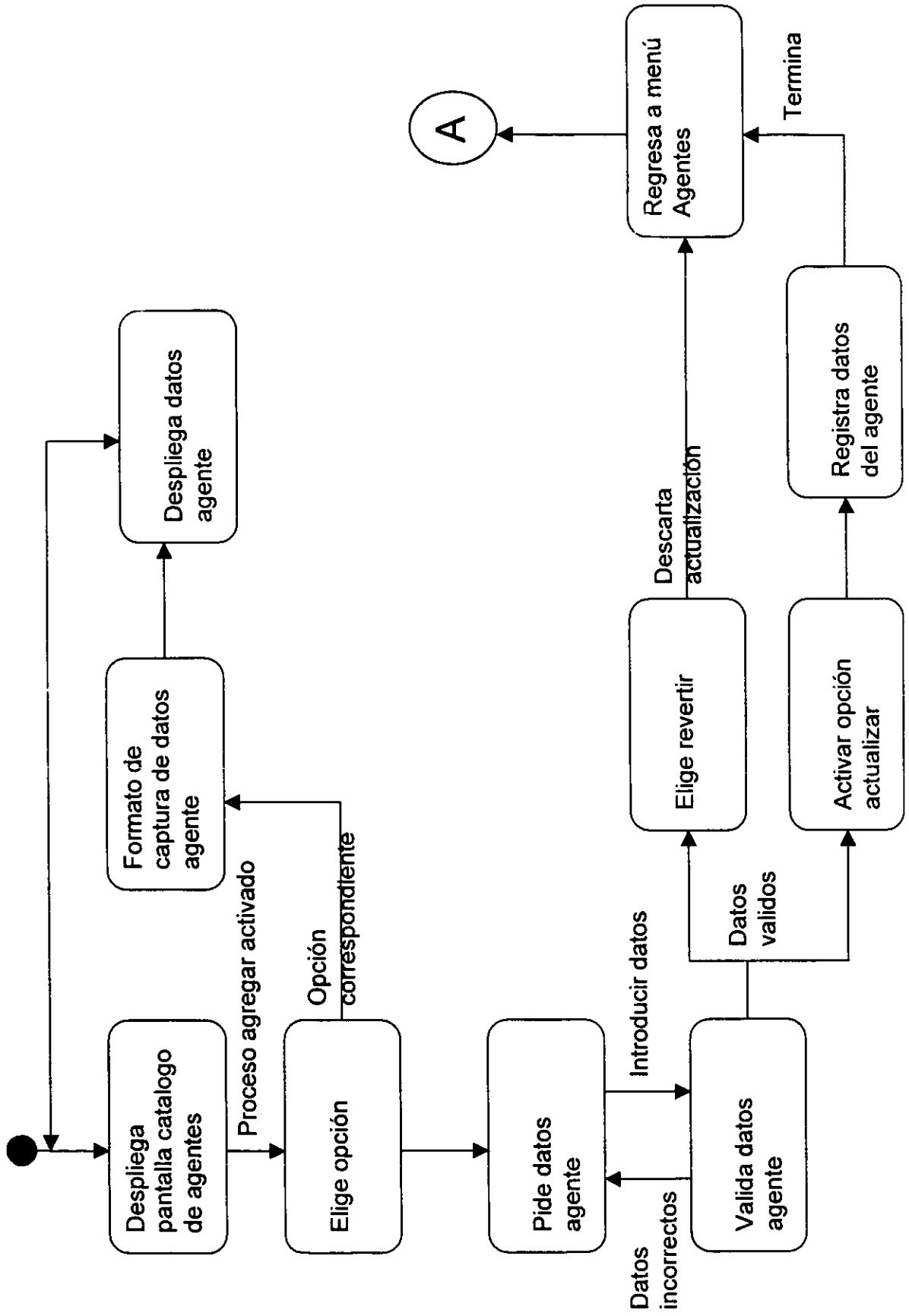


Diagrama de Estado Catálogos
4.1.2 Proceso agentes seleccionar

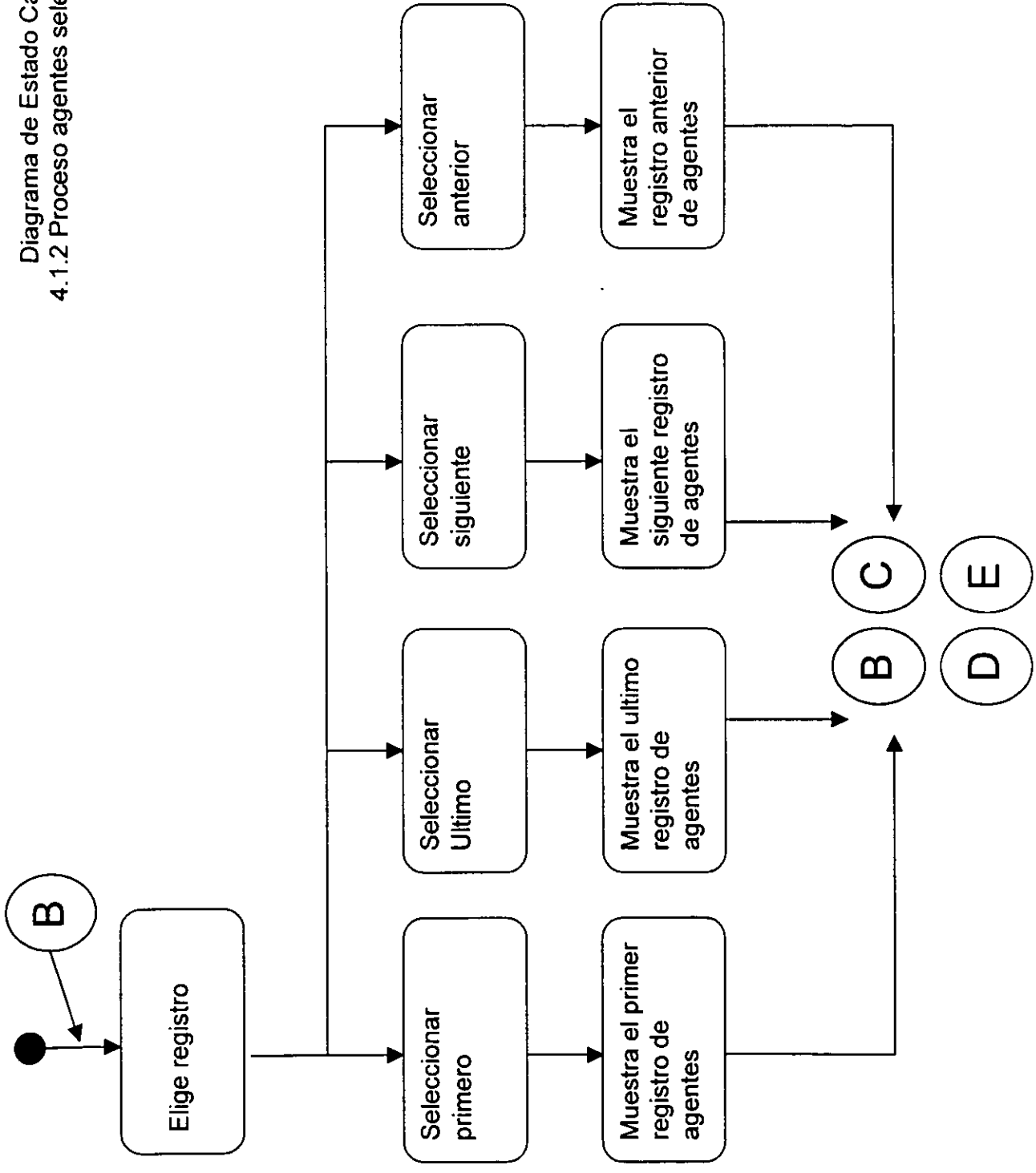


Diagrama de estado catálogos
4.1.3. Proceso Agentes modificar

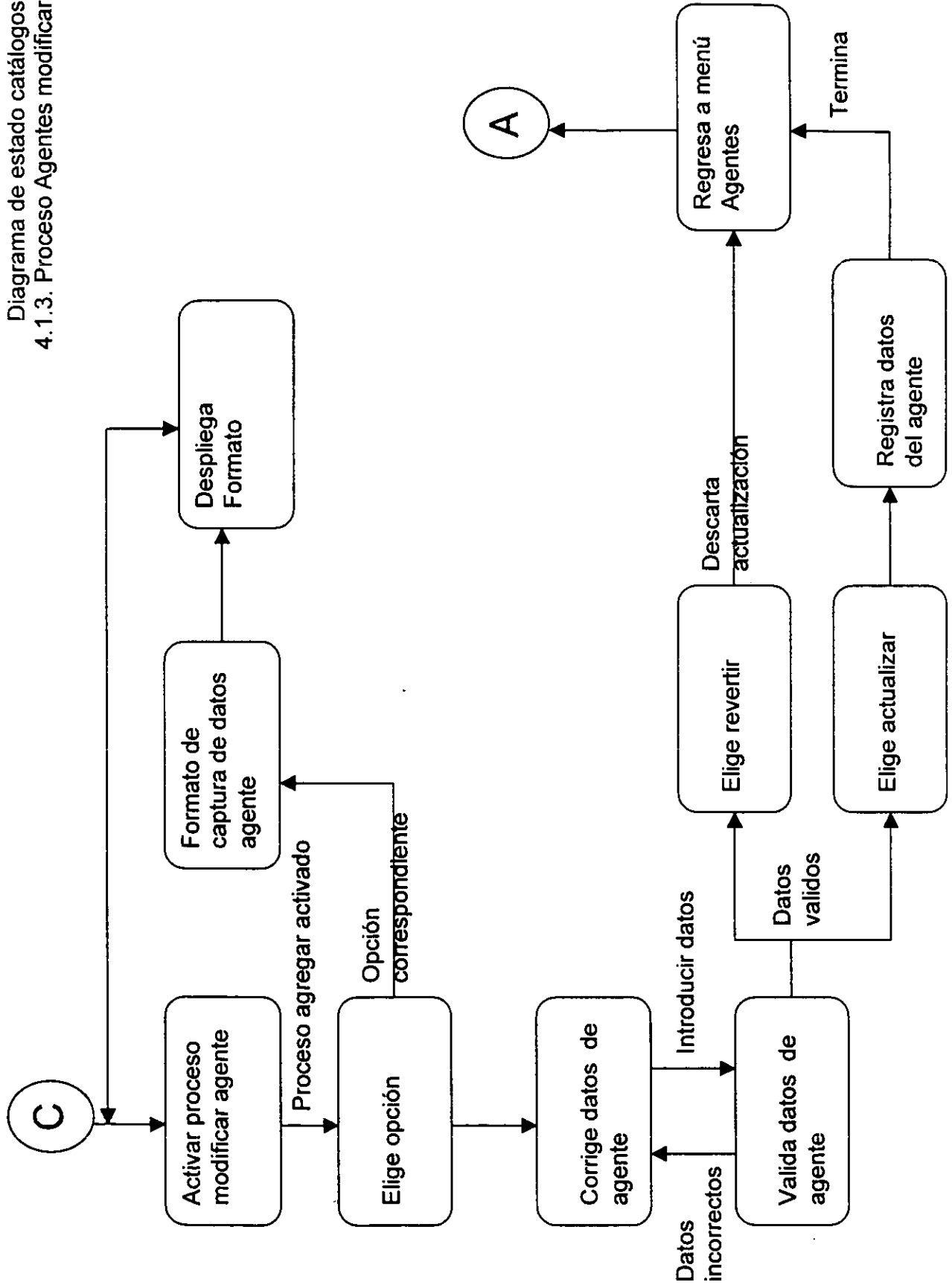


Diagrama de estado Catálogos
4.1.4 Proceso eliminar agentes

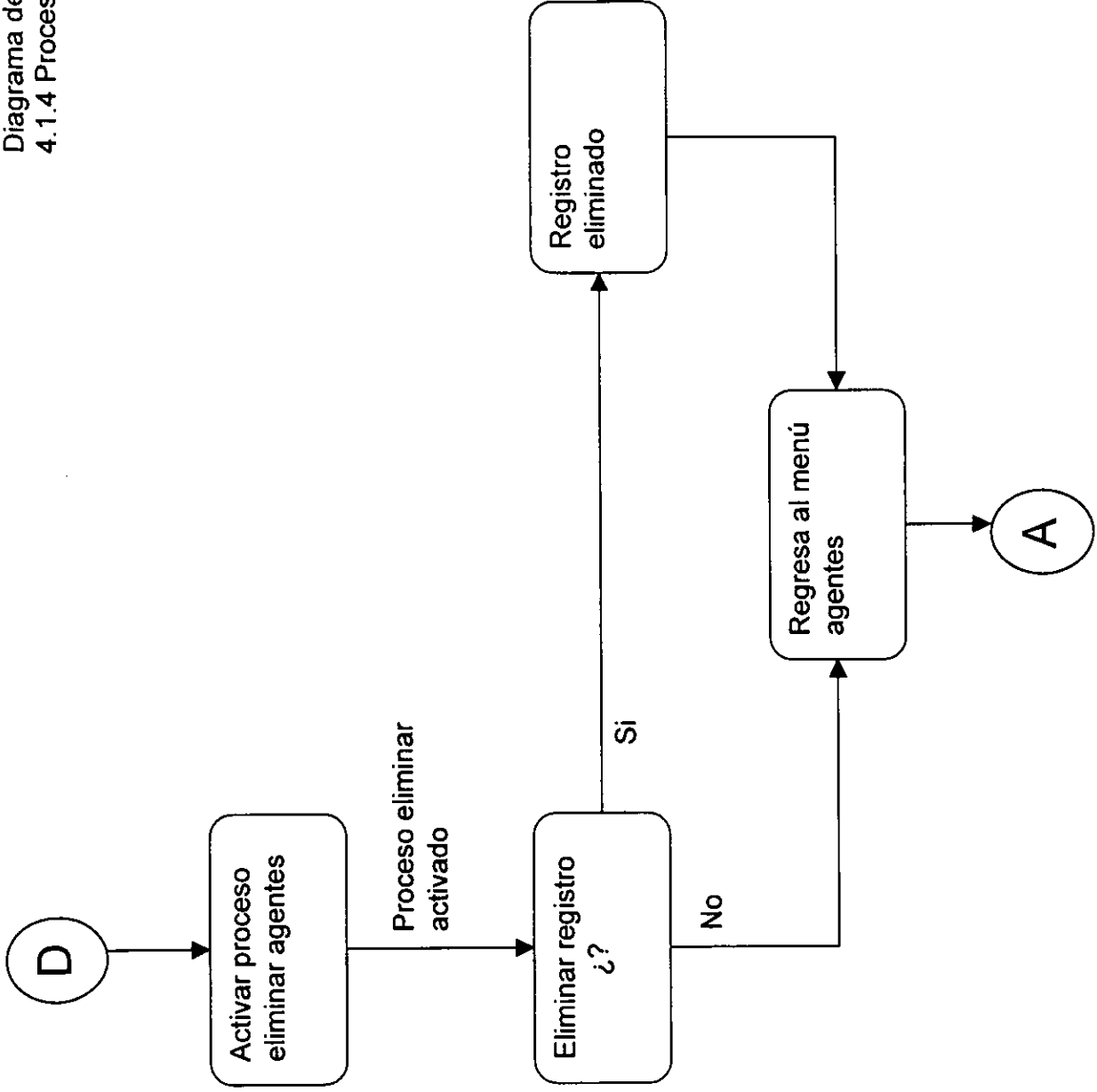


Diagrama de estado catálogos
4.1.5. Proceso Agentes Imprimir

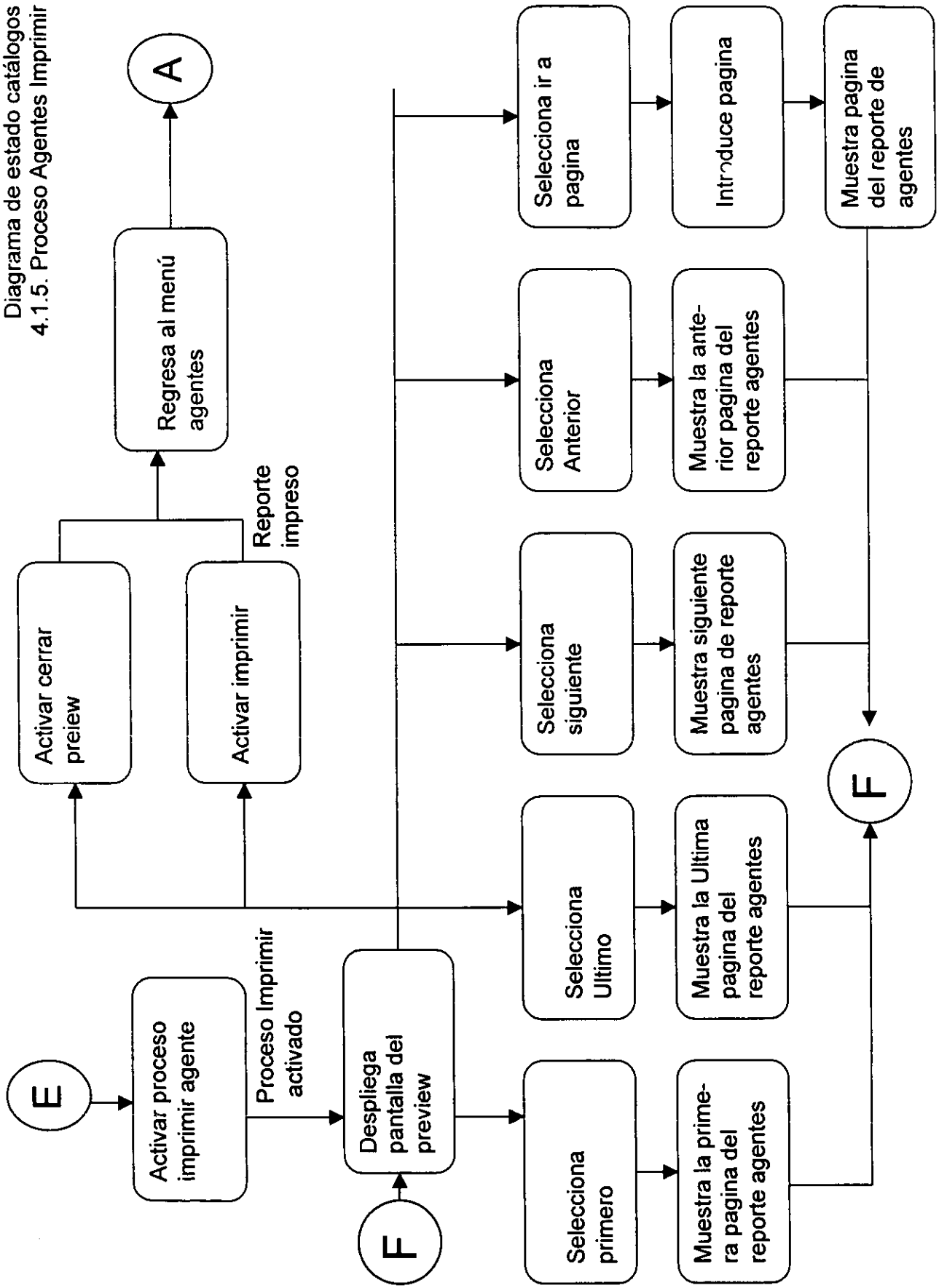


Diagrama de estado Catálogos
4.1.6 Proceso buscar agentes

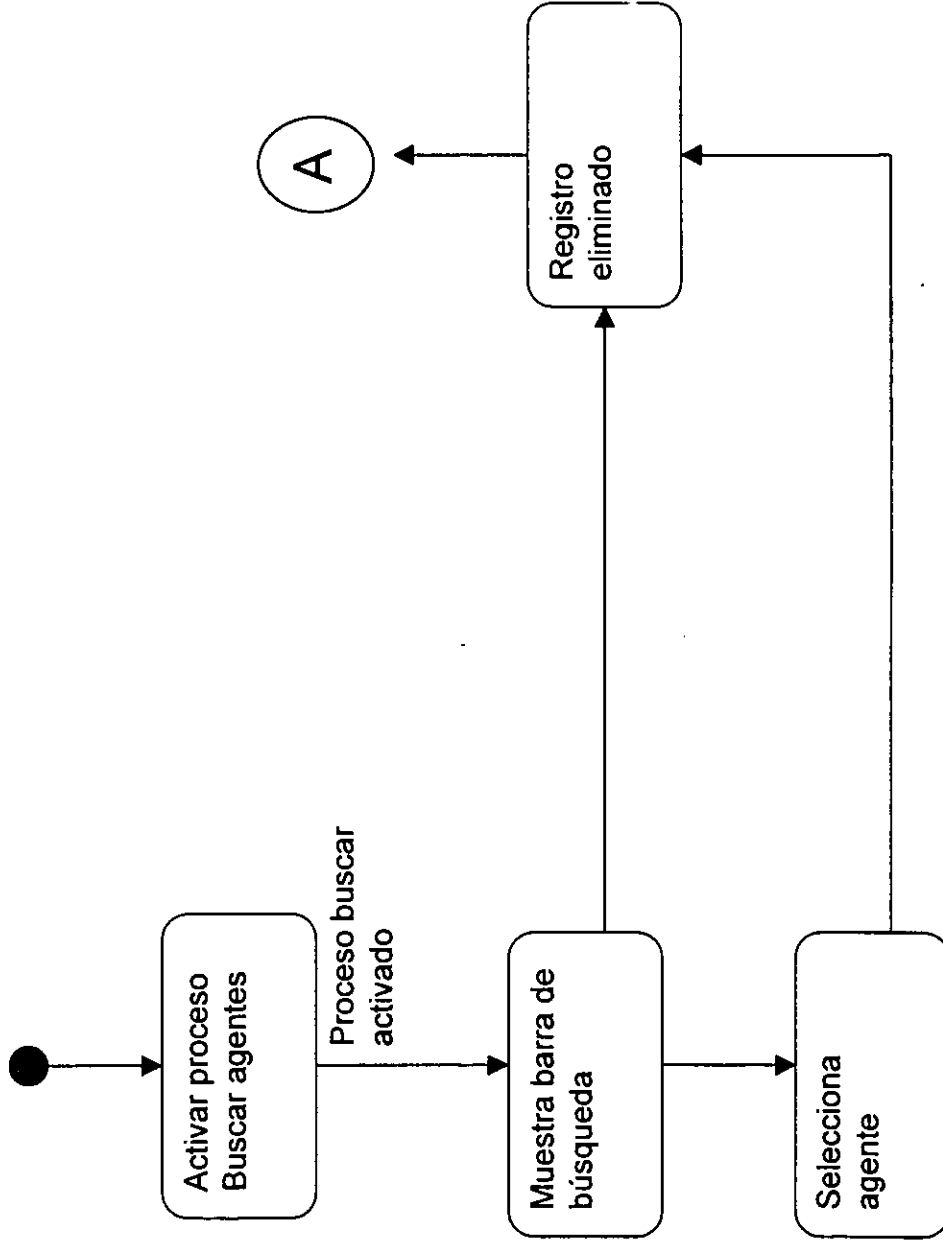


Diagrama de estado Catálogos
4.2 Proceso Estaciones de cobro

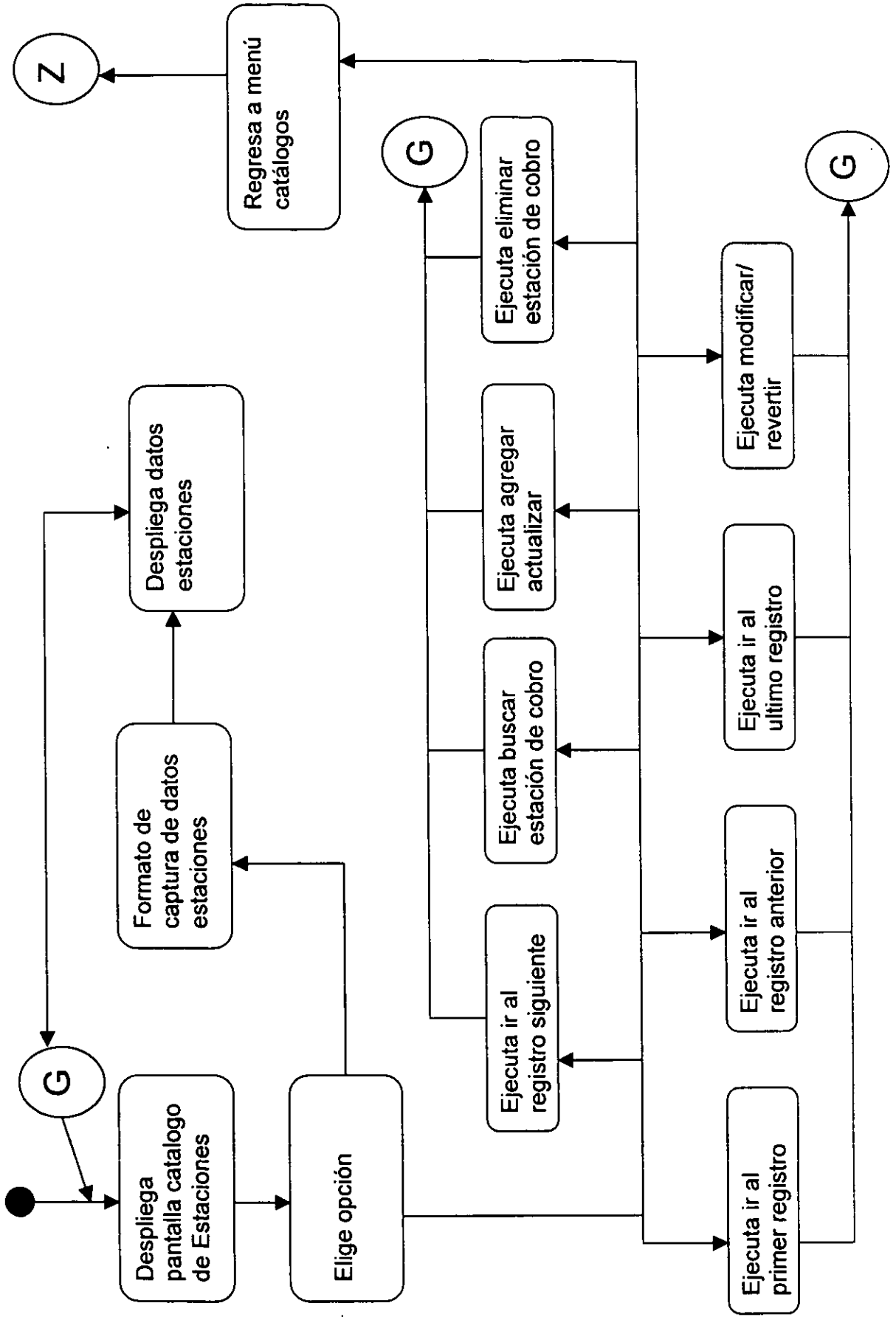


Diagrama de estado
4.2.1 Proceso Estaciones agregar

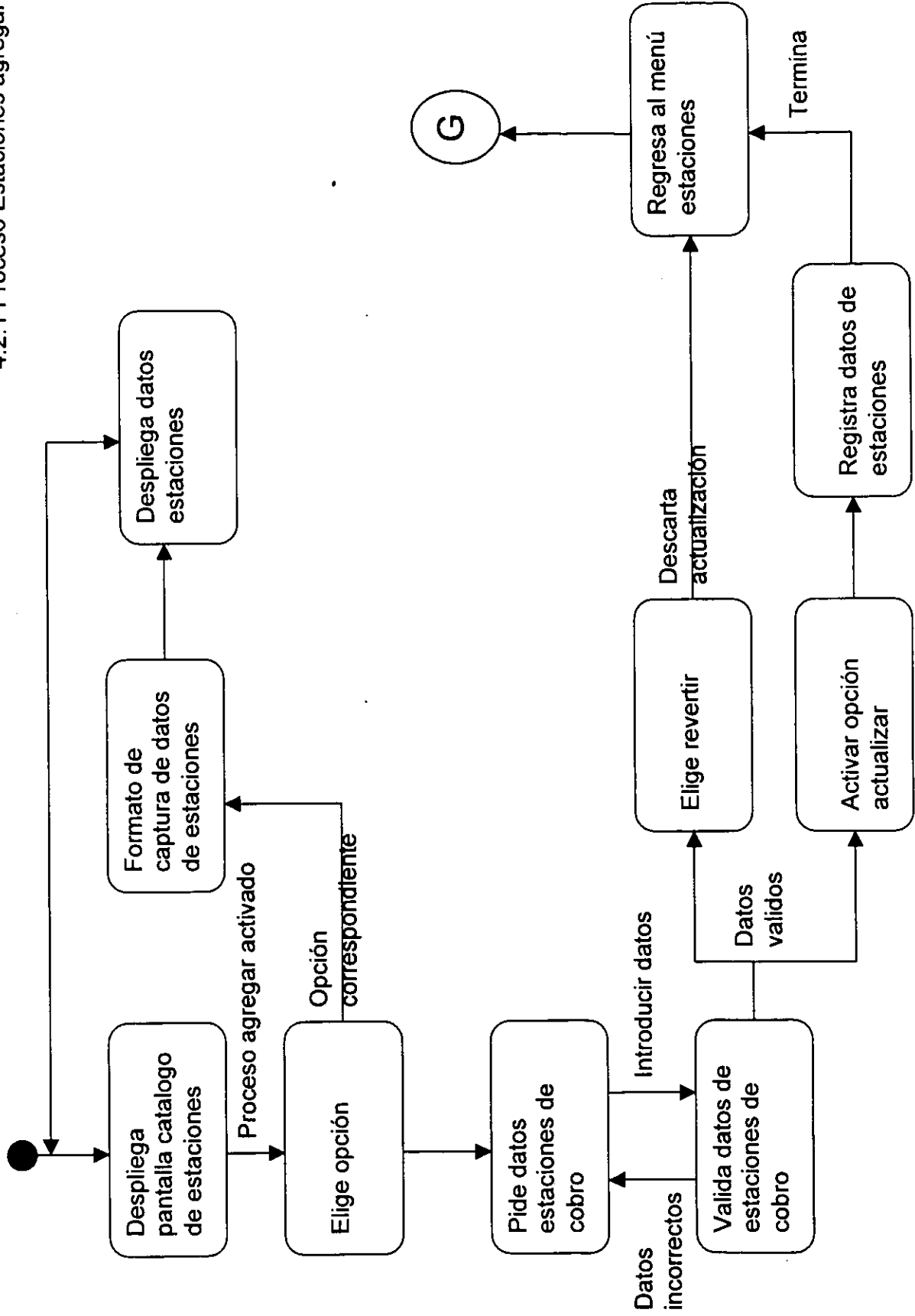


Diagrama de Estado Catálogos
4.2.2 Proceso estaciones seleccionar

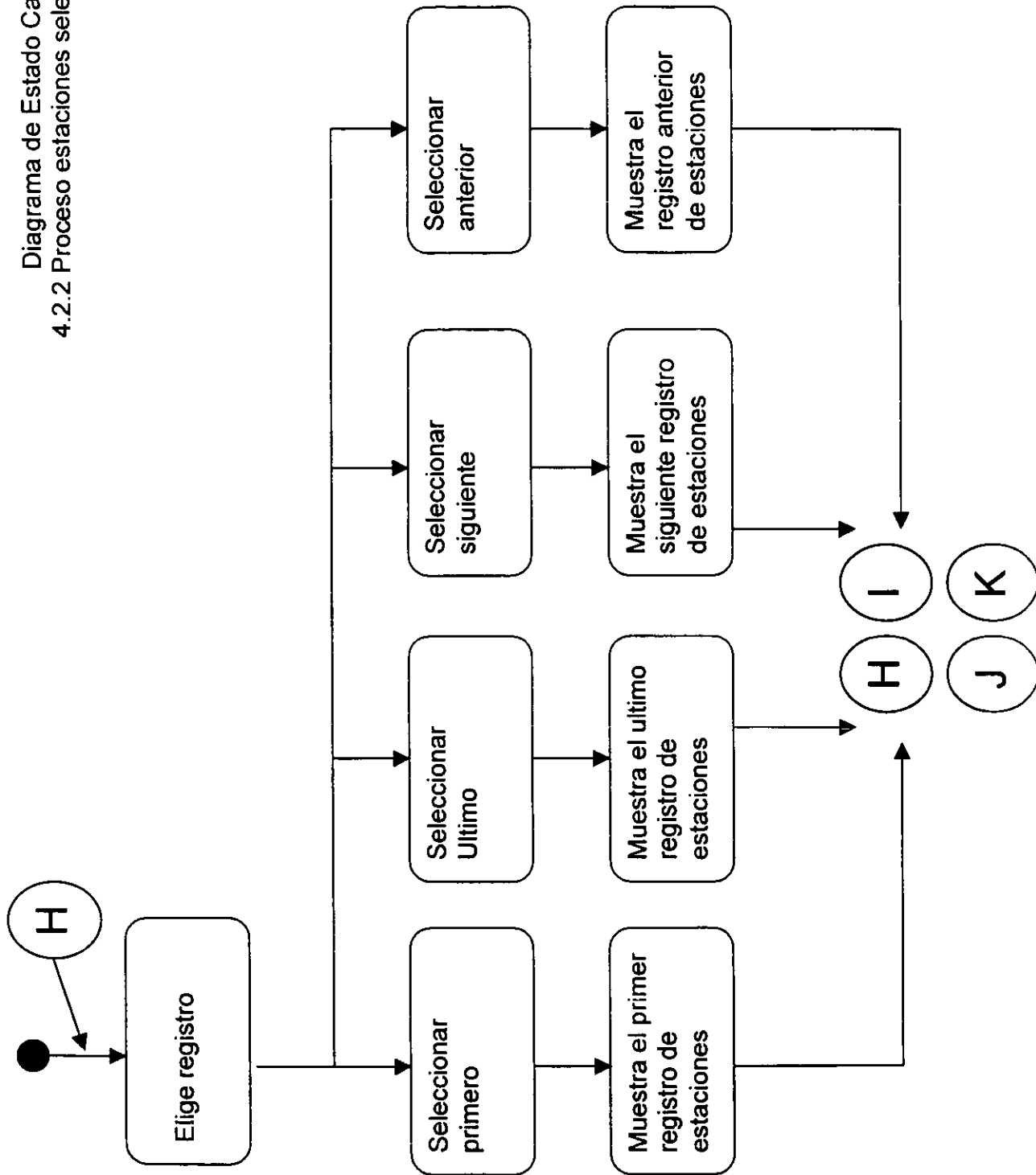


Diagrama de estado catálogos
4.2.3. Proceso Estaciones modificar

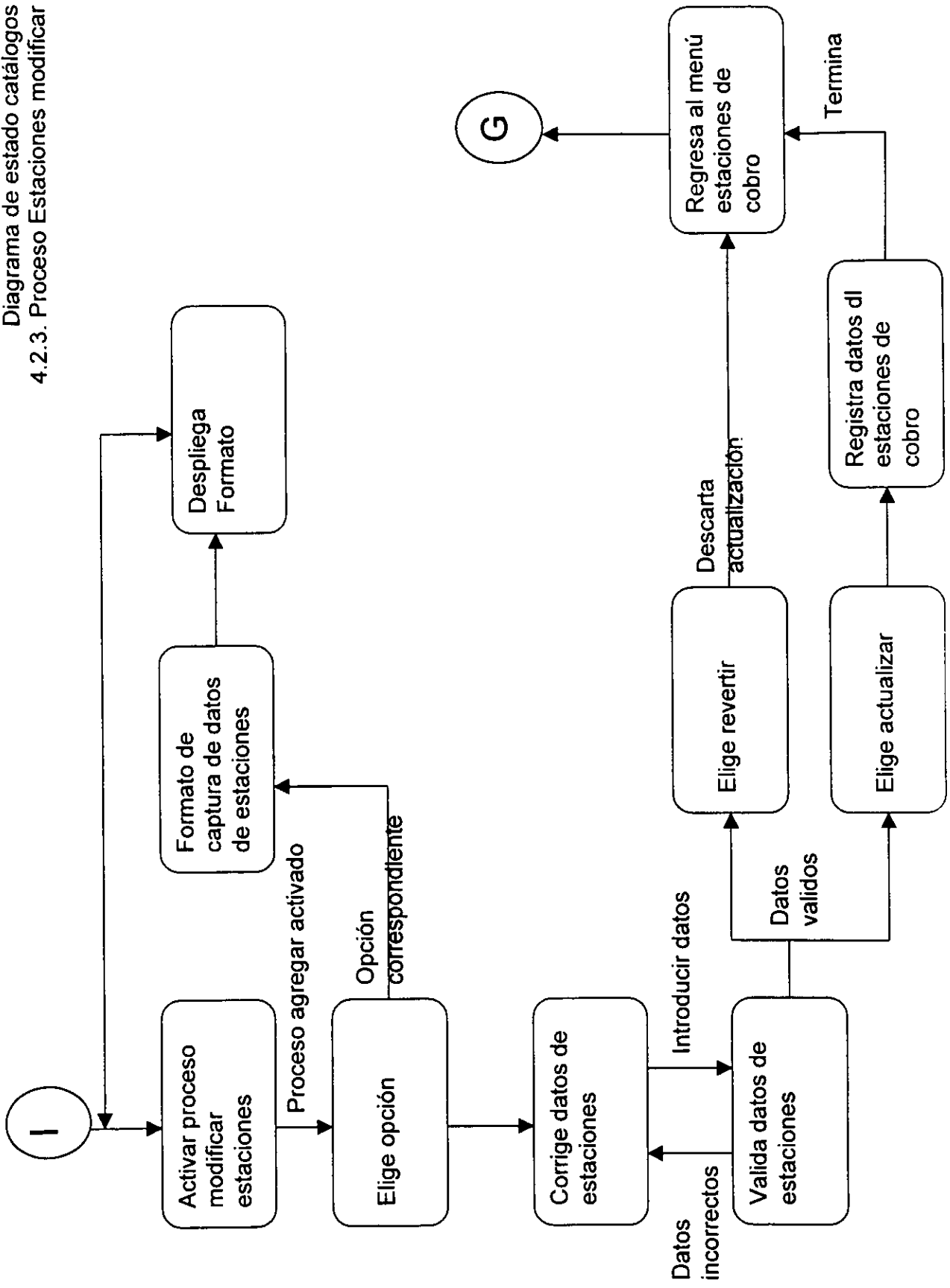


Diagrama de estado Catálogos
4.2.4 Proceso eliminar estaciones

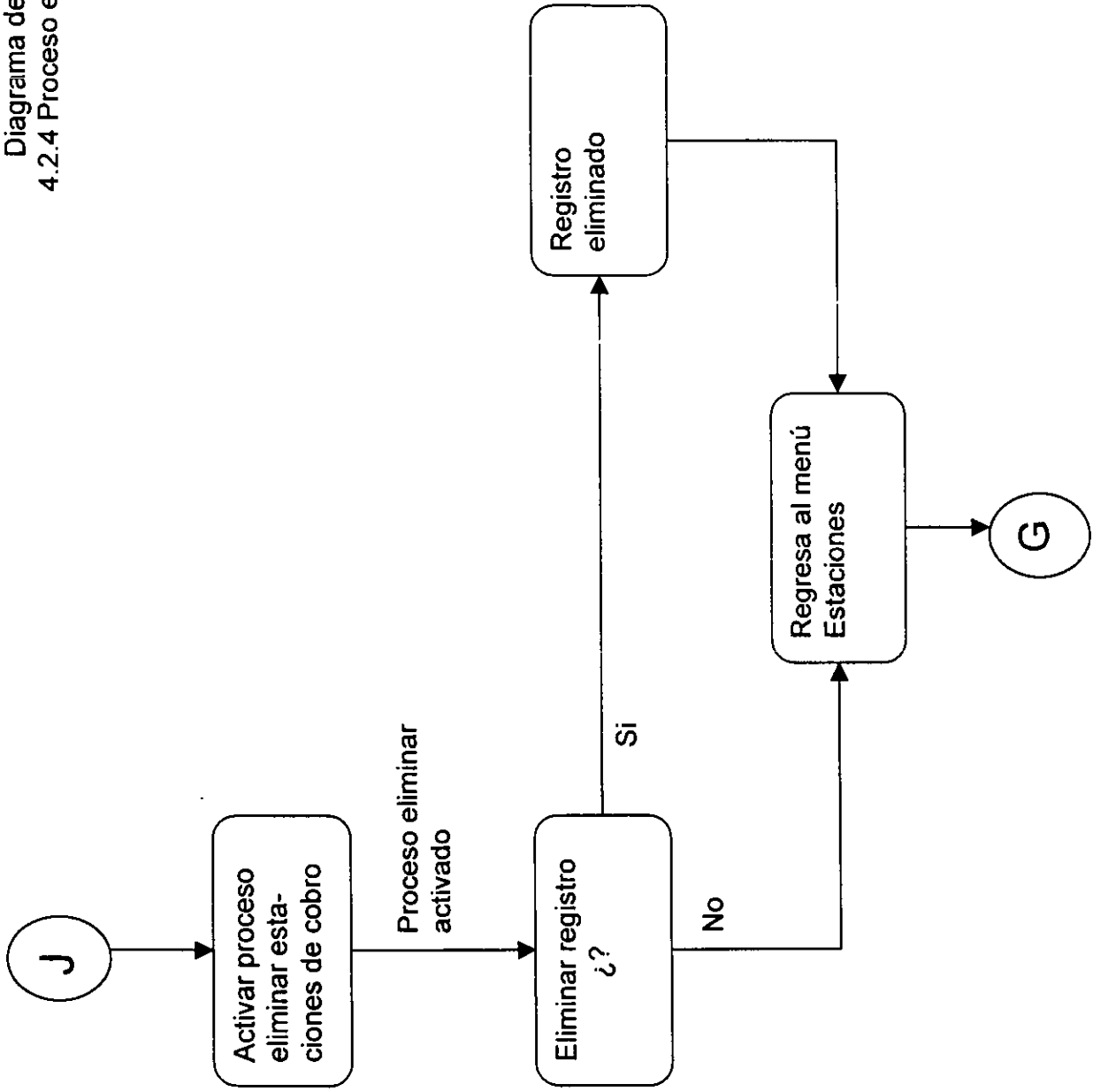


Diagrama de estado catálogos
4.2.5. Proceso Estaciones Imprimir

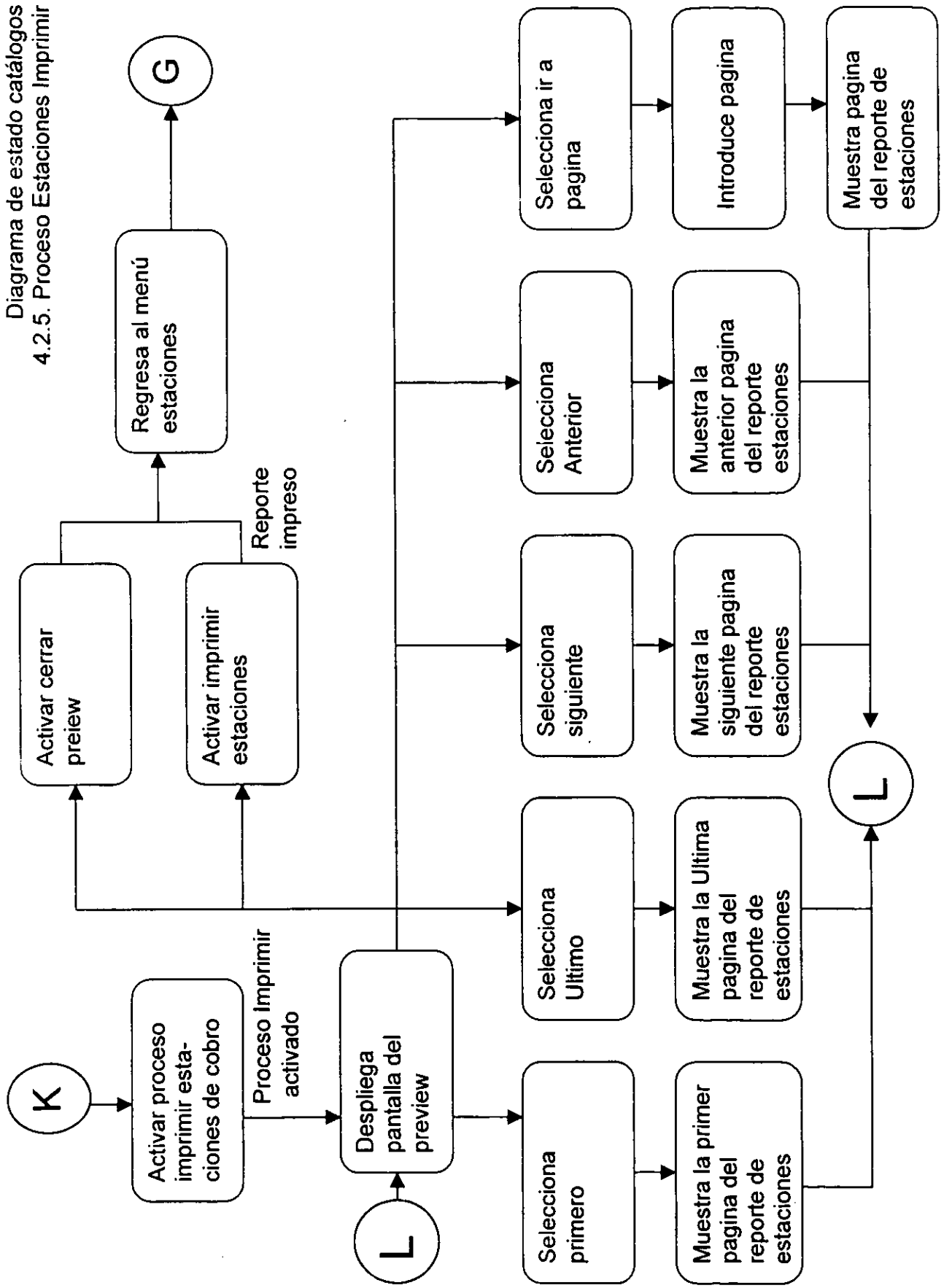


Diagrama de estado Catálogos
4.2.6 Proceso buscar Estación

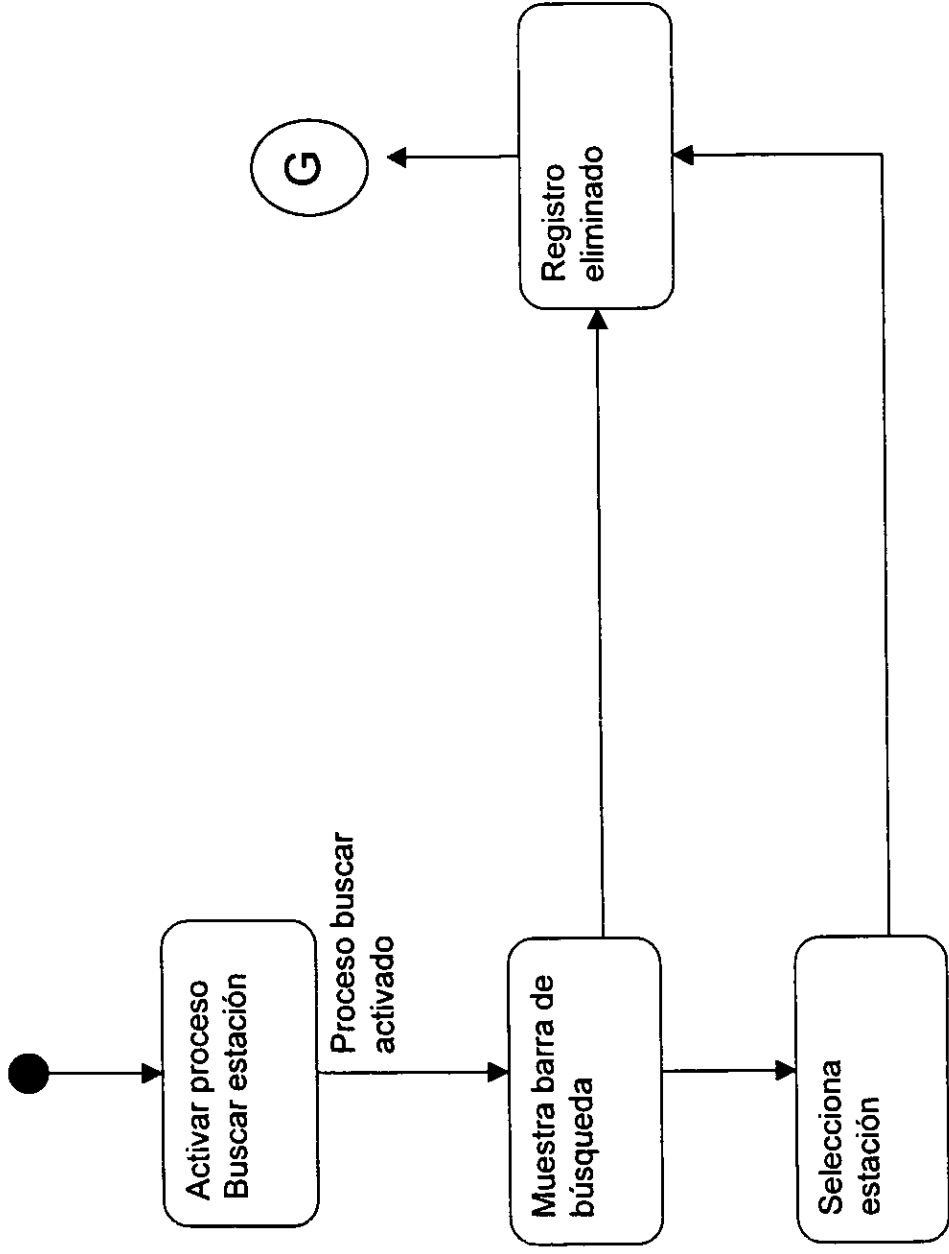


Diagrama de estado Catálogos
4.3 Proceso Usuarios

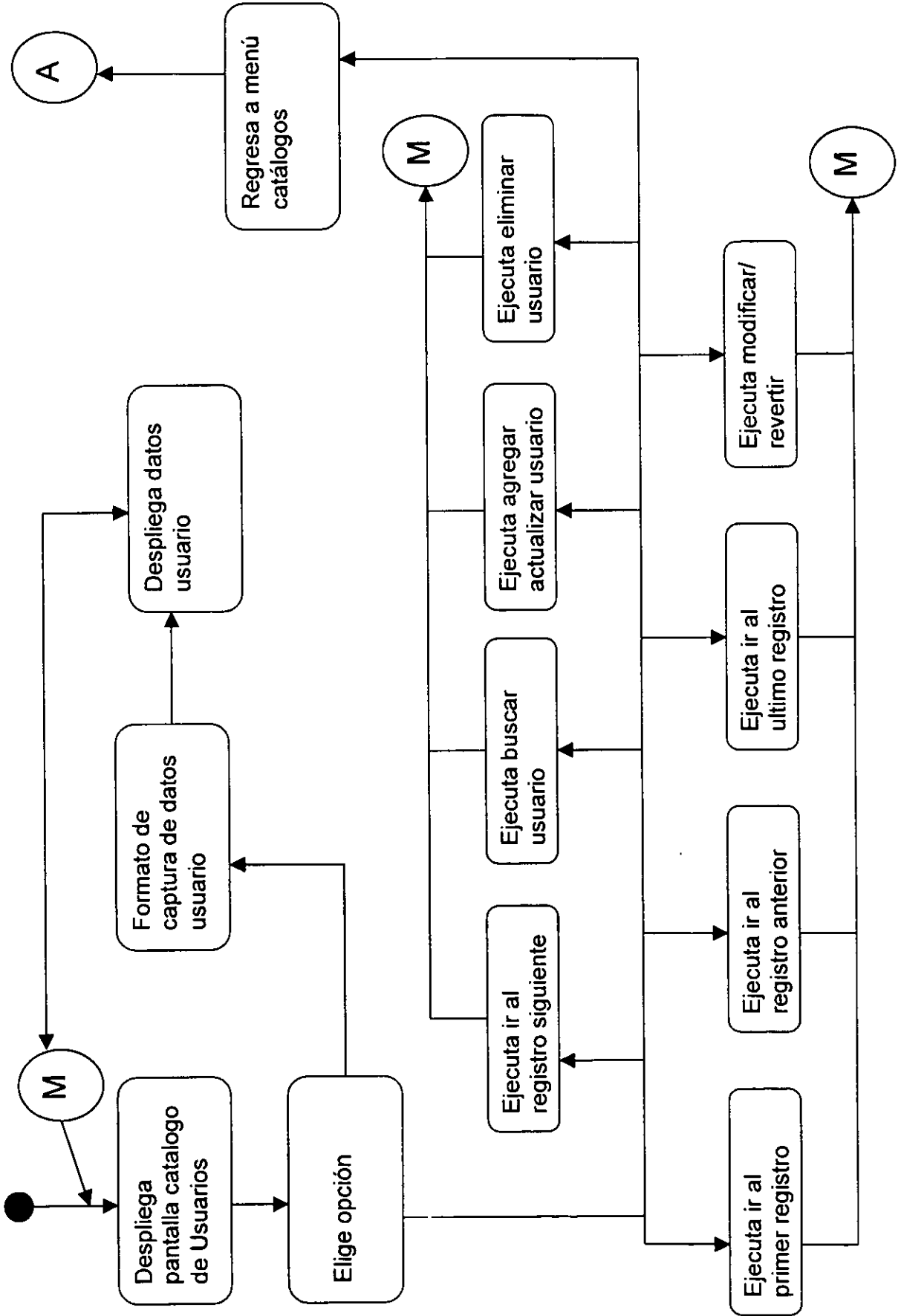


Diagrama de estado
4.3.1 Proceso Usuarios agregar

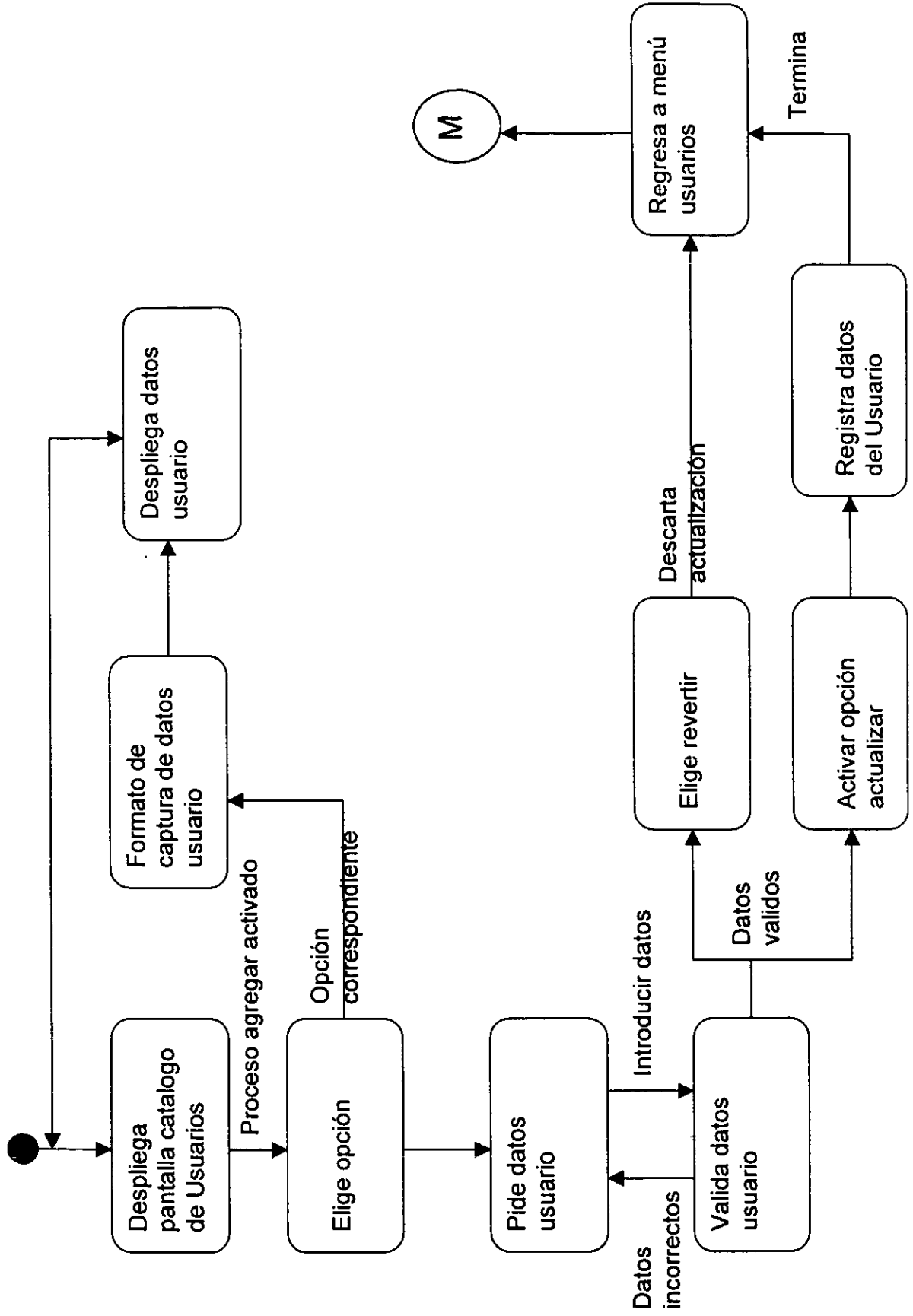


Diagrama de Estado Catálogos
4.3.2 Proceso Usuarios seleccionar

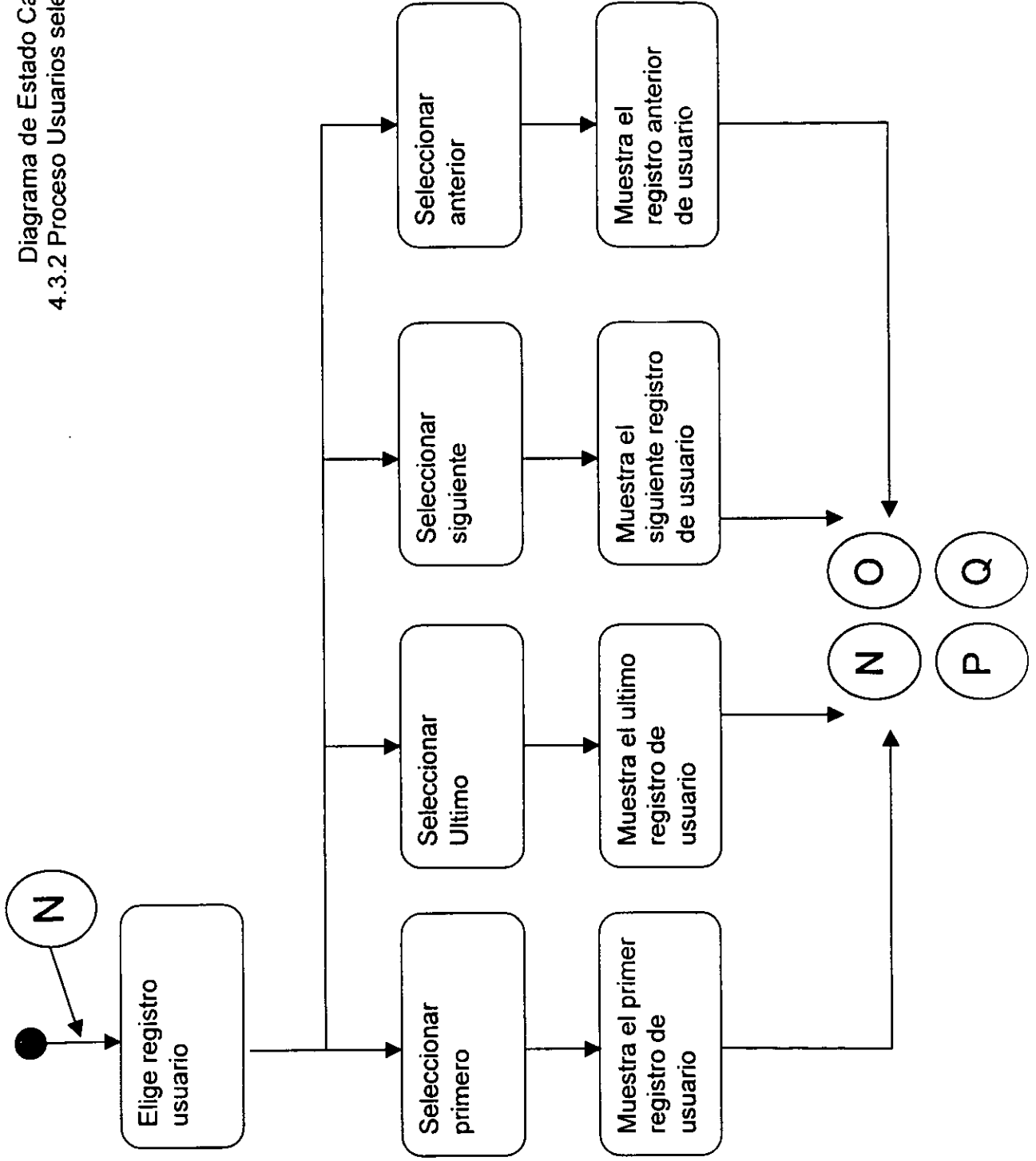


Diagrama de estado catálogos
 4.3.3. Proceso Usuario modificar

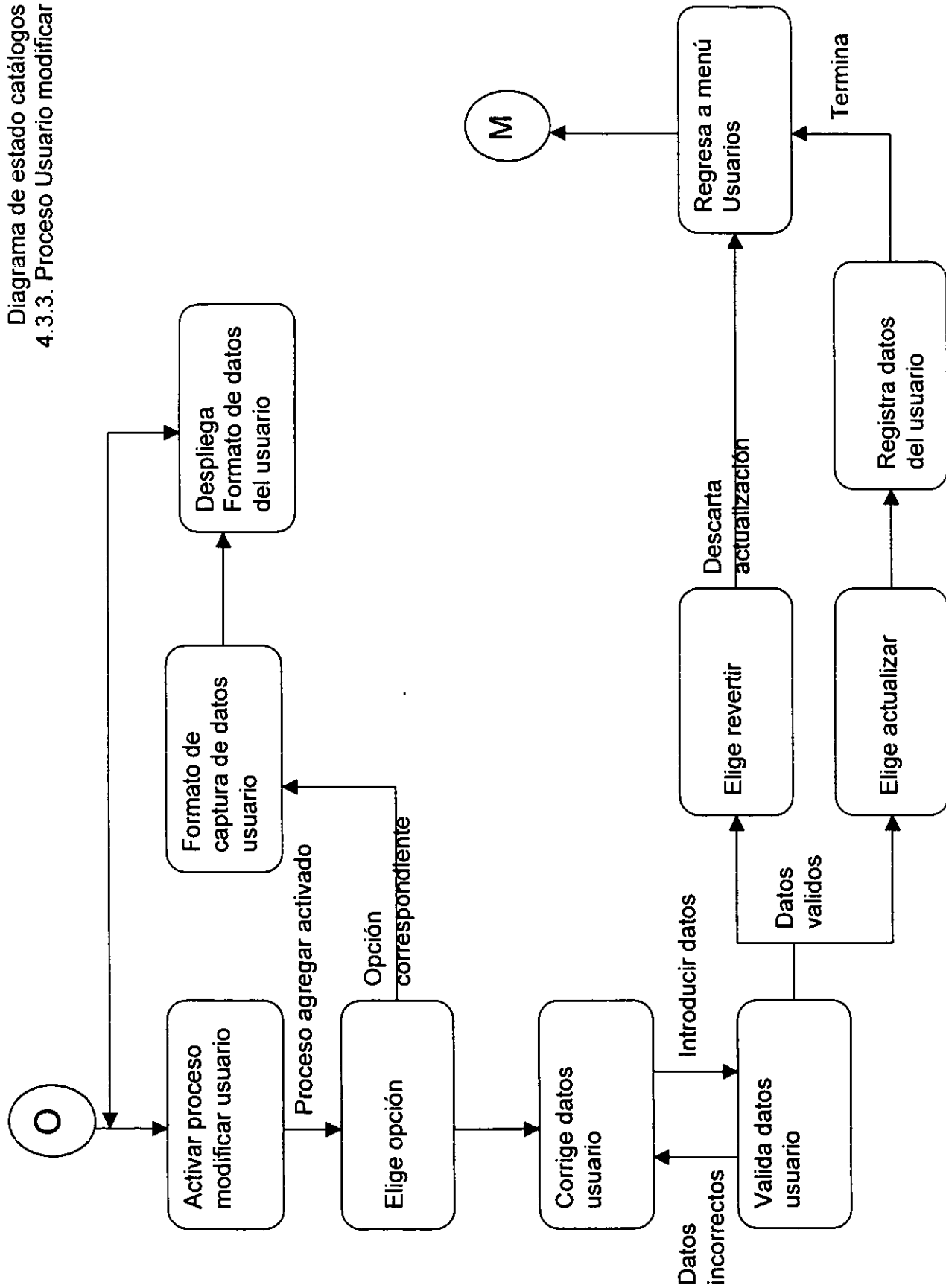


Diagrama de estado Catálogos
4.3.4 Proceso eliminar agentes

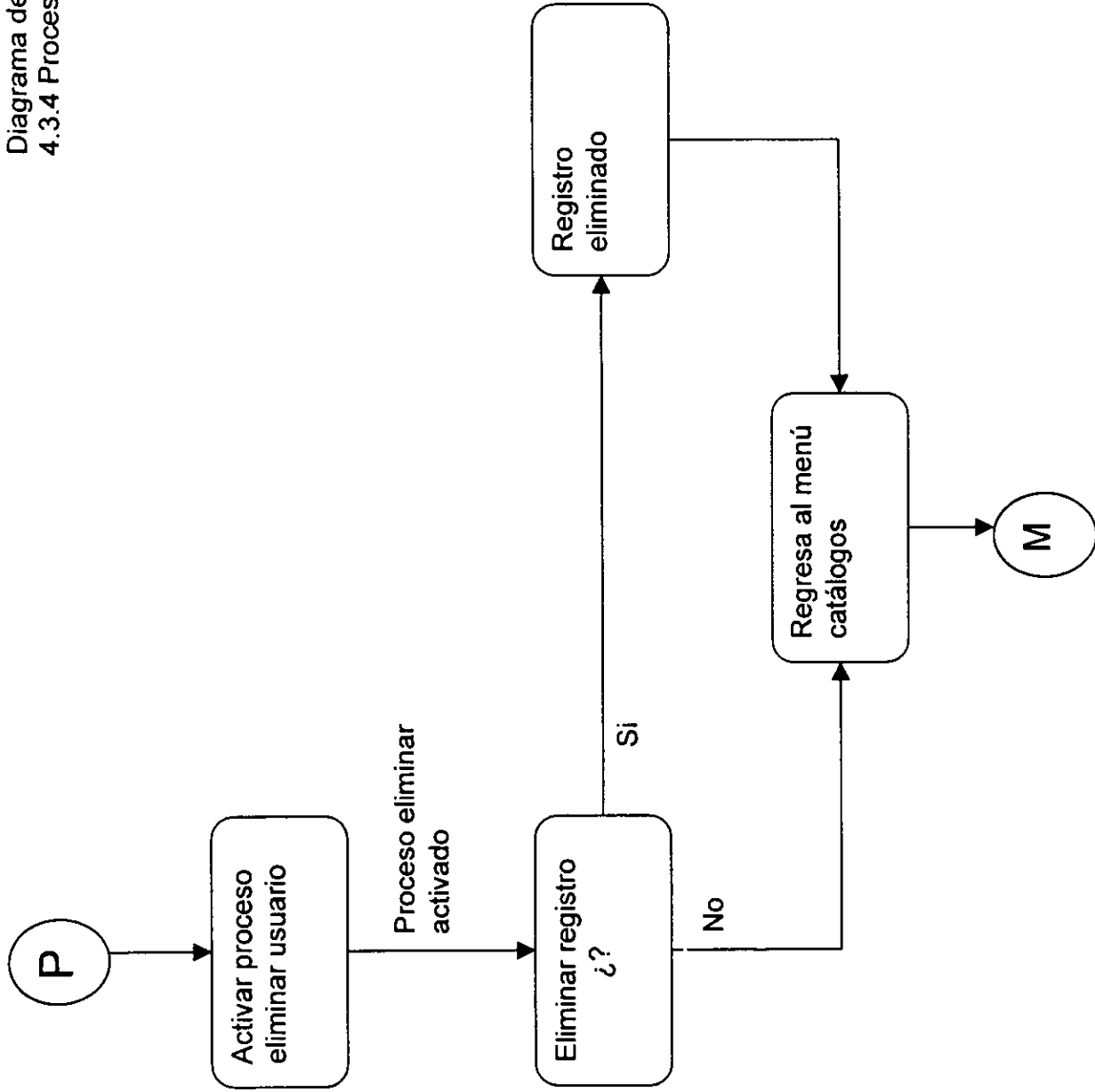


Diagrama de estado catálogos
4.3.5. Proceso Usuarios Imprimir

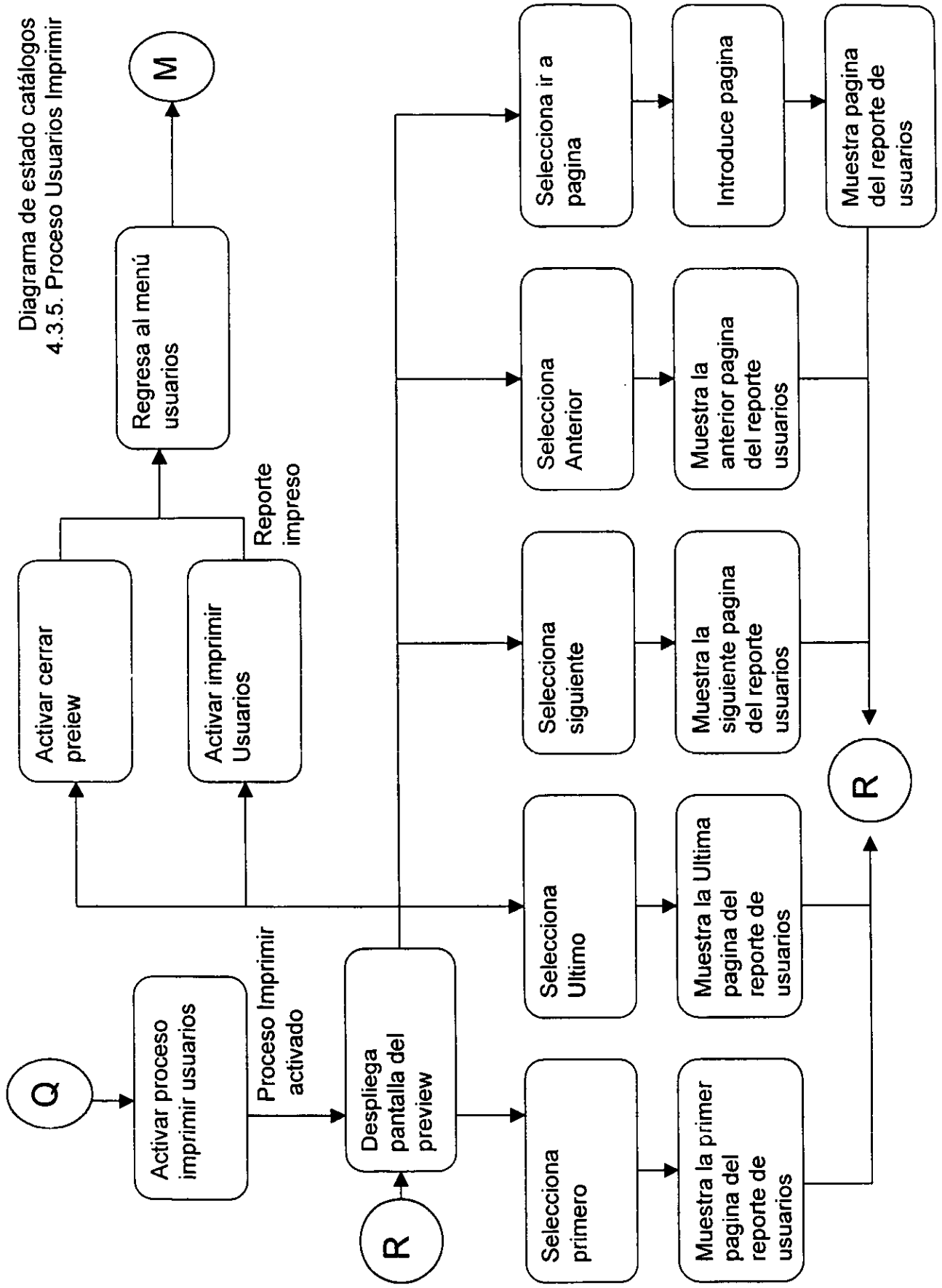


Diagrama de estado Catálogos
4.3.6 Proceso buscar Usuario

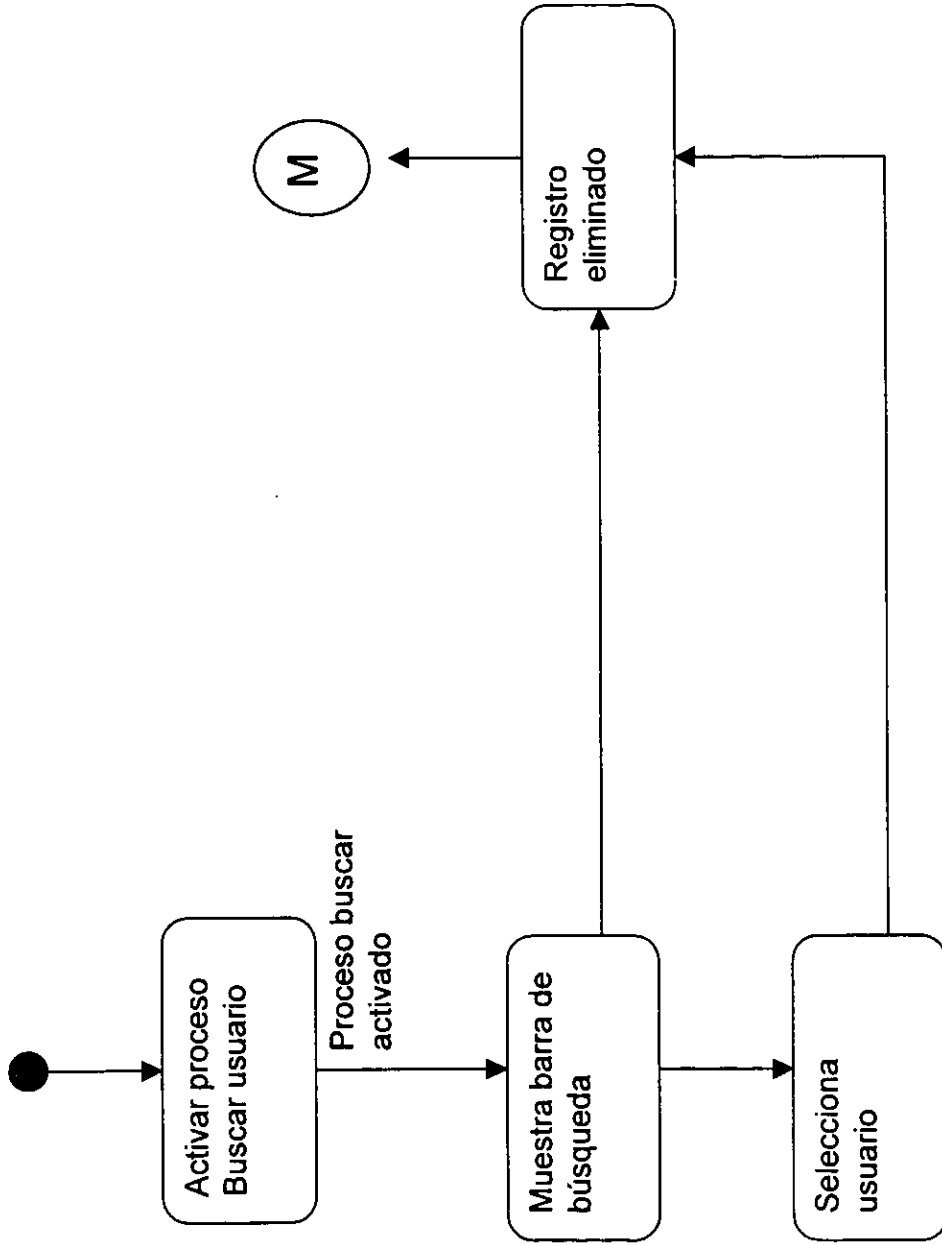


Diagrama de estado Catálogos
4.4 Proceso Grupos

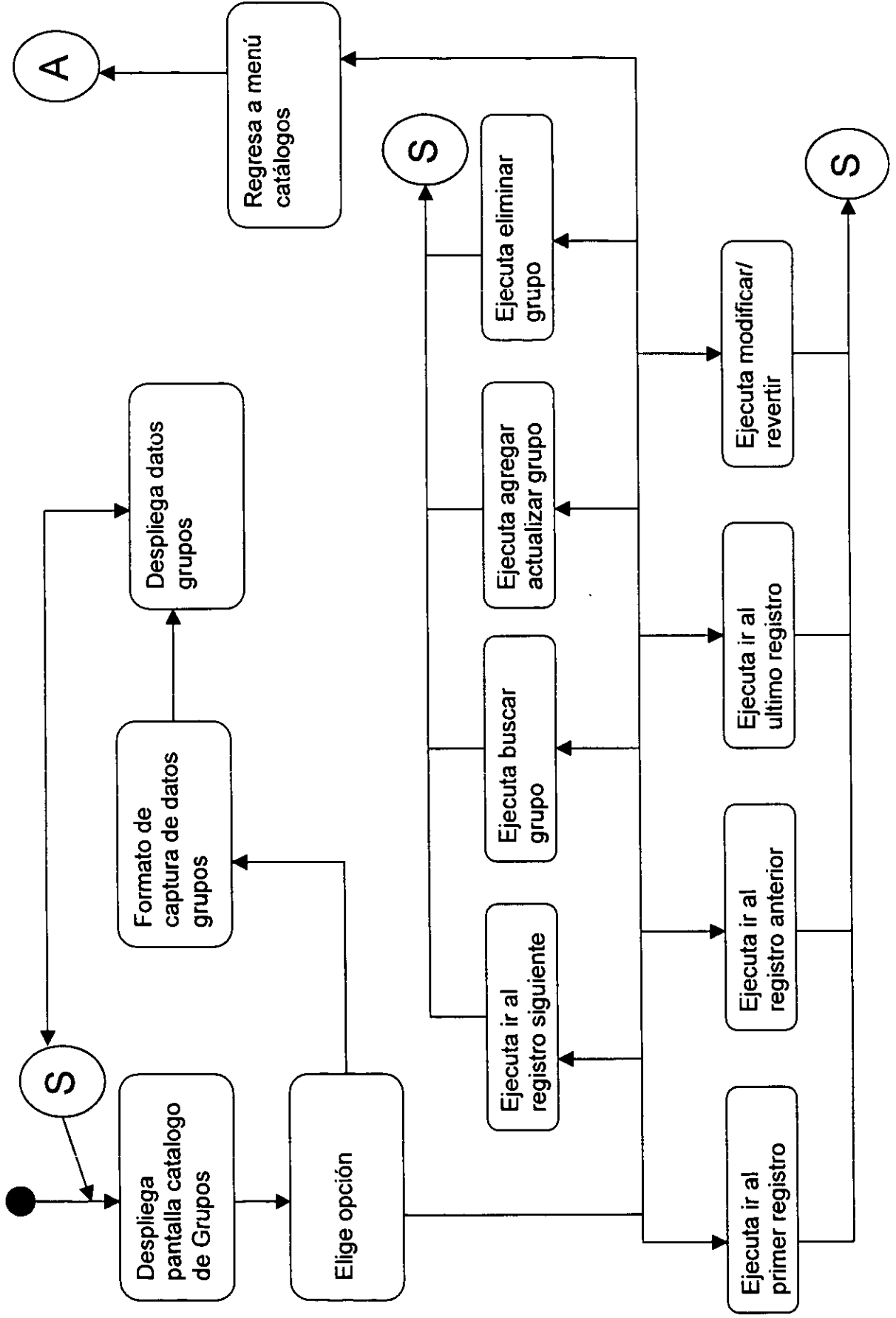


Diagrama de estado
4.4.1 Proceso Grupos agregar

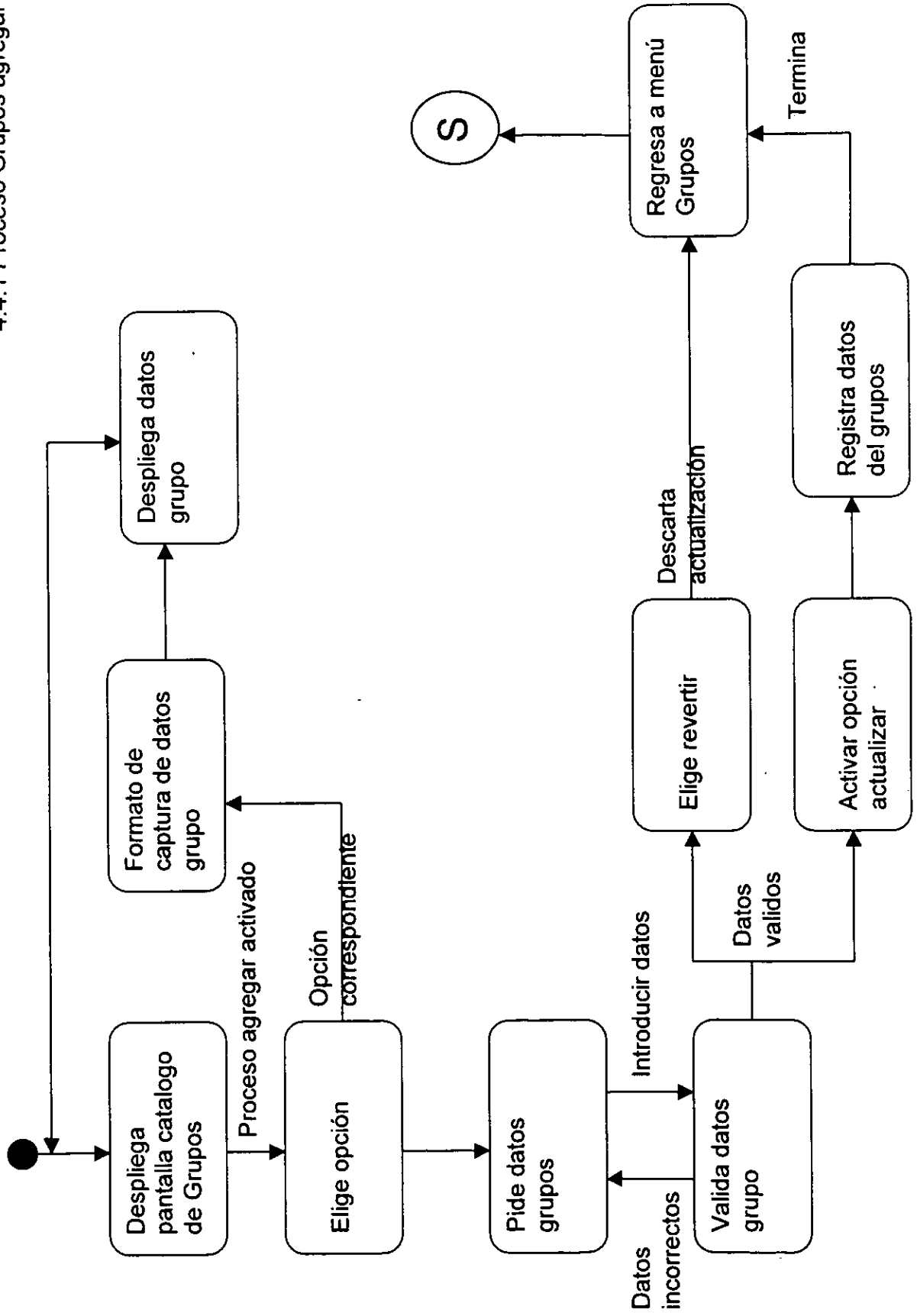


Diagrama de Estado Catálogos
4.4.2 Proceso grupos seleccionar

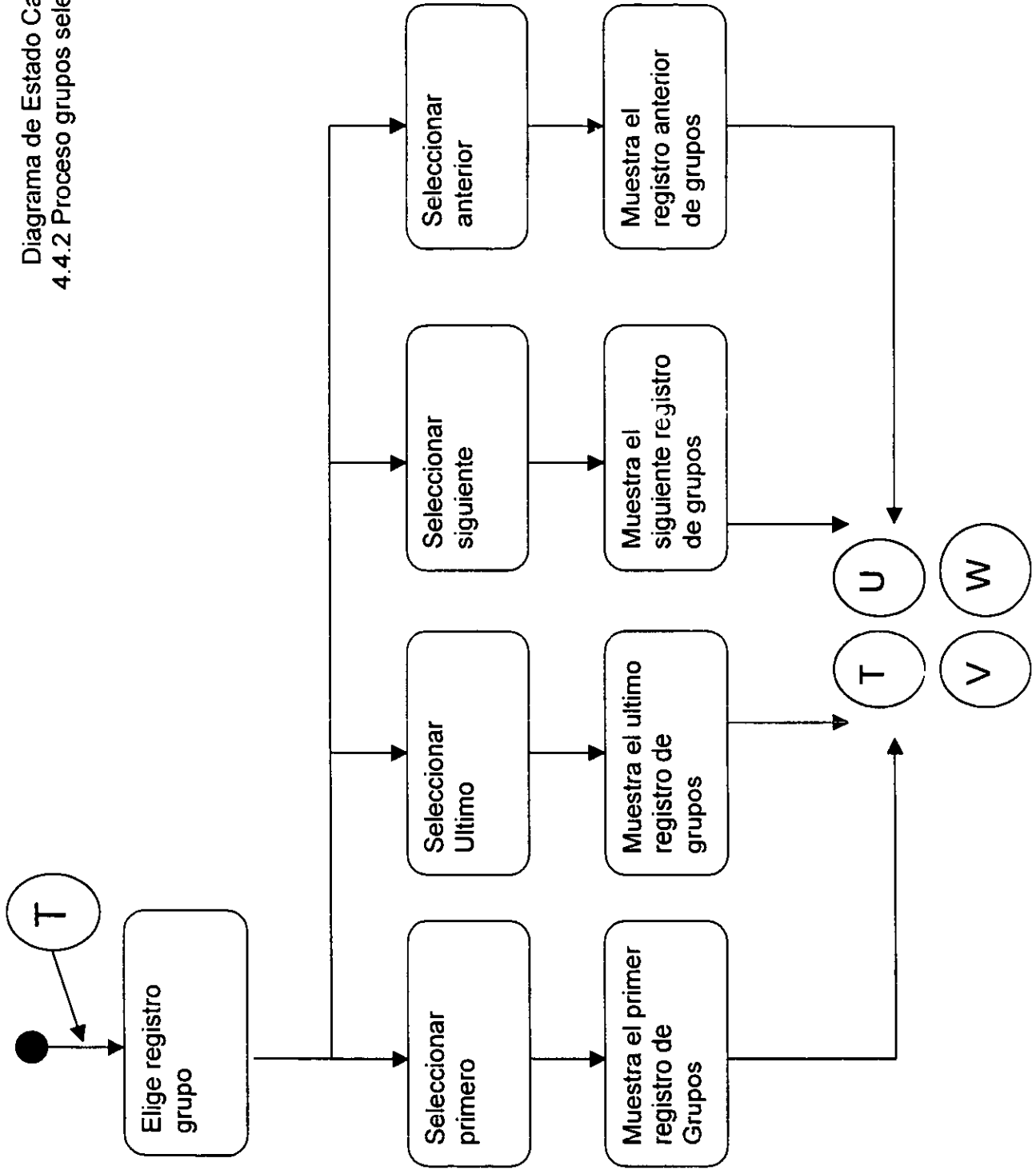


Diagrama de estado catálogos
 4.4.3. Proceso Grupos modificar

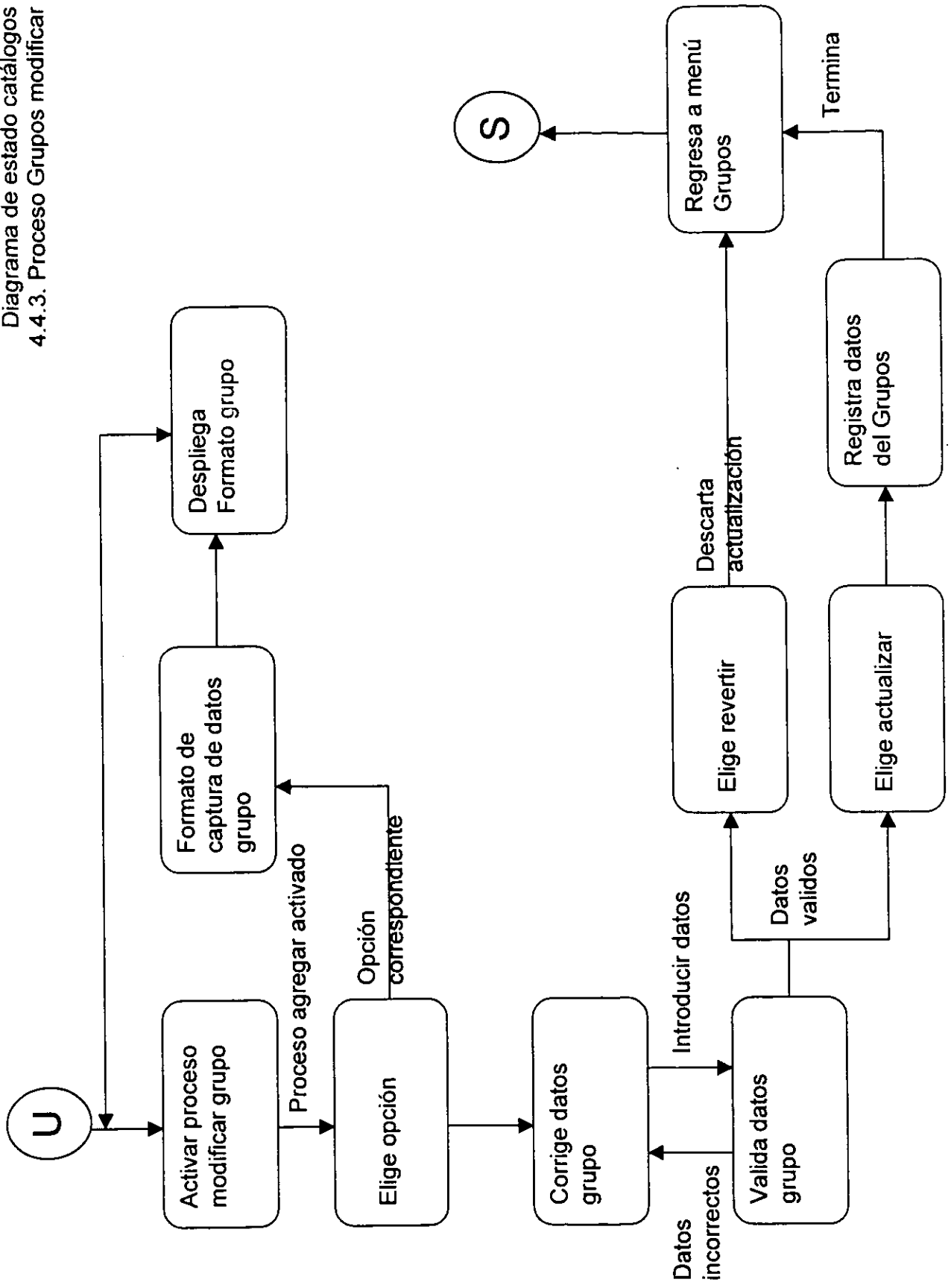


Diagrama de estado Catálogos
4.4.4 Proceso eliminar grupo

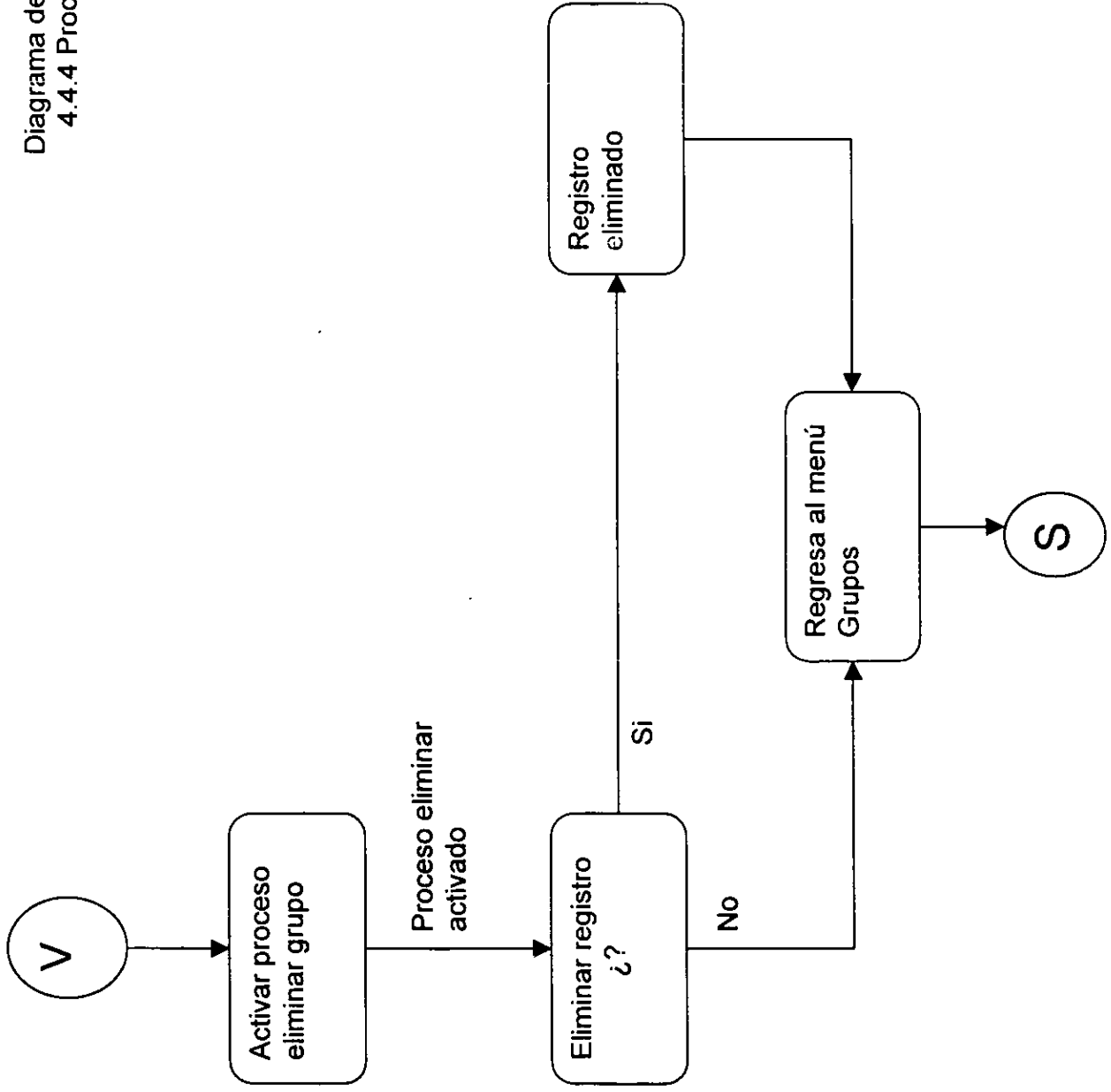


Diagrama de estado catálogos
4.4.5. Proceso grupos Imprimir

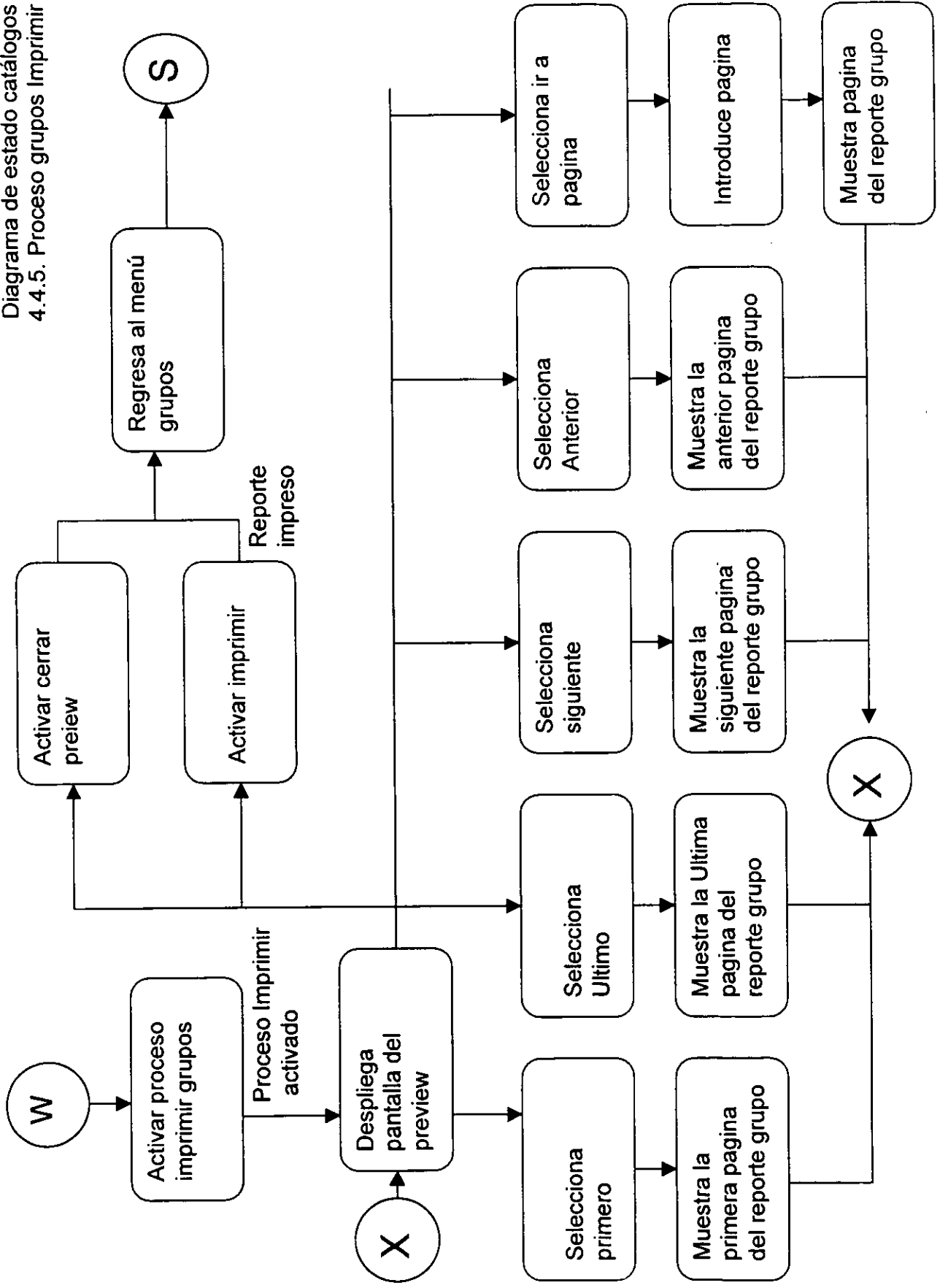


Diagrama de estado Catálogos
4.4.6 Proceso buscar grupos

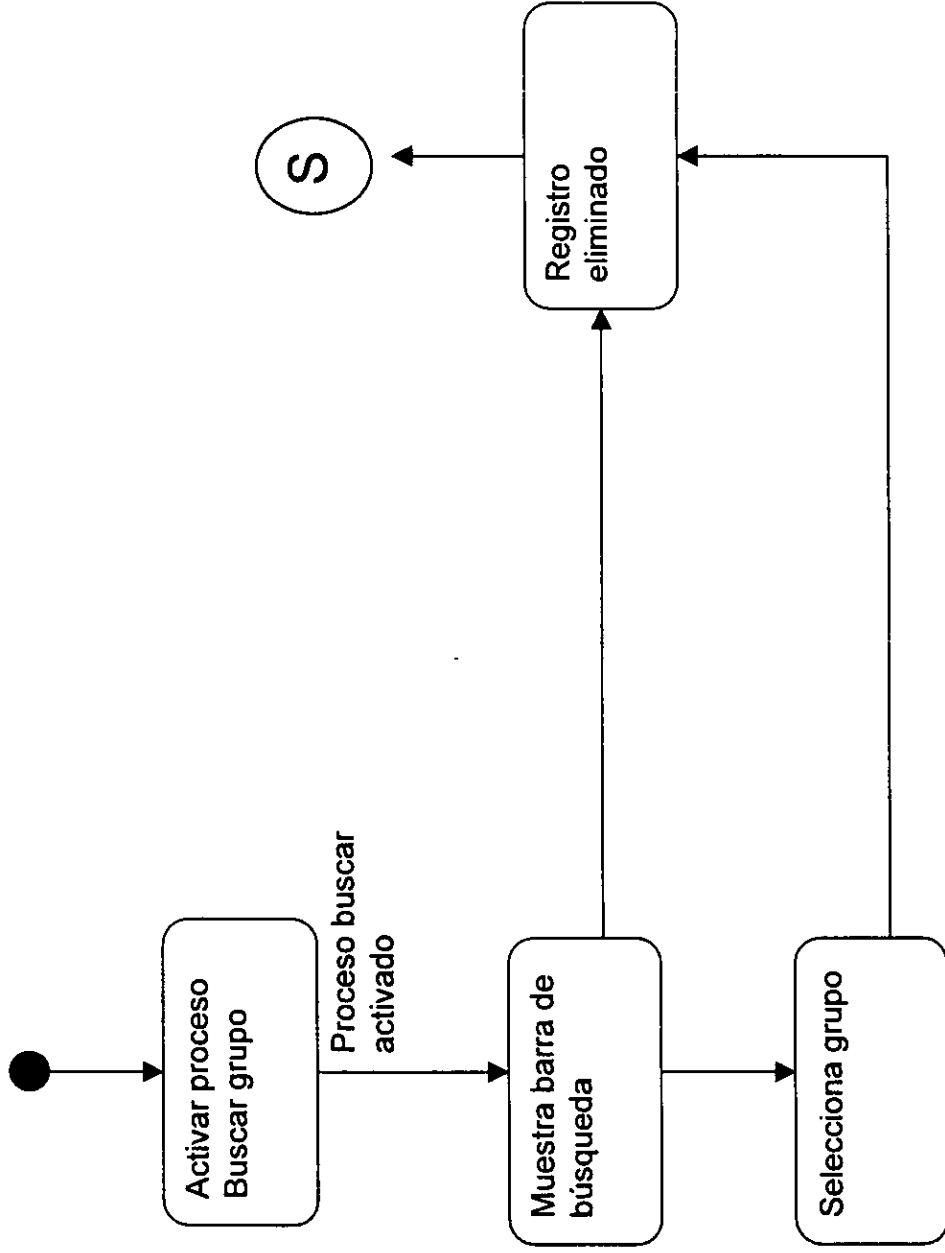


Diagrama de estado
1.5 Menú Generación de reportes

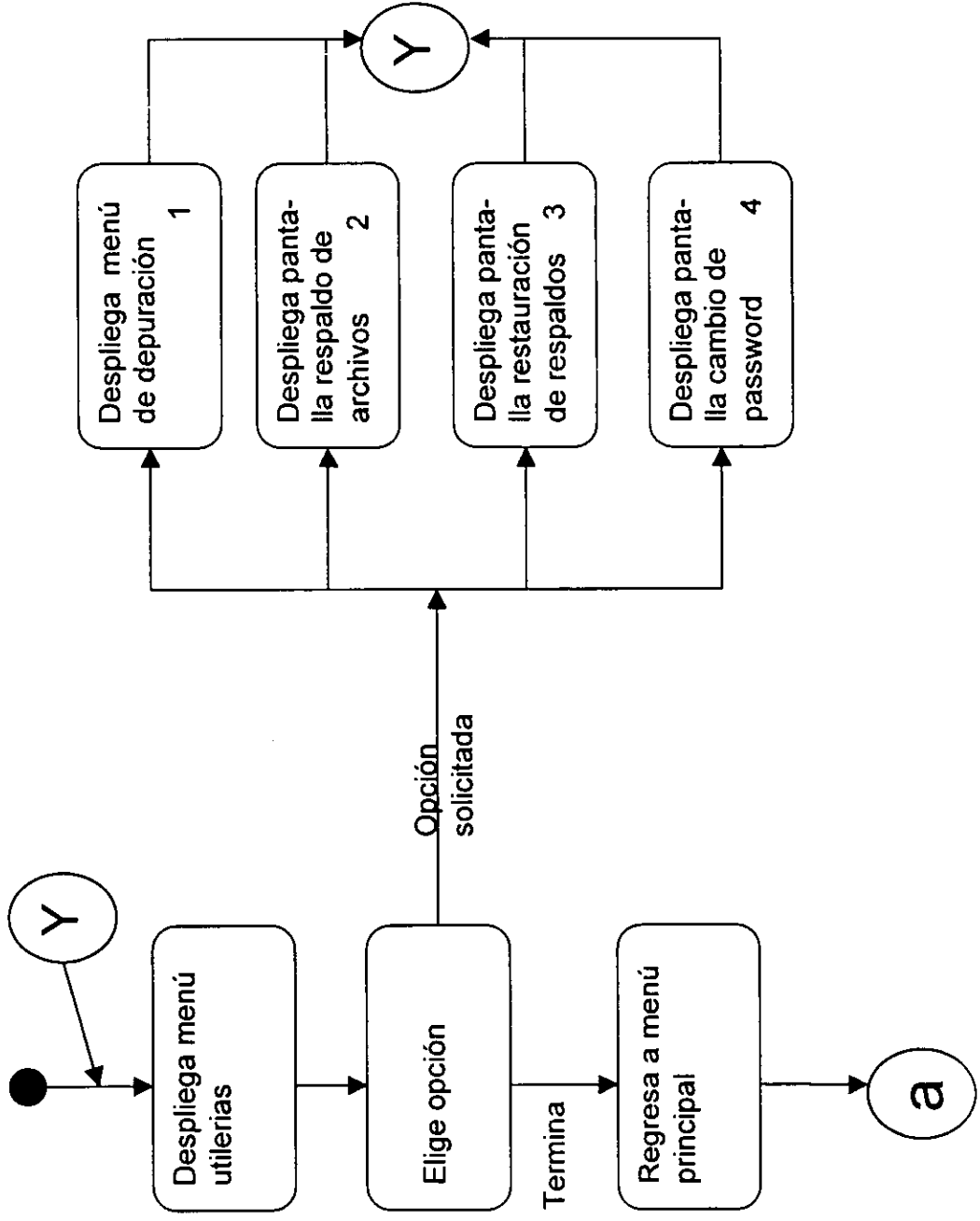


Diagrama de estado
1.5.1 Menú Control de pagos

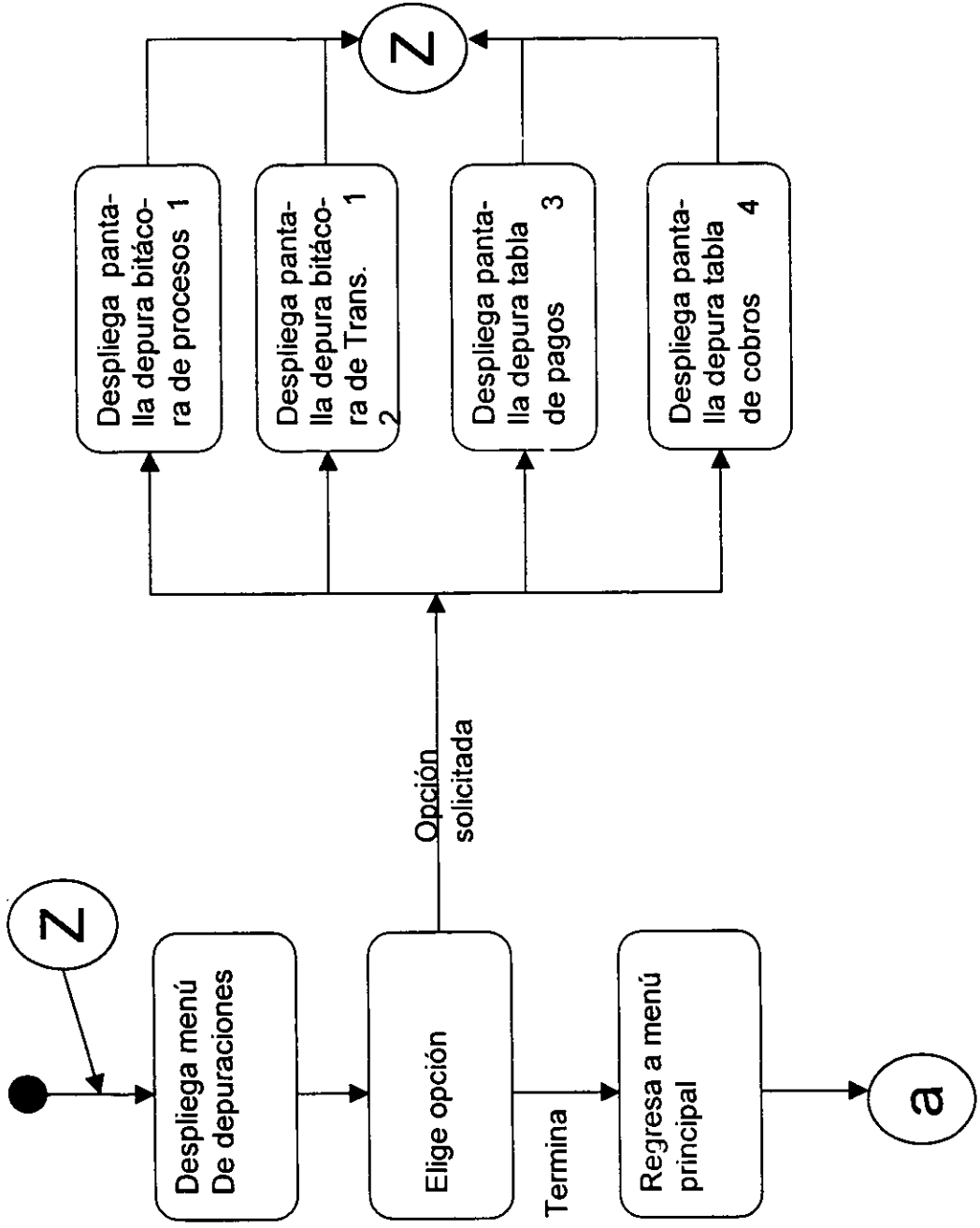


Diagrama de estado

1.5.1.1 Proceso depuración de bitácora de procesos.

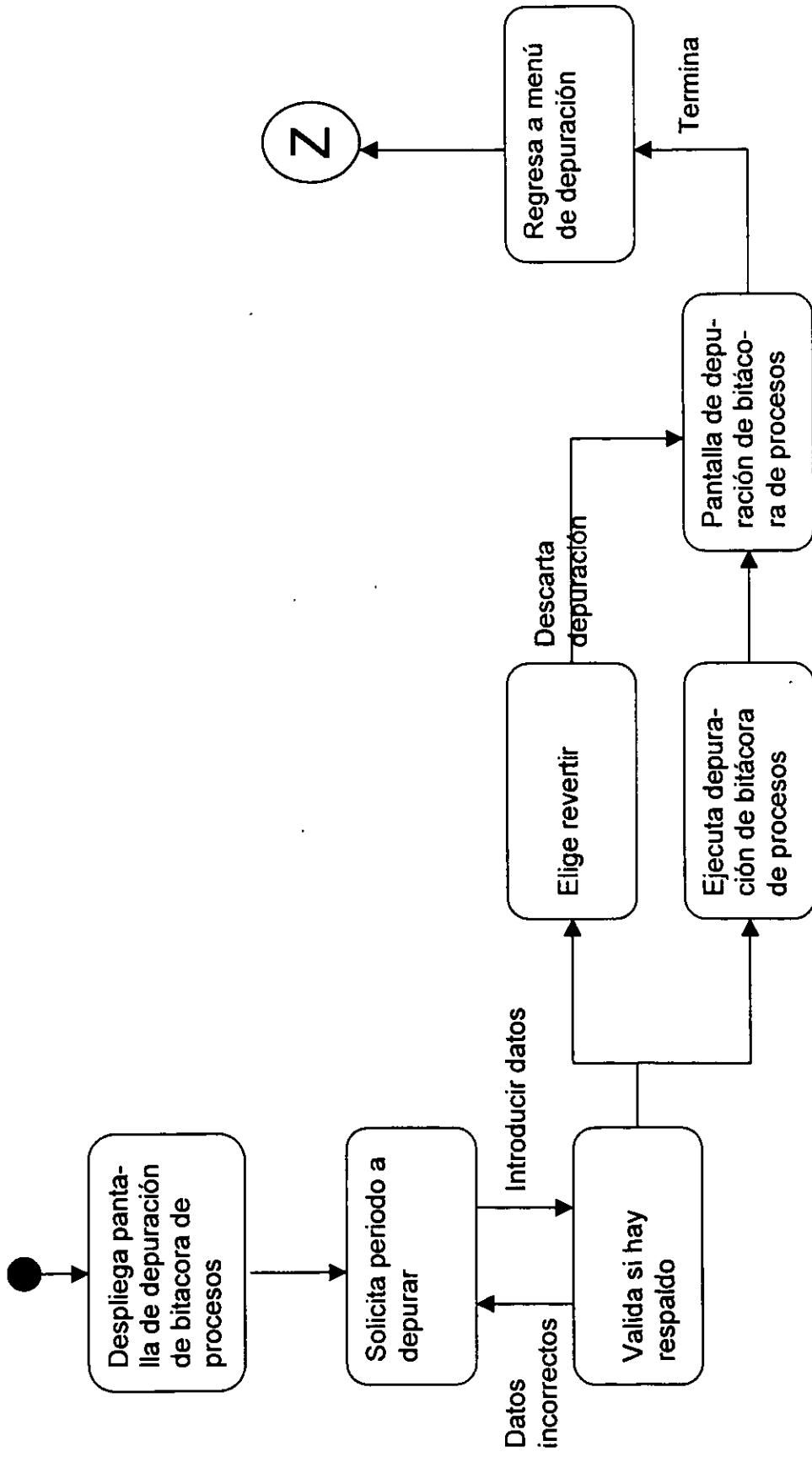


Diagrama de estado

1.5.1.2 Proceso de depuración de bitácora de transmisión

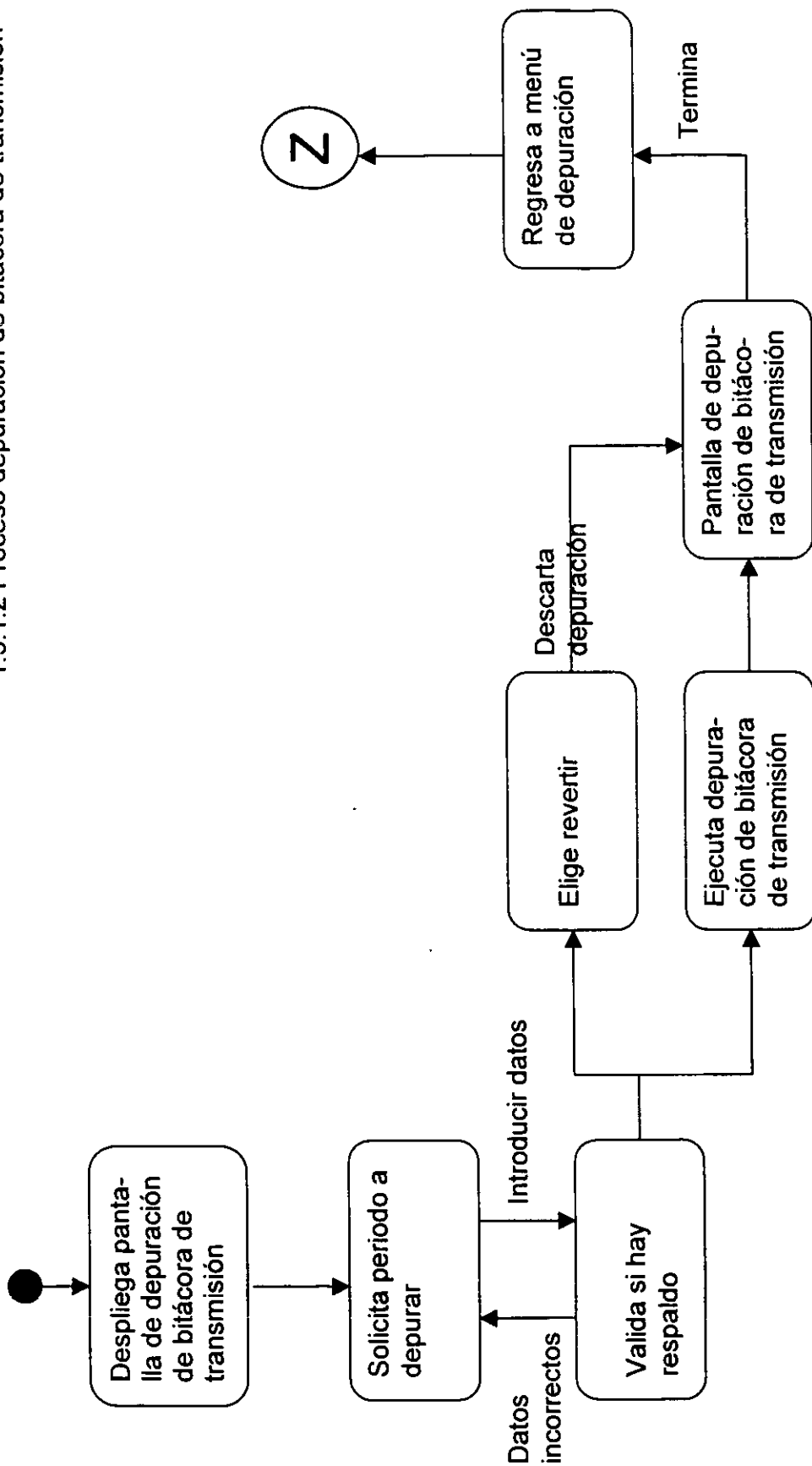


Diagrama de estado

1.5.1.3 Proceso depuración de tabla de pagos

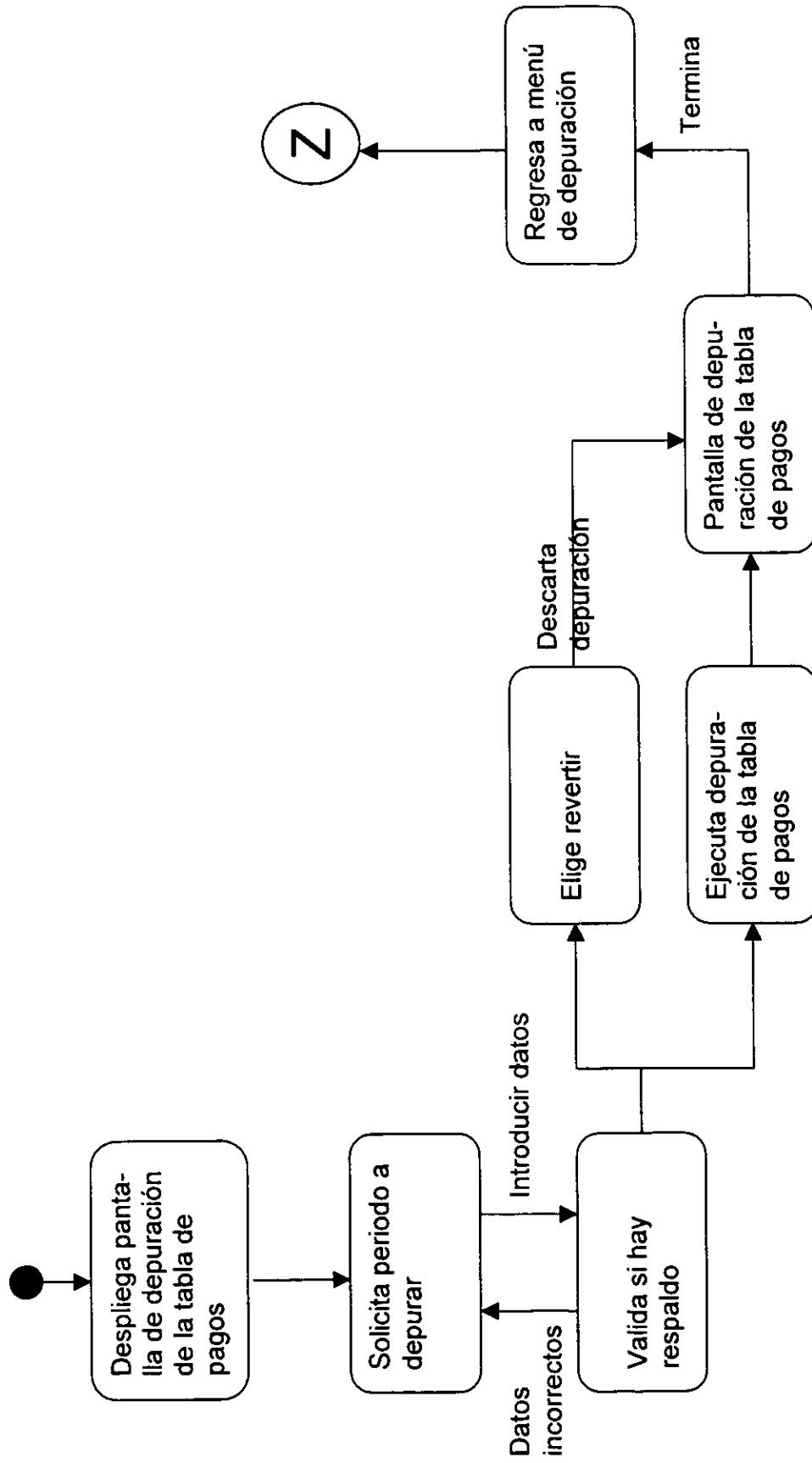


Diagrama de estado

1.5.1.4 Proceso depuración de tabla de cobros

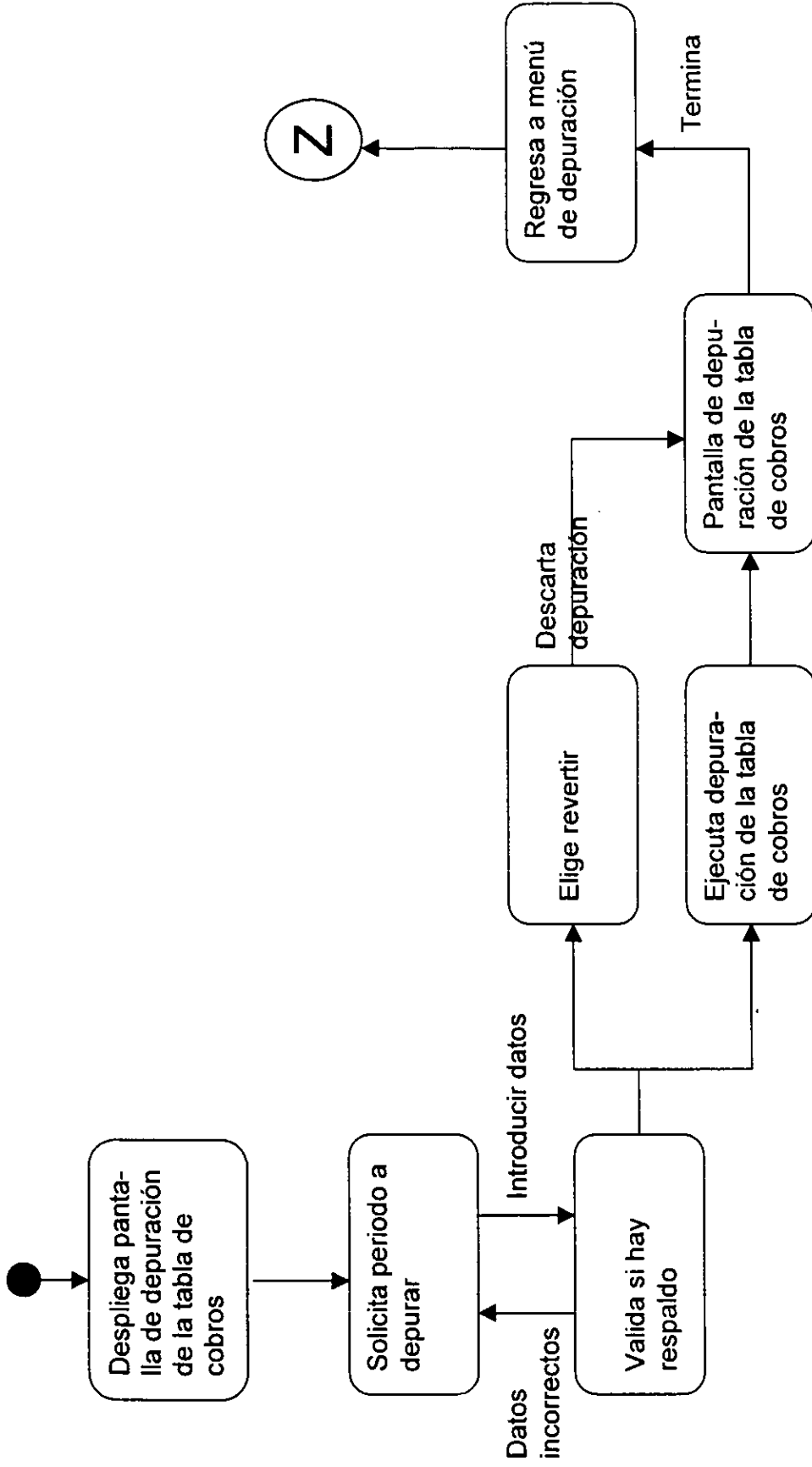


Diagrama de estado

1.5.2 Proceso Respaldo de archivos

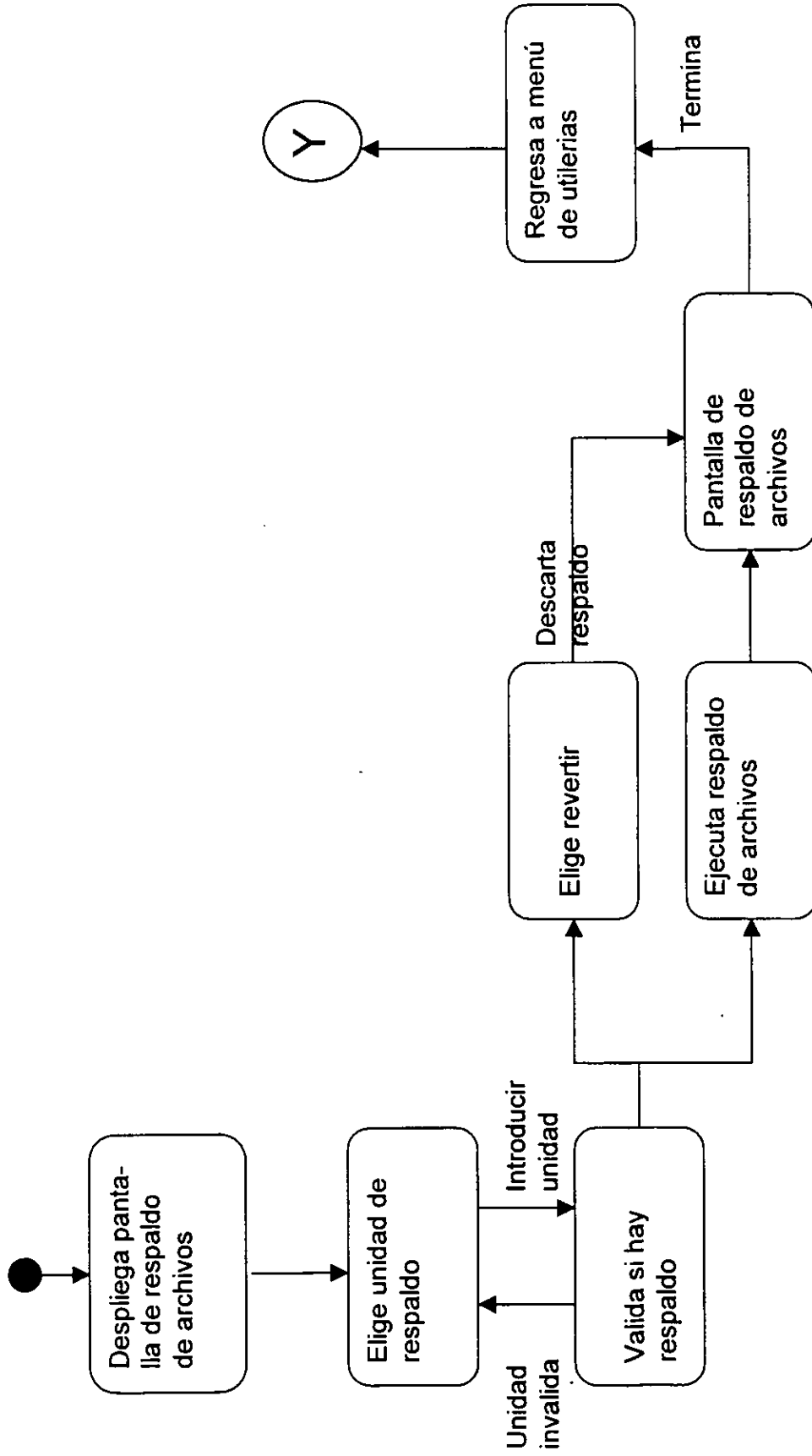


Diagrama de Estado
1.5.3 Proceso Restaurar archivos

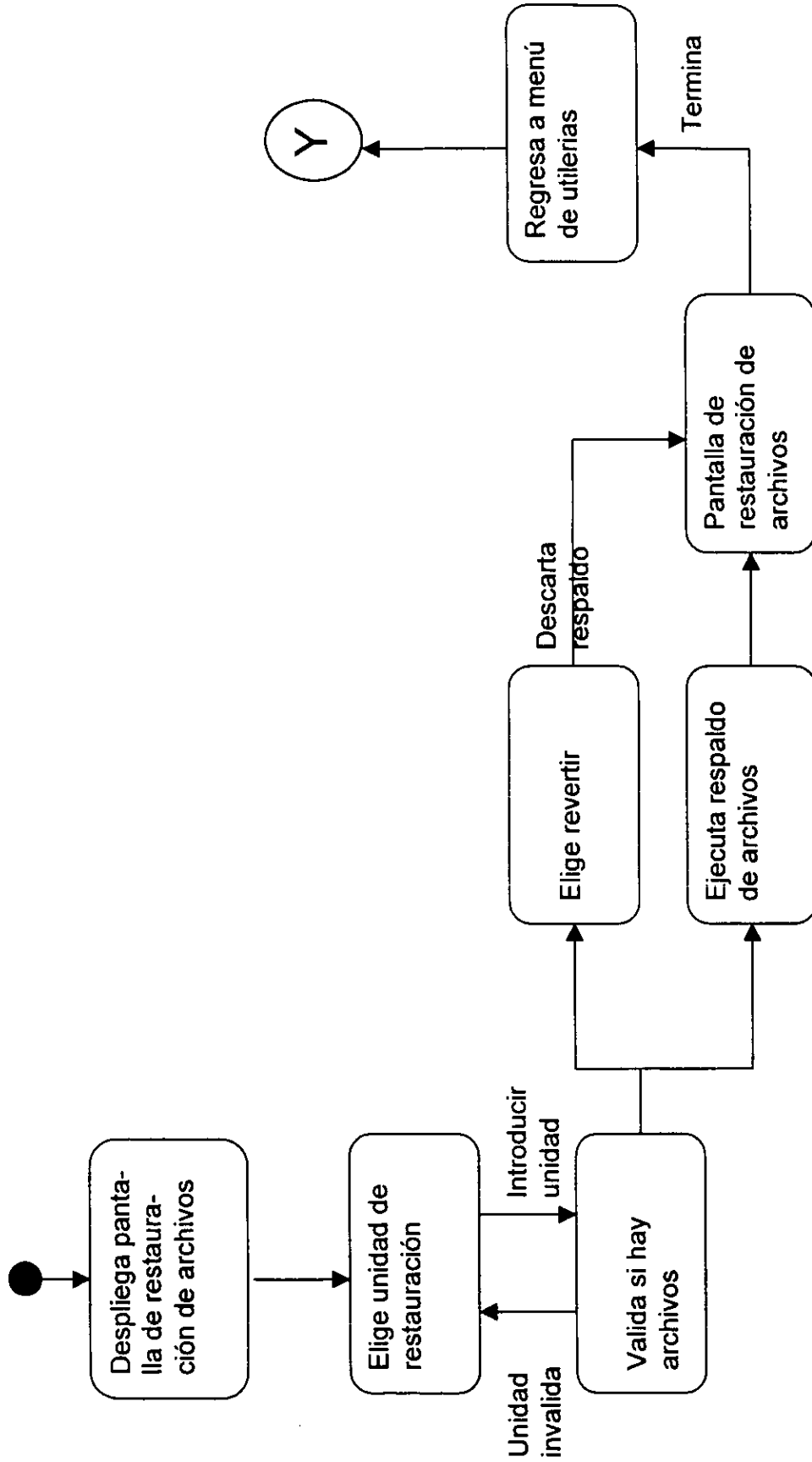


Diagrama de estado
1.2.4.1.3. Proceso Cobros modificar

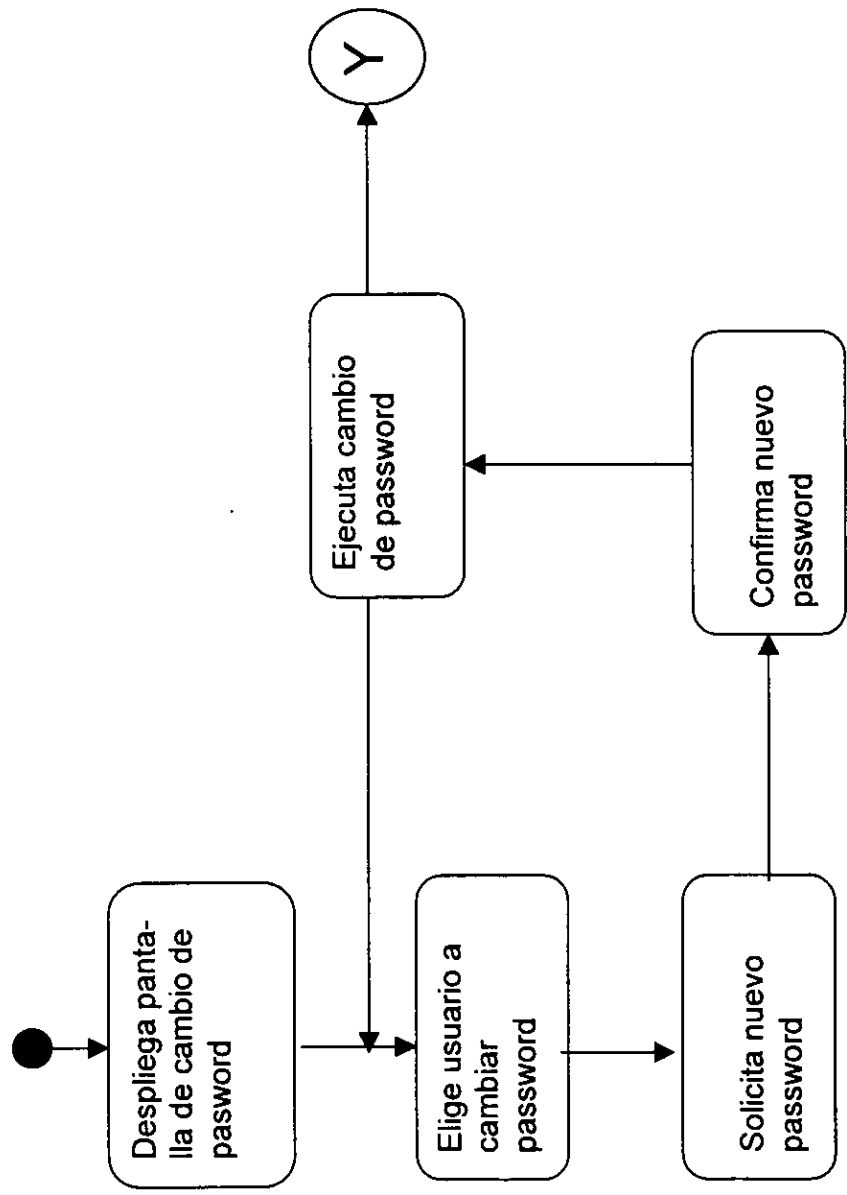


Diagrama de estado
1.6 Menú Bitácoras

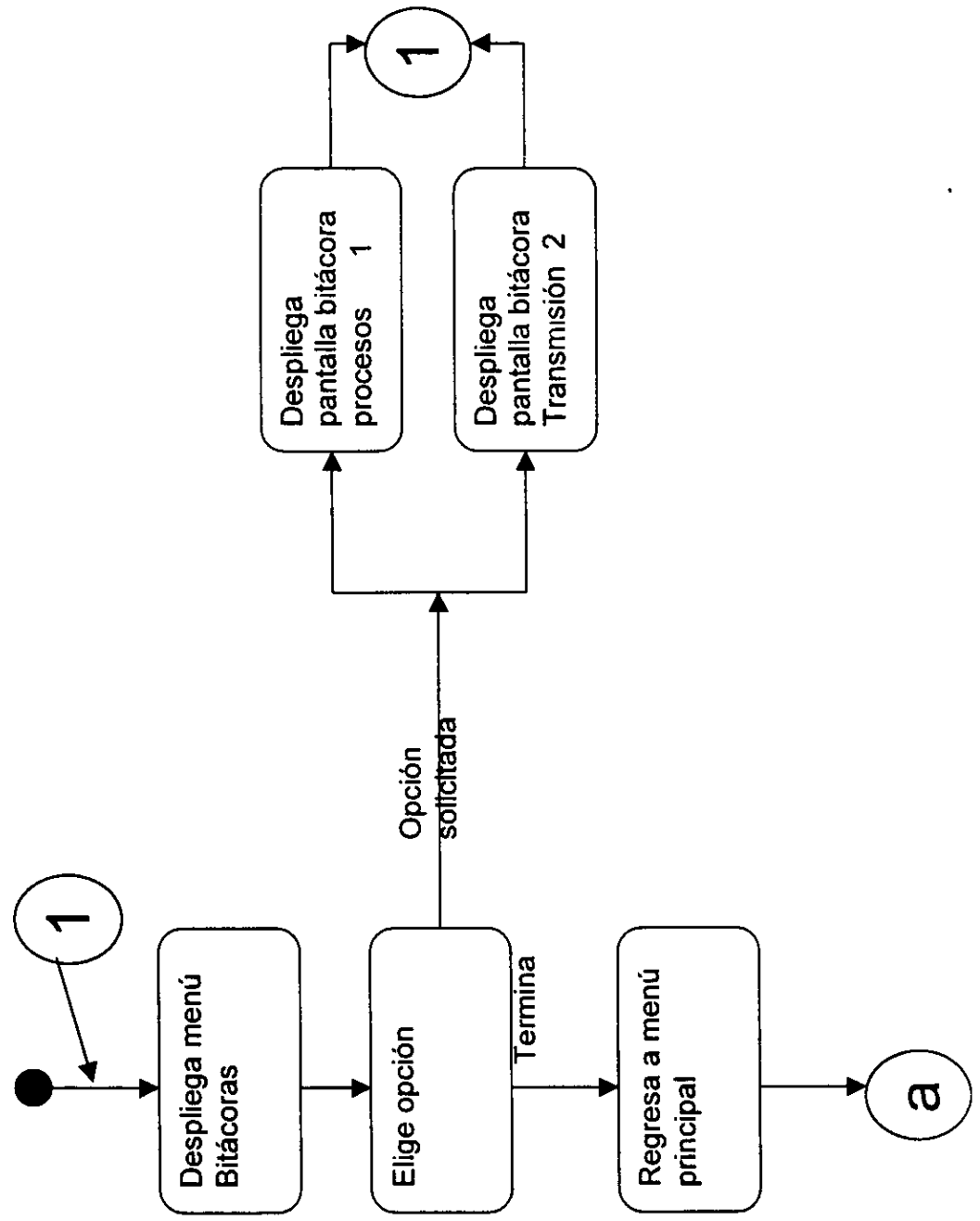


Diagrama de estado
1.61 Bitácora de procesos

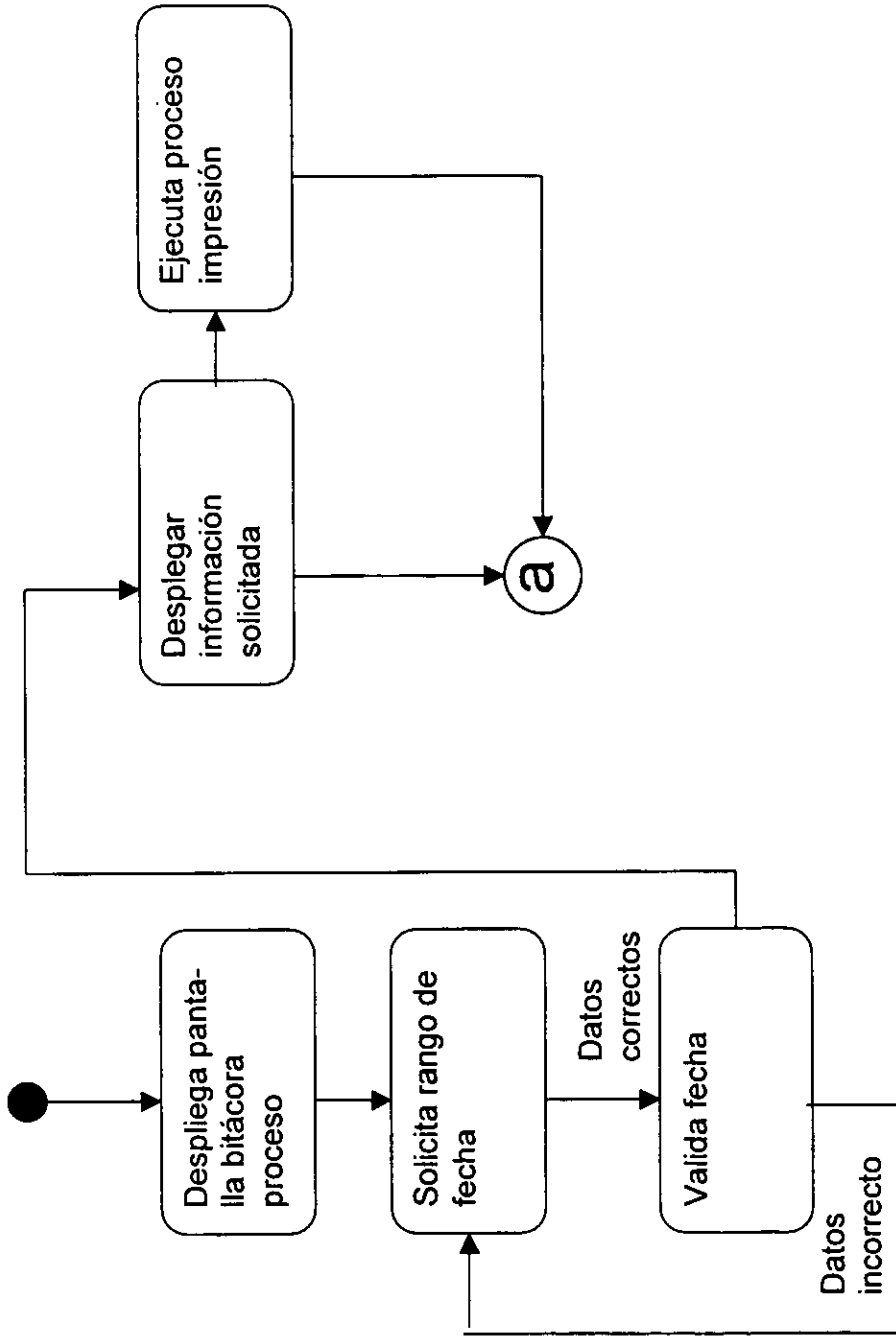
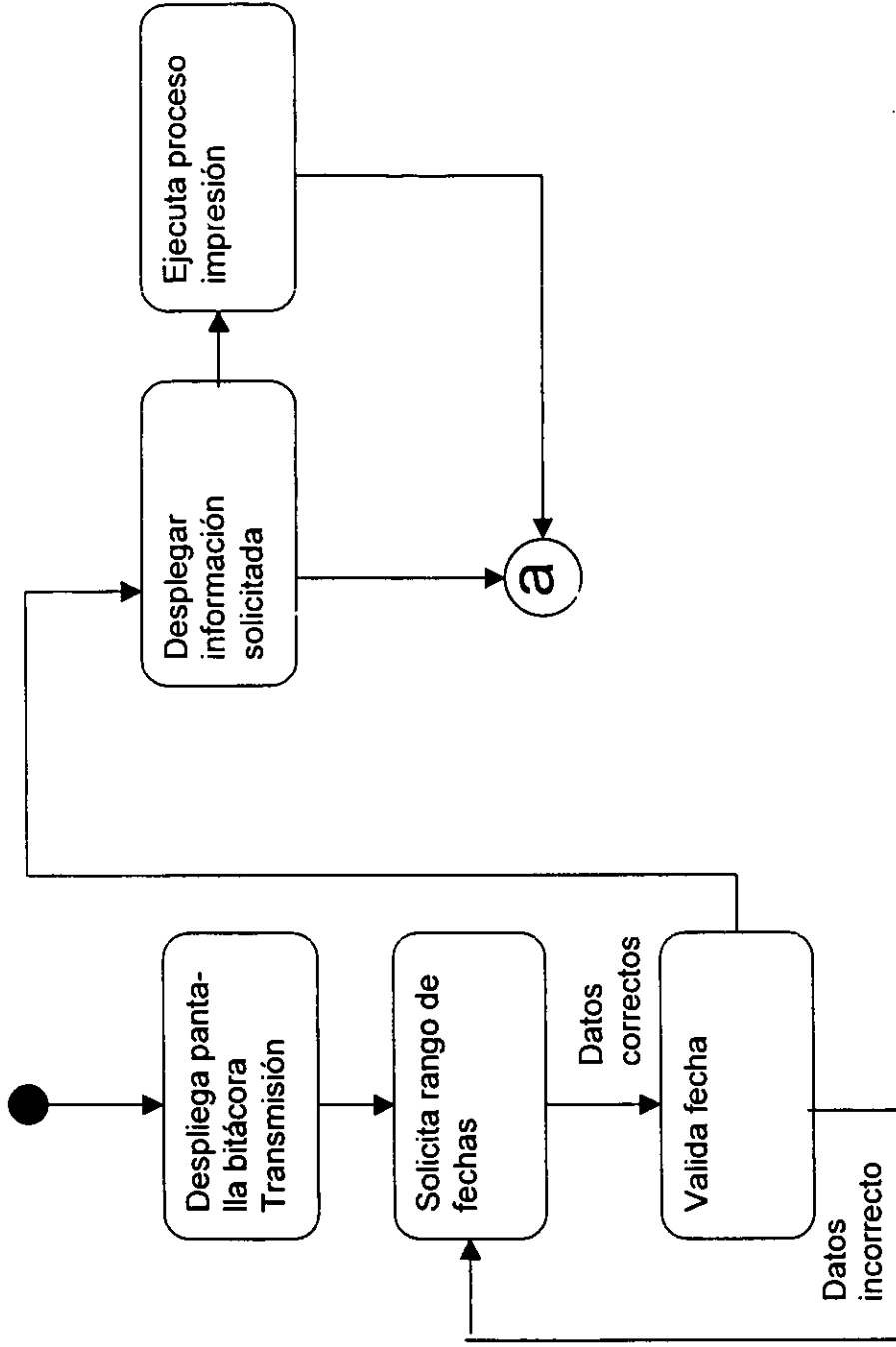


Diagrama de estado
1.62 Bitácora de Transmisión



3.1.4 DICCIONARIO DE DATOS.

Podemos observar en este punto la definición de los datos que conforman la base de datos del sistema. Esto con el propósito de documentar de una manera adecuada todas y cada una de las instancias que conforman la base de datos a utilizar y poder tener una referencia del tipo de información que podemos incluir en dicha base así como el significado de los campos , sus alias y tipo de datos.

DICCIONARIO DE LAS BASES DE DATOS

agentes (catalogo)			
Acrónimo	Descripción	tipo	long
vi_co_ag	Vía de cobro	small	3
nom_agen	Nombre del agente	char	20
vi_lo_ag	Vía de cobro layout	small	3
te_re_ag	Tel respon. vía cobro	char	10
baja_ag	Vía de cobro dado de baja	char	1

Usuarios (catalogo)			
Acrónimo	Descripción	tipo	long
cve_usua	Clave del usuario	char	7
pass_usu	Password	char	5
gpo_usu	Grupo de usuario	char	3
nom_usu	Nombre del usuario	char	20
dir_usu	Dirección de ubicación	char	20
tel_usu	Teléfono de localización	char	10
vi_co_us	Vía de cobro que representa	small	3

Pagos (tabla)			
Acrónimo	Descripción	tipo	long
via_co_pa	Vía de cobro	small	3
fe_in_pa	Fecha cobro inicial	fech	8
fe_fi_pa	Fecha cobro final	fech	8
imp_cob_pa	Importe cobrado	dec	9,2
fech_pag	Fecha de pago	fech	8
liqui_pa	Saldo liquidado	char	2

Estaciones de cobro (catalogo)			
Acrónimo	Descripción	tipo	long
num_ec	No. de estación de cobro	small	3
via_ec	Vía de cobro	small	3
nomb_ec	Nombre de la estación	char	20
sigla_ec	Siglas de la estación de cobro	char	3
direc_ec	Dirección de la estación	char	20
resp_ec	Responsable de la estación	char	20
tel_mod_ec	Teléfono de modem	char	10
tel_re_ec	Tel responsable	char	10

Histórico de cobros (tabla)			
Acrónimo	Descripción	tipo	long
telef_hc	Teléfono de pago	char	10
fech_hc	Fecha de cobro	fech	8
via_co_hc	Vía de cobro	small	3
es_co_hc	Estación de cobro	char	3
fe_rec_hc	Fecha de proceso	fech	8
imp_hc	Importe de cobro	dec	9,2

Cobros (tabla)			
Acrónimo	Descripción	tipo	long
telef_co	Teléfono de pago	char	10
fech_cob	Fecha de cobro	fech	8
via_cobr	Vía de cobro	small	3
est_cobr	Estación de cobro	small	3
fe_re_co	Fecha de proceso	fech	8
imp_cob	Importe de cobro	dec	9,2

bitácora de procesos (tabla)			
Acrónimo	Descripción	tipo	long
tarea_bp	Operación realizada	char	30
usua_bp	Usuario que la desempeño	char	7
fecha_bp	Fecha de operación	fech	8
hora_bp	Hora de operación	hr-mi	4

Grupos (tabla)			
acrónimo	Descripción	tipo	long
cve_gru	Clave de grupo	char	3
acces_gr	Acceso de grupo	char	41

bitácora de transmisión (tablas)			
acrónimo	Descripción	tipo	long
es_co_bt	Estación de cobro	small	3
via_co_bt	Vía de cobro (agente)	small	3
fech_bt	Fecha recepción de cobranza	fech	8
hora_bt	Hora recepción de cobranza	hr-mi	4
estado_bt	Estado de recepción	char	10

Por otro lado las claves que aparecen para definir cada módulo del sistema se definen de la siguiente manera:

DICCIONARIO DE MÓDULOS DEL SISTEMA DE COBRANZA POR TERCEROS
Nombre
Programa principal
Módulo de validación, seguridad del sistema
Transferencia de archivos
Transferencia y extracción de archivos
Extracción de archivos de terceros
Transmisión de archivos PISA
Transferencia de archivos por diskette
Respaldo de archivos PISA
Restaurar archivos de cobranza por terceros
Procesos especiales
Regeneración de archivo PISA
Integración cobranza
Integración terceros faltantes
Módulo de ajustes
Ajustes tablas de cobros
Ajustes tablas de pagos
Generación de reportes
Control de pagos
Captura de pagos de agentes

Diario de agentes
Verificación de pagos de clientes
Reporte diario de caja
Reporte estadístico
Reporte por vía de cobro
Mantenimiento de catálogos
Mantenimiento al catálogo de agentes
Mantenimiento al catálogo de estaciones de cobro
Mantenimiento al catálogo de usuarios
Mantenimiento al catálogo de grupos
Utilerías
Depuración de archivos
Depuración de bitácoras de procesos
Depuración de bitácoras de transmisiones
Depuración de tabla de pagos
Depuración tabla de histórico de cobros
Respaldo de archivos
Restauración de respaldos
Cambio de password
Bitácoras
Consulta a bitácora de procesos
Consulta a bitácora de transmisiones
Salida

Por ultimo describiremos los archivos Layout

HEADER	
Identificación	Observaciones
Clave de registro	0= Registro de control
Fecha de proceso	DDMMAA Fecha de envío a la empresa
Estación de cobro	Sucursales

Vía de cobro	Banco o Comercio: 01=BANAMEX; 02=BANCO LADATEL; 03=SERFIN; 04=BANPAIS; 05=BBV; 06=BANORTE; 07=BANCRECER; 08=BNJERCITO; 09=BANCOMER; 10=BITAL; 11=SANTANDER; 12=GIGANTE; 18=BANSUR; 21=INBURSA; 23=AURRERA; 24=WALL MART; 25=PROMEX; 26=BODEGA GIGANTE; 44=INVERLAT; 45=CITIBANK; 46=SAMBORNS; 47= SEARS.
--------------	--

DETALLE

Identificación	Observaciones
Clave de registro	1=Registro detalle
Importe	Importe del pago
Número telefónico	Teléfono que paga
Fecha de pago	DDMMAA Fecha en que paga el cliente
Fecha de proceso	DDMMAA Fecha de envío a la empresa

TRAILER

Identificación	Observaciones
Clave de registro	9=Registro totales
Número total de cargos	Total de registros
Importe total de cargos	Total del importe del archivo
Clave vía de cobro	Mismas claves de Header

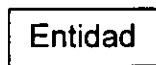
ARCHIVO DE ENVIO A PISA

Identificación	Observaciones
LADA-Teléfono	Clave LADA y teléfono que paga
Fecha de proceso	Fecha en que se envía a PISA
Importe	Importe pagado
Constante	"CE" clave especial manejada por cobros especiales
Constante	"4" clave especial manejada por cobros especiales

3.1.5 Diagrama de entidad relación

Para poder obtener un conocimiento más amplio de las bases de datos y su relación con los procesos dentro de un modelo funcional se cuenta con herramientas de análisis como el diagrama entidad relación. Este diagrama es un modelo que nos permite conocer de modo descriptivo la distribución de datos almacenados en un sistema.

Un diagrama entidad relación esta integrado por dos elementos básicos que son como su nombre lo indica: entidades y relaciones. Y se representa de la siguiente manera.



- La entidad es representada por medio de un rectángulo, lo cual representa una colección o conjunto de objetos reales o abstractos, por ejemplo (eventos, personas, lugares, etc.), que se encuentran en el mundo real, que tiene atributos comunes y que juegan algún papel en el desarrollo del sistema y son identificados de una manera única.

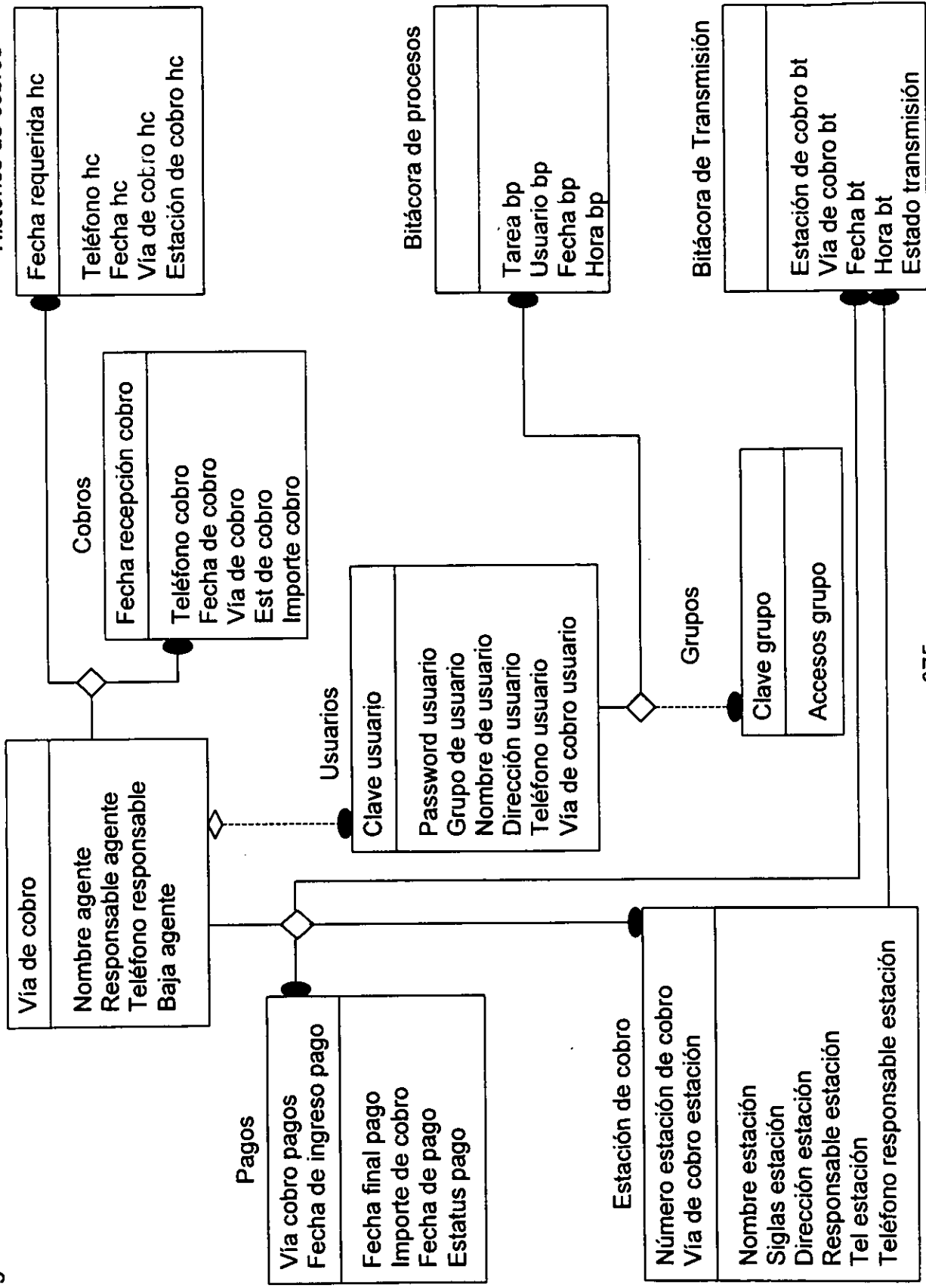


- La relación que se representa arriba es una asociación entre dos entidades y nos muestra un conjunto de conexiones entre objetos, instancias o entidades. Las relaciones se representan a través de líneas, las cuales una vez que se establecen, es evaluado cada extremo para determinar su cardinalidad que puede ser:

- Simples: Para una relación 1:1
- Múltiples : Para relaciones 1:N, 0:n, y N:N

Diagrama entidad relación

Agentes



3.1.6 Normalización

La normalización de datos es un proceso mediante el cual el diseñador de una base de datos es capaz de transformar cualquier estructura de datos no plana. A un conjunto de relaciones normalizadas, es decir a un conjunto de tablas planas que no contengan grupos repetitivos.

Una relación no plana o no normalizada contendrá al menos un dominio que es en realidad otra relación. Una relación ya normalizada tiene únicamente dominios simples, es decir dominios que no son repetitivos o no son a su vez otra relación.

Se normaliza un archivo que no es plano, excepto por un grupo repetitivo, al extraerlo y formar con él una relación separada. Esta relación debe de tener un nombre propio y un campo llave.

La Normalización se utiliza para poder obtener el enfoque relacional que posee bases matemáticas rigurosas, mismas que respaldan la teoría relacional, proporcionando simplicidad en las estructuras de datos utilizadas, con lo que facilitan su uso y modificaciones, además de que evita que los archivos de las bases de datos tengan redundancias, incoherencias o inconsistencias.

A continuación se presentan algunas definiciones, previas realizar la normalización con el propósito de entenderla en lo posible.

- **Dependencia funcional.** Dado un archivo, sean CAMPO1 y CAMPO2 dos campos de ese archivo, diremos que CAMPO2 tiene dependencia funcional de CAMPO1 cuando dado un valor de CAMPO1 se obtiene un valor único de CAMPO2 o que a cada valor de CAMPO2 le corresponde un único valor de CAMPO1.
- **Dependencia funcional completa.** Si un archivo, CAMPO1 es un conjunto de campos, entonces diremos que CAMPO2 tiene dependencia funcional de todo el

conjunto de CAMPO1, si CAMPO1 consta de un único atributo entonces coincide la dependencia funcional y la dependencia funcional completa.

- Llave o candidato. Este es un atributo o conjunto de atributos que sirve para identificar de manera unívoca un registro. Uno de los candidatos o llave se utiliza como llave principal o primaria
- Atributo no llave. Es aquel que no forma parte de la llave o candidato, es un atributo común.
- Determinante. Es un atributo o conjunto de los mismos del que dependen otros atributos.

Primera Forma Normal.

El llegar a la primera forma normal nos permite reducir la redundancia separándola en otro grupo a parte de la entidad debiendo contener el identificador de la entidad original para mantenerse relacionadas, por lo que se tiene una relación en primera forma normal si para cada valor específico de un identificador, existe uno y solo un valor de cada atributo es decir no hay grupos repetitivos, pero se mantiene un alto grado de redundancia.

Segunda Forma Normal.

Sabemos que una entidad se encuentra en segunda forma normal, si el valor de cualquier atributo que no es llave, dependa de todos los atributos que forman la llave . Para normalizar una entidad en la segunda forma, se crea una nueva entidad de los atributos que dependen parcialmente de una llave, siendo parte del identificador de esta nueva entidad el atributo, del cual depende para mantenerla relacionada con la entidad original.

Tercera Forma Normal.

Cuando una relación se encuentra en tercera forma normal el valor de cada atributo depende de toda la llave y no de cualquier otro que no lo sea. En esta forma se buscan los atributos que están dependiendo de otro que no es una llave. Para poner una entidad en tercera forma normal, se crea una entidad con los atributos que no dependen de ningún atributo que forma la llave, siendo el identificador de la nueva entidad, el atributo del cual era dependiente

Los objetivos que persigue la normalización son los de eliminar en lo posible todos aquellos datos que contengan anomalías, conservar toda la información y maximizar la flexibilidad.

La estructura debe permitir colocar todos los datos requeridos. La redundancia que pueda existir deberá ser causada por los elementos que son identificadores o llaves. por lo que se deben elegir aquellos que no estén sujetos a actualizaciones.

Los efectos indeseables son las anomalías que pueden presentarse en las operaciones de actualización, inserción y eliminación. Este tipo de anomalías las describimos enseguida.

Anomalía de inserción. No cabe almacenar nueva información sobre una entidad en particular hasta que se establece su relación con otra entidad.

Anomalía de eliminación. La eliminación de un solo registro puede ocasionar la eliminación de toda la ocurrencia de una entidad.

Anomalía de actualización. Si el valor de un atributo cambia, debe cambiar en los múltiples sitios donde se encuentra definido.

La capacidad de adaptabilidad ante los cambios, maximizan la independencia del uso particular de los datos.

Para normalizar se requieren tres, (de ahí que se denominen Primera, Segunda y Tercera Forma Normal), o mas acciones sobre un atributo de una entidad, hasta que el modelo alcance su ultima etapa de no redundancia, generalmente la normalización esta comprendida hasta la tercera etapa, donde el modelo ya alcanza un estado manejable y entendible para el usuario y el programador.

Lo anterior se establece en las dependencia funcionales que existen entre los atributos y los dominios particulares a esa relación. Las dependencias funcionales las determina directamente el significado o la semántica del contenido de la base de datos según la interpretación del diseñador de esta.

Enseguida se presenta la normalización para realizar la cobranza del servicio telefónico a través de agentes externos (terceros),. estos se muestran en la figura 3.1.6.1.

Numero de teléfono de cobro	Telef_co
Fecha de cobro	Fech_cob
Nombre estación de cobro	Nomb_ec
Importe de cobro	Imp_cob
Agente que realiza el cobro	Nom_agen
Usuario que revisa el cobro	Nom_usu
Pagos de los agentes	Imp_co_pa
Teléfono del responsable agente	Te_re_ag
Responsable agente	Resp_agen
Grupo de usuario	Gpo_usu
Dirección de usuario	Dir_usu
Teléfono del usuario	Tel_usu
Dirección estación de cobro	Direc_ec
Responsable estaciona de cobro	Res_ec
Teléfono responsable estación	Tel_re_ec
Importe de pago	Imp_cob_pa
Fecha de pago de cobros	Fech_pag
Estatus de cobro pago	tarea_bp

Figura 3.1.6.1 datos para cobranza por terceros.

Posteriormente se procede a separar los atributos que se puedan separar, adicionando algunos otros para poder identificar las relaciones que se originen

Cobranza Identificador: Cobros

Via_cobr (id)	Fech_cob	Telef_co	Est_cobr (id)	Fe_re_co
Imp_cob				

Pagos Identificador Pagos

Via_co_pa (id)	Fe_in_pa	Fe_fi_pa	Imp_cob_pa	Fech_pag
Liqui_pa				

Terceros Identificador: Agentes

Via_co_ag (id)	Nom_agen	Vi_lo_ag	Resp_agen	Te_le_ag
Baja_ag				

Usuarios Identificador: Usuarios

Cve_us	Pass_usu	Grp_us (id)	Nom_usu	Dir_usu
Tel_usu	Vi_co_us			

Estaciones de cobros Identificador: est_cobro

Num_ec (id)	Via_ec (id)	Nomb_ec	Sigla_ec	Direc_ec
Resp_ec	Tel_mod_ec	Tel_re_ec		

3.2 GENERACIÓN DE CÓDIGO PARA EL PROCESAMIENTO DE LA INFORMACIÓN

En este capítulo se presenta el código generado del Sistema de Cobranza por Terceros.

```

***** *
C:\COBRATEL\COBRATEL.PRG
.....
.....
*: Procedure File
C:\COBRATEL\COBRATEL.PRG
*:
*:      COBRATEL® Sistema de
cobranza por terceros
*:
*: Documented using Visual FoxPro
Formatting wizard version .05
.....
*: COBRATEL
* MODULO PRINCIPAL
Clear
CD c:\COBRATEL
SET date to ymd
SET century on
SET deleted on
SET talk off
SET SYSMENU Off

DO FORM PRINCIPAL
READ EVENTS
SET SYSMENU ON
Clear EVENTS
RETURN

***** * C:\ARCHIVOS DE
PROGRAMA\DEVSTUDIO\WFPWIZ
ARDS\WIZSTYLE.VCX
RecordSource = "customer"
FontName = "MS Sans Serif"
FontSize = 8
Height = 152
Width = 397
Name = "grid1"
ScaleMode = 3
Top = 6
Left = 6
DoCreate = .T.
wizbuttons = 0
wizbtnpos = 0
Name = "nobtns"
Height = 2
Left = 1
Top = 48
Width = 584
Name = "Shape1"
Height = 2
Left = 3
Top = 264
Width = 584

Name = "Shape2"
FontSize = 14
BackStyle = 0
Caption = "Label1"
Height = 22
Left = 27
Top = 12
Width = 62
AutoSize = .T.
Name = "Label1"
Height = 68
Width = 68
wizeffect = 0
Name = "wizshape"
Width = 517
Height = 205
TabIndex = 0
odimensions = Shape1
olabel1 = Shape2
olabel2 = Shape3
ocol2 = Shape4
Name = "layoutsty"
Height = 205
Left = 0
Top = 0
Width = 517
Name = "Shape1"
Height = 13
Left = 24
Top = 24
Width = 37
Name = "Shape2"
Height = 13
Left = 24
Top = 45
Width = 37
Name = "Shape3"
Height = 13
Left = 228
Top = 24
Width = 37
Name = "Shape4"
FontBold = .F.
FontName = "Arial"
FontSize = 8
BackStyle = 0
Caption = "Label1"
Height = 13
Width = 43
Name = "embossedlabel"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Height = 100
Width = 150
Visible = .T.
Sizable = .T.
Name = "embossedole"

FontName = "MS Sans Serif"
FontSize = 8
Alignment = 0
BackStyle = 0
Caption = "Label1"
Height = 15
Width = 39
AutoSize = .F.
Name = "standardlabel"
MS Sans Serif, 1, 8, 6, 13, 11, 12, 2,
0
Height = 16
Width = 68
FontName = "MS Sans Serif"
FontSize = 9
Alignment = 0
BackStyle = 0
Caption = "Check1"
SpecialEffect = 1
Name = "standardlogic"
MS Sans Serif, 1, 9, 6, 13, 11, 12, 2,
0
Height = 100
Width = 150
Name = "standardole"
Width = 94
Height = 276
Name = "vertxibtns"
cmdPrev.Top = 24
cmdPrev.Left = 0
cmdPrev.Name = "cmdPrev"
cmdNext.Top = 48
cmdNext.Left = 0
cmdNext.Name = "cmdNext"
cmdTop.Top = 0
cmdTop.Left = 0
cmdTop.Name = "cmdTop"
cmdEnd.Top = 72
cmdEnd.Left = 0
cmdEnd.Name = "cmdEnd"
cmdFind.Top = 108
cmdFind.Left = 0
cmdFind.Width = 55
cmdFind.Name = "cmdFind"
cmdPrint.Top = 132
cmdPrint.Left = 0
cmdPrint.Width = 55
cmdPrint.Name = "cmdPrint"
cmdExit.Top = 252
cmdExit.Left = 0
cmdExit.Height = 24
cmdExit.Width = 55
cmdExit.Name = "cmdExit"
cmdAdd.Top = 168
cmdAdd.Left = 0
cmdAdd.Name = "cmdAdd"

```

```

cmdDelete.Top = 216
cmdDelete.Left = 0
cmdDelete.Name = "cmdDelete"
cmdEdit.Top = 192
cmdEdit.Left = 0
cmdEdit.Name = "cmdEdit"
MS Sans Serif, 1, 8, 6, 13, 11, 12, 2,
0
ScaleMode = 3
Height = 293
Width = 401
DoCreate = .T.
AutoCenter = .T.
BackColor = 192,192,192
Caption = "Add Record"
WindowType = 1
Name = "gridaddform"
PROCEDURE Init
LPARAMETERS
cFldKey,cKeyValue,nBtnAction,nAdd
Action,IChildPrimaryKey,IUpdatableP
arentKey,INoSendUpdates

IF PARAMETERS() # 7
    RETURN .F.
ENDIF

this.CommandGroup1.Value = 0
this.Label3.Caption = m.cFldKey
this.cKeyValue = m.cKeyValue

DO CASE
CASE m.InoSendUpdates
    * Send Parent Updates
    this.Optiongroup1.Option1.
Enabled = .F.
    this.Optiongroup1.Option3.
Enabled = .F.
CASE !m.IUpdatableParentKey
    * Updatable parent key
    this.Optiongroup1.Option3.
Enabled = .F.
ENDCASE

* Has a primary key
IF m.IChildPrimaryKey
    this.Optiongroup1.Option2.
Enabled = .F.
ENDIF

DO CASE
CASE TYPE('THIS.cKeyValue') = "C"
    this.Text1.Value = ""
CASE
ATC(TYPE('THIS.cKeyValue'),"NYIB
F") # 0
    this.Text1.Value = 0
CASE
ATC(TYPE('THIS.cKeyValue'),"DT")
# 0
    this.Text1.Value = {}
CASE TYPE('THIS.cKeyValue') = "L"
    this.Text1.Value = .T.
ENDCASE

ENDPROC
AutoSize = .T.
ButtonCount = 3
BackStyle = 0
BorderStyle = 0

```

```

Value = 2
ControlSource = "nAddAction"
Height = 68
Left = 36
SpecialEffect = 0
Top = 48
Width = 218
TabIndex = 1
Name = "Optiongroup1"
Option1.BackStyle = 0
Option1.Caption = "Add record to
parent only"
Option1.Value = 0
Option1.Height = 18
Option1.Left = 5
Option1.Top = 5
Option1.Width = 181
Option1.AutoSize = .T.
Option1.Name = "Option1"
Option2.BackStyle = 0
Option2.Caption = "Add record to
child (grid) only"
Option2.Value = 1
Option2.Height = 18
Option2.Left = 5
Option2.Top = 25
Option2.Width = 208
Option2.AutoSize = .T.
Option2.Name = "Option2"
Option3.BackStyle = 0
Option3.Caption = "Add record to
both"
Option3.Value = 0
Option3.Height = 18
Option3.Left = 5
Option3.Top = 45
Option3.Width = 137
Option3.AutoSize = .T.
Option3.Name = "Option3"
PROCEDURE InteractiveChange
DO CASE
CASE THIS.Value = 2
    thisform.Text1.Value =
THISFORM.cKeyValue
CASE
TYPE('THISFORM.cKeyValue') = "C"
    thisform.Text1.Value = ""
CASE
ATC(TYPE('THISFORM.cKeyValue'),
"NYIBF") # 0
    thisform.Text1.Value = 0
CASE
ATC(TYPE('THISFORM.cKeyValue'),
"DT") # 0
    thisform.Text1.Value = {}
CASE
TYPE('THISFORM.cKeyValue') = "L"
    thisform.Text1.Value = .T.
ENDCASE

thisform.Text1.ReadOnly =
(THIS.Value = 2)

ENDPROC
AutoSize = .T.
ButtonCount = 2
BackStyle = 0
BorderStyle = 0
Value = 1.00
ControlSource = "nBtnAction"

```

```

Height = 39
Left = 120
Top = 248
Width = 163
Name = "Commandgroup1"
Command1.Top = 5
Command1.Left = 5
Command1.Height = 29
Command1.Width = 70
Command1.Caption = "Add"
Command1.TabIndex = 1
Command1.Name = "Command1"
Command2.Top = 5
Command2.Left = 88
Command2.Height = 29
Command2.Width = 70
Command2.Caption = "Cancel"
Command2.TabIndex = 2
Command2.Name = "Command2"
PROCEDURE Click
cKeyValue =
THIS.Parent.Text1.Value
nAddAction =
THIS.Parent.Optiongroup1.Value
nBtnAction = THIS.Value
thisform.Release()
ENDPROC
Alignment = 0
Height = 24
Left = 132
TabIndex = 3
Top = 192
Width = 218
Name = "Text1"
AutoSize = .T.
BackStyle = 0
Caption = "Key value:"
Height = 18
Left = 48
Top = 196
Width = 67
Name = "Label1"
AutoSize = .T.
BackStyle = 0
Caption = "A. Select option for
adding records:"
Height = 18
Left = 24
Top = 24
Width = 225
Name = "Label2"
FontItalic = .T.
BackStyle = 0
Caption = "Label3"
Height = 16
Left = 132
Top = 168
Width = 145
Name = "Label3"
AutoSize = .T.
BackStyle = 0
Caption = "Key field:"
Height = 18
Left = 48
Top = 168
Width = 60
Name = "Label4"
AutoSize = .T.
BackStyle = 0

```

```

Caption = "B: Enter key value linking
parent and child records:"
Height = 18
Left = 24
Top = 132
Width = 325
Name = "Label5"
Arial, 1, 10, 6, 16, 13, 15, 3, 0
Arial, 3, 10, 6, 16, 13, 17, 3, 0
Height = 22
Width = 39
Name = "gridaddrec"
PROCEDURE runaddform
#DEFINE C_NOOBJ_LOC "Failed
to create the Add Record form class.
Check or reinstall the
WIZSTYLE.VCX file."

```

```

PRIVATE
cFldKey,cKeyValue,nBtnAction,nAdd
Action,oGridAddForm

```

```

IF TYPE("THIS.KeyField") # "C"
    this.KeyField = ""
ENDIF

```

```

cFldKey = THIS.KeyField
cKeyValue = THIS.KeyValue
nBtnAction = 1
nAddAction = 1

```

```

oGridAddForm =
CREATE("gridaddform",m.cFldKey,m
.cKeyValue,,
    m.nBtnAction,m.nAddActio
n,THIS.ChildPrimaryKey,THIS.Updat
ableParentKey,THIS.NoSendUpdate
s)

```

```

IF TYPE("m.oGridAddForm") # "O"
    =MESSAGEBOX(C_NOO
BJ_LOC)
    this.AddOption = 0
    RETURN
ENDIF

```

```

IF THIS.NoSendUpdates
    oGridAddForm.Optiongrou
p1.Value = 2
    oGridAddForm.Text1.Valu
e = m.cKeyValue
ENDIF

```

```
oGridAddForm.Show()
```

```

this.AddOption =
IIF(m.nBtnAction=1,m.nAddAction,0)
this.KeyValue = m.cKeyValue

```

```

ENDPROC
Width = 207
Height = 111
Name = "shadowole"
Shape1.Height = 100
Shape1.Left = 54
Shape1.Top = 7
Shape1.Width = 150
Shape1.ZOrderSet = 0
Shape1.Name = "Shape1"
Label1.Height = 16

```

```

Label1.Left = 0
Label1.Top = 3
Label1.Width = 41
Label1.ZOrderSet = 1
Label1.Name = "Label1"
Height = 100
Left = 48
Top = 2
Width = 151
ZOrderSet = 2
Name = "Shape2"
Top = 2
Left = 48
Height = 100
Width = 151
TabIndex = 0
ZOrderSet = 3
Name = "Oleboundcontrol1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Width = 148
Height = 24
Name = "chiselfield"
Shape1.Height = 2
Shape1.Left = 46
Shape1.Top = 20
Shape1.Width = 97
Shape1.Name = "Shape1"
Label1.Height = 13
Label1.Left = 0
Label1.Top = 5
Label1.Width = 45
Label1.Name = "Label1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
BackColor = 192,192,192
BackStyle = 0
BorderStyle = 0
Height = 17
Left = 46
SpecialEffect = 1
TabIndex = 0
Top = 3
Width = 97
DisabledForeColor = 0,0,128
SelectedForeColor = 255,255,255
SelectedBackColor = 0,0,0
Name = "Text1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Arial, 0, 8, 5, 14, 11, 20, 3, 0
Width = 198
Height = 52
Name = "chiselmemo"
Shape1.Height = 2
Shape1.Left = 46
Shape1.Top = 50
Shape1.Width = 150
Shape1.Name = "Shape1"
Label1.Height = 13
Label1.Left = 0
Label1.Top = 3
Label1.Width = 45
Label1.Name = "Label1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
BackColor = 192,192,192
BackStyle = 0
Height = 48
Left = 46

```

```

SpecialEffect = 0
TabIndex = 0
Top = -1
Width = 150
DisabledForeColor = 0,0,128
SelectedForeColor = 255,255,255
SelectedBackColor = 0,0,0
Name = "Edit1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Arial, 0, 8, 5, 14, 11, 20, 3, 0
Width = 200
Height = 106
Name = "chiselole"
Shape1.Height = 2
Shape1.Left = 46
Shape1.Top = 103
Shape1.Width = 150
Shape1.Name = "Shape1"
Label1.Height = 13
Label1.Left = 0
Label1.Top = 3
Label1.Width = 45
Label1.Name = "Label1"
Top = 0
Left = 46
Height = 100
Width = 150
TabIndex = 0
Name = "Oleboundcontrol1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Width = 94
Height = 18
Name = "chisellogic"
Shape1.Height = 2
Shape1.Left = 53
Shape1.Top = 15
Shape1.Width = 13
Shape1.Name = "Shape1"
Label1.Height = 13
Label1.Left = 0
Label1.Top = 1
Label1.Width = 45
Label1.Name = "Label1"
Top = 0
Left = 53
Height = 15
Width = 15
FontName = "MS Sans Serif"
FontSize = 8
BackStyle = 0
Caption = ""
SpecialEffect = 0
TabIndex = 0
Name = "Check1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
MS Sans Serif, 1, 8, 6, 13, 11, 12, 2,
0
Width = 148
Height = 24
BackColor = 192,192,192
BackStyle = 0
BorderWidth = 0
SpecialEffect = 2
Name = "chiselbase"
BackColor = 255,255,255
Height = 2
Left = 46
Top = 20
Width = 97
SpecialEffect = 0

```


Name = "Shape1"
FontName = "Arial"
FontSize = 8
Alignment = 1
BackStyle = 0
Caption = "Label1"
Height = 13
Left = 0
Top = 5
Width = 45
ColorSource = 3
TabIndex = 0
Name = "Label1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Width = 94
Height = 43
Name = "boxfield"
Shape1.Height = 43
Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 94
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
FontSize = 9
BackColor = 255,255,255
BackStyle = 1
BorderStyle = 0
Height = 18
Left = 3
SpecialEffect = 1
TabIndex = 0
Top = 23
Width = 88
SelectedBackColor = 128,128,128
ColorSource = 0
Name = "Text1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 247
Height = 75
Name = "boxmemo"
Shape1.Height = 75
Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 246
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
FontSize = 9
BackColor = 255,255,255
BorderStyle = 0
Height = 50
Left = 3
ScrollBars = 0
SpecialEffect = 1
TabIndex = 0
Top = 23
Width = 241
ColorSource = 0
Name = "Edit1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 156
Height = 125
Name = "boxole"

Shape1.Height = 125
Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 156
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
Top = 23
Left = 3
Height = 100
Width = 150
TabIndex = 0
Name = "Oleboundcontrol1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 88
Height = 43
Name = "boxlogic"
Shape1.Height = 43
Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 88
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
Top = 22
Left = 2
Height = 17
Width = 13
FontSize = 9
BackStyle = 1
Caption = ""
SpecialEffect = 0
TabIndex = 0
ColorSource = 0
Name = "Check1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 94
Height = 43
BackStyle = 0
BorderWidth = 0
SpecialEffect = 2
Name = "boxbase"
BackColor = 255,255,255
BackStyle = 1
FillColor = 255,255,255
FillStyle = 0
Height = 43
Left = 0
Top = 0
Width = 94
SpecialEffect = 1
wizeffect = 1
Name = "Shape1"
FontSize = 9
BackColor = 0,0,255
BackStyle = 0
Caption = "Label1"
ForeColor = 0,0,255
Height = 15
Left = 3
Top = 2
Width = 49
TabIndex = 0
Name = "Label1"

Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 204
Height = 55
Name = "shadowmemo"
Shape1.Top = 7
Shape1.Left = 52
Shape1.Height = 48
Shape1.Width = 152
Shape1.Name = "Shape1"
Label1.Height = 16
Label1.Left = 0
Label1.Top = 4
Label1.Width = 41
Label1.Name = "Label1"
FontName = "Arial"
FontSize = 8
Height = 48
Left = 48
SpecialEffect = 1
TabIndex = 0
Top = 2
Width = 150
ColorSource = 0
BackColor = 255,255,255
Name = "Edit1"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Width = 149
Height = 25
Name = "shadowfield"
Shape1.Top = 5
Shape1.Left = 51
Shape1.Height = 20
Shape1.Width = 97
Shape1.Name = "Shape1"
Label1.Height = 16
Label1.Left = 0
Label1.Top = 3
Label1.Width = 41
Label1.Name = "Label1"
FontName = "Arial"
FontSize = 8
Height = 22
Left = 46
SpecialEffect = 1
TabIndex = 0
Top = 0
Width = 97
BackColor = 255,255,255
DisabledBackColor = 255,255,255
SelectedBackColor = 128,128,128
Name = "Text1"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Width = 94
Height = 18
Name = "shadowlogic"
Shape1.Top = 7
Shape1.Left = 58
Shape1.Height = 11
Shape1.Width = 11
Shape1.Name = "Shape1"
Label1.Height = 16
Label1.Left = 0
Label1.Top = 0
Label1.Width = 41
Label1.Name = "Label1"
Top = 0
Left = 52
Height = 15
Width = 15
BackStyle = 0

```

Caption = ""
SpecialEffect = 1
TabIndex = 0
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "Check1"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
FontBold = .F.
FontName = "Arial"
FontSize = 8
Height = 22
Width = 113
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "embossedfield"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Height = 15
Width = 72
FontBold = .F.
FontName = "Arial"
FontSize = 8
BackStyle = 0
Caption = "Check1"
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "embossedlogic"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
Height = 50
Width = 150
ColorSource = 0
Name = "embossedmemo"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Width = 513
Height = 29
wizbmppath =
(HOME()+"WIZARDS\WIZBMPS\")
Name = "picbtns"
cmdPrev.Top = 4
cmdPrev.Left = 51
cmdPrev.Width = 50
cmdPrev.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
BACK.BMP")
cmdPrev.Caption = ""
cmdPrev.ToolTipText = "Previous
record"
cmdPrev.Name = "cmdPrev"
cmdNext.Top = 4
cmdNext.Left = 102
cmdNext.Width = 50
cmdNext.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
NEXT.BMP")
cmdNext.Caption = ""
cmdNext.ToolTipText = "Next record"
cmdNext.ColorSource = 0
cmdNext.Name = "cmdNext"
cmdTop.Top = 4
cmdTop.Left = 0
cmdTop.Width = 50
cmdTop.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
TOP.BMP")
cmdTop.Caption = ""
cmdTop.ToolTipText = "Top record"
cmdTop.Name = "cmdTop"

```

```

cmdEnd.Top = 4
cmdEnd.Left = 153
cmdEnd.Width = 50
cmdEnd.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
END.BMP")
cmdEnd.Caption = ""
cmdEnd.ToolTipText = "Bottom
record"
cmdEnd.Name = "cmdEnd"
cmdFind.Top = 4
cmdFind.Left = 204
cmdFind.Width = 50
cmdFind.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
LOCATE.BMP")
cmdFind.Caption = ""
cmdFind.ToolTipText = "Find
records"
cmdFind.Name = "cmdFind"
cmdPrint.Top = 4
cmdPrint.Left = 255
cmdPrint.Width = 50
cmdPrint.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
PRINT.BMP")
cmdPrint.Caption = ""
cmdPrint.ToolTipText = "Print report"
cmdPrint.Name = "cmdPrint"
cmdExit.Top = 4
cmdExit.Left = 459
cmdExit.Width = 50
cmdExit.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
CLOSE.BMP")
cmdExit.Caption = ""
cmdExit.ToolTipText = "Exit form"
cmdExit.Name = "cmdExit"
cmdAdd.Top = 4
cmdAdd.Left = 306
cmdAdd.Width = 50
cmdAdd.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
NEW.BMP")
cmdAdd.Caption = ""
cmdAdd.ToolTipText = "Add / Save
record"
cmdAdd.Name = "cmdAdd"
cmdDelete.Top = 4
cmdDelete.Left = 408
cmdDelete.Width = 50
cmdDelete.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
DELETE.BMP")
cmdDelete.Caption = ""
cmdDelete.ToolTipText = "Delete
record"
cmdDelete.Name = "cmdDelete"
cmdEdit.Top = 4
cmdEdit.Left = 357
cmdEdit.Width = 50
cmdEdit.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
EDIT.BMP")
cmdEdit.Caption = ""
cmdEdit.ToolTipText = "Edit / Revert
record"
cmdEdit.Name = "cmdEdit"
PROCEDURE Init

```

```

TxtBtns::Init()
thisform.ShowTips = .T.

ENDPROC
PROCEDURE setcaption
IF !THIS.EditMode
    this.cmdAdd.Picture =
THIS.wizbmppath+"wznew.bmp"
    this.cmdEdit.Picture =
THIS.wizbmppath+"wzedit.bmp"
    this.cmdAdd.DownPicture
= THIS.wizbmppath+"wznew.bmp"
    this.cmdEdit.DownPicture
= THIS.wizbmppath+"wzedit.bmp"
ELSE
    this.cmdAdd.Picture =
THIS.wizbmppath+"wzsave.bmp"
    this.cmdEdit.Picture =
THIS.wizbmppath+"wzundo.bmp"
    this.cmdAdd.DownPicture
= THIS.wizbmppath+"wzsave.bmp"
    this.cmdEdit.DownPicture
= THIS.wizbmppath+"wzundo.bmp"
ENDIF
ENDPROC
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
FontName = "MS Sans Serif"
FontSize = 8
Height = 43
SpecialEffect = 1
Width = 150
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "standardmemo"
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
ScaleMode = 3
DoCreate = .T.
ColorSource = 0
BackColor = 255,255,255
wizlayout = layoutsty
wizfield = standardfield
wizmemo = standardmemo
wizlogic = standardlogic
wizole = standardole
wizmaxcharld = 45
wizlblsuffix = (":")
wizlblcap = proper
wizformstretch = .T.
wizlbldefwid = .T.
wiztitle = Label1
wizuser = 0
wizgrid = grid1
wizverify = .T.
wizbtnpos = 0
wizcodestyle = .T.
wizlabel = standardlabel
wizlblspace = 10
wizcboxlbl = .T.
Name = "standardform"
Top = 6
Left = 9
Width = 561
Height = 253
TabIndex = 0
BackColor = 255,255,255
odimensions = Shape1
Name = "Layoutsty"
Shape1.Top = 0

```

```

Shape1.Left = 0
Shape1.Height = 252
Shape1.Width = 561
Shape1.BackColor = 255,255,255
Shape1.Name = "shape1"
Shape2.Top = 7
Shape2.Left = 16
Shape2.Height = 13
Shape2.Width = 37
Shape2.Name = "shape2"
Shape3.Top = 22
Shape3.Left = 16
Shape3.Height = 13
Shape3.Width = 37
Shape3.Name = "shape3"
Shape4.Top = 7
Shape4.Left = 56
Shape4.Height = 13
Shape4.Width = 37
Shape4.Name = "shape4"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
ErasePage = .T.
PageCount = 2
TabStretch = 1
TabStyle = 1
Width = 287
Height = 169
Name = "boxpages"
Page1.Caption = "Page1"
Page1.Name = "Page1"
Page2.Caption = "Page2"
Page2.Name = "Page2"
Width = 296
Height = 26
BackStyle = 0
BorderWidth = 0
Visible = .T.
TabIndex = 1
nworkarea = 0
previewmode = .F.
oldrefresh = 0
oldrec = 0
oldreprocess = 0
previewinit = .T.
usedataenv = .T.
Name = "gridbtns"
PROCEDURE initvars
#DEFINE C_NOUPDATEVIEW_LOC
    "Edits to one or more of the
Views may not be permanent. "+;
    "To remedy this, ensure
the View's Send SQL Updates
checkbox is checked in the View
Designer."
#DEFINE C_READONLY_LOC
    "The table is
Read-Only. You will not be able to
edit it."

LOCAL
aTablesUsed, nTablesUsed, i, aMems,
nTotMem, cWizFile, IShowedMess, cDataEnvRef
DIMENSION aTablesUsed[1]
DIMENSION aMems[1]

this.nworkarea = SELECT()
this.oldSetDelete =
SET("DELETED")
SET DELETED ON

```

```

this.oldreprocess =
SET("REPROCESS")
SET REPROCESS TO 0

* These properties should not be
used. They are reserved for use by
* the Preview button of the Form
Wizards.
this.previewmode =
IIF(TYPE("THIS.PreviewMode")#"L",
.F., THIS.previewmode)
this.previewinit =
IIF(TYPE("THIS.PreviewInit")#"L", .T.,
THIS.previewinit)

* Check for data environment
DO CASE
CASE
TYPE("THISFORM.DataEnvironment
") = "O"
    this.usedataenv = .T.
    nTotMem =
AMEMBERS(aMems, THISFORM.DataEnvironment, 2)
    cDataEnvRef =
"THISFORM.DataEnvironment"
CASE
TYPE("THISFORMSET.DataEnvironment") = "O"
    this.usedataenv = .T.
    nTotMem =
AMEMBERS(aMems, THISFORMSET.DataEnvironment, 2)
    cDataEnvRef =
"THISFORMSET.DataEnvironment"
    * let's also set the
IF
TYPE("THISFORM")="O" AND
!THISFORM.VISIBLE
    thisform.VISIBLE
= .T.
ENDIF
OTHERWISE
    this.usedataenv = .F.
ENDCASE

IF THIS.usedataenv
WITH
EVAL(m.cDataEnvRef)
    * Check for
Views
FOR i = 1 TO
m.nTotMem
IF
UPPER(EVAL("."+aMems[m.i]+"BaseClass")) = "CURSOR"
WITH
EVAL("."+aMems[m.i])
IF
CURSORGETPROP("SourceType",
Alias)#3
    *
Check if we need to requery for
deleted records.
* -
Note: Parameterized views are not

```

```

required so all records will be
brought over.

* To
remedy this situation you can place a
SET DELETED ON command in the

*
BeforeOpenTables event of the
DataEnvironment.

IF
THIS.oldSetDelete = "OFF" AND
ATC("?", CURSORGETPROP("SQL",
.Alias))-0
    =REQUERY(.Alias)
ENDIF

*
Check if updates are made
IF
!CURSORGETPROP("SendUpdates",
.Alias) AND !m.IShowedMess
    =MESSAGEBOX(C_NOUP
DATEVIEW_LOC)
    IShowedMess = .T.
ENDIF
ENDIF
ENDWITH
ENDFOR
ENDWITH
ENDIF
ENDIF

IF ISREADONLY()
WAIT WINDOW
C_READONLY_LOC TIMEOUT 2
ENDIF

* Disable appropriate buttons
this.cmdAdd.Enabled =
!ISREADONLY()
this.cmdDelete.Enabled =
!ISREADONLY()

IF THIS.previewmode
RETURN
ENDIF

IF THIS.usedataenv
RETURN
ENDIF

* The following code is here to
support forms not using a
DataEnvironment.
this.oldSetFields = SET("FIELDS")
SET FIELDS OFF

```

```

this.oldMultiLocks =
SET("MULTILOCKS")
SET MULTILOCKS ON
this.oldrefresh = SET("REFRESH")
SET REFRESH TO 5
IF !EMPTY(ALIAS())
    this.oldBuffering=CursorGe
tProp("buffering")
    m.nTablesUsed =
AUSED(aTablesUsed)
    FOR i = 1 TO
m.nTablesUsed
        IF
CursorGetProp('sourcetype',aTables
Used[m,i,1])#3    &&skip for views

            =CursorSetProp("buffering"
,5,aTablesUsed[m,i,1])
            &&optimistic table buffering
            ENDIF
        ENDFOR
    ENDFIF
GO TOP

ENDPROC
PROCEDURE updatereows
#DEFINE E_FAIL_LOC
    "Failed to update
table:"
#DEFINE E_TRIGGERFAIL_LOC
    "Trigger failed."
#DEFINE E_FIELDNULL_LOC
    "Field doesn't
accept NULL"
#DEFINE E_FIELDRULE_LOC
    "Field rule
violated"
#DEFINE E_RECORDLOCK_LOC
    "Record in use by another
user"
#DEFINE E_ROWRULE_LOC
    "Row rule
violated"
#DEFINE E_UNIQUEINDEX_LOC
    "Unique index violation"
#DEFINE E_DIRTYREC_LOC
    "Data has been
changed by another user. Overwrite
changes with your edits?"
#DEFINE E_NOFORCE_LOC
    "Could not force table
updates."
#DEFINE E_PROMPT_LOC
    "Error: "
#DEFINE MSGBOX_YES
    6

LOCAL
aErrors,cErrorMessage,aTablesUsed
,nTablesUsed,nTotErr
LOCAL
nFld,i,nOldArea,iSuccess,lnInDBC,iOv
erwrite,iHadMessage

DIMENSION aTablesUsed[1]
DIMENSION aErrors[1]
m.cErrorMessage=""
m.iSuccess = .T.
m.nOldArea = SELECT()

```

```

m.nTablesUsed =
AUSED(aTablesUsed)

* Can wrap everything in transaction
if using strictly DBCs

FOR i = 1 TO m.nTablesUsed
    SELECT
(aTablesUsed[m,i,1])

        m.lnInDBC =
!EMPTY(CURSORGETPROP("Data
base"))
        m.cErrorMessage = ""
        m.iOverwrite = .F.
        m.iHadMessage = .F.

        DO CASE
        CASE
CURSORGETPROP("Buffering") = 1
            * Skip if buffering
not on
                LOOP
                CASE GetFldState(0) = 2
                    &&deleted record
                    * Only delete
current record and force it
                    m.iSuccess =
TableUpdate(.F.,.T.)
                    IF m.iSuccess
                        &&successful update
                        LOOP
                            ENDFIF
                            CASE !m.lnInDBC AND
(ATC("2",GetFldState(-1))#0 OR;
ATC("3",GetFldState(-1))#0)
                                * Field was
edited - in Free Table
                                * Since free
tables are not supported by
transactions,
                                * we must
process record by record
                                m.nModRecord =
GetNextMod(0)
                                DO WHILE
m.nModRecord # 0 &&loop locks all
records
                                    GO
m.nModRecord
                                        m.iSuccess = RLOCK()
                                        &&try to lock record
                                        IF
!m.iSuccess
                                            &&failed to lock record
                                                m.cErrorMessage =
E_RECORDLOCK_LOC
                                                UNLOCK ALL
                                                EXIT
                                        ENDFIF
                                    ENDFOR
                                ENDDO
                                IF m.iSuccess
                                    &&was able to lock all
records
                                        m.iSuccess =
TableUpdate(.T.,m.iOverwrite)
                                        IF
m.iSuccess &&was able to update all
records
                                            LOOP
                                                ENDFIF
                                            UNLOCK ALL
                                            ENDFIF
                                            CASE m.lnInDBC

```

```

IF
!m.iHadMessage    &&so we don't
repeat alert
    * See if record(s) modified
by another user
        FOR m.nFld = 1 TO
FCOUNT()
            IF
TYPE(FIELD(m.nFld)) = "G"    &&skip
for General fields
                LOOP
                    ENDFIF
            IF
OLDVAL(FIELD(m.nFld)) #
CURVAL(FIELD(m.nFld))
                m.iHadMessage = .T.
            IF
MESSAGEBOX(E_DIRTYREC_LOC
,4+48) = MSGBOX_YES
                m.iOverwrite = .T.
            ELSE
                m.iSuccess = .F.
            UNLOCK ALL
            EXIT
        ENDFIF
    ENDFOR
    ENDFIF

    m.nModRecord =
GetNextMod(m.nModRecord)
    ENDDO
    IF m.iSuccess
        &&was able to lock all
records
            m.iSuccess =
TableUpdate(.T.,m.iOverwrite)
            IF
m.iSuccess &&was able to update all
records
                LOOP
                    ENDFIF
                UNLOCK ALL
                ENDFIF
                CASE m.lnInDBC

```

```

BEGIN
TRANSACTION
  * Try to update
  all records in selected table
  m.lSuccess =
  TableUpdate(.T.,.F.)
  &&successful update
  IF m.lSuccess
    END
TRANSACTION
  LOOP
  ENDIF
  ROLLBACK
ENDCASE

* Handle errors
nTotErr
=AERROR(aErrors)
DO CASE
CASE nTotErr = 0

CASE aErrors[1,1] = 1539

  && Trigger failed

  m.cErrorMessage =
E_TRIGGERFAIL_LOC
CASE aErrors[1,1] = 1581

  && Field doesn't accept
NULL

  m.cErrorMessage =
E_FIELDNULL_LOC
CASE aErrors[1,1] = 1582

  && Field rule violated

  m.cErrorMessage =
E_FIELDRULE_LOC
CASE aErrors[1,1] = 1700

  && Record in use by
  another user

  m.cErrorMessage =
E_RECORDLOCK_LOC
CASE aErrors[1,1] = 1583

  && Row rule violated

  m.cErrorMessage =
E_ROWRULE_LOC
CASE aErrors[1,1] = 1884

  && Unique index violation

  m.cErrorMessage =
E_UNIQUEINDEX_LOC
CASE aErrors[1,1] = 1585

  && Record changed by
  another user

  IF m.lInDBC
    &&handle free
  tables above

  *
  Dislpay conflict alert

```

```

IF
MESSAGEBOX(E_DIRTYREC_LOC
,4+48) = MSGBOX_YES

  *Try to force update

  BEGIN TRANSACTION

  m.lSuccess =
  TABLEUPDATE(.T.,.T.)

  IF m.lSuccess

    END

  TRANSACTION

  LOOP

  ELSE

    ROLLBACK

    =MESSAGEBOX(E_NOFOR
    RCE_LOC)

  ENDIF

  ENDIF

  OTHERWISE

  IF

  !EMPTY(m.cErrorMessage) &&for
  free table handling above

  m.cErrorMessage =
  E_PROMPT_LOC+aErrors[1,2]
  ENDIF

  ENDCASE

  * Had an error we couldn't
  handle

  =TABLEREVERT(.T.)
  &&revert all records
  m.lSuccess = .F.
  IF

  !EMPTY(m.cErrorMessage)

  =MESSAGEBOX(E_FAIL_
  LOC+m.cErrorMessage)
  ENDIF

  ENDFOR

  SELECT (m.nOldArea)
  RETURN m.lSuccess

ENDPROC
PROCEDURE Refresh
**** Special Preview Mode Handling
****
IF THIS.previewmode AND
THIS.previewinit

  this.previewinit = .F.
  this.cmdAdd.Enabled = .F.
  this.cmdDelete.Enabled =
.F.

  this.cmdFind.Enabled = .F.
  this.cmdPrint.Enabled = .F.
  this.cmdExit.Enabled = .F.

```

```

this.nworkarea =
SELECT()
ENDIF

ENDPROC
PROCEDURE Init
#DEFINE C_WIZSTYLE

  "WIZSTYLE.VCX"
#DEFINE C_WIZDIR

  "WIZARDS\1"
#DEFINE C_PROMPT1_LOC
"Find: "
#DEFINE E_NOSTYLE_LOC
"The class library
(WIZSTYLE.VCX) needed by this
form could not be found. "+;

  "Please locate."

LOCAL
cWizHomePath,separator,cWizStyFil
e

IF TYPE('THIS.Parent') # "O"
  RETURN
ENDIF

IF SET("TALK") = "ON"
  SET TALK OFF
  this.oldTalk = "ON"
ELSE
  this.oldTalk = "OFF"
ENDIF

IF
ATC(C_WIZSTYLE.SET("CLASSLIB
")) = 0

  * Returns just the
  pathname
  cWizHomePath =
  _WIZARD
  IF \ $ cWizHomePath
  cWizHomePath =
  SUBSTR(m.cWizHomePath,1,RAT(\
,m.cWizHomePath))
  IF
  RIGHT(m.cWizHomePath,1) = \
  AND LEN(m.cWizHomePath) > 1 ;
  AND
  SUBSTR(m.cWizHomePath,LEN(m.c
  WizHomePath)-1,1) <> \
  cWizHomePath =
  SUBSTR(m.cWizHomePath,1,LEN(
  m.cWizHomePath)-1)
  ENDIF
  ELSE
  cWizHomePath = "
  ENDIF

  * Add a backslash unless
  there is one already there.
  Separator =
  IIF(_MAC, "\", "\")
  IF
  !(RIGHT(m.cWizHomePath,1) $ "\")
  AND !EMPTY(m.cWizHomePath)

```

```

        m.cWizHomePath=
m.cWizHomePath+ m.separator
    ENDIF

    DO CASE
    CASE FILE(C_ WIZSTYLE)
        cWizFile =
C_ WIZSTYLE
        CASE
FILE(m.cWizHomePath+C_ WIZSTY
LE)
            cWizFile =
m.cWizHomePath+C_ WIZSTYLE
        CASE
FILE(m.cWizHomePath+C_ WIZDIR+
C_ WIZSTYLE)
            cWizFile =
m.cWizHomePath+C_ WIZDIR+C_ WI
ZSTYLE
        CASE
FILE(HOME()+C_ WIZSTYLE)
            cWizFile =
HOME()+C_ WIZSTYLE
        CASE
FILE(HOME()+C_ WIZDIR+C_ WIZST
YLE)
            cWizFile =
HOME()+C_ WIZDIR+C_ WIZSTYLE
        OTHERWISE
            =MESSAGEBOX(E_ NOST
YLE_ LOC)
            cWizFile =
GETFILE("VCX",C_ PROMPT1_ LOC
+C_ WIZSTYLE)
        ENDCASE

        IF
ATC(C_ WIZSTYLE,m.cWizFile)#0
            SET CLASS TO
(m.cWizFile) ADDITIVE
        ELSE
            * Failed to get
WIZSTYLE.VCX file
            RETURN .F.
        ENDIF

    ENDIF

this.inivars()

ENDPROC
PROCEDURE Error
PARAMETERS nError, cMethod,
nLine
LOCAL aFoxErr,nTotErr
DIMENSION aFoxErr[1]
nTotErr = AERROR(aFoxErr)

DO CASE
CASE INLIST(m.nError,1733,1734)
&&property not found -- traps
SETALL()
RETURN
CASE m.nError=1938 &&no parent
RETURN
CASE nTotErr>0 AND aFoxErr[1,1] =
1420
    * Corrupt Ole object in
General field.
            =MESSAGEBOX(aFoxErr[
1,2])
            RETURN
        ENDCASE

**** Error Dialog *****
=MESSAGEBOX(MESSAGE(1)+CHR
R(13)+;
"Error: "+STR(nError)+CHR(13)+;
MESSAGE()+CHR(13)+;
"Method: "+cMethod+CHR(13)+;
"Line: "+STR(nLine))

RETURN TO MASTER

ENDPROC
PROCEDURE Destroy
* Restore various settings
LOCAL
nTablesUsed,aTablesUsed,i,nDECur
sors,aDECursors,cDataEnvRef
DIMENSION aTablesUsed[1]

IF TYPE('THIS.Parent') # "O"
RETURN
ENDIF

IF TYPE("THIS.oldTalk") = "C" AND
THIS.oldTalk="ON"
SET TALK ON
ENDIF

* OLE Servers can still send data
back to General fields
* even though they are not in Edit
Mode. We need to reset
* buffering to 1 so the buffer is not
updated by the OLE Server.
* Also, folks might exit out while
editing.
IF THIS.usedataenv
DIMENSION
aDECursors[1]
DO CASE
CASE
TYPE("THISFORM.DataEnvironment
") = "O"
nDECursors =
AMEMBERS(aDECursors,THISFOR
M.DataEnvironment,2)
cDataEnvRef =
"THISFORM.DataEnvironment"
CASE
TYPE("THISFORMSET.DataEnviron
ment") = "O"
nDECursors =
AMEMBERS(aDECursors,THISFOR
MSET.DataEnvironment,2)
cDataEnvRef =
"THISFORMSET.DataEnvironment"
ENDCASE
FOR i = 1 TO
m.nDECursors
WITH
EVAL(m.cDataEnvRef + "." +
aDECursors[m,i])
IF
USED(.ALIAS) AND
ATC("CURSOR",.BaseClass)#0 AND
;
CursorGetProp("sourcetype",.ALIAS)
=3 AND ;
CursorGetProp("buffering",.ALIAS)>1
=TableRevert(.T..ALIAS)
=CursorSetProp("buffering"
,1,.ALIAS) &&optimistic
table buffering
ENDIF
ENDWITH
ENDFOR
ENDIF

* Skip if using preview mode
IF THIS.previewmode
RETURN
ENDIF

IF THIS.oldSetDelete = "OFF"
SET DELETED OFF
ENDIF
SET REPROCESS TO
THIS.oldreprocess
SET MESSAGE TO
SELECT (THIS.nworkarea)

IF THIS.usedataenv
RETURN
ENDIF

* The following code is here to
support
* forms not using a
DataEnvironment.
m.nTablesUsed =
AUSED(aTablesUsed)
FOR i = 1 TO m.nTablesUsed
IF
USED(aTablesUsed[m,i,1]) AND
ATC(".TMP",DBF(aTablesUsed[m,i,1
]))=0 &&skip for views
=CursorSetProp("buffering"
,THIS.oldBuffering,aTablesUsed[m,i,
1]) &&optimistic table buffering
ENDIF
ENDFOR

IF THIS.oldMultiLocks = "OFF"
SET MULTILOCKS OFF
ENDIF

IF THIS.oldSetFields = "ON"
SET FIELDS ON
ENDIF

SET REFRESH TO THIS.oldrefresh

ENDPROC
Top = 2
Left = 0
Height = 24
Width = 49
FontName = "MS Sans Serif"
FontSize = 8
Caption = "<Find"
Enabled = .T.

```

```

StatusBarText = "Search for records"
TabIndex = 5
ToolTipText = ""
Name = "cmdFind"
PROCEDURE Click

```

```

LOCAL
IVisChange,IStateChange,oSearchDi
og

```

```

* Check if SDI Window
IF THISFORM.ShowWindow = 2
  IF !_VFP.Visible
    _VFP.Visible =

```

```

.T.
  IVisChange = .T.
  ENDIF
  IF !_SCREEN.WindowState

```

```

= 1
    _SCREEN.WindowState =

```

```

0
    IStateChange =

```

```

.T.
  ENDIF
  ENDIF
ENDIF

```

```

oSearchDlg =
CREATE("searchform")
oSearchDlg.SHOW()
thisform.REFRESH()

```

```

IF m.IVisChange
  _VFP.Visible = .F.
ENDIF
IF m.IStateChange
  _SCREEN.WindowState =

```

```

1
ENDIF
IF THISFORM.ShowWindow = 2
  Activate Window
(THISFORM.Name)
ENDIF

```

```

ENDPROC
Top = 2
Left = 58
Height = 24
Width = 49
FontName = "MS Sans Serif"
FontSize = 8
Caption = "Print"
Enabled = .T.
StatusBarText = "Print to report"
TabIndex = 6
ToolTipText = ""
Name = "cmdPrint"
PROCEDURE Click
#DEFINE C_MAKEREPO_LOC

```

```

"Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC

```

```

"Error opening
table. Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC

```

```

"Select a report to print:"

```

```

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

```

```

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

```

```

#IF 0
* Handling for Private data
sessions
IF m.nSaveSess # 1
  SET
  DATASESSION TO 1
  SELECT 0
  IF
  !EMPTY(m.cSaveData)
  OPEN
  DATABASE (m.cSaveData)
  ENDIF
  IF
  USED(m.cSaveAlias)
  SELECT (m.cSaveAlias)
  IF
  CURSORGETPROP("SourceName")
  #m.cSaveSource
  cDiffSource =
  CURSORGETPROP("SourceName")
  USE IN (m.cSaveAlias)
  SELECT 0
  ENDIF
  IF
  EMPTY(ALIAS())
  USE
  (m.cSaveSource) AGAIN ALIAS
  (m.cSaveAlias) SHARED
  IF
  EMPTY(ALIAS())
  =MESSAGEBOX(C_NOO
  PEN_LOC)
  RETURN
  ENDIF
  ENDIF
  ENDIF
  #ENDIF
  IF FILE(m.cRepName)
  REPORT FORM
  (m.cRepName) PREVIEW NOWAIT
  ELSE
  m.cRepName =
  GETFILE("frx",C_GETFILEPROMPT
  _LOC,"",1)
  IF !EMPTY(m.cRepName)
  IF
  FILE(m.cRepName)

```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```

* User
pressed Open button

```

```

REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE

```

```

* User
pressed New button
DO
(_WIZARD) WITH "AUTOREPORT"
ENDIF
ENDIF
ENDIF

```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```


```

```

MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
APPEND BLANK
thisform.REFRESH()
ENDPROC
Top = 2
Left = 178
Height = 24
Width = 53
FontName = "MS Sans Serif"
FontSize = 8
Caption = "\<Delete"
Enabled = .T.
StatusBarText = "Delete existing
record"
TabIndex = 9
ToolTipText = ""
Name = "cmdDelete"
PROCEDURE Click
#DEFINE MSGBOX_YES
6
#DEFINE C_MSGBOX1
36
#DEFINE C_DELETE_LOC "Do
you want to delete this record?"

IF
MESSAGEBOX(C_DELETE_LOC,C
_MSGBOX1) = MSGBOX_YES
Delete
IF
THIS.Parent.updaterows()
&&success
* Success
IF !EOF()
Skip 1
ENDIF
IF EOF() AND
!BOF()
Skip -1
ENDIF
ENDIF
ENDIF
thisform.REFRESH()
IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
Width = 220
Height = 120
Name = "fancyole"
Label1.Name = "Label1"
Top = 12
Left = 60
Height = 100
Width = 145
Name = "Oleboundcontrol1"
Width = 149
Height = 25
BackStyle = 0
BorderWidth = 0
Name = "shadowbase"
Top = 4
Left = 51

```

```

Height = 20
Width = 97
BackStyle = 0
BorderStyle = 0
BorderWidth = 0
FillStyle = 0
BackColor = 0,255,255
FillColor = 128,128,128
Name = "Shape1"
FontBold = .T.
FontName = "Arial"
FontSize = 8
Alignment = 1
BackStyle = 0
Caption = "Label1"
Height = 16
Left = 0
Top = 3
Width = 41
TabIndex = 0
Name = "Label1"
DoCreate = .T.
BackColor = 0,128,128
wizfield = fancyfield
wizlabel =
wizblcap = lower
wizblsuffix = ("")
wizlogic = fancylogic
wizmemo = fancymemo
wizole = fancyole
wizcboxl = .F.
wizblspace =
wizmaxcharid = 40
Name = "fancyform"
layoutsty.Shape1.Top = 0
layoutsty.Shape1.Left = 0
layoutsty.Shape1.Height = 228
layoutsty.Shape1.Width = 538
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Top = 7
layoutsty.Shape2.Left = 6
layoutsty.Shape2.Height = 13
layoutsty.Shape2.Width = 37
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Top = 27
layoutsty.Shape3.Left = 6
layoutsty.Shape3.Height = 13
layoutsty.Shape3.Width = 37
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Top = 6
layoutsty.Shape4.Left = 111
layoutsty.Shape4.Height = 13
layoutsty.Shape4.Width = 37
layoutsty.Shape4.Name = "Shape4"

layoutsty.Top = 48
layoutsty.Left = 21
layoutsty.Width = 538
layoutsty.Height = 228
layoutsty.Name = "layoutsty"
Label1.FontName = "Tahoma"
Label1.FontSize = 18
Label1.Height = 31
Label1.Left = 21
Label1.Top = 10
Label1.Width = 78
Label1.Name = "Label1"
Shape1.Top = 279
Shape1.Left = 5
Shape1.Height = 1

```

```

Shape1.Width = 569
Shape1.BackColor = 255,255,255
Shape1.BorderColor = 255,255,255
Shape1.Name = "Shape1"
Shape2.Top = 43
Shape2.Left = 5
Shape2.Height = 1
Shape2.Width = 569
Shape2.Name = "Shape2"
Shape3.Top = 314
Shape3.Left = 5
Shape3.Height = 1
Shape3.Width = 569
Shape3.BackColor = 255,255,255
Shape3.BorderColor = 255,255,255
Shape3.Name = "Shape3"
Shape4.Top = 5
Shape4.Left = 5
Shape4.Height = 0
Shape4.Width = 569
Shape4.BackStyle = 0
Shape4.SpecialEffect = 0
Shape4.Name = "Shape4"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
Width = 184
Height = 34
SpecialEffect = 0
Name = "fancybase"
FontName = "Tahoma"
FontSize = 9
Caption = "Label1"
Height = 18
Left = 8
Top = 9
Width = 43
Name = "Label1"
FontName = "MS Sans Serif"
FontSize = 8
Alignment = 3
Height = 22
SpecialEffect = 1
Width = 100
ColorSource = 0
Name = "standardfield"
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
ColumnCount = 1
AllowHeaderSizing = .F.
AllowRowSizing = .F.
DeleteMark = .F.
HeaderHeight = 0
Height = 178
Highlight = .F.
HighlightRow = .F.
RecordMark = .F.
RowHeight = 300
ScrollBars = 2
SplitBar = .F.
Width = 369
BackColor = 192,192,192
wizbtnstyle = gridbtns
Name = "scrollgrid"
Column1.Width = 500
Column1.Sparse = .F.
Column1.BackColor = 192,192,192
Column1.Name = "Column1"
Caption = "Header1"
Name = "Header1"
Top = 23
Left = 15

```


Width = 200
 Height = 200
 BackStyle = 1
 Name = "Container1"
 Arial, 0, 9, 5, 15, 12, 13, 3, 0
 DataSession = 1
 ScaleMode = 3
 Height = 320
 Width = 580
 DoCreate = .T.
 AutoCenter = .T.
 Caption = "Form1"
 Enabled = .T.
 wizlayout =
 wizfield =
 wizmemo =
 wizlogic = 0
 wizole =
 wizmaxcharfld = 0
 wizblsuffix = 0
 wizblcap = 0
 wizformstretch = 0
 wizbldefwid = 0
 wiztitle =
 wizuser = 0
 wizgrid =
 wizverify = 0
 wizbuttons = 0
 wizbtnpos = 0
 wizcodestyle = 0
 wizbtnlayout = 0
 wizcaptions = .T.
 wizbuffering = 5
 wizlabel =
 wizblspace = 0
 wizpages = 1
 wizpagestyle =
 Name = "baseform"
 Arial, 0, 9, 5, 15, 12, 13, 3, 0
 ScaleMode = 3
 DoCreate = .T.
 ColorSource = 0
 BackColor = 192,192,192
 wizlayout = layoutsty
 wizfield = boxfield
 wizmemo = boxmemo
 wizlogic = boxlogic
 wizole = boxole
 wizmaxcharfld = 45
 wizblsuffix = (":")
 wizblcap = proper
 wizformstretch = .T.
 wizbldefwid = .F.
 wiztitle = Label1
 wizuser = 0
 wizgrid = grid1
 wizverify = .T.
 wizcodestyle = .T.
 wizpages = 2
 wizpagestyle = boxpages
 Name = "boxform"
 Top = 62
 Left = 5
 Width = 571
 Height = 195
 TabIndex = 0
 Name = "Layoutsty"
 Shape1.Top = 0
 Shape1.Left = 0
 Shape1.Height = 194

Shape1.Width = 570
 Shape1.Name = "shape1"
 Shape2.Top = 1
 Shape2.Left = 16
 Shape2.Height = 13
 Shape2.Width = 37
 Shape2.Name = "shape2"
 Shape3.Top = 13
 Shape3.Left = 16
 Shape3.Height = 13
 Shape3.Width = 37
 Shape3.Name = "shape3"
 Shape4.Top = 1
 Shape4.Left = 52
 Shape4.Height = 13
 Shape4.Width = 37
 Shape4.Name = "shape4"
 Top = 0
 Left = 0
 Height = 61
 Width = 580
 FillStyle = 0
 BackColor = 0,0,255
 FillColor = 0,0,255
 Name = "Shape1"
 AutoSize = .T.
 FontName = "Arial"
 FontSize = 20
 BackStyle = 0
 Caption = "Label1"
 Height = 35
 Left = 21
 Top = 17
 Width = 83
 ForeColor = 255,255,255
 BackColor = 255,255,255
 Name = "Label1"
 Top = 261
 Left = 0
 Height = 61
 Width = 580
 FillStyle = 0
 BackColor = 0,0,255
 FillColor = 0,0,255
 Name = "Shape2"
 Arial, 0, 9, 5, 15, 12, 13, 3, 0
 ScaleMode = 3
 DoCreate = .T.
 ColorSource = 0
 BackColor = 192,192,192
 wizlayout = layoutsty
 wizfield = chiselfield
 wizmemo = chiselmemo
 wizlogic = chisellogic
 wizole = chisole
 wizmaxcharfld = 45
 wizblsuffix = (":")
 wizblcap = proper
 wizformstretch = .T.
 wizbldefwid = .T.
 wiztitle = Label1
 wizuser = 0
 wizgrid = grid1
 wizverify = .T.
 wizbtnpos = 0
 wizcodestyle = .T.
 Name = "chiselform"
 Top = 53
 Left = 2
 Width = 573

Height = 213
 TabIndex = 0
 ZOrderSet = 0
 odimensions = Shape1
 Name = "Layoutsty"
 Shape1.Top = 0
 Shape1.Left = 0
 Shape1.Height = 212
 Shape1.Width = 572
 Shape1.Name = "shape1"
 Shape2.Top = 7
 Shape2.Left = 16
 Shape2.Name = "shape2"
 Shape3.Top = 22
 Shape3.Left = 16
 Shape3.Name = "shape3"
 Shape4.Top = 7
 Shape4.Left = 255
 Shape4.Name = "shape4"
 Top = 270
 Left = 0
 Height = 2
 Width = 584
 ZOrderSet = 1
 Name = "Shape2"
 Top = 0
 Left = 0
 Height = 50
 Width = 580
 FillStyle = 0
 FillColor = 0,0,0
 BorderColor = 0,0,0
 ZOrderSet = 2
 Name = "Shape3"
 AutoSize = .T.
 FontSize = 20
 BackStyle = 0
 Caption = "Label1"
 Height = 35
 Left = 19
 Top = 10
 Width = 83
 ForeColor = 255,255,255
 BackColor = 0,0,0
 ZOrderSet = 3
 Name = "Label1"
 Arial, 0, 9, 5, 15, 12, 13, 3, 0
 ScaleMode = 3
 DoCreate = .T.
 ColorSource = 0
 BackColor = 192,192,192
 wizlayout = layoutsty
 wizfield = shadowfield
 wizmemo = shadowmemo
 wizlogic = shadowlogic
 wizole = shadowole
 wizmaxcharfld = 45
 wizblsuffix = (":")
 wizblcap = proper
 wizformstretch = .T.
 wizbldefwid = .T.
 wiztitle = Label1
 wizuser = 0
 wizgrid = grid1
 wizverify = .T.
 wizbtnpos = 0
 wizcodestyle = .T.
 Name = "shadowform"
 Top = 54
 Left = 7

Width = 564
 Height = 205
 TabIndex = 0
 odimensions = Shape1
 Name = "Layoutsty"
 Shape1.Top = 0
 Shape1.Left = 0
 Shape1.Height = 205
 Shape1.Width = 563
 Shape1.Name = "shape1"
 Shape2.Top = 7
 Shape2.Left = 16
 Shape2.Height = 13
 Shape2.Width = 37
 Shape2.Name = "shape2"
 Shape3.Top = 22
 Shape3.Left = 16
 Shape3.Height = 13
 Shape3.Width = 37
 Shape3.Name = "shape3"
 Shape4.Top = 7
 Shape4.Left = 252
 Shape4.Height = 13
 Shape4.Width = 37
 Shape4.Name = "shape4"
 Top = 48
 Left = 0
 Height = 2
 Width = 584
 Name = "Shape1"
 Top = 264
 Left = 2
 Height = 2
 Width = 584
 Name = "Shape2"
 AutoSize = .T.
 FontBold = .T.
 FontSize = 14
 BackStyle = 0
 Caption = "Label1"
 Height = 25
 Left = 24
 Top = 12
 Width = 64
 Name = "Label1"
 Arial, 0, 9, 5, 15, 12, 13, 3, 0
 ScaleMode = 3
 DoCreate = .T.
 wizlayout = layoutsty
 wizfield = embossedfield
 wizmemo = embossedmemo
 wizlogic = embossedlogic
 wizole = embossedole
 wizmaxcharfld = 45
 wizblsuffix = (":")
 wizblcap = proper
 wizformstretch = .T.
 wizbldefwid = .T.
 wiztitle = Label1
 wizuser = 0
 wizgrid = grid1
 wizverify = .T.
 wizbtnpos = 0
 wizcodestyle = .T.
 wizcboxbl = .T.
 wizlabel = embossedlabel
 Name = "embossedform"
 Top = 54
 Left = 4
 Width = 571

Height = 205
 TabIndex = 0
 Name = "Layoutsty"
 Shape1.Top = 0
 Shape1.Left = 0
 Shape1.Height = 205
 Shape1.Width = 570
 Shape1.Name = "shape1"
 Shape2.Top = 7
 Shape2.Left = 6
 Shape2.Height = 13
 Shape2.Width = 37
 Shape2.Name = "shape2"
 Shape3.Top = 22
 Shape3.Left = 6
 Shape3.Height = 13
 Shape3.Width = 37
 Shape3.Name = "shape3"
 Shape4.Top = 7
 Shape4.Left = 260
 Shape4.Height = 13
 Shape4.Width = 37
 Shape4.Name = "shape4"
 AutoSize = .T.
 FontBold = .T.
 FontName = "Times New Roman"
 FontSize = 20
 BackStyle = 0
 Caption = "Label1"
 Height = 34
 Left = 12
 Top = 8
 Width = 83
 Name = "Label1"
 Top = 42
 Left = 5
 Height = 1
 Width = 568
 BackColor = 128,128,128
 BorderColor = 128,128,128
 Name = "Shape1"
 Top = 43
 Left = 5
 Height = 1
 Width = 568
 BackColor = 255,255,255
 BorderColor = 255,255,255
 Name = "Shape2"
 Top = 264
 Left = 6
 Height = 1
 Width = 568
 BackColor = 128,128,128
 BorderColor = 128,128,128
 Name = "Shape3"
 Top = 265
 Left = 6
 Height = 1
 Width = 568
 BackColor = 255,255,255
 BorderColor = 255,255,255
 Name = "Shape4"
 Arial, 0, 9, 5, 15, 12, 13, 3, 0
 Width = 184
 Height = 34
 Name = "fancyfield"
 Label1.Name = "Label1"
 FontName = "Tahoma"
 Height = 23
 Left = 59

Top = 5
 Width = 113
 DisabledForeColor = 0,0,0
 NullDisplay = ("n/a")
 Name = "Text1"
 Tahoma, 0, 9, 5, 14, 12, 17, 2, 0
 Width = 88
 Height = 34
 Name = "fancylogic"
 Label1.Name = "Label1"
 Top = 8
 Left = 56
 Height = 18
 Width = 17
 Caption = ""
 DisabledForeColor = 0,0,0
 Name = "Check1"
 Arial, 0, 9, 5, 15, 12, 13, 3, 0
 Width = 240
 Height = 90
 Name = "fancymemo"
 Label1.Name = "Label1"
 FontName = "Tahoma"
 Height = 78
 Left = 54
 Top = 5
 Width = 177
 ForeColor = 0,0,0
 DisabledForeColor = 0,0,0
 IntegralHeight = .T.
 Name = "Edit1"
 Tahoma, 0, 9, 5, 14, 12, 17, 2, 0
 Width = 452
 Height = 160
 BackStyle = 0
 TabIndex = 1
 BackColor = 192,192,192
 remot delimiter = (.F.)
 Name = "searchclass"
 PROCEDURE searchexpr
 LOCAL
 cGetExpr1,cGetExpr2,cJoin,cGetExp
 r

 m.cGetExpr1 =
 THIS.SearchItem(THIS.cboFields1,T
 HIS.cboOperators1,THIS.txtExpr1)
 m.cGetExpr2 =
 THIS.SearchItem(THIS.cboFields2,T
 HIS.cboOperators2,THIS.txtExpr2)
 m.cJoin =
 IIF(THIS.optGrpAndOr.value = 2,"
 OR "," AND ")

 DO CASE
 CASE EMPTY(m.cGetExpr1) AND
 EMPTY(m.cGetExpr2)
 m.cGetExpr = ""
 CASE EMPTY(m.cGetExpr2)
 m.cGetExpr =
 m.cGetExpr1
 CASE EMPTY(m.cGetExpr1)
 m.cGetExpr =
 m.cGetExpr2
 OTHERWISE
 m.cGetExpr =
 m.cGetExpr1+m.cJoin+m.cGetExpr2
 ENDCASE

 RETURN m.cGetExpr

```

ENDPROC
PROCEDURE dataexpr
LPARAMETER cDataType,cFldExpr

LOCAL cTmpExpr

DO CASE
CASE
INLIST(m.cDataType,"M","G","P","O",
,"U")
RETURN ""

CASE m.cDataType = "C"
IF
TYPE("Test'="+m.cFldExpr) # "L"
IF
THIS.remotedelimiter

cTmpExpr =
""+m.cFldExpr+"" ELSE

cTmpExpr =
["'+m.cFldExpr+'"]
ENDIF
ELSE
cTmpExpr =
m.cFldExpr
ENDIF

* Check for case sensitive
IF
THIS.chkCaseSensitive.Value = 0
m.cTmpExpr=
"UPPER("+m.cTmpExpr+)"
ENDIF
RETURN m.cTmpExpr

CASE
INLIST(m.cDataType,"N","F","I","Y",
"B")
* Check for any commas
and remove
RETURN
ALLTRIM(STR(VAL(STRTRAN(m.cF
ldExpr,","),16,4))

CASE INLIST(m.cDataType,"D","T")
RETURN
"{'+CHRTRAN(m.cFldExpr,"{}",'"')+"}"

OTHERWISE
RETURN ""

ENDCASE

ENDPROC
PROCEDURE SearchItem
LPARAMETERS oField,oOp,oExpr
LOCAL
cExpr,cDataType,cOp,cFldName,cFl
dExpr,cRetExpr,aExprs,nTotExprs,i

* Check to make sure proper
parameters passed
IF TYPE("m.oField")#"O" OR
TYPE("m.oOp")#"O" OR
TYPE("m.oExpr")#"O"
RETURN ""

```

```

ENDIF

m.cFldName =
ALLTRIM(m.oField.Value)
m.cFldExpr =
ALLTRIM(m.oExpr.Value)

* If empty expression return empty.
IF EMPTY(m.cFldExpr) AND
!INLIST(m.oOp.listitemid,5,6)
RETURN ""
ENDIF

* Get data type of field
m.cDataType =
aWizFList(m.oField.listitemid,2)

IF m.cDataType = "C" AND
THIS.chkCaseSensitive.Value = 0
m.cFldName =
"UPPER("+m.cFldName+)"
ENDIF

C:\COBRATEL\CPT0100.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú
transferencia de archivos."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializar
thisform.bs.Option1.value=valor1
thisform.bs.Option2.value=valor2
DO case
CASE valor1=1
thisform.Command1.visible=.T.
thisform.Command2.visible=.T.
thisform.Command3.visible=.F.
thisform.command4.visible=.F.
CASE valor2=1
thisform.Command1.visible=.F.
thisform.Command2.visible=.F.
thisform.Command3 .isible=.T.
thisform.command4.visible=.T.
ENDCASE
thisform.Label1.visible=.T.
thisform.Label2.visible=.T.
thisform.Label3.visible=.F.
thisform.Label4.visible=.F.
thisform.refresh

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Transferencia de
archivos"
Height = 21
Left = 100
Top = 40
Width = 290
ForeColor = 255,255,255
Name = "Label10"

```

```

AutoSize = .T.
Top = 320
Left = 650
Height = 59
Width = 84
WordWrap = .T.
Picture = arw09it.ico
Caption = "Menú principal"
StatusBarText = "Regresar al menú
principal."
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
* Boton Salir
thisform.release
DO form menuuprin
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
ButtonCount = 2
BackStyle = 0
Value = 1
Height = 50
Left = 200
Top = 100
Width = 300
Name = "bs"
Option1.BackStyle = 0
Option1.Caption = "Transmisión y
extracción de archivos"
Option1.Value = 1
Option1.Height = 17
Option1.Left = 5
Option1.Top = 5
Option1.Width = 255
Option1.ForeColor = 255,255,255
Option1.Name = "Option1"
Option2.BackStyle = 0
Option2.Caption = "Operaciones con
archivos vía diskette"
Option2.Height = 17
Option2.Left = 5
Option2.Top = 24
Option2.Width = 245
Option2.ForeColor = 255,255,255
Option2.Name = "Option2"
PROCEDURE Option1.Click
* Via modem
IF this.value=1 then
thisform.Command1.visible=.T.
thisform.Command2.visible=.T
thisform.Command3.visible=.F
thisform.command4.visible=.F.
thisform.Label1.visible=.T.
thisform.Label2.visible=.T.
thisform.Label3.visible=.F.
thisform.Label4.visible=.F.
ENDIF

```

```

ENDPROC
PROCEDURE Option2.Click
* Via modem
IF this.value=1 then
  thisform.Command1.visible=.F.
  thisform.Command2.visible=.F.
  thisform.Command3.visible=.T.
  thisform.Command4.visible=.T.
  thisform.Label1.visible=.F.
  thisform.Label2.visible=.F.
  thisform.Label3.visible=.T.
  thisform.Label4.visible=.T.
ENDIF

ENDPROC
AutoSize = .F.
Top = 200
Left = 200
Height = 59
Width = 80
WordWrap = .T.
Picture = files09.ico
Caption = ""
StatusBarText = "Extracción de
archivos."
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton Transferencia de archivos
thisform.release
valor1=1
valor2=0
DO form cpt0111

ENDPROC
Top = 280
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = mail03.ico
Caption = ""
StatusBarText = "Transmisión de
archivos PISA."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus

```

```

thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Procesos especiales
thisform.release
valor1=1
valor2=0
DO form cpt0112
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
Top = 200
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = bckup.ico
DownPicture =
Caption = ""
StatusBarText = "Respaldo de
archivos PISA."
TabIndex = 5
ToolTipText = ""
Name = "Command3"
PROCEDURE LostFocus
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton respaldo de archivos PISSA
thisform.release
valor1=0
valor2=1
DO form cpt0121
ENDPROC
Top = 280
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = resto.ico
Caption = ""
StatusBarText = "Cobranza por
terceros."

```

```

TabIndex = 6
ToolTipText = ""
Name = "Command4"
PROCEDURE LostFocus
thisform.Label4.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label4.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Mantenimiento de catálogos
thisform.release
valor1=0
valor2=1
DO form cpt0122
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
BackStyle = 0
Caption = "Extracción de archivos"
Height = 21
Left = 290
Top = 220
Width = 160
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Transmisión de archivos
PISA"
Height = 21
Left = 290
Top = 300
Width = 180
ForeColor = 255,255,255
Name = "Label2"
BackStyle = 0
Caption = "Respaldo de archivos
PISA"
Height = 21
Left = 290
Top = 220
Width = 170
ForeColor = 255,255,255
Name = "Label3"
BackStyle = 0
Caption = "Restauración de archivos
PISA"
Height = 21
Left = 290
Top = 300
Width = 275
ForeColor = 255,255,255
Name = "Label4"
Picture = net01.ico
Stretch = 1
BackStyle = 1
Height = 40
*****

```

Nota: La totalidad del código generado se presenta en los apéndices A (Generación de código del Back End) y B (Generación de código del Front End).

3.3 DISEÑO Y CONSTRUCCIÓN DEL FRONT END

Introducción

Las diversas aplicaciones que van a interactuar directamente con los usuarios, es decir, la parte de la aplicación que el usuario visualiza y maneja directamente, es lo que se conoce como Front End (vista final).

En el presente capítulo se describe la forma en que fue realizado el Front End del Sistema de Cobranza por Terceros, utilizando las herramientas de diseño de Visual Foxpro 5.0.

Diseño del Front End

El diseño del Front End debe basarse en los requerimientos del usuario, satisfaciendo sus necesidades y proporcionando sencillez y facilidad de uso.

Es por ello que se ha diseñado un Front End de fácil operación a través del uso del mouse o del teclado, se han utilizado botones de comando, listas desplegables (Combo Boxes) entre otras, con el fin de que el usuario realice de manera sencilla y rápida las operaciones con información (altas, cambios, consultas etc.).

Se incluyen barras de navegación estándares que contienen las opciones más comunes de operación de una base de datos como son; agregar, eliminar, modificar, avanzar, retroceder, realizar búsqueda, imprimir, salvar etc.

El Front End incluye también Ayuda a través de mensajes cortos (Tooltips) que se despliegan al apuntar sobre determinado botón de comando, zona de pantalla y barras de estado, con lo cual se amplía la información de la actividad a realizar.

El Front End se ha diseñado utilizando las ventajas que proporciona la programación orientada a objetos con la cual se permite la generación de clases propias, reutilización

de código fuente, polimorfismo en formularios (pantallas de captura) y encapsulamiento de código en objetos entre otras.

En general el Front End diseñado cumple con las siguientes características:

- Sencillez
- Facilidad de uso
- Modular
- Totalmente Visual

Construcción de un proyecto

Un proyecto contiene todos los componentes de un sistema, como son formularios, tablas de la base de datos, reportes, vistas (locales o remotas) etc.

Para construir un nuevo proyecto debe seguir el siguiente procedimiento:

1. Inicie Visual Foxpro
2. Seleccione File-New-Proyect
3. Pulse el botón de NewFile
4. Asigne un nombre
5. Asigne una carpeta o directorio donde se guardará el proyecto

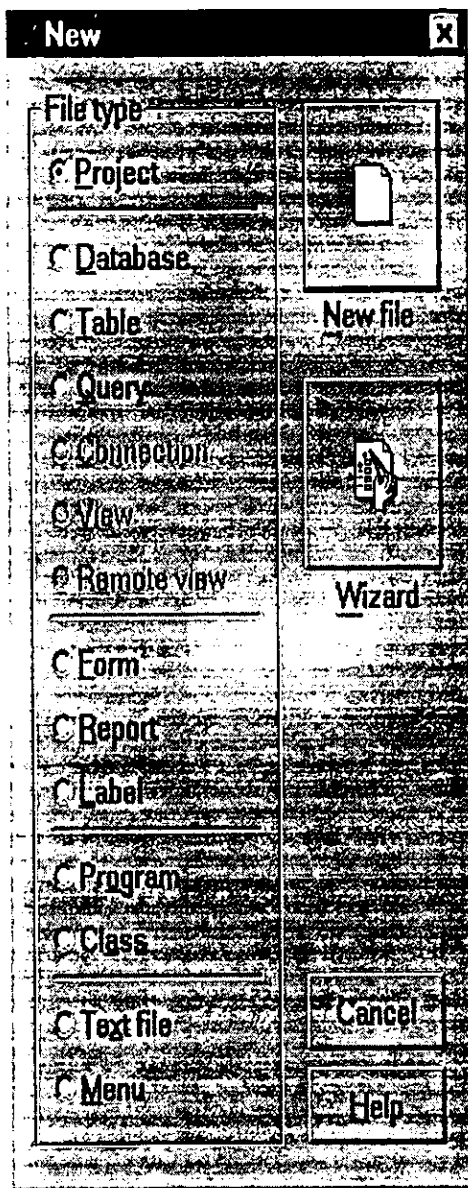


Figura 3.3.1 Pantalla New

Construcción de un formulario

Un formulario es una pantalla de captura de información, en visual Foxpro se diseñan mediante la herramienta "Form Designer" junto con la barra de herramientas del mismo "Form Controls"

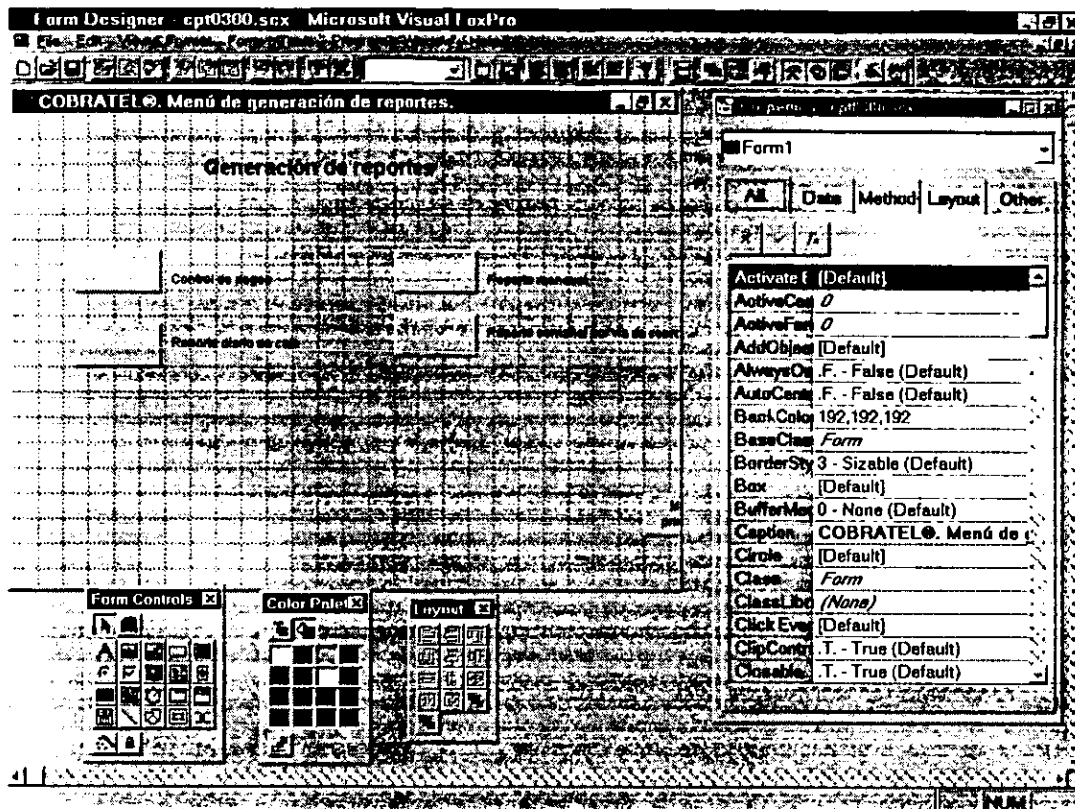


Figura 3.3.2 Pantalla Form Designer

Un formulario puede incluir campos de bases de datos, selecciones múltiples, barras de selección, folders, contenedores de información, barras de navegación, botones de comando y cualquier tipo de control que permita al usuario ejecutar alguna acción.

Utilización de Wizards (asistentes) para construir formularios

Un asistente (Wizard) es una herramienta que nos sirve para desarrollar aplicaciones, permite diseñar y construir fácilmente desde una base de datos hasta un formulario de captura con solo ir proporcionando el tipo de elemento que se quiere agregar al proyecto y seleccionando de un catálogo previamente definido la apariencia final que se desea.

Para crear un nuevo formulario utilizando Wizards debe seguir los siguientes pasos:

1. En el "Project Manager" seleccione la carpeta "Documents"
2. De click en "Forms"
3. Presione el botón "New"
4. Seleccione la tabla de la base de datos del origen del formulario
5. Elija el tipo de formulario
6. Seleccione el tipo de ordenamiento de la tabla de datos del formulario
7. Escriba el nombre que designará al formulario

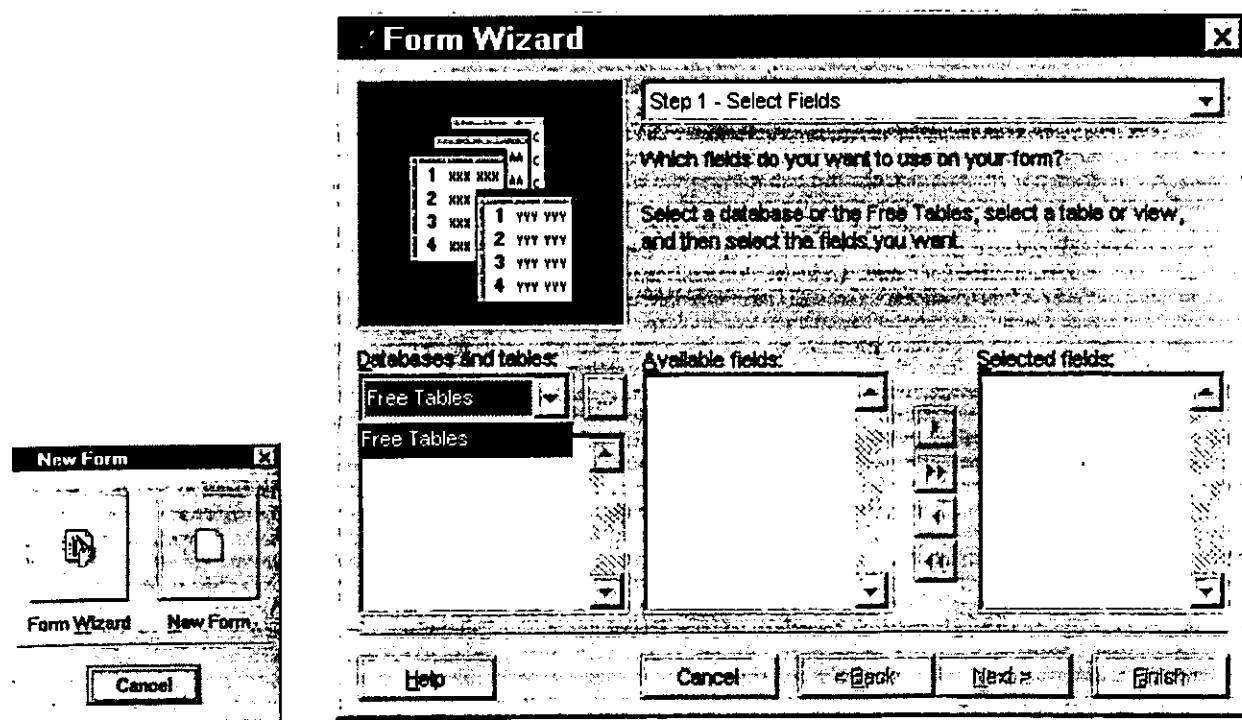


Figura 3.3.3 Pantallas New Form y Form Wizard

El asistente mostrará el nuevo formulario que se ha construido, sin embargo es muy probable que se deseen cambios, ya que el asistente solo construye versiones simples de formularios, por lo que si se desea modificar el formulario se tendrá que realizar de forma manual haciendo uso de las "herramientas de diseño" de formularios si lo que se realizará es agregar controles.

Por otra parte si se desea cambiar la apariencia del formulario o el código de las acciones que deben ser ejecutadas al ser presionado un botón de comando, se tiene que hacer uso de las "propiedades de los objetos" a cambiar.

A continuación se explica el desarrollo de las pantallas principales que integran el Sistema de Cobranza por Terceros.

- Pantalla inicial del sistema "COBRATEL ® Sistema de Cobranza por Terceros"
- Pantalla "Finaliza", salida del sistema, finalizar sesión de trabajo en el sistema.

El Front End de estas pantallas consiste en desplegar un mensaje de bienvenida y de despedida respectivamente, para lo cual se utilizaron etiquetas (Label), al dar click sobre ellas, la pantalla de inicio de sistema nos lleva a la siguiente pantalla, mientras que la pantalla de salida del sistema nos saca completamente del Sistema de Cobranza por Terceros.

El código para el diseño del Front End se realiza en base a las propiedades de las herramientas de diseño (Form Controls) utilizadas en un formulario.

El código desarrollado para este formulario en la etiqueta utilizada se presenta a continuación.

* Salir del formulario
thisform.release

no_intentos= 1
do form validar

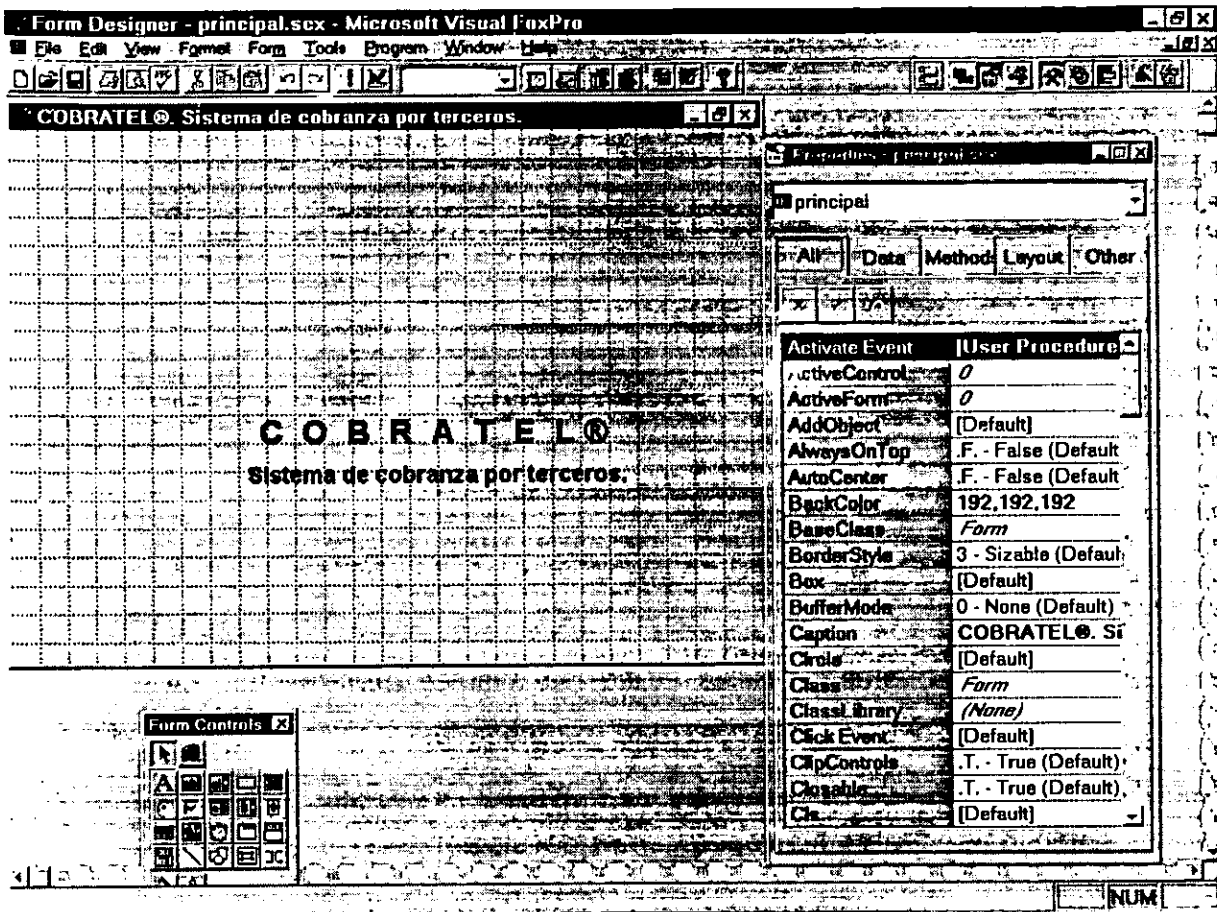


Figura 3.3.4 Pantalla COBRATEL Sistema de Cobranza por Terceros

A continuación se muestran los nombres de las pantallas que integran el menú principal del Sistema de Cobranza por Terceros y las herramientas que se utilizaron para la construcción de su Front End.

- Menú principal
- Menú transferencia de archivos
Transmisión y extracción de archivos

Operaciones con archivos vía Diskette

- Menú de procesos especiales
Menú de ajustes
- Menú de generación de reportes
Menú de control de pagos
- Mantenimiento a catálogos
- Menú utilerías
Menú Depuración
- Bitácoras

Cada una de las anteriores pantallas utiliza las mismas herramientas de diseño de formularios (Form Controls), las mismas propiedades de los objetos utilizados (Properties) y los eventos de los objetos, ya que transfieren el flujo de control a las pantallas en donde se realizan las operaciones del sistema.

De forma general el Front End para cada una de estas pantallas y en el sistema en su totalidad se ha creado con las siguientes características:

- El Front End se diseño con etiquetas (Label) y botones de comando (Command Button).
- En los botones de comando se describe la acción a realizar mediante el texto descriptivo incluido dentro del botón de comando (Caption), en algunos otros casos serán incrustadas figuras descriptivas de la acción.

- Dentro de las propiedades del objeto (StatusBarText) se ha incluido un mensaje descriptivo de cada acción a realizar y la cual es desplegada en la parte inferior de la pantalla.
- Un evento asignado a este formulario y en general a la mayoría de los formularios de el Sistema de Cobranza por Terceros es el de que al movimiento del mouse (Mouse Move) por cualquiera de los botones de comando va transfiriendo el control de flujo y esto se indica al poner la etiqueta (Label) descriptiva del botón de comando, de otro color al tomar el foco (GotFocus) y al perderlo (LostFocus).
- El orden de tabulación (Tab Order), es utilizado en la gran mayoría de los formularios de este sistema, es decir que se ha cuidado el orden que debe seguirse al dar la tecla <Tab> dentro de las pantallas del sistema.
- Todas las pantallas del sistema presentan un botón de comando con la opción de regresar al menú que les antecedió, lógicamente en los primeros módulos solo se presenta la opción de Salir.

En cada botón de comando utilizado se transfiere el control de flujo al módulo correspondiente del sistema, tomando como ejemplo los botones de utilerías y salir tenemos el siguiente código:

Botón Utilerías

```
* Boton Utilerías
thisform.release
do form cpt0500
```

Botón Salir

```
* Boton Salir
```

thisform.release

do form finaliza

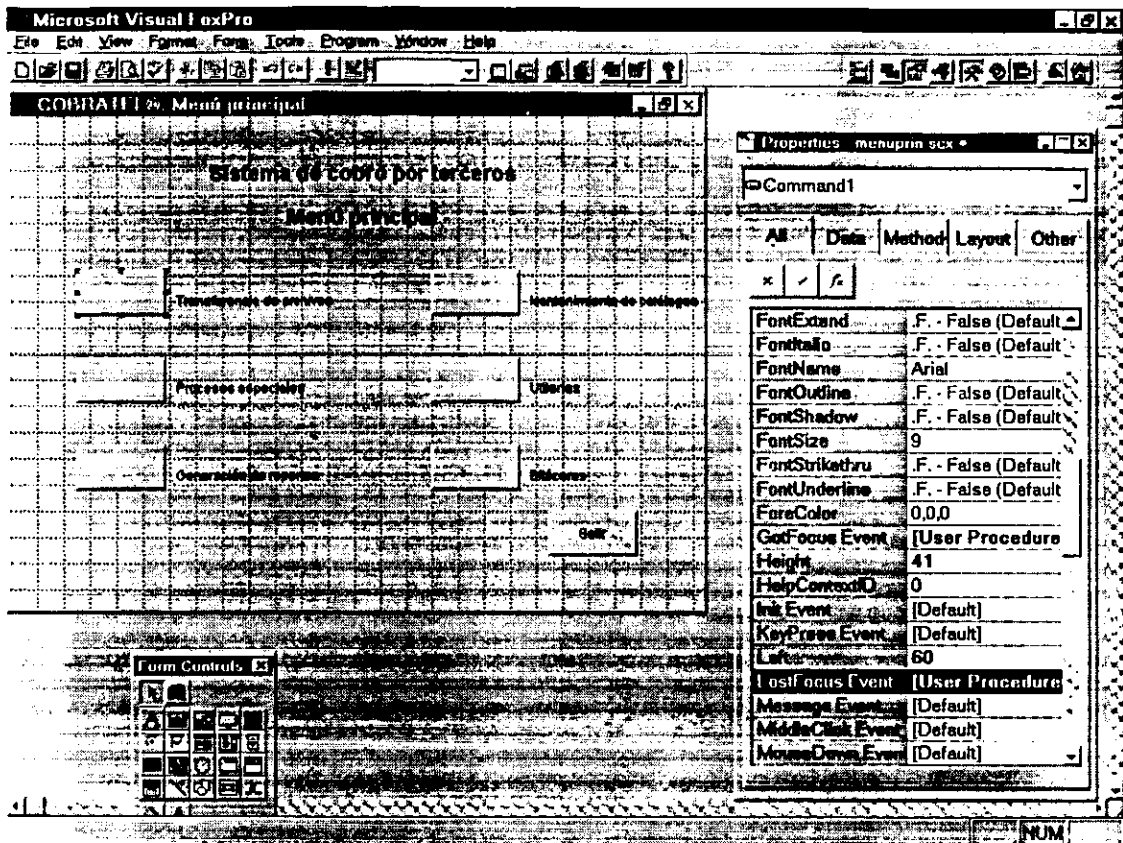


Figura 3.3.5 Pantalla Menú principal

Pantalla "Menú transferencia de archivos"

Se utilizaron para la construcción de este Front End, botones de comando (Command Button), etiquetas (Label) y se incluye un grupo de opciones (Option group).

El grupo de opciones (Option Group) se utiliza en este caso para seleccionar entre las tareas de Transmisión y extracción de archivos y la opción de Operaciones con archivos vía diskette.

En esta pantalla se utiliza una de las ventajas de visual FoxPro; El polimorfismo, esta ventaja se aplica al seleccionar una u otra opción ya que aparecen solo las posibles pantallas a las que puede transferirse el control de flujo.

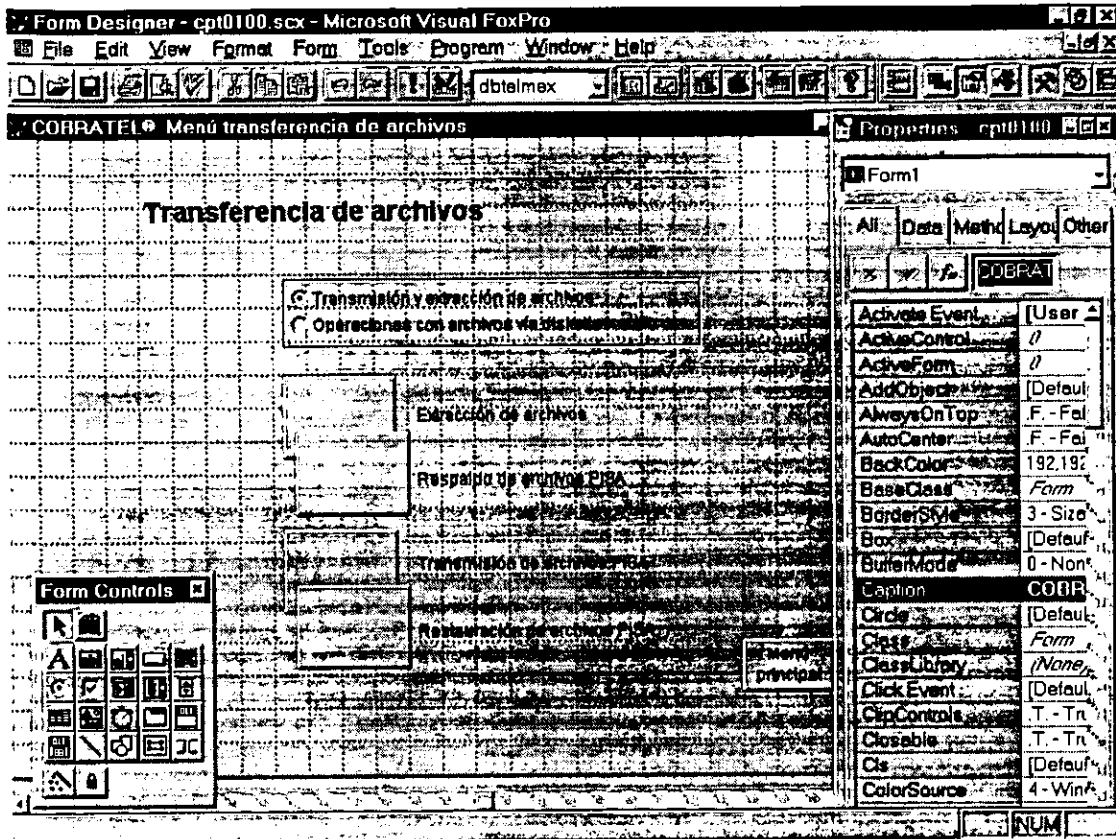


Figura 3.3.6 Pantalla de Menú transferencia de archivos.

En la ejecución del programa al seleccionar la primera opción solo aparecerán los botones de comando con las opciones de Extracción de archivos y Transmisión de archivos PISA.

Al seleccionar la segunda opción solo aparecen los botones de comando con las opciones de Respaldo de archivos PISA y Restauración de archivos PISA.

El código utilizado en el grupo de opciones es el siguiente:

Option 1

* Transmisión y extracción de archivos

```
if this.value=1 then
  thisform.command1.visible=.T.
  thisform.command2.visible=.T.
  thisform.command3.visible=.F.
  thisform.command4.visible=.F.
  thisform.label1.visible=.T.
  thisform.label2.visible=.T.
  thisform.label3.visible=.F.
  thisform.label4.visible=.F.
endif
```

Option 2

* Operaciones con archivos vía diskette

```
if this.value=1 then
  thisform.command1.visible=.F.
  thisform.command2.visible=.F.
  thisform.command3.visible=.T.
  thisform.command4.visible=.T.
  thisform.label1.visible=.F.
  thisform.label2.visible=.F.
  thisform.label3.visible=.T.
  thisform.label4.visible=.T.
endif
```

El Front End de las siguientes pantallas se realizo de manera similar, estas pantallas son las siguientes:

- Validación de entrada al sistema

- Captura de pagos de agentes
- Captura diario de agentes
- Verificación de pagos de clientes
- Reporte diario de caja
- Reporte estadístico
- Reporte por vía de cobro
- Depuración de bitácora de procesos
- Depuración de bitácora de transmisiones
- Depuración de tabla de pagos
- Depuración de tabla de cobros
- Respaldo de archivos
- Restauración de respaldos
- Cambio de password
- Consulta bitácora de procesos
- Consulta bitácora de transmisiones

Además de las características presentadas anteriormente, incluyen algunas otras, estas se presentan a continuación:

Las operaciones del sistema se realizan en un rango de fechas o en una fecha determinada, para lo cual se utilizaron a excepción de la pantalla de validación del sistema en la cual se valida la clave del usuario y el password.

En estas pantallas se utilizan dentro del formulario etiquetas (Label), cajas de texto (Text box) para los datos, como pueden ser las fechas, la clave de usuario etc.

Posteriormente se ejecuta la acción deseada dando click sobre los botones de comando (Command Button).

En el caso de la pantalla de cambio de password, se utiliza también una lista desplegable (Combo Box), al dar click sobre ella despliega la lista de usuarios del sistema.

El código utilizado en las cajas de texto utilizadas en este formulario se presenta a continuación:

Texto Box 1

*Validar que la fecha no sea mayor a la del sistema

```
if this.value > date() then
    messagebox ('La fecha que ha proporcionado no es válida. ¡Verifique por favor!',0+16+0,'Error.')
    return .f.
else
    return .t.
endif
```

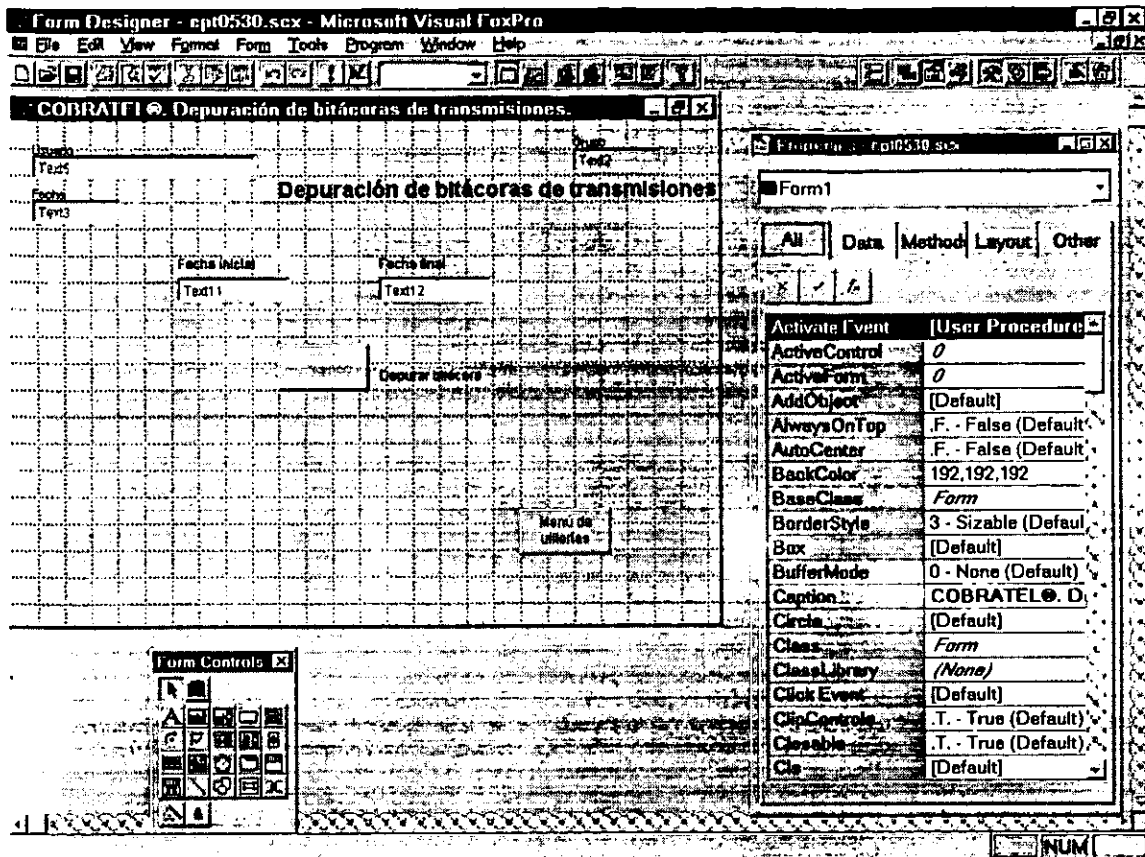


Figura 3.3.7 Pantalla Depuración de bitácora de transmisiones

Text Box 2

* Validar rango de fechas

```
if (this.value>=thisform.text11.value) .and. (this.value<=date()) then
```

```
return .t.
```

```
else
```

```
messagebox('La fecha proporcionada no es válida. ¡Verifique por favor!',0+16+0,'Error.')
```

```
return .f.
```

```
endif
```

Para la generación de reportes se utilizó el generador de reportes (Report Designer) y las herramientas de diseño de reportes (Report Controls).

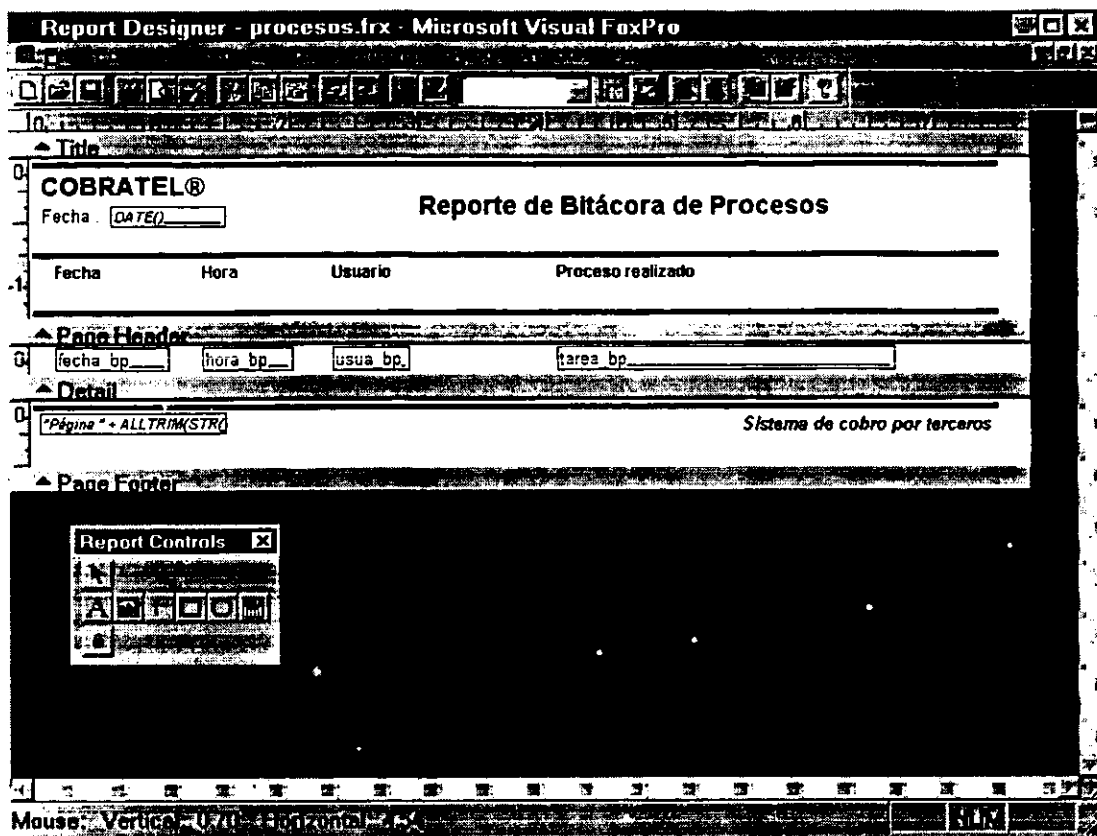


Figura 3.3.8 Pantalla Reporte de bitácora de procesos

El diseño del Front End de las siguientes pantallas del Sistema de Cobranza por terceros, tienen en común que se encargan del mantenimiento de los datos de agentes, estaciones de cobro, usuarios y grupos del sistema, estas pantallas son las siguientes:

- Mantenimiento al catálogo de agentes
- Mantenimiento al catálogo de estaciones de cobro
- Mantenimiento al catálogo de usuarios
- Mantenimiento al catálogo de grupos

En estas pantallas se realizan las altas, bajas, cambios, búsquedas e impresión de los datos de los catálogos; de agentes, de usuarios, de estaciones de cobro y de grupos.

Como ayuda para realizar estas operaciones, se creó una herramienta que está creada con botones de comando que realizan dichas operaciones, la denominaremos Barra de navegación (Navegar).

La barra de navegación, es una barra estándar de funciones comunes para operar registros de las tablas de la base de datos del sistema.

Esta barra de navegación aparece en los formularios mencionados anteriormente y en los cuales se tiene integración directa con los registros de la base de datos.

Se integra por las funciones más comunes en la operación de tablas de bases de datos.



Figura 3.3.9 Barra de navegación

De izquierda a derecha, se describe la función de cada botón de comando de la barra de navegación:

- Muestra los datos del primer registro
- Proporciona los datos del registro anterior al actual
- Proporciona los datos del siguiente registro al actual
- Muestra los datos del último registro

- Búsqueda de registros
- Imprimir reporte
- Agregar o adicionar registros a la base de datos
- Corrección de datos del registro
- Eliminar el registro actual
- Salir del formulario actual o regresar al formulario anterior

En estas pantallas se utilizan cajas de texto y etiquetas.

Se utiliza también una lista desplegable (Combo Box), al dar click sobre ella despliega la información del campo a buscar (agente, usuario, grupo y estación de trabajo) en la pantalla respectiva.

El código desarrollado en el Combo Box (lista desplegable) se muestra a continuación:

* Cuando encuentra el elemento buscado

```
go record recno()
regactual=recno()
thisform.busca.visible=.F.
thisform.navegar.visible=.T.
* Reset from prior
thisform.Refresh()
```

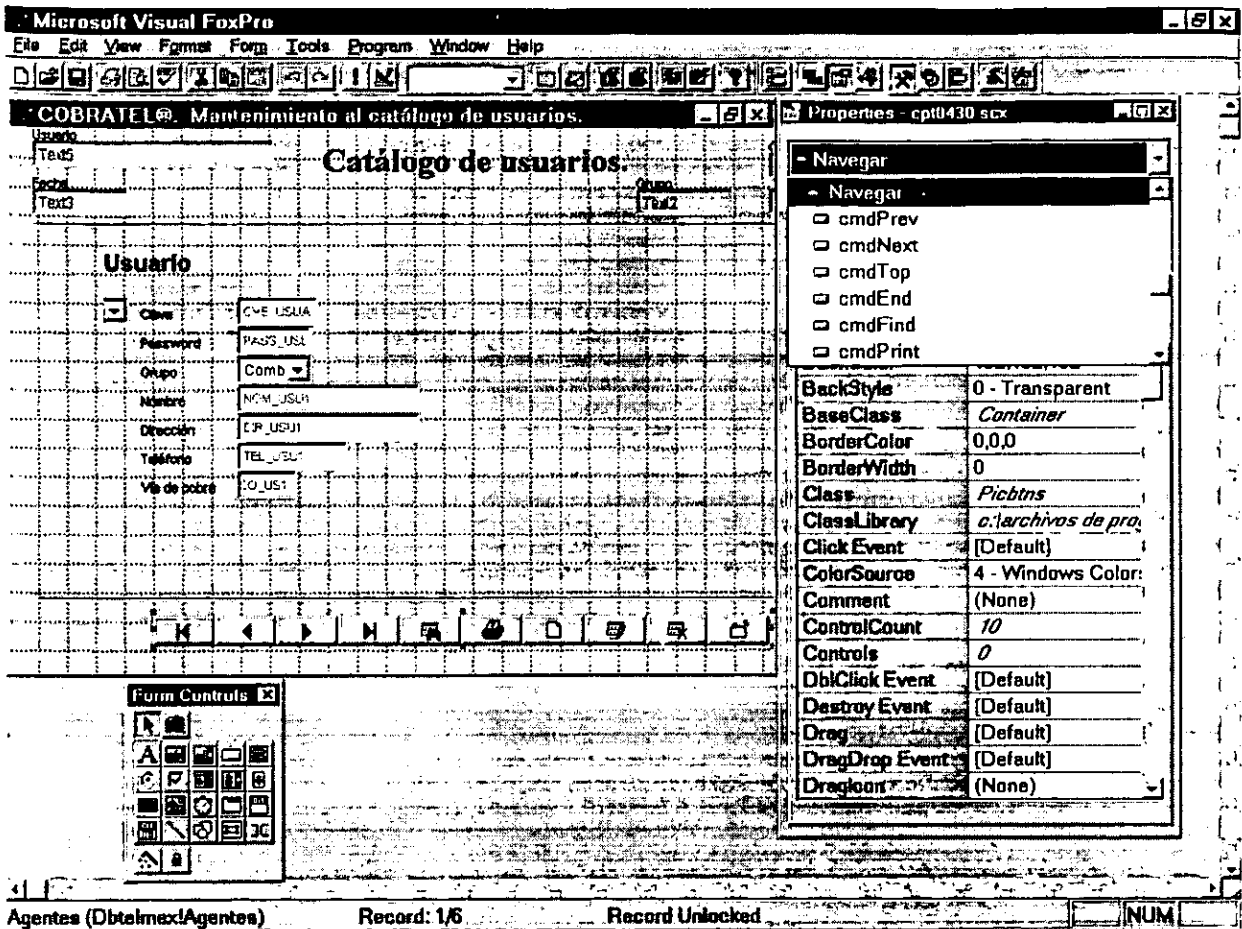


Figura 3.3.10 Pantalla Catálogo de usuarios

Tomando como ejemplo el botón del comando Borrar de la barra de navegación tenemos que el código desarrollado para el Front End, al dar click sobre el es el siguiente:

Botón Borrar

*botón borra

* Tipo de cuadro de diálogo

* 4 + 32 + 256

* 4 = Botones SI y No

- * 32 = Icono de PREGUNTA
- * 256 = el segundo botón es el de default
- * Respuestas
- * 6 = SI
- * 7 = NO

IF MESSAGEBOX('¿ Desea eliminar este registro ?', 4+32+256 , 'Confirme por favor...') = 6

DELETE

IF THIS.Parent.UpdateRows() &&success

* Success

IF !EOF()

SKIP 1

ENDIF

IF EOF() AND !BOF()

SKIP -1

ENDIF

regactual=recno()

ENDIF

THISFORM.LockScreen = .T.

THIS.Parent.ButtonRefresh()

THIS.Parent.NavRefresh()

THISFORM.LockScreen = .F.&&.....registro en bitácora

SELECT procesos

append blank

replace tarea_bp with "Baja al catálogo de usuarios"

replace usua_bp with usuario

replace fecha_bp with date()

replace hora_bp with time()

SELECT USUARIOS &&..... cerrar procesos

ENDIF


```

IF THISFORM.ShowWindow = 2
    Activate Window (THISFORM.Name)
ENDIF

```

En las pantallas siguientes:

- Respaldo de archivos
- Restauración de respaldos

Se utiliza un objeto OLE que despliega el avance en la realización de las tareas de respaldo y restauración de información.

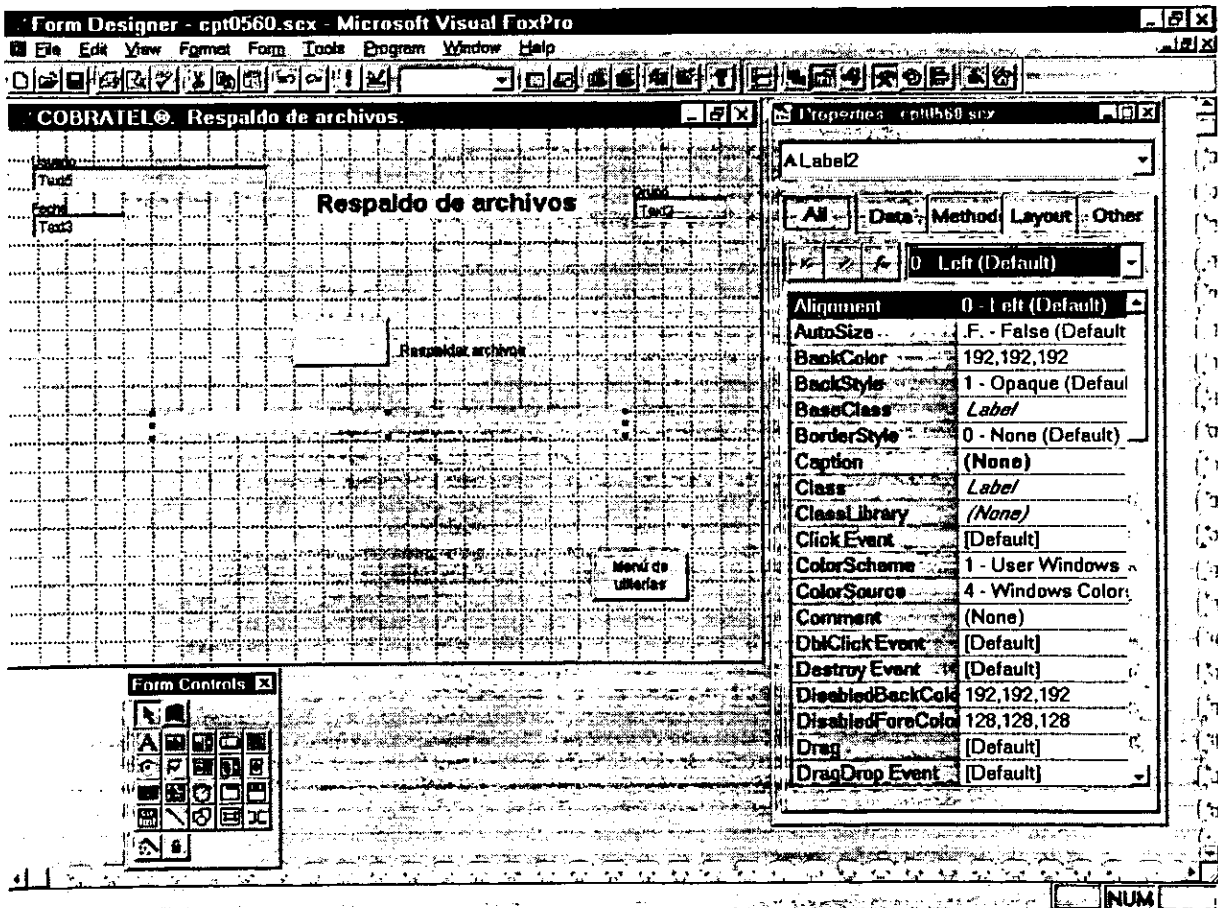


Figura 3.3.11 Pantalla respaldo de archivos

También se utilizan en estas pantallas botones de comando (Command Button), cajas de texto (Text Box) y etiquetas (Labels).

Como puede observarse, el desarrollo de un Front End en Visual Foxpro puede realizarse, hasta cierto punto de manera sencilla y rápida, además las herramientas que proporciona son bastante prácticas y todo esto en sí facilita el desarrollo del sistema.

Nota: El código generado para el diseño y construcción del Front End, se encuentra en el Apéndice B Generación de Código del Front End.

3.4 DISEÑO E IMPLEMENTACIÓN DE RUTINAS DE DIAGNOSTICO Y EVALUACIÓN DE LOS DATOS.

Introducción

Para diagnosticar y comprobar que el sistema realiza correctamente las operaciones para las cuales fue diseñado, se han desarrollado una serie de rutinas de diagnóstico para cada uno de los módulos que lo integran, así como una evaluación de los datos del Sistema de Cobranza por Terceros.

Implementación de rutinas de diagnóstico

A continuación se implementan las rutinas de diagnóstico por cada uno de los módulos siguientes:

- Módulo de Inicio "Principal"
- Módulo "Validar"
- Menú Principal
- Barra de navegación
- Transmisión y extracción de archivos
 - ◆ Extracción de archivos
 - ◆ Transmisión de archivo PISA
- Operaciones con archivos vía Diskette

- ≈ Respaldo de archivos Pisa
- ≈ Restauración de archivos PISA
- Procesos especiales
 - ◆ Regeneración de archivo PISA
 - ◆ Integración de cobranza
 - ◆ Integración de Terceros Faltantes
 - ◆ Módulo de Ajustes
 - ≈ Ajuste a la tabla de Cobros
 - ≈ Ajuste a la tabla de Pagos
- Generación de Reportes
 - ◆ Control de Pagos
 - ≈ Captura de Pagos de Agentes
 - ≈ Diario de Agente
 - ≈ Verificación de Pagos de Clientes
 - ◆ Reporte Diario de Caja
 - ◆ Reporte Estadístico

- ◆ Reporte por Vía de Cobro
- Mantenimiento de Catálogos
 - ◆ Mantenimiento al Catálogo de Agentes
 - ◆ Mantenimiento al Catálogo de Estaciones de Cobro
 - ◆ Mantenimiento al Catálogo de Usuarios
 - ◆ Mantenimiento al Catálogo de Grupos
- Utilerias
 - ◆ Depuración
 - ≈ Depuración de Bitácora de Procesos
 - ≈ Depuración de Bitácora de Transmisiones
 - ≈ Depuración de la tabla de Pagos
 - ≈ Depuración de la tabla de Cobros
 - ◆ Respaldo de Archivos
 - ◆ Restauración de Respaldos
 - ◆ Cambio de Password
- Bitácoras

- ◆ Bitácora de Procesos
- ◆ Bitácora de Transmisiones
- Salida del sistema (Finaliza)

Rutinas de diagnóstico

Es necesario mencionar que en todos los módulos las rutinas de diagnóstico se comprueban los siguientes puntos en común:

- La ejecución de la(s) operaciones(s) básicas de cada módulo.
- Cada módulo debe realizar correctamente la transferencia al o a los siguientes módulos.
- Al movimiento del mouse sobre cualquier botón de comando debe desplegarse el mensaje de la función que realiza este en la parte inferior de la pantalla.
- El control de flujo a través del movimiento del mouse, el cual debe indicarse también con el cambio en el color del texto descriptivo del botón de comando.
- El control de flujo a través de la tecla tabulador <Tab>
- Al estar el flujo de control sobre cierto botón de comando se pueda ejecutar pulsando la tecla <Enter> o dando un click sobre él con el mouse.
- Despliegue de mensajes de advertencia, error e información cuando sean necesarios.

- Todas las operaciones realizadas en el sistema como pueden ser la entrada al sistema y la transferencia de archivos deben registrarse en la bitácora del sistema.

Módulo de inicio “Principal”

(Pantalla de entrada (bienvenida) al sistema)

Al iniciar el programa aparece una pantalla de bienvenida, en donde se tendrá que dar un click en el texto que aparece (COBRATEL Sistema de cobranza por terceros) para pasar a la siguiente pantalla (Validación de entrada al sistema)

La rutina consiste en verificar que al ejecutar el programa se presente la pantalla de bienvenida y al dar click en el texto nos transfiera al módulo de Validar.

Validar

(Validación de la Clave de usuario)

En este módulo se valida la clave del usuario para su acceso al sistema COBRATEL, el usuario debe teclear su clave de acceso al sistema en lugar indicado y después dar click en el botón de comando Validar con lo que nos transfiere a la pantalla del Menú Principal, también puede digitar la tecla <Enter> con lo que se realiza la misma función.

Se verifica que solo se tengan tres posibles intentos fallidos para ingresar al sistema, en caso afirmativo el sistema presenta el siguiente mensaje “Ud. no tiene acceso al sistema” y presenta la pantalla de salida (Finaliza).

Se verifica también que en los intentos erróneos realizados se indique un mensaje descriptivo del error, el mensaje es el siguiente; “¡ Clave de usuario errónea, Intente nuevamente por favor!”.

Para terminar se comprueba que al dar click sobre el botón de comando salir nos lleve al módulo de salida del sistema (Finaliza).

Menú principal

(Presenta los módulos generales que componen el sistema)

Se comprueba en esta pantalla que cada uno de los botones de comando que aparecen transfieran a la sección indicada, los módulos son los siguientes:

- Transferencia de archivos
- Procesos especiales
- Generación de reportes
- Mantenimiento de catálogos
- Utilerias
- Bitácoras
- Salir

Barra de navegación

(Barra estándar de funciones comunes para operar tablas de la base de datos)

Esta barra de navegación aparece en varios formularios dentro del sistema en los cuales se tiene integración directa con los registros de la base de datos.

Se integra por las funciones más comunes en la operación de tablas de bases de datos.

La rutina de diagnóstico consiste en comprobar que cada botón realice la función asignada.



Figura 3.4.1 Barra de navegación

De izquierda a derecha, se describe la función de cada botón de la barra de navegación:

- Muestra los datos del primer registro
- Proporciona los datos del registro anterior al actual
- Proporciona los datos del siguiente registro al actual
- Muestra los datos del último registro
- Búsqueda de registros
- Imprimir reporte
- Agregar o adicionar registros a la base de datos
- Corrección de datos del registro
- Eliminar el registro actual
- Salir del formulario actual o regresar al formulario anterior

Transferencia de archivos

(Módulo donde se realizan todas las funciones de transmisión de archivos de cobranza de Terceros - Servidor - Servidor, PISA)

la rutina implementada comprueba que este módulo presente y transfiera a los módulos de:

- Extracción de archivos
- Transmisión de archivos PISA

Extracción de archivos.

Extracción de archivos de Terceros a Servidor

(Extracción de archivos de terceros, el sistema le solicitara los siguientes datos:
Vía de Cobro - Estación de Cobro - Nombre del archivo (sin extensión))

La rutina verifica que se realicen las siguientes operaciones:

Obtener el nombre de archivo a extraer, vía de cobro y estación de cobro.

Buscar vía de cobro y estación de cobro en el catálogo de estaciones de cobro por llave compuesta de estación de cobro y vía de cobro.

Si no existe la llave compuesta de estación de cobro en el catálogo de estaciones de cobro:

Buscar vía de cobro en el catálogo de agentes, de no existir la vía de cobro en este catálogo, desplegar el mensaje "vía de cobro inexistente" y solicitar nueva vía de cobro.

Transmisión de archivos PISA

(Módulo donde se realiza la transmisión del archivo que contiene toda la cobranza del día a PISA)

La rutina de diagnóstico para este módulo consiste en comprobar que se realicen sus operaciones básicas, esta son las siguientes:

Obtener fecha de cobranza

Buscar archivo de transacciones PISA de esa fecha, en caso de no existir archivo de esa fecha, deberá desplegarse el mensaje "No existe archivo correspondiente a esa fecha" y regresa a la pantalla Transferencia de archivos PISA.

Si el archivo de esa fecha si existe, se realizará la conexión con PISA, de no realizarse exitosamente la conexión, se presente el mensaje "No se pudo establecer conexión, desea intentar de nuevo? s/n". Si se responde con "s", se intenta establecer nuevamente la conexión con PISA, en caso de responder con "n", regresa a la pantalla de transmisión de archivos PISA.

Una vez que la conexión con PISA fue exitosa, se transfiere el archivo pisa en el directorio correspondiente de la máquina de PISA, se despliega el mensaje "Archivo exitosamente transferido"

Operaciones con archivos vía Diskette

(Módulo de contingencia que tiene como objetivo respaldar o restaurar los archivos de cobranza del Servidor para su envío a PISA)

Se comprueba que se presenten y se ejecuten correctamente los módulos de:

- Respaldo de archivos PISA
- Restauración de archivos PISA

Respaldo de archivos PISA

(Respalda archivo de la cobranza total para su envío a PISA)

Las operaciones básicas de este módulo son las siguientes:

Obtener fecha de cobranza y dirección en donde se va a respaldar.

Buscar archivo de transacciones PISA de esa fecha.

De no existir el archivo, se despliega el mensaje "No existe archivo correspondiente a esa fecha" y regresa a la pantalla anterior.

En caso de existir el archivo, se debe verificar que exista el directorio especificado y que este lista la unidad de disco, si se presenta error de unidad o directorio inexistente, se despliega el mensaje "Unidad de disco no preparada", "Directorio invalido", "Desea intentar de nuevo? s/n". Si la respuesta es "s" verifica nuevamente la unidad de disco y directorio, si la respuesta es "n" regresa a la pantalla de Respaldo de archivo PISA.

Una vez verificada la unidad de disco y directorio, se copia el archivo PISA en el directorio correspondiente o en la unidad especificada con lo que se despliega el mensaje "Archivo exitosamente respaldado".

Restauración de archivos PISA

(Restaura archivo de cobranza al servidor para integrar información a la base de datos)

Se comprueban las operaciones de este módulo:

Obtener nombre de archivo a restaurar y dirección de donde se encuentra

Verificar que exista el directorio especificado y que este lista la unidad de disco, si la unidad de disco no esta lista o el directorio no existe, se despliega el mensaje "Unidad de disco no preparada", "Directorio Invalido", "Desea intentar de nuevo s/n". Si la respuesta es "s" verifica la unidad de disco o dirección, si la respuesta es "n" regresa a la pantalla de Restaura archivo de cobranza terceros.

Se busca el archivo de cobranza en esa dirección, se verifica que el archivo se encuentre en dicha dirección, de no estar presente se despliega el mensaje "No existe archivo con ese nombre" y regresa a la pantalla anterior.

Si el archivo si esta presente en esa dirección, se copia el archivo de cobranza al directorio correspondiente y se despliega el mensaje "archivo exitosamente restaurado".

Procesos especiales

(Procesos especiales del sistema que realizan ciertas actividades extraordinarias y de ayuda para el usuario.)

Se comprueba en esta pantalla que cada uno de los botones de comando que aparecen, transfieran al módulo indicado, los módulos son los siguientes:

- Regeneración de archivo PISA
- Integración de cobranza
- Integración de terceros faltantes
- Módulo de Ajustes

Regeneración de archivo PISA

(Genera nuevamente el archivo que se envía a PISA con la información de la Base de Datos)

La rutina de diagnóstico comprueba las funciones básicas de este módulo.

Solicitud de la fecha del archivo a regenerar.

Verificamos que la fecha sea valida y que se encuentre en la base de datos. Si existe la fecha.

Procedemos a seleccionar la información.

Integramos la información seleccionada a una tabla temporal.

Generamos la salida en un archivo de tipo texto para el archivo de Pisa.

Borramos la tabla temporal.

Despliega mensaje de "Proceso Terminado".

En otro caso. Si la fecha es escrita incorrectamente despliega mensaje "fecha incorrecta, intente nuevamente".

Si no existe la fecha en la base de datos despliega el mensaje " fecha inexistente"

Integración de cobranza

(Realiza la integración de la cobranza de los terceros de un día en específico a la Base de Datos)

las operaciones que se verifican en este módulo son las siguientes:

Obtener fecha de proceso.

Validar que la fecha sea escrita correctamente.

Si la fecha es escrita incorrectamente. Despliega mensaje "fecha incorrecta".

En otro caso:

Busca archivos de cobranza de proceso.

Si no existen archivos con la fecha de proceso.

Manda el mensaje "información inexistente".

En otro caso:

Carga de uno de los archivos en tablas temporales donde se realizará la validación de los datos.

Si todos los datos son correctos se integran a la tabla de cobros así como a la tabla de pagos.

En caso contrario:

Se mandan los archivos realizados al directorio de rechazos

Integración de terceros faltantes

(Una vez recuperados los archivos de los terceros por la opción extracción de archivos de terceros a servidor se procede a integrar la información a la Base de Datos)

La rutina de diagnóstico consiste en comprobar las operaciones que realiza este módulo.

Obtenemos fecha de proceso.

Si la fecha es escrita incorrectamente.

Desplegamos mensaje "Fecha incorrecta".

En otro caso.

Buscamos archivos de cobranza con fecha de proceso.

Si no existen archivos con fecha especificada. Desplegamos el mensaje "No existe información con esa fecha"

En otro caso

Cargamos archivos a tablas temporales para validar la información.

Si la información es incorrecta. Movemos archivos incorrectos a directorio de rechazos.

En otro caso.

Se integran a la tabla de cobros así como a la tabla de pagos, se genera archivo de envío PISA.

Módulo de Ajustes

(Poder hacer Altas, Bajas, Modificaciones y Consultas a las tablas de Cobros, Histórico de Cobros y Pagos)

Se comprueba en esta pantalla que cada uno de los botones de comando que aparecen, transfieran al módulo indicado:

- **Ajustes a tabla de Cobros**
- **Ajustes a tabla de Pagos**

Ajustes a tabla de Cobros

(Altas, Bajas, Cambios, Consultas a la tabla de Cobros)

la rutina de diagnóstico comprueba las operaciones que se realizan a la tabla de Cobros, estas son las siguientes:

- **Altas**
- **Bajas**
- **Cambios**
- **Consultas**

Ajustes a tabla de Pagos

(Poder hacer cambios a la tabla de Pagos)

Se comprueba que se puedan realizar correctamente cambios en la tabla de Pagos del sistema.

Generación de Reportes

(Módulo generador de todos los reportes del sistema)

Se comprueba en esta pantalla que cada uno de los botones de comando que aparecen, transfieran al módulo indicado, los módulos son los siguientes:

- Control de Pagos
- Reporte Diario de Caja
- Reporte Estadístico
- Reporte por Vía de Cobro

Control de Pagos

(Nos refleja el total cobrado por semana y el total pagado por los terceros así como nos proporciona estación, teléfono, importe por un día en específico)

La rutina de diagnóstico comprueba que se transfiera correctamente a los módulos siguientes:

- Captura Pagos de Agentes
- Diario de Agente
- Verificación de Pagos de Clientes

Captura Pagos de Agentes

(Nos refleja el total cobrado por semana y el total pagado por los terceros)

la rutina de diagnóstico verifica la realización correcta de las operaciones que ejecuta este módulo

Obtener fecha inicial y fecha final a consultar o capturar.

Usar en forma exclusiva la tabla de pagos.

Realizar selección o filtro en la tabla de pagos por los campos fechas de cobro inicial y fecha de cobro final.

Si no existe información en la tabla de pagos relativa al periodo especificado. Desplegar el mensaje "No existe información para el periodo de " fecha inicial "a" fecha final.

Regresamos a pantalla de pagos.

En otro caso.

Realizar la suma del campo de importe de cobro del filtro realizado.

Desplegar el mensaje "Se ha cobrado" suma de importe de cobro.

Calcular las diferencias entre la fecha de deposito menos fecha de cobro.

Desplegar pantalla de consulta de pagos con la selección de fechas realizada, conteniendo los campos de la tabla de pagos más la diferencia calculada.

Si el usuario tiene derecho para actualizar pagos.

Activar modo de actualización de pagos.

En cada captura de nuevos importes de cobro o de deposito volver a calcular diferencias.

Diario de Agente

(De un día en específico nos indica la estación de cobro, el número telefónico y el importe pagado)

Operaciones básicas del módulo:

Obtener fecha de cobro y vía de cobro

Usar en forma exclusiva la tabla de cobros.

Si no existe vía de cobro en el catálogo de agentes:

Desplegar el mensaje "Vía de cobro inexistente".

Regresar al campo de vía de cobro

Si no existe información en la tabla de cobros relativa a la fecha y vía de cobro.

Desplegar el mensaje "No existe información relativa a la fecha" fecha capturada.

Regresar a pantalla de diario de Agente.

En otro caso

Realizar suma del campo de importe de recibo del filtro realizado.

Desplegar el mensaje "El importe de recibo es de " suma de importe de recibos.

Desplegar pantalla de consulta de recibos con selección realizada, conteniendo los campos número; de estación, teléfono e importe.

Verificación de Pagos de clientes

(Solicitando la fecha, la vía de cobro y el teléfono, podemos saber datos esenciales de un cliente, como por ejemplo su población, etc.)

Se comprueban las operaciones que realiza este módulo.

Obtener fecha de cobro, vía de cobro, lada y teléfono.

Usar en forma exclusiva la tabla de cobros.

Si no existe vía de cobro en el catálogo de agentes.

Desplegar el mensaje "Vía de cobro inexistente".

Volver a ingresar vía de cobro.

Si no existe información en la tabla de cobros relativa a la fecha, vía de cobro, lada y teléfono.

Desplegar mensaje "No existe información para la fecha" capturada "agente" vía de cobro "y teléfono" lada + teléfono.

Regresar a pantalla de verificación de pagos.

En caso contrario:

Desplegar pantalla de consulta de verificación de pagos con selección realizada, conteniendo los campos teléfono, importe, fecha de facturación, dígito verificador de teléfono, fecha de cobro y vía de cobro.

Reporte Diario de Caja

(Nos indica el total cobrado por estación de cobro, el número total de recibos cobrados así como el total por vía de cobro)

La rutina de diagnóstico comprueba las operaciones que realiza este módulo.

Obtener fecha de cobro y vía de cobro

Usar en forma exclusiva la tabla de cobros.

Si no existe vía de cobro en catalogo de agentes:

Desplegar mensaje "Vía de cobro inexistente".

Volver a ingresar vía de cobro

Si no existe información en la tabla de cobros relativa a la fecha, vía de cobro.

Desplegar mensaje "No existe información para la fecha" fecha capturada" agente "vía de cobro".

Regresar a pantalla de reporte diario de caja.

En otro caso

Generar reporte diario de caja conteniendo los cortes de número de estación de cobro, número de recibos e importe de recibos y los totales de número de recibos, importe total por vía de cobro y población de agente.

Reporte Estadístico

(Nos proporciona la fecha de cobranza, el no. de recibos y el importe por día de una vía de cobro (agente) de todo un mes)

Se realiza la rutina de diagnóstico comprobando que se realicen correctamente las operaciones que ejecuta este módulo.

Obtener mes y año a reportar.

Usar en forma exclusiva la tabla de Cobros y Pagos.

Si no existe información en la tabla de Cobros relativa al periodo especificado.

Desplegar mensaje "No existe información para el periodo del año "año" y mes "mes".

Regresar a pantalla de reporte Mensual.

En otro caso.

Generar reporte Mensual conteniendo fecha de cobro, número de recibos, importe diario, número total de recibos e importe total por agente y total.

Reporte por Vía de Cobro

(Proporciona los siguientes datos: Estación de cobro, Día y fecha de la semana consultada, Total por semana y por vía.)

Se comprueban las funciones que realiza este módulo.

Obtener mes, año, y vía de cobro a reportar.

Usar en forma exclusiva la tabla de cobros.

Si no existe vía de cobro en el catálogo de agentes:

Desplegar el mensaje "Vía de cobro inexistente".

Volver a ingresar vía de cobro.

Si no existe información en la tabla de cobros relativa al periodo especificado:

Desplegar el mensaje "No existe información para el periodo del año "año", mes "mes" y vía de cobro "vía de cobro".

Regresar a la pantalla de reporte Semanal por Vía de cobro.

En otro caso:

Generar reporte por vía de cobro conteniendo población de agente, estación de cobro, importe por día y total por semana por estación de cobro y con totales de importes por vía de cobro y globales.

Mantenimiento de Catálogos

(Módulo de mantenimiento a los catálogos del sistema)

La rutina de diagnóstico comprueba que se transfiera correctamente a los módulos siguientes:

- Mantenimiento al Catálogo de Agentes
- Mantenimiento al Catálogo de Estaciones de cobro
- Mantenimiento al Catálogo de Usuarios
- Mantenimiento al Catálogo de Grupos

La rutina de diagnóstico para estos módulos consiste en verificar que se realicen correctamente las operaciones que se ejecutan al catálogo de agentes, catálogo de usuarios y al catálogo de grupos respectivamente, estas operaciones son las siguientes:

- Altas

- Bajas
- Modificaciones
- Consultas

Utilerías

(Módulo de utilerías del sistema)

Se comprueba en esta pantalla que cada uno de los botones de comando que aparecen, transfieran al módulo indicado, los módulos son los siguientes:

- Depuración
- Respaldo de Archivos
- Restauración de Respaldos
- Cambio de Password

Depuración

(Proceso de depuración de la bitácora de procesos, bitácora de transmisiones, tabla de pagos y tabla de cobros)

La rutina de diagnóstico comprueba que se transfiera correctamente a los módulos de Depuración de las bitácoras del sistema (procesos y transmisiones), así como a los de Depuración de tablas (Pagos y Cobros).

Para cada uno de estos módulos de depuración la rutina de diagnóstico comprueba que se realice, en el periodo deseado la eliminación de registros en cada uno de estos, indicando mediante mensajes que se realice el respaldo de dicha información antes de

ejecutar la tarea y mantener la integridad de la información de estas bitácoras y tablas del sistema.

- Depuración de Bitácora de Procesos (Proceso que depura la tabla que contiene la bitácora de procesos para su optima utilización y minimizar el tiempo de consulta de los procesos)
- Depuración de Bitácora de Transmisiones (Proceso que depura la tabla que contiene la bitácora de transmisiones para su optima utilización y minimizar el tiempo de consulta de transmisiones)
- Depuración de la tabla de Pagos (Proceso que depura la tabla de Pagos cuando el número de registros contenidos en dicha tabla es muy grande)
- Depuración de la tabla de Cobros (Proceso que depura la tabla de Cobros en el momento en que el volumen de registros es considerable)

Respaldo de Archivos

(Realiza el respaldo de la semana tanto de los archivos de la cobranza tanto de envío como de los terceros recibidos así como las tablas y catálogos.)

La rutina de diagnóstico implementada, verifica que se realice correctamente el respaldo de los movimientos de la cobranza, las operaciones a realizar son:

Obtener periodo a respaldar

Buscar en la tabla de cobros la información correspondiente al periodo capturado.

Si no existe información de ese periodo:

Desplegar el mensaje "No existe información del periodo "periodo".

Regresar a la pantalla de Respaldo de archivos.

En caso de existir información de ese periodo:

Verificar que este lista la unidad de respaldo

Si hay error de unidad, desplegar el mensaje "Unidad de respaldo no disponible, desea intentar de nuevo ?, s/n".

Si la respuesta es afirmativa:

Volver a verificar la unidad de respaldo

Si la respuesta es "n":

Regresar a la pantalla de Respaldo de Archivos

En otro caso:

Realizar la copia (respaldo) de la información de las tablas de la base de datos

Desplegar el mensaje "Operación de respaldo exitosa"

Restauración de Respaldos

(Realiza la restauración de los archivos de la cobranza así como de las tablas y catálogos)

La rutina de diagnóstico comprueba las operaciones que realiza este módulo.

Obtener periodo a restaurar

Verificar que este lista la unidad de respaldo.

Si hay error de unidad, desplegar el mensaje "Unidad de respaldo no disponible, desea intentar de nuevo ?, s/n".

Si la respuesta es afirmativa:

Volver a verificar la unidad de respaldo.

Si la respuesta es negativa:

Regresar a la pantalla de Restauración de respaldos

En otro caso:

Buscar en la unidad de respaldo la información correspondiente al periodo capturado.

Si no existe información de ese periodo:

Desplegar el mensaje "No existe información del periodo"

Regresar a la pantalla de Restauración de respaldos.

En caso de existir información de ese periodo:

Copiar la información correspondiente al periodo indicado a cada una de las tablas de la base de datos.

Desplegar el mensaje "Operación de restauración exitosa"

Cambio de Password

(Cambiar el password de los usuarios dados de alta en el sistema)

En este módulo, la rutina de diagnóstico consiste en verificar que se realiza correctamente el cambio de password de un usuario definido anteriormente en el sistema.

Bitácora

(Módulo de consulta de bitácora de todos los procesos y acciones hechas en el sistema, así como de las estaciones de cobro recibidas en una fecha específica)

En este módulo se guardan todas las operaciones realizadas en el sistema, es decir que cada módulo guarda en esta bitácora los datos importantes de los procesos realizados, lo cual fue comprobado en cada rutina de diagnóstico aplicada a cada módulo.

En el presente módulo, se comprueba que se presenten y se ejecuten correctamente los módulos siguientes:

- Bitácora de Procesos
- Bitácora de Transmisiones

Bitácora de Procesos

(Muestra todas las acciones registradas proporcionando usuario, fecha, hora y acción realizada)

Para este módulo la rutina de diagnóstico comprueba que se realicen las operaciones para las que fue diseñado este módulo.

Obtener fecha a consultar.

Realizar selección o filtro en la bitácora de procesos donde la fecha capturada es igual a la fecha del proceso.

Si no existe información en la bitácora de procesos relativa a la fecha capturada:

Desplegar el mensaje "no existe información para la fecha" fecha capturada.

Regresar a la pantalla anterior.

En otro caso:

Desplegar pantalla de consulta de la bitácora de procesos con la selección realizada.

Bitácora de Transmisiones

(Muestra todas las estaciones de cobro recibidas en una fecha específica)

Para este módulo la rutina de diagnóstico comprueba que se realicen las operaciones para las que fue diseñado este módulo.

Obtener fecha a consultar.

Realizar selección o filtro en la bitácora de transmisiones donde la fecha capturada es igual a la fecha del proceso.

Si no existe información en la bitácora de transmisiones relativa a la fecha capturada:
Desplegar el mensaje "No existe información para la fecha" "fecha capturada".
Regresar a la pantalla anterior.

En otro caso:

Desplegar pantalla de consulta de la bitácora de transmisiones con la selección realizada.

Módulo Finaliza

(Módulo de salida del sistema)

Se comprueba que este módulo finalice la sesión de trabajo o salir del sistema.

Evaluación de los datos

La evaluación de los datos es una parte importante para la realización de pruebas y la implementación del sistema en sí, es por ello que se describen a continuación los datos a utilizar en el sistema de Cobranza por Terceros.

De forma general se enuncian los datos importantes a utilizar en algunas de las operaciones de entrada y salida, así como de entidades del sistema.

Evaluación de datos de Entrada

Realización de Respaldos

Datos	Tipo	Longitud	Decimales	Repeticiones
Número telefónico	Numérico	7	0	1

Dígito verificador	Numérico	1	0	1
Fecha de facturación	Numérico	3	0	1
Fecha de cobranza	Alfanumérico	10	0	1
Vía de cobro	Numérico	3	0	1
Población	Alfanumérico	13	0	1
Población Agente	Alfanumérico	3	0	1
Número de estación de cobro	Numérico	3	0	1
Fecha de recepción	Alfanumérico	10	0	1
Importe de recibo	Numérico	11	0	1

Tabla 3.4.1 Datos para la realización de respaldos

Transacciones de Archivos PISA

Datos	Tipo	Longitud	Decimales	Repeticiones
Número telefónico	Numérico	7	0	1
Importe	Numérico	11	0	1
Fecha de pago	Fecha	0	0	1
Vía de cobro	Numérico	3	0	1
Dígito constante	Numérico	1	0	1

Tabla 3.4.2 Datos para realizar las transacciones de archivos PISA

Transacciones de Archivo de cobranza

Datos	Tipo	Longitud	Decimales	Repeticiones
Número telefónico	Numérico	7	0	1
Código de población	Numérico	7	0	1
Turno	Numérico	1	0	1
Tipo	Numérico	1	0	1

Dígito verificador	Numérico	1	0	1
Importe	Numérico	11	0	1
Fecha de facturación	Numérico	3	0	1
Número de lote	Numérico	3	0	1
Número de secuencia	Numérico	4	0	1
Verificador de importe	Numérico	1	0	1
Tipo de captura		1	0	1
Lada	Numérico	4	0	1
Vía de cobro	Numérico	3	0	1

Tabla 3.4.3 Datos para realizar las transacciones de archivos de cobranza

Actualización de agentes

Datos	Tipo	Longitud	Decimales	Repeticiones
Vía de cobro	Numérico	3	0	1
Nombre del agente	Alfanumérico	20	0	1
Nombre del responsable	Alfanumérico	20	0	1
Teléfono del responsable	Numérico	11	0	1

Tabla 3.4.4 Datos para realizar la actualización de agentes

Actualización de usuarios

Datos	Tipo	Longitud	Decimales	Repeticiones
Clave de usuario	Alfanumérico	7	0	1
Password	Alfanumérico	7	0	1
Grupo	Alfanumérico	3	0	1
Nombre del usuario	Alfanumérico	20	0	1
Dirección	Alfanumérico	20	0	1

Teléfono del usuario	Numérico	11	0	
Siglas	Alfanumérico	3	0	1
Vía de cobro	Numérico	3	0	1

Tabla 3.4.5 Datos para realizar la actualización de usuarios

Actualización de estaciones de cobro

Datos	Tipo	Longitud	Decimales	Repeticiones
Número de estación de cobro	Numérico	3	0	1
Vía de cobro	Numérico	3	0	1
Nombre de estación de cobro	Alfanumérico	20	0	1
Dirección de estación de cobro	Alfanumérico	20	0	1
Población de estación	Alfanumérico	3	0	1
Responsable de estación de cobro	Alfanumérico	20	0	1
Teléfono de estación de cobro	Alfanumérico	11	0	1
Teléfono del responsable	Numérico	11	0	1

Tabla 3.4.6 Datos para realizar la actualización de estaciones de cobro

Evaluación de datos de Salida

Consulta de bitácora de terceros

Datos	Tipo	Longitud	Decimales	Repeticiones
Número de estación de cobro	Numérico	3	0	1
Fecha de transacción	Alfanumérico	8	0	1
Hora	Alfanumérico	5	0	1

Tabla 3.4.7 Datos para realizar consultas a la bitácora de terceros o de transacciones

Respaldo de archivos

Datos	Tipo	Longitud	Decimales	Repeticiones
Número telefónico	Numérico	7	0	1
Dígito verificador	Numérico	1	0	1
Fecha de facturación	Numérico	3	0	1
Fecha de cobranza	Alfanumérico	10	0	1
Vía de cobro	Numérico	3	0	1
Población	Alfanumérico	13	0	1
Población agente	Alfanumérico	3	0	1
Número de estación de cobro	Numérico	3	0	1
Fecha de recepción	Alfanumérico	10	0	1
Importe de recibo	Numérico	11	0	1

Tabla 3.4.8 Datos para realizar respaldo de archivos

Archivos de cobranza

Datos	Tipo	Longitud	Decimales	Repeticiones
Código de población	Numérico	7	0	1
Dígito verificador	Numérico	1	0	1
Fecha de facturación	Numérico	3	0	1
Importe	Numérico	11	0	1
Lada	Numérico	4	0	1
Número telefónico	Numérico	7	0	1
Número de lote	Numérico	3	0	1
Número de secuencia	Numérico	4	0	1
Tipo	Numérico	1	0	1
Tipo de captura	Alfanumérico	1	0	1
Turno	Numérico	1	0	1

Verificador de importe	Numérico	1	0	1
------------------------	----------	---	---	---

Tabla 3.4.9 Datos para realizar transacciones con archivos de cobranza

Archivos de proceso PISA

Datos	Tipo	Longitud	Decimales	Repeticiones
Número telefónico	Numérico	7	0	1
Fecha de pago	Fecha	0	0	1
Importe	Numérico	11	0	1
Vía de cobro	Numérico	3	0	1
Dígito constante	Numérico	1	0	1

Tabla 3.4.10 Datos para realizar transacciones de archivos PISA

Consulta de pagos

Datos	Tipo	Longitud	Decimales	Repeticiones
Vía de cobro	Numérico	3	0	1
Diferencia	Numérico	11	0	1
Fecha deposito	Fecha	0	0	1
Importe liquidado	Numérico	11	0	1
Importe del periodo	Numérico	11	0	1

Tabla 3.4.11 Datos para realizar consulta de pagos

Reporte diario de caja

Datos	Tipo	Longitud	Decimales	Repeticiones
Nombre de estación de cobro	Alfanumérico	20	0	1

Importe total	Numérico	11	0	1
Importe de recibos	Numérico	11	0	1
Total de recibos	Numérico	5	0	1
Número de recibos	Numérico	3	0	1
Fecha de cobranza	Alfanumérico	10	0	1
Vía de cobro	Numérico	3	0	1
Población	Alfanumérico	13	0	1

Tabla 3.4.12 Datos para realizar el reporte diario de caja

Consulta de diario de agente

Datos	Tipo	Longitud	Decimales	Repeticiones
Fecha de cobranza	Alfanumérico	10	0	1
Importe total diario	Numérico	11	0	1
Número telefónico	Numérico	7	0	1
Número de estación de cobro	Numérico	3	0	1
Importe	Numérico	11	0	1

Tabla 3.4.13 Datos para realizar consulta del diario de agente

Consulta de verificación de pagos

Datos	Tipo	Longitud	Decimales	Repeticiones
Fecha de cobranza	Alfanumérico	10	0	1
Importe total diario	Numérico	11	0	1
Nombre de estación de cobro	Alfanumérico	20	0	1
Número telefónico	Numérico	7	0	1
Importe	Numérico	11	0	1

Tabla 3.4.14 Datos para realizar consulta de verificación de pagos

Reporte estadístico

Datos	Tipo	Longitud	Decimales	Repeticiones
Fecha inicial	Fecha	0	0	1
Fecha final	Fecha	0	0	1
Nombre del agente	Alfanumérico	20	0	1
Fecha de cobranza	Alfanumérico	10	0	1
Numero de recibos	Numérico	3	0	1
Importe	Numérico	11	0	1
Importe semanal	Numérico	11	0	1
Recibos por semana	Numérico	4	0	1
Total de recibos	Numérico	5	0	1
Importe total	Numérico	11	0	1

Tabla 3.4.15 Datos para realizar el reporte estadístico

Reporte por vía de cobro

Datos	Tipo	Longitud	Decimales	Repeticiones
Fecha inicial	Fecha	0	0	1
Fecha final	Fecha	0	0	1
Vía de cobro	Numérico	3	0	1
Población agente	Alfanumérico	3	0	1
Número de estación de cobro	Numérico	3	0	1
Importe total diario	Numérico	11	0	1
Importe semanal	Numérico	11	0	1
Importe diario	Numérico	11	0	1
Total importe semanal	Numérico	11	0	1

Tabla 3.4.16 Datos para realizar el reporte por vía de cobro

Bitácora de procesos

Datos	Tipo	Longitud	Decimales	Repeticiones
Tarea	Alfanumérico	35	0	1
Clave de usuario	Alfanumérico	7	0	1
Fecha de transacción	Alfanumérico	8	0	1
Hora	Alfanumérico	5	0	1

Tabla 3.4.17 Datos para realizar la consulta de la bitácora de procesos

Consulta de Catálogo de agentes

Datos	Tipo	Longitud	Decimales	Repeticiones
Vía de cobro	Numérico	3	0	1
Nombre del agente	Alfanumérico	20	0	1
Nombre del responsable	Alfanumérico	20	0	1
Teléfono del responsable	Numérico	11	0	1

Tabla 3.4.18 Datos para realizar la consulta del catálogo de agentes

Consulta Catálogo de usuarios

Datos	Tipo	Longitud	Decimales	Repeticiones
Numero de estación de cobro	Numérico	3	0	1
Vía de cobro	Numérico	3	0	1
Nombre de estación de cobro	Alfanumérico	20	0	1
Dirección de estación de cobro	Alfanumérico	20	0	1
Población de estación	Alfanumérico	3	0	1
Responsable de estación de cobro	Alfanumérico	20	0	1
Teléfono de estación de cobro	Alfanumérico	11		1

Teléfono del responsable	Numérico	11	0	1
--------------------------	----------	----	---	---

Tabla 3.4.19 Datos para realizar consulta del catálogo de usuarios

Consulta Catálogo de estaciones de cobro

Datos	Tipo	Longitud	Decimales	Repeticiones
Número de estación de cobro	Numérico	3	0	1
Vía de cobro	Numérico	3	0	1
Nombre de estación de cobro	Alfanumérico	20	0	1
Dirección de estación de cobro	Alfanumérico	20	0	1
Población de estación	Alfanumérico	3	0	1
Responsable de estación de cobro	Alfanumérico	20	0	1
Teléfono de estación de cobro	Alfanumérico	11		1
Teléfono del responsable	Numérico	11	0	1

Tabla 3.4.20 Datos para realizar consulta al catálogo de estaciones de cobro

Archivos transferidos

Datos	Tipo	Longitud	Decimales	Repeticiones
Clave de usuario	Alfanumérico	7	0	1
Password	Alfanumérico	7	0	1
Grupo	Alfanumérico	3	0	1
Nombre del usuario	Alfanumérico	20	0	1
Dirección	Alfanumérico	20	0	1
Teléfono del usuario	Numérico	11	0	1
Siglas	Alfanumérico	3	0	1
Vía de cobro	Numérico	3	0	1

Tabla 3.4.21 Datos para realizar consulta del catálogo de usuario

Evaluación de los datos de entidades

Catálogo de agentes

Datos	Tipo	Longitud	Decimales	Repeticiones
Vía de cobro	Numérico	3	0	1
Nombre del agente	Alfanumérico	20	0	1
Nombre del responsable	Alfanumérico	20	0	1
Teléfono del responsable	Numérico	11	0	1
Comprobación	Numérico	1	0	1

Tabla 3.4.22 Datos de catálogo de agentes

Bitácora de procesos

Datos	Tipo	Longitud	Decimales	Repeticiones
Clave de usuario	Alfanumérico	7	0	1
Tarea	Alfanumérico	35	0	1
Hora	Alfanumérico	5	0	1
Fecha de transacción	Alfanumérico	8	0	1

Tabla 3.4.23 Datos de la bitácora de procesos

Bitácora de transmisiones

Datos	Tipo	Longitud	Decimales	Repeticiones
Número de estación de cobro	Numérico	3	0	1
Fecha de transacción	Alfanumérico	8	0	1

Hora	Alfanumérico	5	0	1
Vía de cobro	Numérico	3	0	1

Tabla 3.4.24 Datos de bitácora de transacciones

Transacciones de cobranza

Datos	Tipo	Longitud	Decimales	Repeticiones
Número telefónico	Numérico	7	0	1
Fecha de facturación	Numérico	3	0	1
Verificador de teléfono	Numérico	3	0	1
Vía de cobro	Numérico	3	0	1
Fecha de cobranza	Alfanumérico	10	0	1
Código de población	Numérico	7	0	1
Importe de recibo	Numérico	11	0	1
Fecha de recepción	Alfanumérico	10	0	1
Número de estación de cobro	Numérico	3	0	1
Población agente	Alfanumérico	3	0	1

Tabla 3.4.25 Datos de las transacciones de cobranza

Histórico de cobros

Datos	Tipo	Longitud	Decimales	Repeticiones
Número telefónico	Numérico	7	0	1
Verificador de teléfono	Numérico	3	0	1
Fecha de facturación	Numérico	3	0	1
Fecha de cobranza	Alfanumérico	10	0	1
Vía de cobro	Numérico	3	0	1
Población	Alfanumérico	13	0	1

Población agente	Alfanumérico	3	0	1
Número de estación de cobro	Numérico	3	0	1
Fecha de recepción	Alfanumérico	10	0	1
Importe	Numérico	11	0	1

Tabla 3.4.26 Datos de la tabla de histórico de cobros

Tabla de Pagos

Datos	Tipo	Longitud	Decimales	Repeticiones
Vía de cobro	Numérico	3	0	1
Importe de pago	Numérico	11	2	1
Fecha de deposito	Fecha	10	0	1
Importe de cobro	Numérico	11	2	1
Fecha de cobro final	Fecha	10	0	1
Fecha de cobro inicial	Fecha	10	0	1

Tabla 3.4.27 Datos de la tabla de pagos

Catálogo de Usuarios

Datos	Tipo	Longitud	Decimales	Repeticiones
Vía de cobro	Numérico	3	0	1
Siglas	Alfanumérico	3	0	1
Teléfono del usuario	Numérico	11	0	1
Dirección	Alfanumérico	20	0	1
Nombre del usuario	Alfanumérico	20	0	1
Grupo	Alfanumérico	3	0	1

Password	Alfanumérico	7	0	1
Clave del usuario	Alfanumérico	7	0	1

Tabla 3.4.28 Datos del catálogo de usuarios

Catálogo de estaciones de cobro

Datos	Tipo	Longitud	Decimales	Repeticiones
Número de estación de cobro	Numérico	3	0	1
Vía de cobro	Alfanumérico	3	0	1
Dirección de estación de cobro	Alfanumérico	20	0	1
Nombre de estación de cobro	Alfanumérico	20	0	1
Teléfono de estación de cobro	Alfanumérico	11	0	1
Responsable de estación de cobro	Alfanumérico	20	0	1
Población de estación	Alfanumérico	3	0	1
Teléfono del responsable	Numérico	11	0	1

Tabla 3.4.29 Datos del catálogo de estaciones de cobro

3.5 INTEGRACIÓN Y PRUEBAS DEL SISTEMA

Introducción

Las pruebas son parte integral y vital del ciclo de vida del desarrollo de sistemas. Se realizan con el propósito de encontrar fallas o errores y se establecen para mejorar la calidad del sistema.

Las pruebas permiten:

- Determinar las bases para encontrar los objetivos y un plan específico de pruebas.
- Asegurar la obtención y formalización de los requerimientos del usuario y verificar que son adquiridos de una manera completa, correcta y consistente.
- Verificar los requerimientos funcionales y estructurales para establecerlos como fundamento para la realización de las pruebas.
- Buscar y registrar fallas o defectos asociados a los requerimientos establecidos.
- Documentar los reportes para las pruebas realizadas.

En este capítulo se describen los fundamentos para la realización de pruebas, categorías, consideraciones necesarias, recursos humanos e infraestructura para la realización de pruebas, concluyendo con la aplicación e integración de las pruebas realizadas al Sistema de cobranza por terceros.

Consideraciones importantes para la ejecución de las pruebas

Riesgos y suposiciones para las pruebas

Los riesgos son aquellos factores que pueden afectar negativamente la ejecución de las pruebas.

Las suposiciones son las premisas que pueden afectar positiva o negativamente la ejecución de las pruebas complicando o facilitando las actividades de pruebas.

Condiciones y restricciones

Generalmente son limitaciones o problemas de naturaleza técnica y están relacionadas con el desarrollo del proyecto en sí, la tecnología de pruebas, el estado de los ambientes de prueba, etc.

Cobertura funcional de las pruebas

Dentro de la cobertura funcional de las pruebas se deben describir y listar de manera clara y concisa las funciones a probar, así como aquellas funciones a no ser probadas aun siendo parte del proyecto, ya que son necesarias especialmente cuando se requiere explicar el porque de su exclusión definiendo el alcance de las pruebas y delimitando responsabilidades. Además se debe documentar el ciclo del sistema a ser simulado con el objeto de ejecutar cada una de las funciones objeto de las pruebas. Esta simulación puede ser realizada con muestras de datos fuera de especificaciones.

Descripción de la arquitectura del sistema

Para ello se consideran las especificaciones del software de base sobre el cual esta construido el sistema tal como la plataforma, el software de base de datos, el sistema operativo, el lenguaje de programación etc.

Fundamentos para la ejecución de pruebas

Los fundamentos de las pruebas definen los objetivos para las pruebas del sistema. En esta fase se intenta demoler todo el software que se ha construido en las etapas de análisis y diseño.

Las pruebas requieren que se descarten las ideas acerca de lo correcto que es el software desarrollado y que al descubrir los errores se logre superar cualquier conflicto.

Categoría de pruebas

Las categorías de pruebas se toman directamente de la lista de requerimientos funcionales y estructurales del plan de pruebas. La especificación de las categorías de pruebas a usar nos sirven para determinar la infraestructura de pruebas requeridas.

Las categorías de las pruebas pueden ser:

- De integración
- Unitarias
- De regresión
- De volumen
- De concurrencia
- De aceptación del usuario
- Del sistema
- Manuales o automáticas

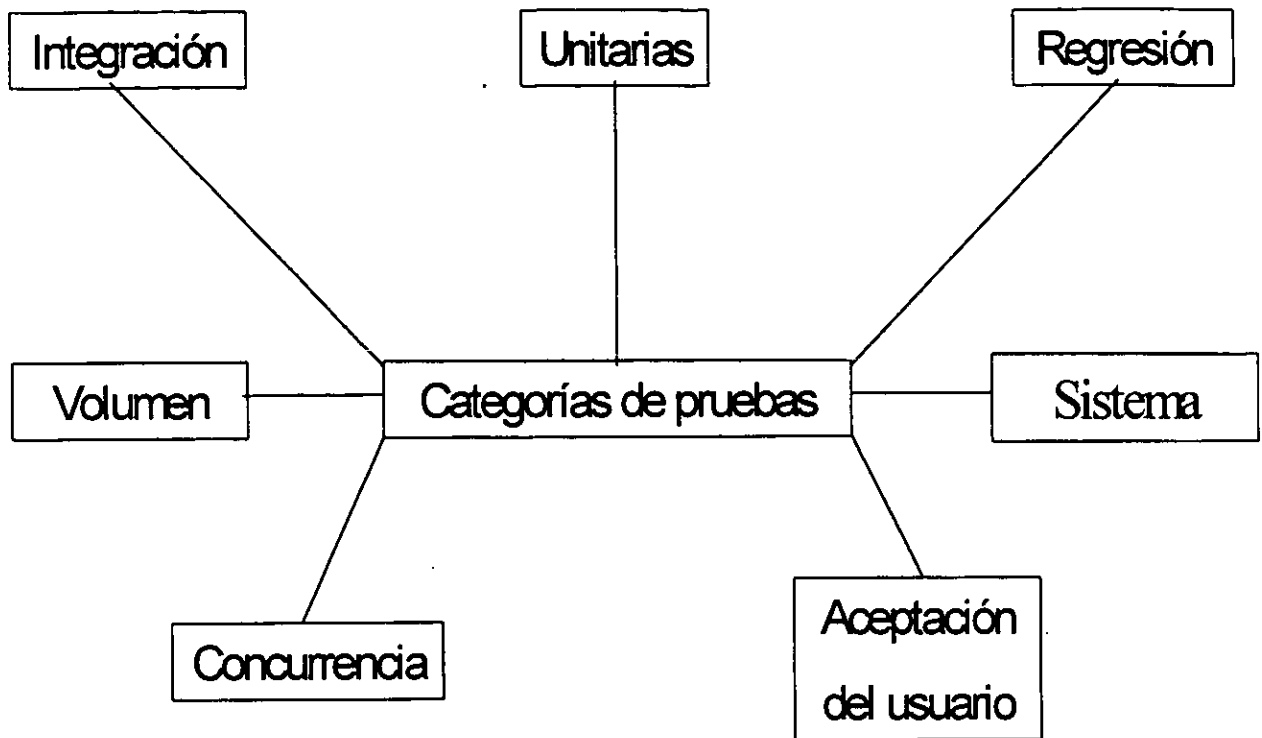


Figura 3.5.1 Categorías de las pruebas

Pruebas de integración

Categoría de pruebas realizadas a un grupo de programas para asegurar que los datos y controles pasen adecuadamente entre programas.

Es una técnica sistemática para construir la estructura del programa mientras que, se llevan a cabo al mismo tiempo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados en unidad y construir una estructura de programas que este de acuerdo con lo que dicta el diseño.

Pruebas unitarias

Es aquella categoría de pruebas realizadas sobre un programa o módulo con la finalidad de encontrar problemas funcionales en la lógica y problemas técnicos en el código.

La prueba de unidad basa el proceso de verificación en la menor unidad de diseño del software, es decir, el módulo.

Utilizando la descripción del diseño detallado como guía, se prueban los caminos de control, a fin de descubrir errores dentro del módulo.

La complejidad relativa de las pruebas y de los errores descubiertos esta limitada por el alcance estricto establecido por la prueba de unidad. Esta prueba esta orientada a la caja blanca y se puede llevar a cabo en paralelo para múltiples módulos.

Pruebas de regresión

Categoría de pruebas selectivas para detectar fallas que se hallan introducido durante las modificaciones a un sistema o componente, permiten verificar que estas modificaciones no repercutan en forma negativa y que se cumpla con los requerimientos preestablecidos.

Pruebas de volumen

Categoría de pruebas realizadas para verificar el comportamiento adecuado de una aplicación bajo condiciones de volumen (número de operaciones), competencia de recursos (conurrencia) y carga máxima (velocidad de petición de ejecución de una operación). También se verifica el correcto comportamiento bajo las condiciones de volumen máximo (cantidad de datos).

Pruebas de aceptación del usuario

Categoría de pruebas finales ejecutada por el usuario, asegura que el sistema satisfaga las necesidades de la empresa o usuario final.

Validan que el sistema desarrollado es el correcto.

Pruebas estructurales

Validan la arquitectura del sistema confirmando que todas sus partes funcionen sincronizadamente y que la tecnología este siendo utilizada apropiadamente.

Se refieren a las características técnicas, como su comportamiento con grandes volúmenes de información, tiempos de respuesta etc.

Pruebas funcionales

Verifican los requerimientos de la organización (lo que se supone que el sistema debe hacer), se utiliza para descubrir errores cometidos en la implantación de los requerimientos.

Pruebas de caja blanca

Estas pruebas se basan en el conocimiento sobre la lógica y estructura interna del sistema.

Pruebas de caja negra

Pruebas funcionales basadas en los requerimientos sin conocimiento sobre como fue construido el sistema y usualmente dirigida a los datos.

Pruebas estáticas

Consiste en la revisión y la validación de los documentos generados en las distintas fases de la vida de un proyecto.

Verificación realizada sin ejecutar el código del sistema.

Recursos humanos

Identificar las personas específicas a ocupar los puestos en la organización de pruebas. Esta información es de utilidad ya que al asociar personas a los puestos se descubre los factores que les hace falta para el efectivo cumplimiento de sus responsabilidades.

Preparación de las pruebas

Es necesario realizar todas las actividades que aseguren la disponibilidad de la infraestructura de pruebas de acuerdo con las categorías de pruebas a ejecutar

Los criterios a tener en cuenta para decidir que tipo de plan necesitamos son los siguientes

- Tamaño y complejidad del proyecto
- Lista de funciones, requerimientos, condiciones generales (funcionales y estructurales)
- Arquitectura de la aplicación
- Condiciones y restricciones
- Ambiente de pruebas

Criterios para iniciar la ejecución de las pruebas.- Se deben especificar los eventos de carácter general a realizar, con el objeto de iniciar la ejecución del plan de pruebas.

Preparación y verificación del ambiente de pruebas.- Especificar las actividades más importantes para la preparación del ambiente de pruebas.

Una vez especificadas las actividades es posible identificar los procedimientos de pruebas necesarios (ya existentes y los que requieran ser creados).

A partir de los requerimientos funcionales y estructurales se deben generar los casos de pruebas.

Casos de pruebas

Los casos de pruebas son las condiciones de pruebas ejecutables, que incluye un conjunto de datos de entrada, resultados esperados y los resultados obtenidos. Los casos de pruebas contemplados en la realización de estas se toman a partir de los requerimientos funcionales y estructurales con el siguiente criterio:

- Datos dentro de especificaciones
- Datos fuera de especificaciones

Procedimientos para las pruebas

Así como la metodología de pruebas, en caso de no existir un procedimiento dentro de la organización para la realización de las pruebas, se hace necesario elaborar los pasos a seguir para terminar esta fase con óptimos resultados.

Infraestructura de pruebas

La infraestructura de pruebas contempla los siguientes puntos:

- Ambiente de pruebas
- Organización de las pruebas
- Metodología de las pruebas
- Herramientas de pruebas

- Puntos de control y aprobaciones
- Criterios de suspensión y conclusión de las pruebas

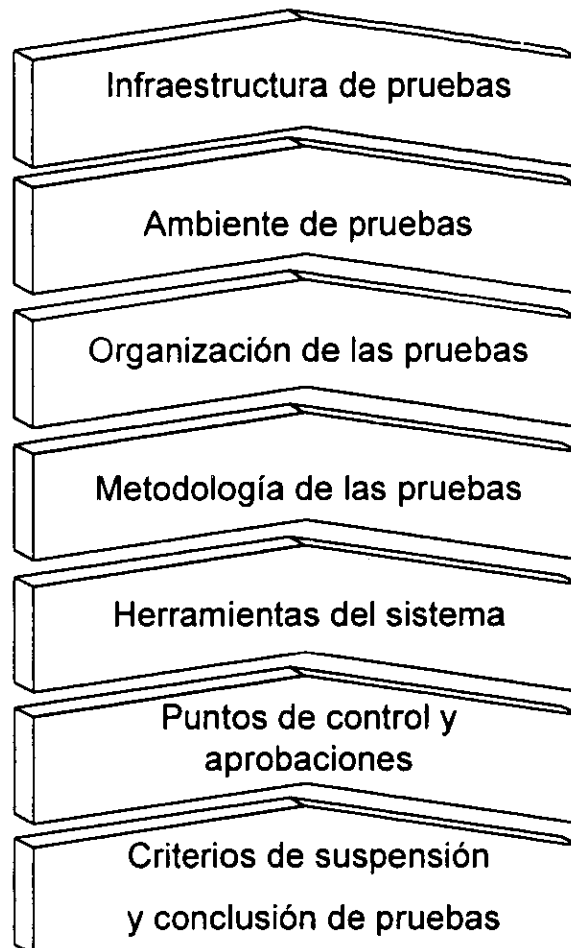


Figura 3.5.2 Infraestructura de pruebas

Ambiente de pruebas

Definen los ambientes donde se ejecutaran las pruebas, mencionan las características generales de los datos de prueba (que datos se necesitan y como se obtendrán) tomando como base el modelo de datos del proyecto.

Esta mención de los datos de prueba es importante para saber cuantos y cuales datos serán seleccionados y para la estimación de la carga de trabajo necesaria para generarla.

Organización de las pruebas

Definir la organización, puestos y responsabilidades que son requeridos para la construcción y ejecución de las pruebas.

Metodología de las pruebas

En este punto es importante determinar si existe un procedimiento de pruebas dentro de la organización o si es necesario elaborarlo y en que medida este procedimiento esta integrado con el resto de los sistemas de desarrollo y mantenimiento. Gracias a esta información, es posible estimar el esfuerzo adicional requerido para la construcción de estos procedimientos, con el objeto de realizar las pruebas.

Se deberá localizar los procedimientos de pruebas de uso general que puedan ser de utilidad.

Herramientas de pruebas

En este punto es necesario identificar los productos a utilizar y el uso específico que se hace con ellos.

Básicamente se debe determinar si es preciso vigilar todos los componentes o solamente algunos, Vigilar a determinados componentes se realiza por la necesidad de checar cual es el comportamiento interno de dicho componente, esto es, como realiza el procesamiento de la información. En el caso de estar solamente interesados en las entradas y salidas de los procesos, es suficiente muchas veces el revisar estas sin tener en cuenta exactamente como se leen y como se generan.

La importancia de este punto radica en la decisión de tomar el enfoque de la caja blanca (vigilando el comportamiento interno de los componentes) lo cual implicará una mayor carga de trabajo debido al número adicional de casos de prueba.

Puntos de control y aprobaciones

Se deben especificar los puntos de control en el transcurso de la construcción y ejecución de las pruebas, tal como el determinar los puestos de las personas que tendrán que autorizar la continuación de las pruebas de acuerdo con el plan original o asumiendo las variaciones incorporadas al mismo.

Criterios de suspensión y conclusión de las pruebas

Estos criterios se refieren a la suspensión o terminación de la ejecución de los casos de prueba cuando son necesarios otros componentes que no se tienen listos o cuando el número de los defectos encontrados exceden el límite de los esperados para lo cual es necesario regresar a la etapa de desarrollo y verificar las especificaciones.

Pruebas realizadas

Las pruebas que se realizaron al Sistema de Cobranza por Terceros fueron las siguientes:

- Pruebas unitarias
- De integración
- De volumen
- De caja blanca

- De caja negra
- De aceptación del usuario final

Dichas pruebas se realizaron tanto con datos dentro de especificaciones, como con datos fuera de ellas para garantizar la integridad y validez de la información procesada.

Consideraciones

Como el sistema fue desarrollado en forma modular basándose en formularios independientes, y estos poseen en común una barra de navegación que permite realizar las funciones más frecuentes para el manejo de información de la base de datos (como moverse entre registros, buscar, editar, agregar, eliminar, etc.) dicha barra ya fue descrita y probada en el capítulo 3.3 Diseño y construcción del Front End, por lo que solo se describirá lo necesario en las pruebas funcionales realizadas.

Ciclo del sistema

El ciclo del sistema a ser probado consiste en

- Registrar la información necesaria del sistema

Cobros o pagos

Agentes

Usuarios

Grupos

Estaciones de cobro

- Verificar que se realicen las funciones principales del sistema

Transferencia de archivos a través de PISA y Vía Diskette.

Extracción de archivos de terceros

Respaldo y restauración de archivos

- Realización de procesos especiales:

Regeneración de archivos

Integración de archivos de cobranza y de terceros faltantes

Ajuste de las tablas de pagos e histórico de pagos

- Generar los reportes del sistema; control de pagos (captura, diario y verificación), diario de caja, estadístico y por vía de cobro.
- Administración del sistema (depuración de bitácoras y tablas), Mantenimiento de catálogos (agentes, usuarios, grupos y estaciones de trabajo)
- El registro de las operaciones realizadas en la Bitácora para su posterior consulta.

Puntos de control para pruebas unitarias, de caja blanca y de integración

Estas pruebas fueron realizadas tomando en cuenta los siguientes puntos:

- De forma Modular

- Utilizando datos dentro de especificaciones
- Utilizando datos fuera de especificaciones
- Detección y corrección de errores funcionales en la lógica y problemas técnicos
- Verificando el Control de flujo
- Detección y corrección de errores en la estructura

Estas pruebas se realizaron en cada módulo del sistema, basándonos en los datos dentro de especificaciones y fuera de ellas, es decir introduciendo valores correctos e incorrectos, con el fin de probar que el módulo realiza con datos correctos las operaciones para las que fue creado y en caso contrario para detectar posibles errores funcionales, de lógica o técnicos en el código, para posteriormente realizar los cambios pertinentes para que los errores encontrados no se presenten nuevamente y permitir mejorar la ayuda y funcionalidad del sistema mediante la inclusión de mensajes de error, advertencia e información. De igual manera se comprueba que el control de flujo pase adecuadamente entre cada módulo, comprobándose así la estructura interna del programa.

Puntos de control para pruebas funcionales, de caja negra y de aceptación del usuario final

Ambiente de las pruebas

Para cubrir esta fase de pruebas fue necesario instalar el sistema en el equipo del cliente, ingresar catálogos, cobros y solicitarle a este que lo utilizará por un mes realizando sus labores normales de cobranza por terceros pero apoyándose ahora con el Sistema de Cobranza por Terceros, de acuerdo con los procedimientos descritos en el manual del usuario y observando el orden modular que tiene el sistema.

De igual forma se le pidió reportar cualquier falla presentada y cualquier duda, con el fin de mejorar o corregir el error presentado o aclarar la duda correspondiente. Así mismo se le pide nos de las sugerencias y comentarios acerca de la funcionalidad y facilidad de uso del sistema.

Los resultados fueron bastantes favorables, los errores presentados fueron por bloqueos de equipo y no del sistema en sí, dudas casi no surgieron, aunque si los comentarios de facilidad de uso, sencillez y rapidez del sistema.

3.6 FACTIBILIDAD TÉCNICA Y OPERATIVA

Introducción

Este tema es parte fundamental para la implementación final del Sistema de Cobranza por Terceros, en el se describe la forma en que se realizaron, por ejemplo, la puesta a punto del servidor, la instalación del programa en el cual se desarrollo el sistema así como la instalación del programa en sí, se describen también algunos puntos importantes para la eficiente operación del sistema, como son: la administración, el control de cambios, el mantenimiento y el soporte técnico.

Factibilidad técnica

El Sistema de Cobranza por Terceros requiere de realizar lo siguiente para su implementación:

- Puesta a punto del servidor Windows NT a utilizar.
- Instalación de Visual Foxpro 5.0 y del Sistema de cobranza por Terceros

La infraestructura de la red dentro de la compañía y dentro de la red local del área de cobranza en donde se instalará el servidor del sistema ya esta implementada y cumple con la normatividad establecida por la empresa para las redes que utilicen Windows NT como sistema operativo de red dentro de la empresa.

Dada esta normatividad, la implementación del servidor a utilizar para el Sistema de Cobranza por Terceros debe cumplir en su totalidad con dichas normas.

A continuación se presenta la normatividad de la empresa para redes con S. O. Windows NT.

Normatividad para redes bajo Windows NT como S. O. de red en la empresa.

Esta normatividad deberá seguirse por todas aquellas personas encargadas directamente de la instalación y puesta a punto de servidores Windows NT Server.

El cumplimiento de esta normatividad garantizará la operación óptima del servidor y evitará cualquier problema con la interconexión, además de permitir la creación de una plataforma eficiente y estandarizada para proporcionar los servicios de red corporativos.

La presente normatividad está enfocada fuertemente a lograr reducir al máximo las labores relacionadas con la administración de redes.

- Requerimientos de HARDWARE del servidor

La parte más importante y crítica de una red lo constituye el servidor. La función del servidor es la de compartir sus recursos a la red, ya sean estos servicios de disco, impresión o aplicaciones cliente/servidor. También debe proporcionar una plataforma robusta para la ejecución de aplicaciones de misión crítica.

Por todo lo anterior, se hace necesario que la computadora destinada a ser servidor, cumpla con requisitos específicos para realizar de manera eficiente todas las funciones anteriormente mencionadas. Estos incluyen los concernientes a sus requerimientos mínimos de hardware y software, sus requerimientos eléctricos y su ubicación física.

Con el fin de ajustarse a diferentes requerimientos de uso y aplicación se han definido dos tipos de servidores: el servidor departamental y el servidor corporativo.

El servidor departamental esta enfocado a la ejecución de los servicios básicos de red (archivos, impresión, software institucional) y esta proyectado para dar servicio hasta 200 usuarios.

Por otra parte el servidor corporativo esta enfocado no solamente a la ejecución de los servicios básicos de red, sino también a la ejecución de aplicaciones de misión crítica (servidores SQL, aplicaciones cliente/servidor etc.) y para redes con más de 200 usuarios.

Los requerimientos mínimos de hardware para el servidor departamental son:

- Servidor Pentium II (350 Mhz)
- Memoria RAM ECC de 128 MB
- Unidad de CD-ROM 8X (SCSI) -interna- booteable
- Unidad de Respaldo 24 GB (SCSI) -interna-
- Disco duro de 9 GB (SCSI)
- Drive interno de 3.5"
- 3 bahías disponibles para crecimiento futuro
- 3 ranuras de expansión libres PCI
- Tarjeta de red PCI (Operación Dual 10BASET/100BASETX)
- 2 puertos seriales y 1 puerto paralelo
- Mouse (bus)
- Monitor SVGA color 15"
- UPS con interface al servidor (1200 Watts)

Los requerimientos mínimos de hardware para el servidor corporativo son:

- Servidor Dual Pentium II (350 Mhz)
- Memoria RAM ECC de 256 MB
- Unidad de CD-ROM 8X (SCSI) -interna- booteable
- Unidad de Respaldo 24 GB (SCSI) -interna-
- RAID con 3 discos duros Hot-Swap de 9 GB Hot-Swap
- Drive interno de 3.5"
- 3 bahías Hot-Swap disponibles para crecimiento futuro
- 3 ranuras de expansión libres PCI

Tarjeta de red PCI (Operación Dual 10BASET/100BASETX)
2 puertos seriales y 1 puerto paralelo
Mouse (bus)
Monitor SVGA color 15"
UPS con interface al servidor (1200 Watts)

Es importante señalar que las marcas y modelos a adquirir (CPU, UPS, tarjeta de red) estén registrados en el HCL (Hardware Compatibility List) de Windows NT Server.

- Requerimientos de SOFTWARE del servidor

Los requerimientos mínimos de software son:

Windows NT Server v4.x -English-

Windows NT Service Pack 3 -English-

- Requerimientos eléctricos del servidor

El servidor debe estar alimentado por un UPS, y a su vez contar con una interfaz a este para protegerlo de forma automática en caso de falla en el suministro eléctrico. Gracias a esta interface, es posible apagar de forma segura al servidor, evitándose así pérdida de información y posibles daños al equipo.

- Ubicación física del servidor

El servidor debe ser colocado en una área exclusiva y libre del paso (de preferencia en un lugar cerrado, y con aire acondicionado), con el fin de que pueda ser operado sin dificultad cuando sea necesario.

- **Requerimientos de HARDWARE de las estaciones de trabajo**

La estación de trabajo es la computadora donde el usuario ejecuta aplicaciones y desde donde accesa y utiliza los recursos del servidor.

Para ejecutar todo el software institucional de manera rápida y eficiente las estaciones de trabajo deben cumplir con ciertos requerimientos mínimos de hardware y software.

Los requerimientos mínimos de hardware son:

- Procesador Pentium -100 MHz-
- Memoria RAM de 32 MB
- Disco duro de 1 GB
- 2 ranuras de expansión libres ISA
- 2 ranuras de expansión libres PCI
- Tarjeta de red PCI (Operación Dual 10BASET/100BASETX)
- Monitor color SVGA
- Mouse (bus)
- 2 puertos seriales y 1 puerto paralelo

- **Requerimientos SOFTWARE de las estaciones de trabajo**

Los requerimientos mínimos de software son:

Windows NT Workstation 4.x (de preferencia) ó

Windows 95

Así mismo deberán utilizar la configuración de red en modo protegido (32 bits).

- Impresoras

Aquellas impresoras que sean conectadas directamente a la red (impresoras con tarjeta de red o a través de una interfaz externa) deberán soportar (y por lo tanto configurarse) con el protocolo TCP/IP. Las impresoras y los respectivos adaptadores de red tienen que estar en el HCL de Windows NT y Windows 95.

- Puesta a punto

Una vez que se tienen listos el servidor, las estaciones de trabajo e impresoras con sus respectivas tarjetas de red, configuradas e intercomunicadas a través del cableado, se debe realizar la puesta a punto, la cual comprende la instalación y configuración inicial del servidor.

Con el fin de simplificar, optimizar y homogeneizar los servidores Windows NT de la empresa, se han normalizado los siguientes puntos:

- A. Instalación
- B. Nombre del servidor y rol asignado
- C. Nombres de las estaciones de trabajo y grupos de trabajo
- D. Nombres de cuentas de usuarios (login-name)
- E. Nombres de recursos
- F. Dominios
- G. Configuración TCP/IP

- H. Configuración IIS
- I. Servicios WINS/DNS/DHCP
- J. Servicio FPNW
- K. Directorio de aplicaciones (APPS)
- L. Directorio de grupos (GROUPS)
- M. Tareas y responsabilidades del administrador
- N. Configuración de las estaciones de trabajo

A. Instalación

Deberá consultarse antes de la instalación, todas los puntos contenidos en esta sección a fin de hacer una adecuada selección de los parámetros de instalación.

Los siguientes son los primeros parámetros que deberán ajustarse:

Directorio de instalación Windows NT	C:\WINNT
Modo de licenciamiento	Per Seat

Tabla 3.6.1 parámetros de instalación

Se requiere que la partición del sistema (C:) tenga al menos 1 GB de espacio.

Recuerde, antes de proceder con la instalación, deberá contar con un nombre y rol asignados, la configuración TCP/IP correcta, deberá conocer así mismo los servicios a

utilizar y asegurarse que el servidor este apropiadamente conectado a un nodo operacional.

B. Nombre del servidor y rol asignado

El servidor debe contar con un nombre único que lo identifique y evite problemas de comunicación hacia otras redes.

Un servidor Windows NT puede configurarse bajo tres esquemas diferentes de seguridad: como PDC (Primary Domain Controller), BDC (Backup Domain Controller) y como Server (Standalone Server). Este esquema de seguridad es conocido como Dominio. El nombre del servidor, así como el rol asignado deberá solicitarlo y registrarlo a la Gerencia de Comunicaciones en Sistemas.

Para servidores que pertenezcan al dominio de la empresa y que trabajen bajo el esquema de seguridad de Server, el nombre se formará de acuerdo a las siguientes reglas:

SRV<nombre-asignado><consecutivo>

donde:

SRV	Para indicar que es un nodo tipo servidor dentro del dominio de la empresa
<nombre asignado>	Corresponden al nombre asignado (siglas, ubicación física, proyecto etc.). Este deberá tener un máximo de 8 caracteres. Deberá solicitarlo a la Gerencia de Comunicaciones.
<consecutivo>	El consecutivo estará en función del número de servidores que existan en la misma dependencia, empezando por el número 1.

Ejemplos:

SRVFC1

SRVSOUP1

SRVRCDT1

C. Nombre de las estaciones de trabajo y grupos de trabajo

El nombre asignado a las estaciones de trabajo (nombres NETBIOS) debe ser igual al login-name del usuario. Si algún usuario tiene asignado más de un equipo se deberá utilizar números consecutivos empezando por el número dos.

Ejemplos:

LLUIS

GOSORIO

JVILLALO

La configuración del grupo de trabajo deberá ser igual al de sus siglas organizativas, mínimos a nivel Subdirección (siglas correspondientes a su área).

D. Nombres de cuentas de usuarios (login-name)

El nombre del usuario (login-name) se formará con un máximo de 8 caracteres (todos ellos deben ser caracteres ASCII de 7 bits, esto significa que no deberán utilizarse caracteres como la ñ o letras acentuadas). El primero de ellos corresponderá a la primer letra del primer nombre del usuario, los subsecuentes caracteres, corresponderán a su apellido paterno hasta completar un máximo de 8. Si se diera el caso de un login-name repetido se tomará entonces como segundo carácter la primer letra de su apellido paterno y los restantes corresponderán a su apellido materno hasta completar un máximo de 8 caracteres.

Solo en caso de que existan login-names repetidos aún después de aplicar la regla anterior, se deberá aplicar la siguiente nomenclatura: El apellido paterno hasta un máximo de 7 caracteres y la primera letra de su primer nombre.

Ejemplos:

Nombre del usuario:	Login-Name:
Irma Gabriela Martínez Rocha	IMARTINE
Ricardo Hernández Hernández	RHERNAND
Rosa Hernández Ortega	RHORTEGA
María Teresa Nuñez Ruíz	MNUNEZ
Julio López García	JLOPEZ
Javier López Ortíz	JLORTIZ
Juan López Ortíz	LOPEZJ

Tabla 3.6.2 Ejemplo de Login-names

La asignación de las cuentas de usuario (login-names) definitivos para el dominio de la empresa los asignará la Gerencia de Comunicaciones. Aquí se asignan de tal forma que sean únicos, no solo a nivel de servidor, sino a nivel corporativo. Por lo tanto si los usuarios tienen acceso al servicio de correo electrónico, o algún servicio que requiera la validación dentro del dominio empresarial, su nombre de cuenta de usuario debe ser solicitado al área de correo electrónico.

E. Nombre de recursos

Un recurso es cualquier parte de una computadora interconectada a la red, como una impresora, disco duro, unidad de CD-ROM que puede ser compartido.

Ahora bien, con el fin de asegurar el acceso homologado a ciertos recursos de los servidores Windows NT, se deberán utilizar los nombres que a continuación se señalan:

CDROM	Primera unidad de CD-ROM
CDROM2	Segunda unidad de CD-ROM
EVERYONE	Directorios de acceso común (\groups\everyone)

Tabla 3.6.3 Nombres de recursos reservados

Para el caso de los nombres de recursos asignados a impresoras la terminología se deja abierta, sin embargo se recomienda utilizar la siguiente nomenclatura: <marca><modelo><tecnología>-<ubicación física>

Ejemplos:

Hplj4laser-of1513

Hpdj602ink-of1402

F. Dominios

En Windows NT, al conjunto de servidores que comparten una base de datos de usuarios común se le llama dominio. Un dominio provee acceso centralizado a las cuentas de usuarios y de grupos. Todo esto es controlado por un administrador de dominio. Cada dominio debe tener un nombre único.

Ahora bien, el configurar servidores NT bajo un mismo dominio es una poderosa herramienta que facilita la administración del sistema, ya que posibilita "ver" a un conjunto de servidores NT como si fueran un sistema único.

En una seguridad basada en dominios un servidor NT puede jugar uno de tres papeles: como PDC (Primary Domain Controller), BDC (Backup Domain Controller) y como Server (Standalone Server). Solo puede haber un PDC en un dominio.

En la empresa se sigue un esquema de seguridad llamado Master Domain Model, el que básicamente consiste en tener un dominio de cuentas (Domain Account) y dominios de recursos (Resources Domains).

Existe solo un dominio maestro (Account Domain) en el cual están contenidos las cuentas de todos los usuarios (a nivel corporativo).

Se sugiere que todos los servidores NT instalados se integren en este esquema de seguridad participando en un rol de tipo Server. La cuenta para poder agregar un servidor al dominio empresarial (solicitada durante el momento de instalación) es el nombre de la compañía, con letras minúsculas. Deberá solicitar el password a la Gerencia de Comunicaciones.

Todos los dominios, así como el rol desempeñado por cada servidor Windows NT instalado se deberá solicitar y registrar en la Gerencia de Comunicaciones de Sistemas.

G. Configuración TCP/IP

Uno de los puntos claves en la arquitectura bajo la cual se diseñó la red de servidores corporativos utilizando Windows NT Server, es la capacidad de este sistema de trabajar todos sus servicios a través del protocolo TCP/IP. Esto está estrechamente relacionado con los proyectos y esfuerzos orientados a construir una red WAN uniprotocolo, lo cual representa incrementar sustancialmente la velocidad de la red, reducir los tiempos de administración de esta al simplificar enormemente los parámetros de operación y configuración, para lograr así una red más robusta, segura y eficiente.

Por tal motivo, solo deberá habilitar el protocolo TCP/IP en el servidor. Otros protocolos como IPX/SPX, NETBEUI, DLC no deberán instalarse.

Los servidores Windows NT deben tener las direcciones hosts en el rango del 245 al 249, configurarse bajo el dominio DNS internal.<nombre de la Cía.>.net, con un

hostname igual al nombre NetBIOS (computer name) del servidor y apuntar hacia los servidores de resolución de nombres WINS/DNS registrados.

H. Configuración IIS

El IIS (Internet Information Server) es un servicio que permite publicar páginas WEB en la red. De hecho ofrece la plataforma ideal para crear la INTRANET de la empresa, al ofrecer los servicios de WWW (World Wide Web) y FTP (File Transfer Protocol) a los servidores Windows NT Server.

Deberá consultar la Normatividad para Intranet emitida por la Gerencia de Desarrollo de Software. Esta normatividad tiene como propósito establecer las políticas, normas y recomendaciones a seguir para la implantación de Intranet's en la compañía.

Los nombres para los directorios de los servicios INTRANET deben ser los siguientes (son los directorios por omisión)

\inetPub\ftproot (Home-Directory para los servicios de FTP)
\inetPub\gophroot
\inetPub\wwwroot

Así mismo es necesario crear un directorio virtual (Virtual Directory) llamado Everyone que apunte al directorio \GROUPS\EVERYONE y con el fin de hacerlo visible al software FTP, se deberá crear el directorio \inetPub\ftproot\Everyone.

I. Servicios WINS/DNS/DHCP

El servicio de DHCP (Dynamic Host Configuration Protocol), ofrece la posibilidad de configurar automáticamente todos los parámetros TCP/IP en las estaciones de trabajo (Windows NT Workstation y Windows 95). DHCP provee una manera segura, confiable y simple para configurar TCP/IP en la red a la vez que ayuda a prevenir conflictos de direcciones, ayudando así a conservar el uso de las direcciones IP y posibilitando la

administración centralizada de esta importante función. Actualmente el rango de direcciones de 13.x.x.120 a 13.x.x.219 esta reservado para la asignación de direcciones dinámicas.

WINS (Windows Internet Name Service) y DNS (Domain Name Service) son los servicios de resolución de nombres. Estos se utilizan para resolver nombres NETBIOS (los nombres utilizados por los servidores y estaciones de trabajo) a direcciones IP.

A través de la red de servidores corporativos de la Gerencia de Comunicaciones ya se están proporcionando estos servicios, por lo que no deberán instalarse.

La información de los servidores WINS/DNS registrados, así como de los servidores DHCP y los segmentos habilitados con este servicio los podrá consultar en el sitio WEB de la Gerencia de Comunicaciones.

J. Servicio FPNW

El FPNW (File and Print Services for Netware) es el servicio que permite operar a un servidor Windows NT como un servidor Novell Netware. En la primera revisión de esta normatividad se enunciaba la necesidad de instalar este servicio a fin de facilitar los procesos de migración de plataforma, ya que permite el acceso a servicios de disco e impresión a los usuarios de DOS y Windows v3.x a través de la tradicional configuración para acceder servidores Novell Netware.

Sin embargo, bajo esta nueva normatividad **NO DEBERA INSTALARSE** este servicio, ya que en este momento se están realizando esfuerzos por detener el incremento de servicios IPX/SPX, y la eliminación gradual del enrutamiento de este protocolo en la red.

K. Directorio de aplicaciones (APPS)

En la primera revisión de esta normatividad, se pretendía que todos los servidores tuvieran instaladas la mayor parte del software institucional (Office, Windows 95, Antivirus, Utilerias etc.) bajo una estructura de directorios estandarizada con el fin de ofrecer a los usuarios de estos servidores un acceso rápido y transparente al software.

Ahora, en base a las nuevas políticas para el acceso al software institucional, surgidas de las recomendaciones de las auditorías realizadas en la empresa, la instalación de este software será exclusivamente a través de los servidores administrados por el área de Distribución de Software.

Por tal motivo los servidores NO deberán ser utilizados para la instalación del software institucional y el directorio asignado para este propósito (VAPPS) no se requiere. Deberá obtener del área de distribución de software los procedimientos para el acceso a estos servidores, así como las cuentas y passwords autorizados.

Finalmente, en base a la nueva filosofía de trabajo en red, el software institucional debe instalarse completo en el disco duro del usuario, NO como instalación de red. Los beneficios en términos de menor tráfico en la red y mayor tolerancia a fallas compensan el consumo de espacio utilizado en el disco de los usuarios.

L. Directorio de grupos (GROUPS)

Este constituye el directorio de datos y aplicaciones específicas. A fin de uniformizar los nombres de subdirectorios bajo GROUPS (\GROUPS) se deberán seguir las siguientes reglas en la asignación de nombres de grupo.

Al menos se deberá crear el directorio \GROUPS\EVERYONE en el que todos los usuarios tengan acceso, esto con el fin de que al menos exista un lugar donde cualquier usuario pueda compartir información con otro. Será necesario informar a los usuarios que no existe ningún compromiso de confidencialidad y permanencia de información sobre este directorio y que será completamente depurado en cuanto se requiera. Es

importante mencionar que además este directorio puede ser accesado a través de FTP, para así posibilitar a los usuarios UNIX o bien a cualquiera que cuente con un servicio FTP acceder y grabar información en este directorio.

Los nombres de subdirectorios de grupos deberán llamarse igual a los grupos para los cuales se tiene acceso. Así mismo se recomienda crear un recurso compartido por cada uno de estos directorios.

Si se tienen privilegios de sólo lectura, para un mismo directorio, se deberá de añadir el sufijo –RO al nombre de estos grupos.

M. Tareas y responsabilidades del administrador

Cada área deberá de nombrar a su administrador, el cual se encargará de todas las tareas relacionadas con la administración y mantenimiento del servidor.

El administrador deberá asumir las siguientes responsabilidades:

- Administración de usuarios y grupos (altas/bajas/cambios).
- Administrar la seguridad del servidor (asignación de privilegios y passwords).
- Administrar los servicios de archivos e impresión.
- Respaldos de la información.
- Soporte técnico de primer nivel a sus usuarios.
- Instalación y soporte del software institucional a sus usuarios.
- Asegurar el cumplimiento de las normatividades de Sistemas.

Para cualquier problema relacionado con los puntos anteriores el administrador puede solicitar asesoría y apoyo al área de Sistemas que le asesora o bien solicitar soporte a través del HELP DESK. Para cualquier problema relacionado con la comunicación con otras redes, el supervisor deberá comunicarse a la Gerencia de Comunicaciones. Para ello se deberán proporcionar a esta todos los datos del supervisor, con el fin de coordinar con él la solución a cualquier problema existente.

N. Configuración de las estaciones de trabajo

Como se mencionó anteriormente los clientes que se recomiendan para trabajar en redes NT son Windows NT Workstation y Windows 95. Es importante que estos sistemas se configuren con el software de red apropiado, es decir, que se utilicen drivers de red en modo protegido (32 bits) específicamente diseñados para el sistema operativo que se trate.

Todos estos clientes deben configurarse con el stack de protocolo TCP/IP y de preferencia configurarse a través de los servicios de DHCP (siempre que sea posible). Así mismo, para asegurar la correcta resolución de nombres en la red se deberán configurar los servidores primarios y secundarios de WINS y DNS.

- Soporte Técnico

A fin de facilitar, en el mayor grado posible, el acceso a información técnica, utilerías, actualizaciones y en general a todo tipo de información referente a Windows NT Server se han levantado los siguientes servicios:

Sitio WEB de la Gerencia de Comunicaciones

Este sitio WEB es parte de la INTRANET de la compañía y lo puede acceder a través de un visualizador, como el Microsoft Internet Explorer. La dirección es <http://Comunicaciones>.

Aquí podrá obtener, entre otras cosas, lo siguiente:

Documentos actualizados de esta normatividad

Service Pack para Windows NT y Windows 95

Información técnica

Listas de compatibilidad de hardware (HCL)

Documentos técnicos y de configuración

Acceso a la base de datos de soporte técnico y utilerías administrativas

Mediante el acceso a las bases de datos de soporte técnico (MS TechNet) podrá encontrar todo tipo de información sobre Windows NT y en general para un amplio conjunto de productos, definición de términos y mensajes de error. También está disponible un directorio de utilerías administrativas.

Todo esto lo puede acceder desde:

Servidor	TMXDFUNIV5	
Dominio	<Nombre de la Cía.>	
Cuenta	<Su login name>	
Password	<Su password>	
Recurso	CDROM	MS-TechNet
Recurso	ADMIN	utilerías admnistrativas

Tabla 3.6.4 Servidor de ayudas y utilerías

Se recomienda estar consultando de forma periódica el sitio WEB de la Gerencia de Comunicaciones, ya que ahí se informará de cualquier cambio en la manera de acceder de esta información.

Puesta a punto del servidor del Sistema de Cobranza por Terceros

La puesta a punto del servidor en donde se instalará la aplicación, contempla todas y cada una de las normas establecidas y presentadas anteriormente, dado que el servidor con el que cuenta el área de Control de Ingresos de la empresa es de uso departamental, se considero necesaria la implementación de otro servidor para uso corporativo, se registro también el nombre y el rol de este (Sistema de Cobranza por Terceros).

El nombre asignado al servidor y debido a que se firmará dentro del dominio empresarial será el siguiente:

COBRATEL

Se pidió la asignación de una dirección I. P. al área respectiva, otorgando la siguiente:

Dirección I.P. 13.53.1.245

De forma general los requisitos mínimos de hardware y software son cumplidos satisfactoriamente, la ubicación del servidor será la misma en donde esta actualmente el servidor departamental del área de cobranza (Control de Ingresos) y que cumple con las condiciones de clima y eléctricas establecidas en la normatividad.

La estructura de directorios y nombres de dispositivos también cumple con las normas establecidas.

El cumplimiento de esta normatividad permitirá:

- Una operación óptima de los servidores.
- Evitar conflictos de operación.
- La integración e interconexión en la red.
- Facilitar actualizaciones futuras.
- La estandarización de los parámetros de operación.
- Facilitar el soporte técnico.

Instalación de Visual FoxPro 5.0 y del Sistema de Cobranza por Terceros

La instalación de Visual Foxpro 5.0 es sumamente sencilla ya que cuenta con guión de instalación, se inicia mediante la ejecución del archivo Setup.exe, las posibles formas de instalación son las siguientes:

- Portátil (15 Mb)
- Completa (100 Mb).
- Personalizada (240 Mb)

Se deberá realizar la instalación total del software, esto se realiza seleccionando la opción de instalación "Personalizada" y en la pantalla siguiente dando click en "Seleccionar todo", con lo cual se instala totalmente Visual Foxpro.

La instalación del Sistema de Cobranza por Terceros, consistirá en pasar del equipo de pruebas actualmente utilizado y en donde actualmente está el sistema, al servidor COBRATEL una vez que se haya puesto a punto.

La instalación del software Visual Foxpro y del Sistema de Cobranza por Terceros, en cumplimiento con la normatividad para los servidores Windows NT, se instalará bajo el directorio /APPS bajo los directorios VFP (Visual Foxpro) y COBRATEL (Sistema de Cobranza por Terceros) respectivamente.

Administración del sistema

Para la administración del Sistema de Cobranza por Terceros y en general del servidor COBRATEL, se ha sugerido que el encargado de estas labores, sea el propio administrador del servidor del área de Control de Ingresos de la empresa.

Esta determinación se tomo, dadas la características y/o perfil (capacidad técnica, conocimientos de informática) que se requieren para realizar las labores de administración tanto del servidor como del Sistema de Cobranza por Terceros.

El administrador deberá asumir además de las responsabilidades descritas en la normatividad de la empresa para servidores bajo sistema operativo Windows NT, las siguientes responsabilidades para el Sistema de Cobranza por Terceros:

- Administración de agentes, estaciones de trabajo, usuarios y grupos (altas/bajas/cambios).
- Administrar la seguridad del sistema, mediante la asignación de privilegios y passwords.
- Realizar los respaldos de la información, (se recomienda cada 15 días).
- Aclaración y atención de dudas, comentarios, funcionamiento y mejoras propuestas por los usuarios del sistema.
- Brindar soporte técnico a los usuarios del sistema.

- Instalación y soporte del software del Sistema de Cobranza por Terceros a usuarios.
- Llevar un control de los cambios realizados al sistema, así como altas bajas y cambios en la información de las bases de datos del sistema.

Control de Cambios

Aun cuando el sistema esta creado bajo los requerimientos del usuario, no están descartadas posibles modificaciones al sistema, con las que puedan aumentar su capacidad o agregar algún requerimiento más.

Es por esta razón que deberá llevarse un control de cambios realizados al sistema, indicando datos importantes como son; la fecha, motivo de la modificación, tipo de modificación, solución propuesta, persona que la autorizo y persona que realizo el cambio.

Este control servirá para poder llevar un histórico del sistema y de cada una de sus modificaciones por fallas, limitaciones del diseño y/o cualquier aumento de necesidades.

El Sistema de Cobranza por terceros contiene un módulo dedicado totalmente al registro de todas las operaciones realizadas al sistema (Bitácoras de procesos y bitácora de transmisiones).

Esta bitácora sirve para poder revisar que operaciones fueron realizadas en cierto período de tiempo y poder verificar o encontrar posibles fallas, determinando claramente en que fecha se realizaron dichas operaciones y quien o que usuario las realizó.

Tipos de mantenimiento

Una parte fundamental para el buen funcionamiento y posible crecimiento de un sistema se basa en el mantenimiento que a este se le brinde, a continuación se presenta la descripción de algunos tipos de mantenimiento.

- **Mantenimiento perfectivo.**

El mantenimiento perfectivo comprende los cambios solicitados al programador del sistema.

A medida que se usa el software, se reciben de los usuarios recomendaciones sobre nuevas posibilidades acerca de modificaciones a funciones ya existentes. Para satisfacer estas peticiones se lleva a cabo el mantenimiento perfectivo.

- **Mantenimiento Preventivo.**

En este tipo de mantenimiento se previenen errores.

Este mantenimiento se da cuando se realizan cambios en el software con el fin de mejorar algún proceso.

Se puede considerar el mantenimiento a la información que se maneja para garantizar que los resultados dados por el sistema sean los correctos.

- **Mantenimiento adaptativo.**

El mantenimiento adaptativo se debe a cambios en el ambiente del programa y a la adaptación de nuevas unidades o módulos.

modificaciones y/o cambios, para lo cual deberán realizarse mantenimientos adaptativos.

En cuanto a los mantenimientos correctivos, estos serán realizados en cuanto se presente un problema o error en el sistema, para lo cual se pide a los usuarios del sistema informen cualquier anomalía.

Mejoras y reingeniería del sistema

El objetivo de las mejoras al sistema es modificar o amplificar el sistema de aplicaciones como respuesta a las necesidades cambiantes de la empresa.

Los objetivos de la reingeniería son o bien adaptar el sistema ante un cambio tecnológico importante y arreglar el sistema antes de que falle o bien hacer el sistema más sencillo para cuando tenga que ser adaptado. Estos objetivos pueden relacionarse con los bloques elementales de los sistemas de información del modo siguiente:

- Personas

En su mayoría, las mejoras a los sistemas son propuestas por los usuarios de los sistemas, aunque los analistas, diseñadores y programadores de sistemas también pueden detectar posibles problemas técnicos relativos al rendimiento, la seguridad y los controles internos.

En su mayor parte la reingeniería es llevada a cabo por personal técnico y de sistemas de información.

- Datos

Muchas mejoras de los sistemas se derivan por la necesidad de obtener nueva información que puede obtenerse de datos almacenados existentes. Algunas mejoras de datos pueden requerir la ampliación del almacenamiento de estos.

Los proyectos de reingeniería son debidos a las necesidades de reestructurar los datos almacenados, ya sea para hacerlos más flexibles y fáciles de adaptar o para convertirlos a un nuevo entorno tecnológico.

- Procesos

En su mayoría, las mejoras requieren la modificación de programas existentes o la creación de nuevos programas para ampliar al ámbito general de sistemas de aplicación.

En la reingeniería se intenta reestructurar o reorganizar programas de aplicación para hacerlos más fáciles de mantener o convertirlos a un nuevo entorno tecnológico (por ejemplo, el lenguaje). Muchos otros modifican los métodos de entradas y salidas en los programas (por ejemplo, de modo batch a modo on-line, o de modo on-line a interfaces gráficas de usuario).

- Redes

Algunos proyectos de aplicación buscan modificar las aplicaciones para adaptarlas a una nueva tecnología de redes.

- Tecnología

La mayoría de las mejoras a los sistemas y la reingeniería de proyectos se deben a cambios en la tecnología o a la necesidad de aprovechar mejor la tecnología existente.

Las mejoras al Sistema de cobranza por terceros, serán realizadas conforme las necesidades de los usuarios y del medio en general vayan cambiando.

El sistema se realizo en Visual Foxpro y utilizando la programación orientada a objetos pensando y previendo estos cambios.

El sistema de Cobranza por Terceros debe tener por lo menos una vida útil de 5 a 10 años, realizándole los mantenimientos sugeridos y las adecuaciones necesarias como pueden ser incluir más módulos o mejoras sustanciales al sistema.

La realización de mejoras al sistema y/o cambios en su funcionalidad, como pueden ser la adición de módulos o cambios en la información de las bases de datos, las podremos realizar los desarrolladores del sistema, sin embargo tendrán su debido costo.

Capacitación de los usuarios y administrador del sistema

Se ha determinado brindar la capacitación necesaria para los usuarios de las áreas de Control de Ingresos, Cobros Especiales y al administrador del sistema.

Para los usuarios se contempla la siguiente capacitación.

- Problemática en los procesos que desarrollan actualmente.
- Solución y/o sistema propuesto.
- Ventajas que proporciona el sistema.
- Funcionamiento del Sistema de Cobranza por Terceros (Manual de usuario)

El administrador del sistema deberá tomar la capacitación que se les brinde a los usuarios del sistema.

Al administrador y a dos personas más de las áreas involucradas al proceso de cobranza por terceros (Cobros especiales y Control de ingresos), se les brindará la siguiente capacitación.

- Características y ventajas principales de Visual Foxpro
- Desarrollo e implementación del sistema.
- Administración y mantenimiento del programa.
- Manual técnico

Soporte Técnico

Durante los primeros tres meses de la puesta en marcha del Sistema de Cobranza por Terceros, los desarrolladores del sistema brindaremos el soporte técnico necesario, este se realizará a través del administrador del sistema, es decir, si se encuentran errores de funcionalidad, los usuarios deberán informar al administrador del sistema de estos, con el fin de que los resuelva, en caso de que este no pueda resolverlas consultará con nosotros para que se le brinde la asesoría necesaria y/o realizar el mantenimiento correctivo respectivo y solucionar de inmediato el o los errores presentados.

Análisis del costo y venta del sistema.

El costo total del sistema puede realizarse en base a los siguientes puntos:

- Número de líneas de código
- Por hora – hombre
Analista

Diseñador
Programador

- Por tiempo de desarrollo

El costo de una línea de código oscila entre los 20 y 30 dólares. Para el Sistema de Cobranza por Terceros se tienen aproximadamente 2000 líneas de código.

El costo del Sistema de Cobranza por Terceros obtenido mediante esta opción y tomando el valor por línea de 30 dólares sería de: \$ 60000.00 dólares, al tipo de cambio actual.

El tiempo de desarrollo, se basa en sí, en las horas - hombre utilizadas para el análisis, desarrollo e implementación del sistema, por lo cual realizamos lo siguiente:

El tiempo de desarrollo del Sistema de Cobranza por terceros fue de 3.5 meses, desde el análisis, diseño y programación hasta la implementación del sistema.

El analista y el diseñador del sistema trabajaron durante 1 mes, 4 horas al día.

Considerando los días hábiles únicamente, se tienen por mes 22 días, por lo tanto el total de horas - hombre por mes es de 88.

Los programadores (2) trabajaron durante 2.5 meses, 4 horas al día, por lo tanto el número de horas por los 2.5 meses laborados = 220 horas – hombre, por cada uno de ellos, dando un total de 440 horas- hombre.

El costo por hora – hombre para los analistas, diseñadores, programadores y administradores de sistemas varía entre los \$ 70.00 y 120.00 dólares.

A continuación se presenta el análisis de precios realizado por horas - hombre.

Concepto	Costo Unitario	Unidad de Medición	Cantidad	Costo por Concepto
Analista	\$120.00	Hora-Hombre	\$ 88.00	\$ 10560.00
Diseñador	\$120.00	Hora-Hombre	\$ 88.00	\$ 10560.00
Programadores (2)	\$120.00	Hora-Hombre	\$ 440.00	\$ 52800.00
TOTAL				\$ 73920.00

Tabla 3.6.5 Análisis de costos por hora - hombre

Realizando un ajuste en los costos anteriores y basándonos en el tiempo de desarrollo del sistema, tenemos lo siguiente:

Concepto	Costo Unitario	Unidad de Medición	Cantidad	Costo por Concepto
Tiempo de desarrollo (Análisis, Diseño, Programación e Implementación)	\$ 21428.57	Mes	3.5	\$ 75000.00
TOTAL				\$ 75000.00

Tabla 3.6.6 Análisis de costos por Tiempo de desarrollo

La empresa a la cual se le esta desarrollando el Sistema de Cobranza por Terceros, ya tiene la infraestructura de red necesaria para la implementación del sistema, así como el hardware y software necesarios, por lo tanto no se incluyen estos conceptos en el análisis de costos.

Para empresas que no cuenten con la estructura de red, el hardware (Servidor) y software (Visual Foxpro) necesarios para la implementación del Sistema de Cobranza por Terceros, estos conceptos se agregarán al costo del sistema.

Comparación de costos de acuerdo al tiempo de con proveedores que han desarrollado sistemas para la Compañía en anteriores ocasiones.

Proveedor	Tiempo de desarrollo	Costo (dólares)
EDS	3.5 meses	\$ 89967.00
EPI	2.5 meses	\$ 70050.00
SOFTEK	3 meses	\$ 84658.54

Tabla 3.6.7 Comparación de costos con empresas desarrolladoras de software

Basándonos en el análisis de costos anterior, se determina que la operación de compra-venta del sistema puede realizarse de las siguientes formas:

- Venta del sistema por licencia
- Venta total del sistema
- El costo de venta por licencia sería de: Seiscientos cincuenta dólares \$ 650.00, al tipo de cambio actual.
- El costo total del sistema será de: Setenta y cinco mil dólares \$ 75000.00, al tipo de cambio actual.

La empresa para la cual se desarrollo el Sistema de Cobranza por Terceros tiene como política el adquirir totalmente los derechos de los programas realizados, es decir, requiere le sea entregado la totalidad del código del sistema, manuales de usuario y manuales técnicos.

Por lo tanto la operación de compra-venta del Sistema de Cobranza por Terceros se realizará de forma Total.

MANUAL TÉCNICO

Manual técnico

Introducción

Este manual describe las actividades necesarias para mantener la integridad de la información y en sí del Sistema de Cobranza por Terceros, algunas de estas labores son el respaldo de la información de las tablas de la base de datos del sistema y el mantenimiento a dichas tablas (altas, bajas, cambios, etc.). También se indica quien será el encargado de la realización de cada una de estas actividades.

Se describe también la forma en que deberá instalarse el Sistema de Cobranza por Terceros para los usuarios del sistema.

Instalación del Sistema de Cobranza por Terceros en el equipo del usuario

Esta labor deberá realizarla el administrador del sistema.

En el servidor se encuentra el directorio del sistema (COBRATEL), el cual contiene el sistema en su totalidad (Base de datos, archivos ejecutables, archivos de Visual Foxpro, etc.), también se encuentra la carpeta siguiente:

USUARIO - Contiene el archivo ejecutable del sistema y archivos necesarios de Visual Foxpro para que se ejecute correctamente el sistema.

Los archivos de la carpeta (USUARIOS) deben copiarse a una carpeta con el nombre del sistema (COBRATEL) en el equipo del usuario.

Lo anterior indica que no es necesario instalar Visual Foxpro en cada equipo del usuario, con lo cual se optimizan recursos en los equipos clientes.

En resumen, los pasos para realizar la instalación en el equipo del usuario, son los siguientes:

1. Crear una carpeta en el equipo del usuario con el nombre del sistema (COBRATEL)
2. Ingresar al servidor COBRATEL
3. Copiar los archivos de la trayectoria F:\COBRATEL\USUARIOS a la carpeta del usuario C:\COBRATEL.
4. Crear el acceso directo para el archivo COBRATEL.EXE.

Módulos de mantenimiento y utilerías del sistema

El sistema de cobranza por terceros incluye módulos especialmente desarrollados para realizar las labores de mantenimiento de la información contenida en la base de datos del sistema (agentes, estaciones de cobro, usuarios, grupos y tablas), así como utilerías para depurar la información de las bitácoras y operaciones básicas como son el respaldo y la restauración de archivos.

A continuación se describe cada uno de estos módulos del Sistema de Cobranza por Terceros, así como la barra de navegación de funciones comunes para operar la información de la base de datos.

Barra de navegación

(Barra estándar de funciones comunes para operar tablas de la base de datos)

Esta barra de navegación aparece en varios formularios dentro del sistema en los cuales se tiene integración directa con los registros de la base de datos.

Se integra por las funciones más comunes en la operación de tablas de bases de datos.



Figura 1 Barra de navegación

De izquierda a derecha, se describe la función de cada botón de la barra de navegación:

- Muestra los datos del primer registro
- Proporciona los datos del registro anterior al actual
- Proporciona los datos del siguiente registro al actual
- Muestra los datos del último registro
- Búsqueda de registros
- Imprimir reporte
- Agregar o adicionar registros a la base de datos
- Corrección de datos del registro
- Eliminar el registro actual
- Salir del formulario actual o regresar al formulario anterior

Mantenimiento de catálogos

Este módulo del sistema se encarga de transferir el control de flujo para realizar el mantenimiento a los catálogos siguientes:

- Agentes
- Estaciones de cobro
- Usuarios
- Grupos.

En estos módulos se pueden realizar las operaciones siguientes:

- Altas/Agregar
- Bajas/Eliminar
- Modificaciones/Cambios
- Consultas
- Búsquedas
- Impresión

El mantenimiento a los catálogos de agentes y estaciones de cobro, los puede realizar tanto el administrador, por orden directa del coordinador del área y/o jefe inmediato superior y el o los usuarios con los privilegios de usuario para realizar cambios en este módulo (002 Cobros Especiales).

El mantenimiento a los catálogos de usuarios y grupos solo lo puede realizar el administrador del sistema.

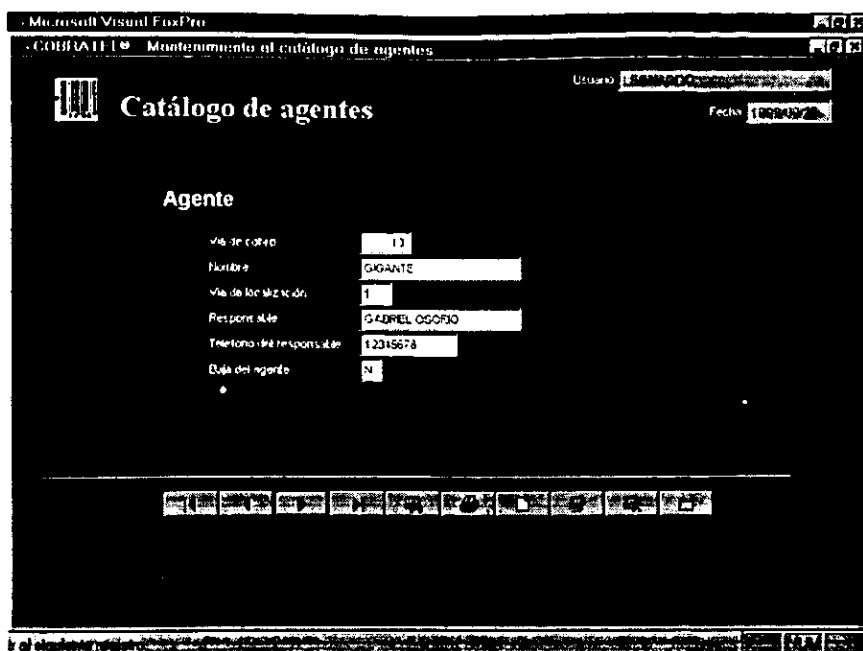


Figura 2 Pantalla de Mantenimiento al catálogo de agentes

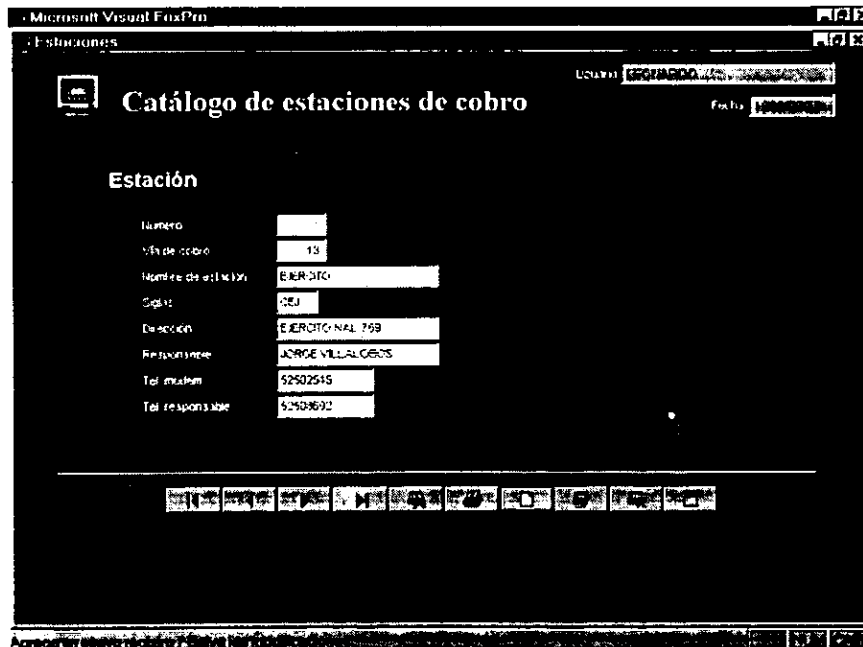


Figura 3 Pantalla de Mantenimiento al catálogo de estaciones de cobro

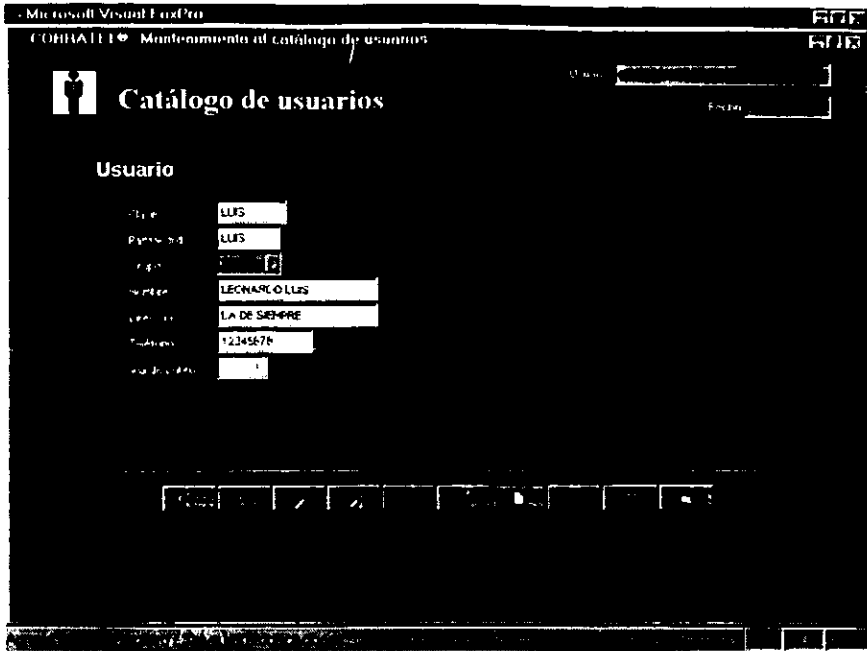


Figura 4 Pantalla de Mantenimiento al catálogo de usuarios

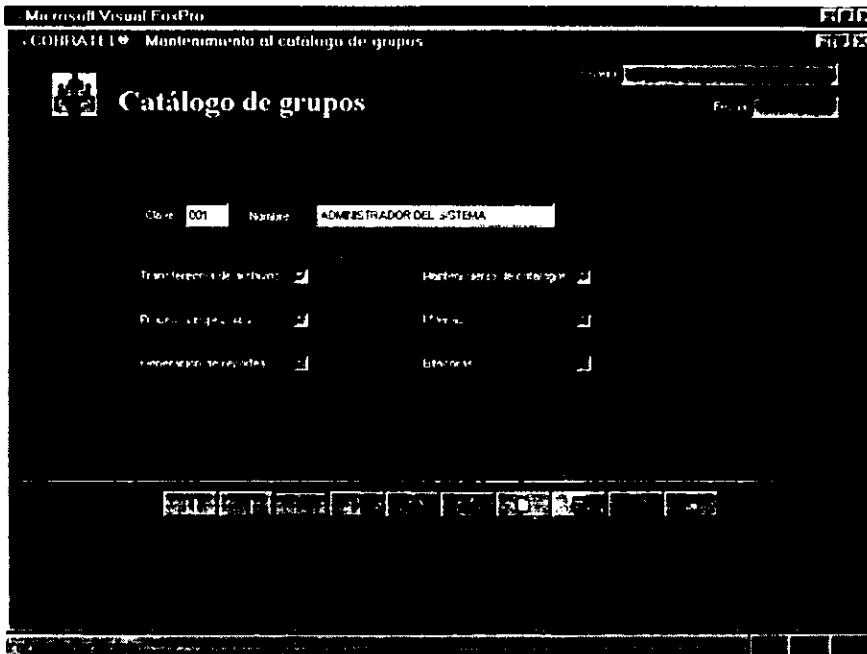


Figura 5 Pantalla de Mantenimiento al catálogo de grupos

Utilerías

Todas las operaciones que se realizan en este módulo, solo las puede realizar el administrador del sistema, ya que tienen impacto directo en la información de las tablas de las bases de datos.

Este módulo contiene los módulos siguientes:

- **Depuración**

 - **Depuración de la bitácora de procesos**

 - **Depuración de la bitácora de transmisiones**

 - **Depuración de la tabla de pagos**

 - **Depuración de la tabla de cobros**

- **Respaldo de archivos**

- **Restauración de respaldos**

- **Cambio de password**

El módulo de depuración, se encarga de eliminar los registros de la base de datos en el rango de fechas proporcionado, pudiendo depurar (eliminar) así la información de las bitácoras (de procesos y de transmisiones) y de las tablas de (pagos e histórico de pagos).

Al ejecutarse cualquiera de estas opciones, el sistema primero verifica que se haya realizado previamente el respaldo de la información, para no afectar la integridad de la misma en el Sistema de Cobranza por Terceros.

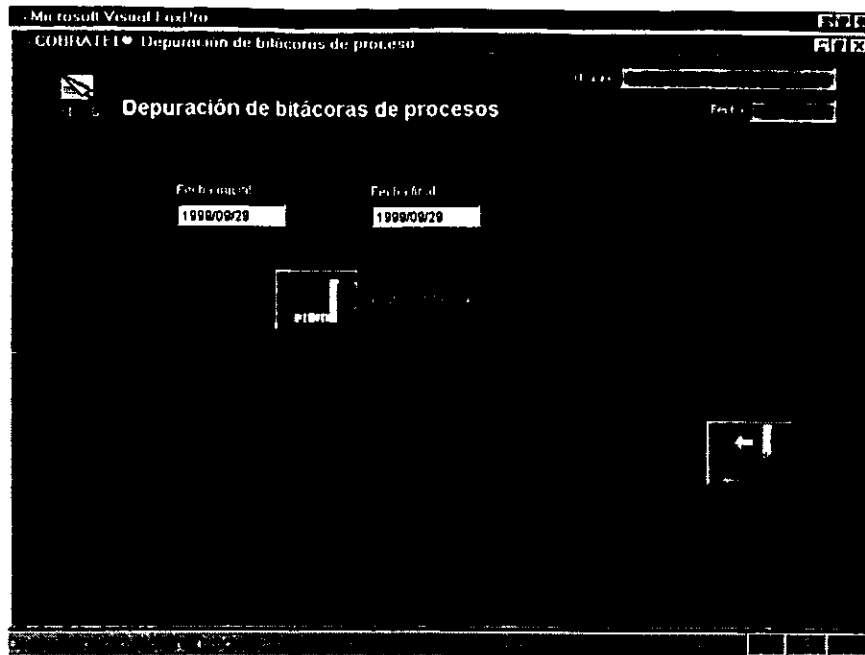


Figura 6 Pantalla de Depuración de la bitácora de procesos

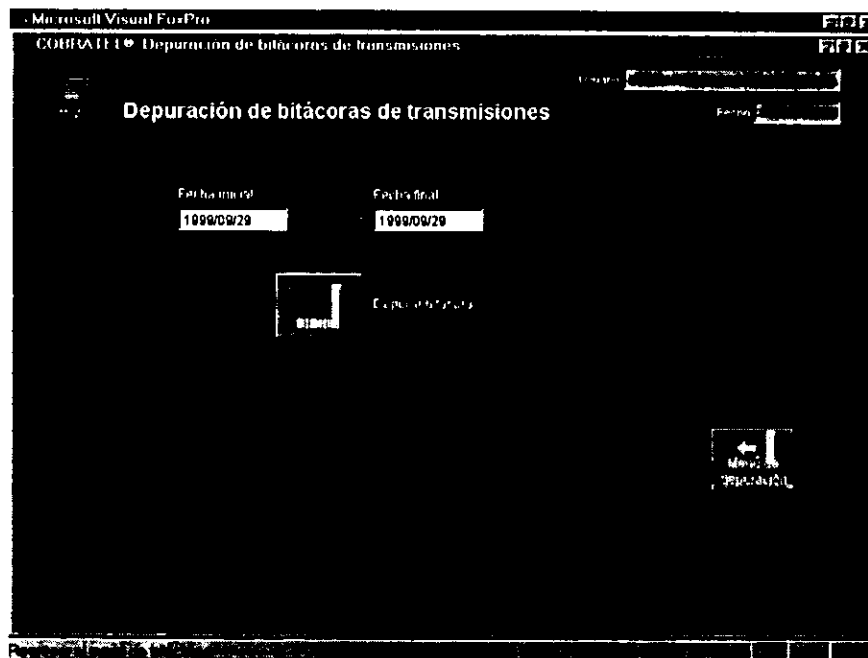


Figura 7 Pantalla de Depuración de la bitácora de transmisiones

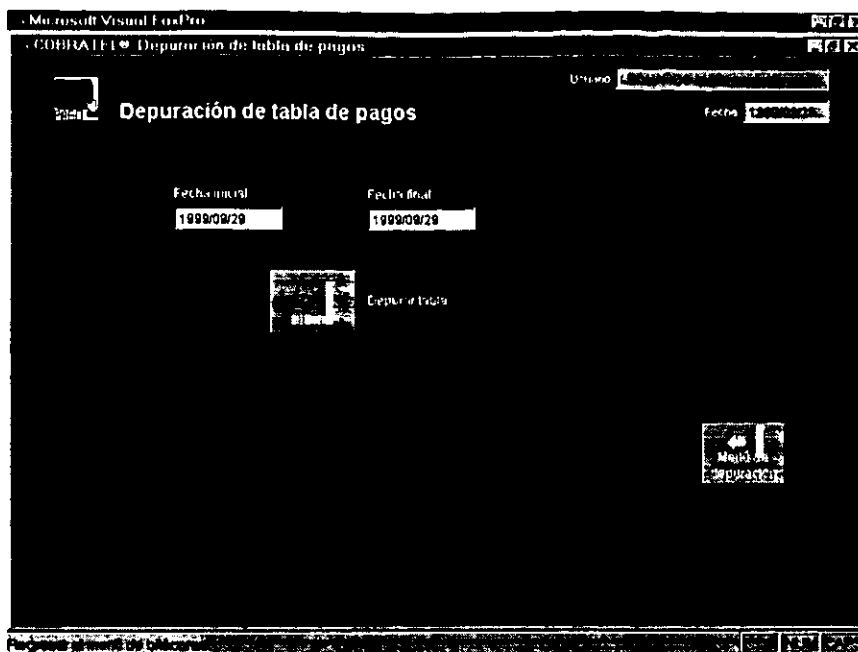


Figura 8 Pantalla de Depuración de la tabla de pagos



Figura 9 Pantalla de Depuración de la tabla de cobros

Para realizar el respaldo de la información del sistema, se debe ejecutar el módulo de Respaldo de archivos, con lo cual se puede guardar la información deseada en la unidad de respaldo, en la unidad de disco duro y/o en diskette.

Se recomienda guardar la información total del sistema por lo menos una vez al mes.

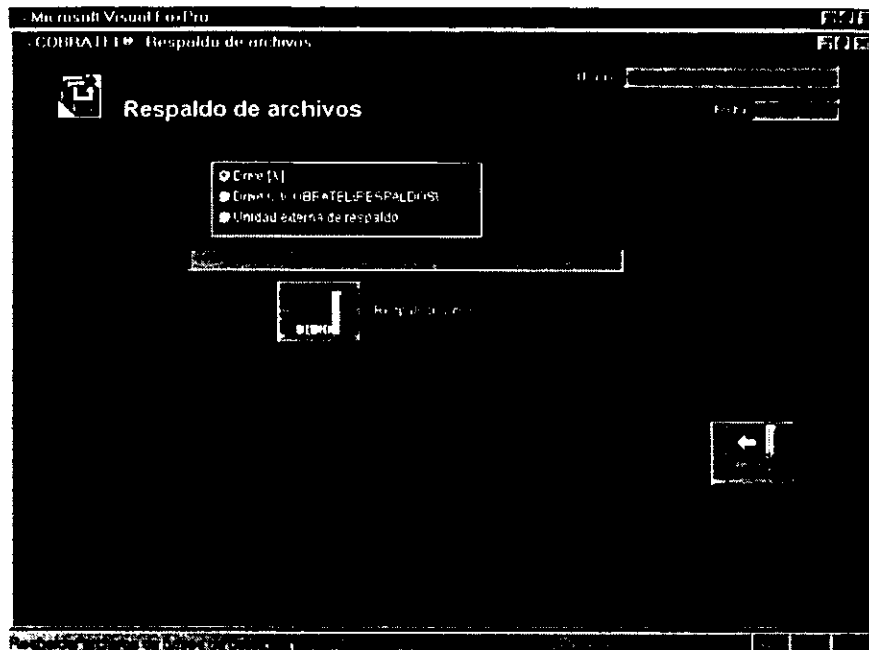


Figura 10 Pantalla de Respaldo de archivos

Para el caso en que se requiera información (respaldada) de meses anteriores, se puede realizar la operación de restaurar la información deseada, mediante el módulo de Restauración de respaldos.

El último módulo de estas utilerías es el de Cambio de password, por medio del cual se puede cambiar la contraseña de acceso al sistema para los usuarios.



Figura 11 Pantalla de Restauración de archivos

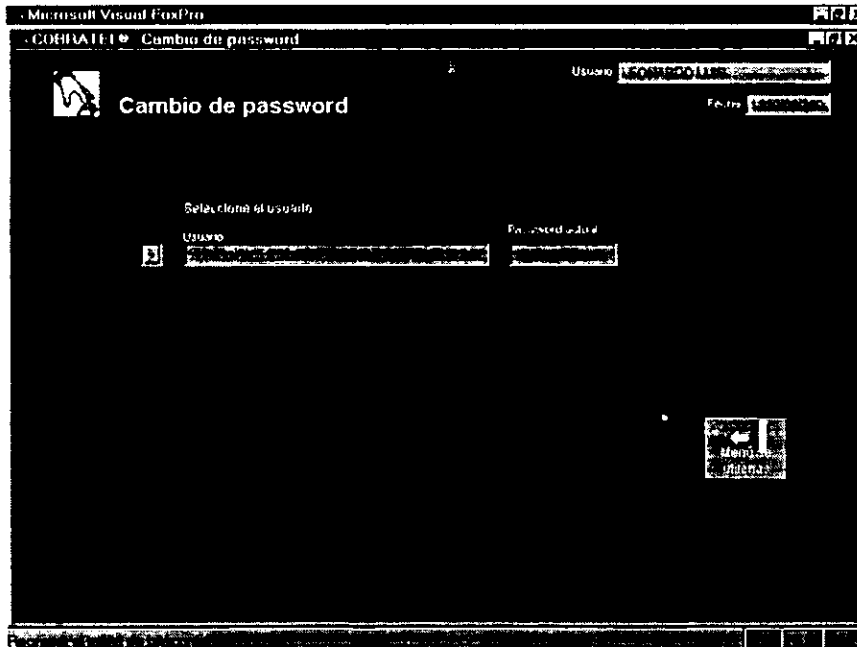


Figura 12 Pantalla de Cambio de password

A continuación se describen los módulos en los cuales también se realizan respaldos y restauraciones de archivos, así como altas, bajas, cambios y consultas en las tablas de pagos y cobros (Módulo de ajustes).

Respaldo de archivos Pisa

En este módulo se realizan copias de los archivos enviados a través de PISA.

Restauración de archivos de cobranza por terceros

Extrae copias de los archivos de cobranza almacenados anteriormente.

Módulo de Ajustes

Se encarga de realizar las operaciones necesarias a la tablas siguientes:

- Ajustes a la tabla de cobros
- Ajuste a la tabla de pagos

Las operaciones que realizan son:

≈ Altas/Agregar

≈ Bajas/Eliminar

≈ Modificaciones

≈ Consultas

Estas labores la realizan los usuarios del sistema con privilegios para ingresar a este módulo y pertenecen al grupo (002 Cobros Especiales), el cual les da la autorización debida para la utilización de estos módulos del Sistema de Cobranza por Terceros.

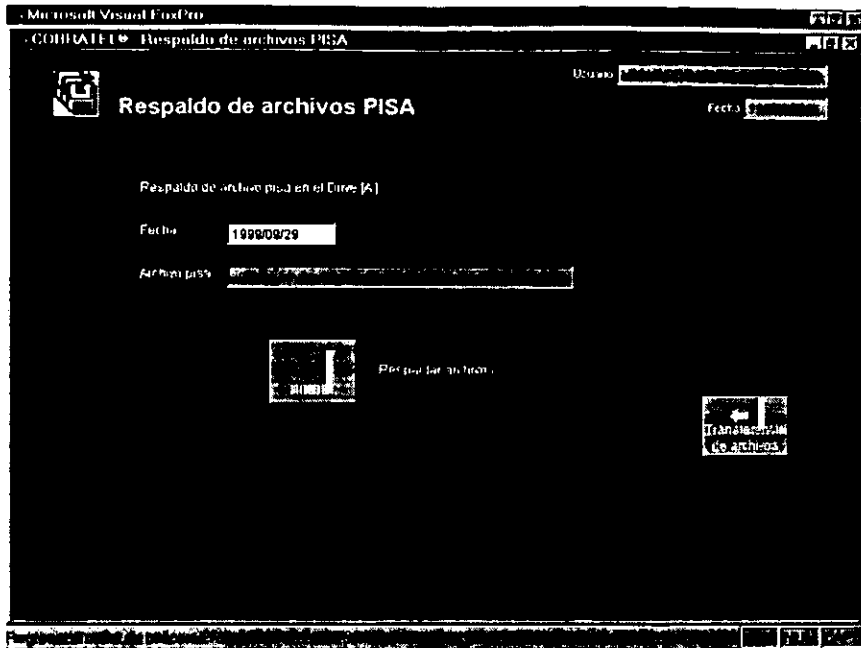


Figura 13 Pantalla de Respaldo de archivos PISA



Figura 14 Pantalla de Restauración de archivos PISA

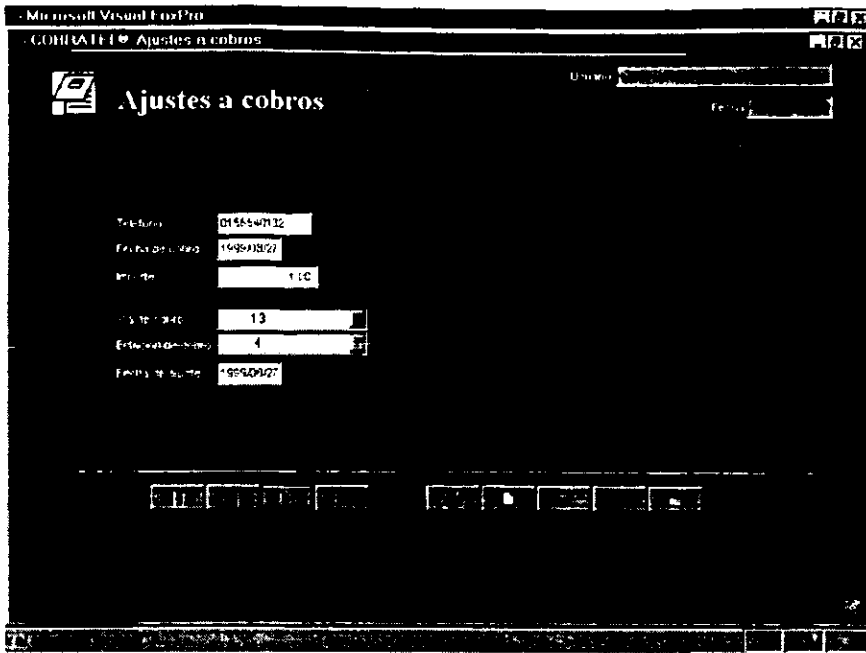


Figura 15 Pantalla de Ajuste a la tabla de cobros

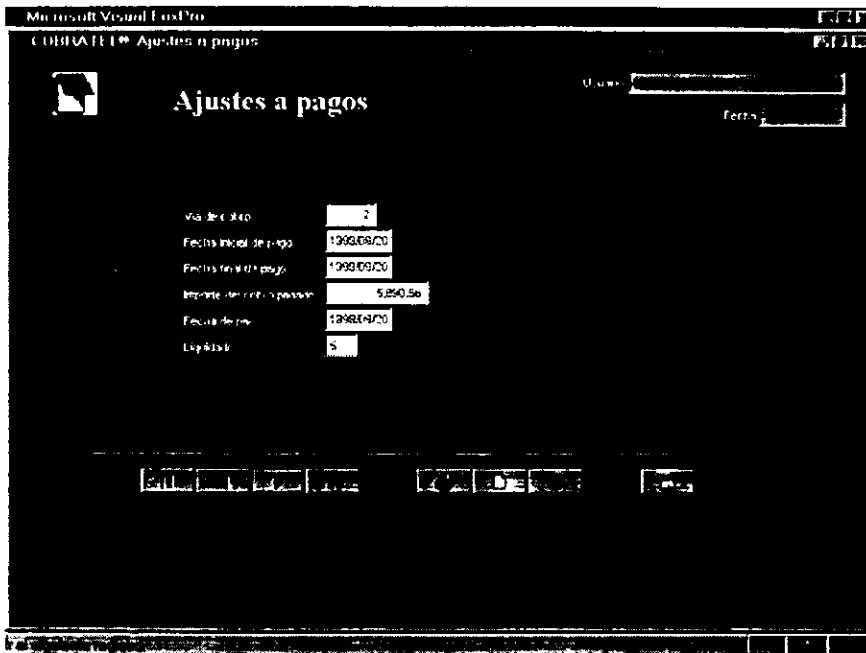


Figura 16 Pantalla de Ajuste a la tabla de pagos

MANUAL DE USUARIO

ÍNDICE

Estructura del directorio Cobranza Por Terceros

Diagrama de flujo

Procesamiento de información de la cobranza diaria

Procesamiento de información de cobranza

Normatividad del sistema

Descripción de procesos

Transferencia de archivos

 Transferencia y extracción de archivos

 Extracción de archivos

 Transmisión de archivos PISA

 Operaciones con archivos vía diskette

 Respaldo de Archivo PISA

 Restauración de archivos PISA

Procesos especiales

 Regeneración de archivo PISA

 Integración de cobranza

 Integración de terceros faltantes

 Módulo de ajustes

 Ajustes a la tabla de cobros

 Ajuste a la tabla de pagos

Generación de reportes

 Control de pagos

 Captura de pagos de agentes

 Diario de agentes

 Verificación de pagos de clientes

 Reporte diario de caja

 Reporte estadístico

 Reporte por vía de cobro

Mantenimiento de catálogos

Mtto. Catálogo de agentes

Mtto. Catálogo de Est. De Cobro

Mtto. Catálogo de usuarios

Mtto. Catálogo de grupos

Utilerias

Depuración

Depuración de bitácora de procesos

Depuración de bitácora de transmisiones

Tabla de pagos

Tabla de Cobros

Respaldo de archivos

Restauración de respaldos

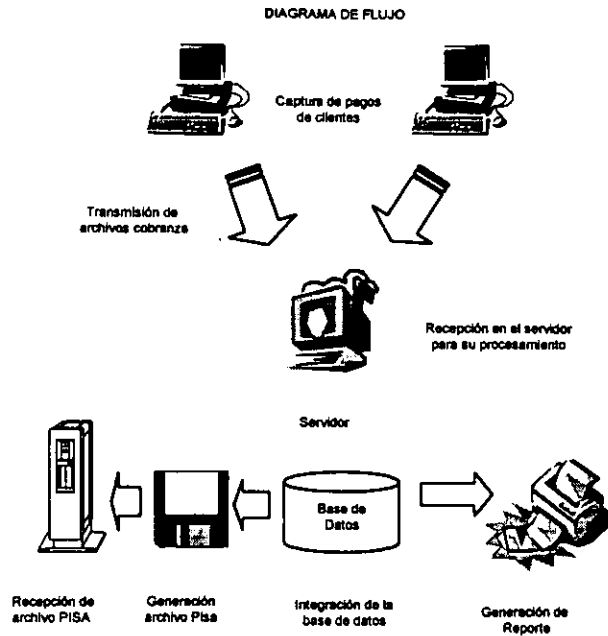
Cambio de password

Bitácoras

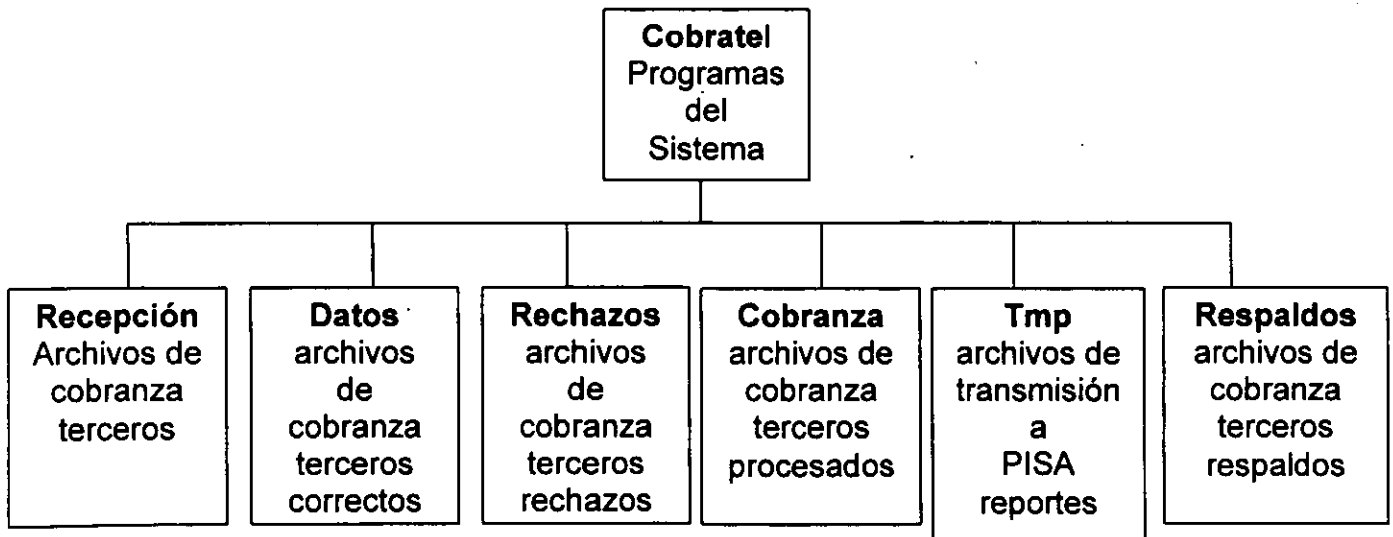
Consulta bitácoras de procesos

Consulta bitácora de transmisiones

Batch integración automática



ESTRUCTURA DEL DIRECTORIO COBRANZA POR TERCEROS



En el directorio **COBRATEL** se tienen todos los programas ejecutables del sistema pantallas y utilerías del mismo.

En el directorio **Recepción** es donde se reciben todos los archivos de la cobranza del día, de donde se tomarán para ser validados por el sistema.

En el directorio **Datos** se almacenan los archivos de cobranza que fueron validados y que se encuentran listos para ser procesados.

En el directorio **Rechazos** se almacenan los archivos de cobranza que fueron validados y a los cuales se les encontró errores y que están listos para ser corregidos.

En el directorio **Cobranza** se almacenan los archivos de cobranza que fueron procesados e integrados a la base de datos y que se encuentran listos para el respaldo.

En el directorio **Tmp.** se guardan todos los archivos generados por el sistema como son reportes, archivos de transmisión etc.

En el directorio **Respaldos** se guardan todos los archivos de cobranza para tener un respaldo de ellos.

PROCESAMIENTO DE INFORMACIÓN DE COBRANZA

Los diferentes terceros cobran los recibos de los clientes en sus puntos de venta, generando un archivo de cobranza por día el cual será transmitido vía FTP o bien por Diskette al servidor de la empresa COBRATEL y será depositado en el directorio /COBRATEL/Recepción.

El servidor COBRATEL procesara la información por lo menos tres veces al día, validara los archivos que se encuentran en ese momento en el directorio /COBRATEL/Recepción.

Los archivos correctos los depositara en el directorio /COBRATEL/Datos.

Los archivos incorrectos los depositara en el directorio /COBRATEL/Rechazos, existirá también un archivo llamado errores en el directorio /COBRATEL que desplegara los errores encontrados en los archivos, este archivo acumulará diariamente los errores

encontrados, pudiendo ser utilizados como histórico de errores, este no se eliminará hasta que el administrador del sistema lo borre, creandose automáticamente el proceso de integración: los errores de los archivos deberán ser movidos al directorio /COBRATEL/Recepción manualmente por el supervisor después de que haya concluido el proceso para que puedan ser procesados por medio de la opción: "Integración de terceros faltantes": Esta opción integrará uno por uno los archivos y no generará el archivo de transmisión PISA, teniendo que regenerar el archivo "Regeneración Archivo PISA" y posteriormente transmitirlo por la opción "Transferencia de archivo PISA"

NORMATIVIDAD EN EL SISTEMA.

- El sistema esta creado bajo la plataforma de Visual FoxPro 5.0 y en un ambiente Windos9X o NT, UNIX, SCO
- Cuenta con menús principales en forma horizontal y menús secundarios en forma vertical.
- Para salir de cada opción o menú es digitando el icono de salida
- El formato de todas las fechas de sistema es DDMMAAAA
- La sintaxis de los archivos de cobranza es ddmmaaaa.cxx

Donde:

dd	es	día
mm	es	mes
aaaa	es	año
c	es	cobranza
xx	es	siglas de la estación de cobro

- La sintaxis de los archivos de transmisión PISA es: Pisaddmmaa.vc

Donde:

vc	es	vía de cobro del archivo
999	es	todas las vías de cobro

DESCRIPCIÓN DE PROCESOS.

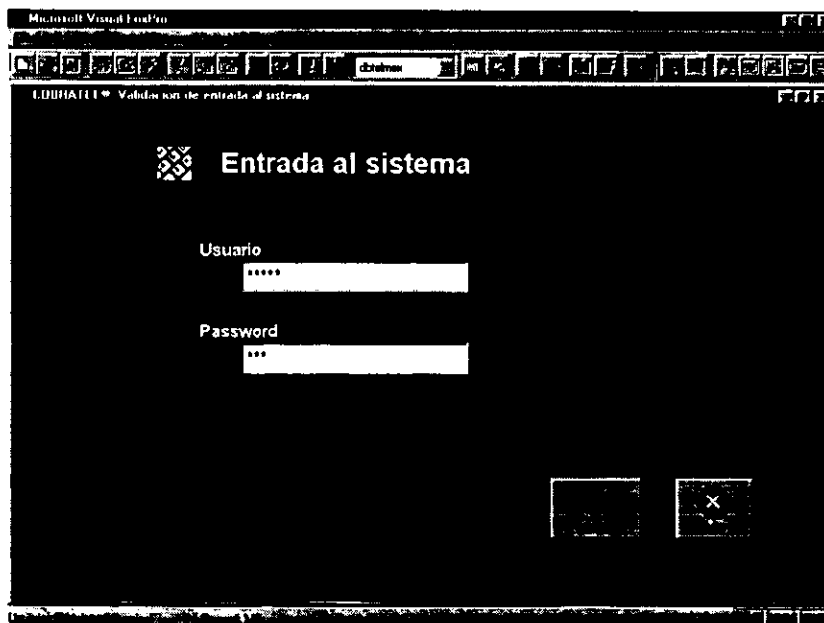
El programa se ejecutará inmediatamente al entrar con el usuario y password. Al salir del sistema, también saldrá de la sesión del servidor.

La primera pantalla del sistema se encarga de la seguridad ya que solo los usuarios autorizados tendrán la posibilidad de navegar en el Sistema de Cobranza por Terceros.

La pantalla cuenta con dos campos que deberán ser llenados sin excepción.

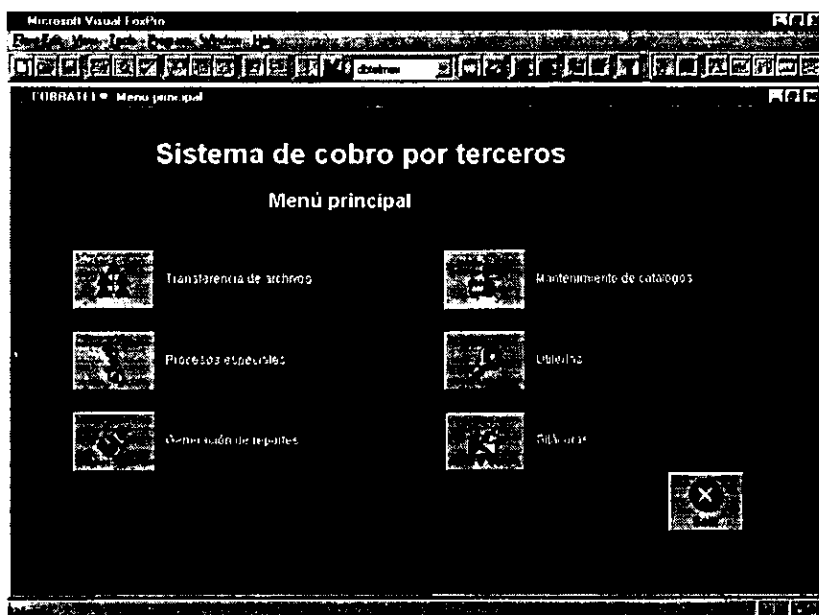
El primer campo solicita la clave de usuario

El segundo campo solicita el password de acceso, este deberá ser sumamente confidencial para evitar que otra persona ajena haga mal uso del sistema y por consiguiente su mal funcionamiento.



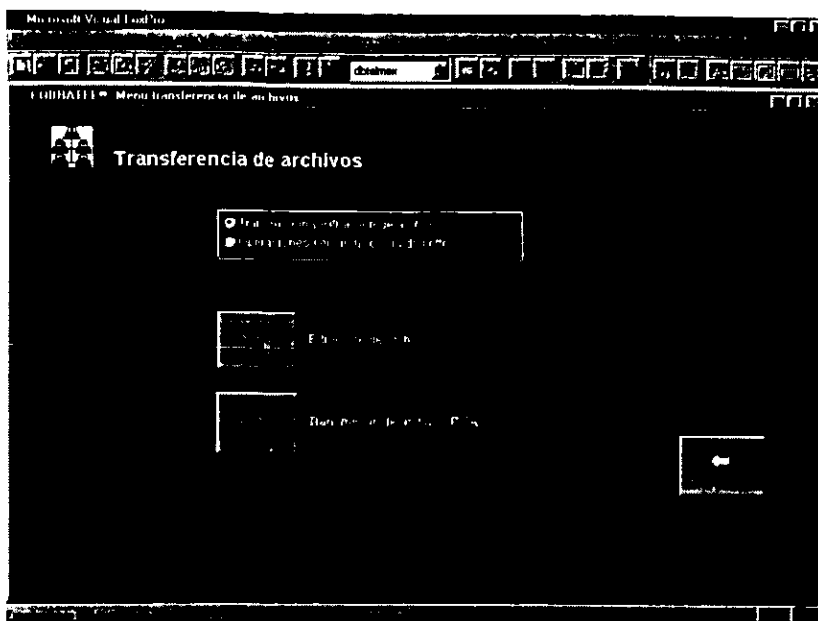
El usuario después de haber tecleado la clave de usuario tiene 3 intentos para poder teclear correctamente el password en caso contrario terminará la sesión con el servidor.

En caso que haya entrado al sistema, se ejecutara un programa que mostrara el menú principal.



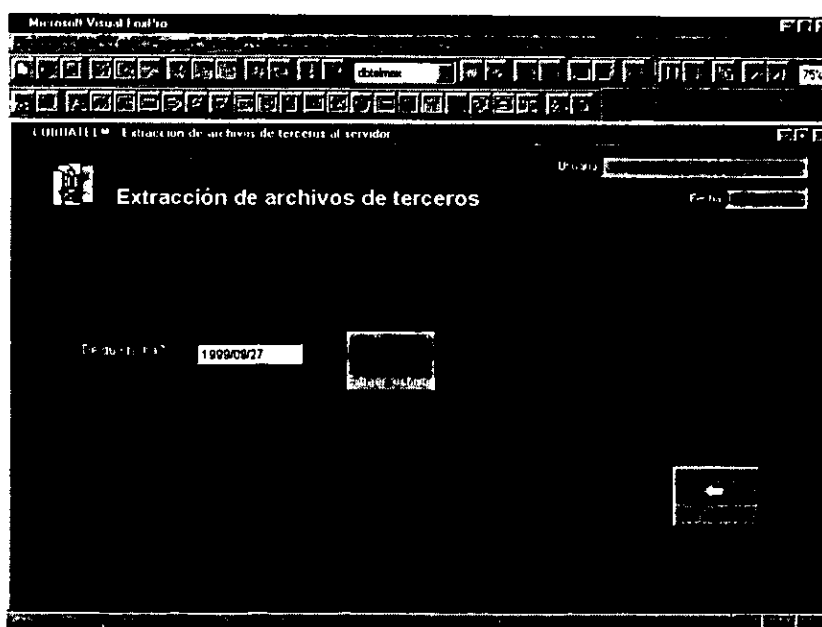
TRANSFERENCIA DE ARCHIVOS.

Objetivo: Mover los archivos del sistema PISA y cobranza por medio de transmisión vía FTP, así como por respaldo de archivos, para trasladarlos de un dispositivo a otro y así ser procesados.



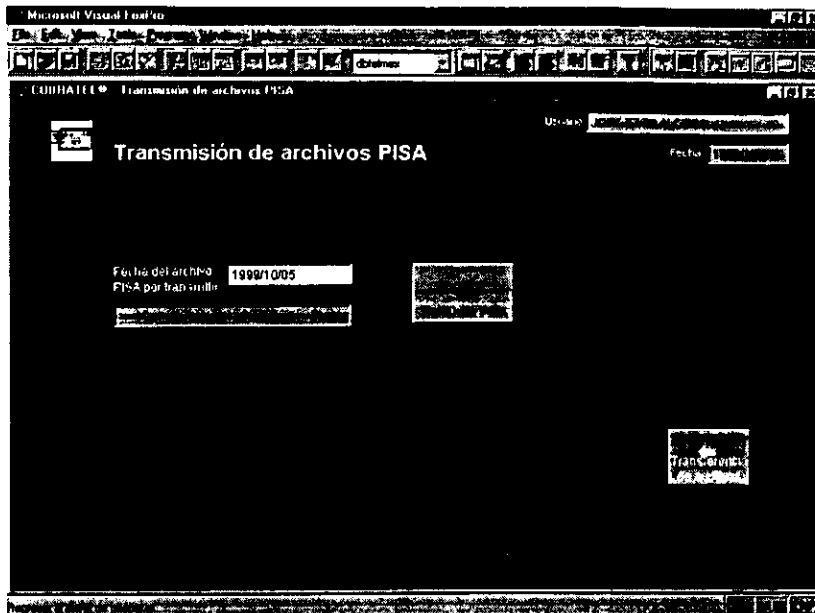
EXTRACCIÓN DE ARCHIVOS.

Objetivo: Sustraer el archivo de cobranza del tercero y depositarlo en el directorio /COBRATEL/Recepción para ser procesado.



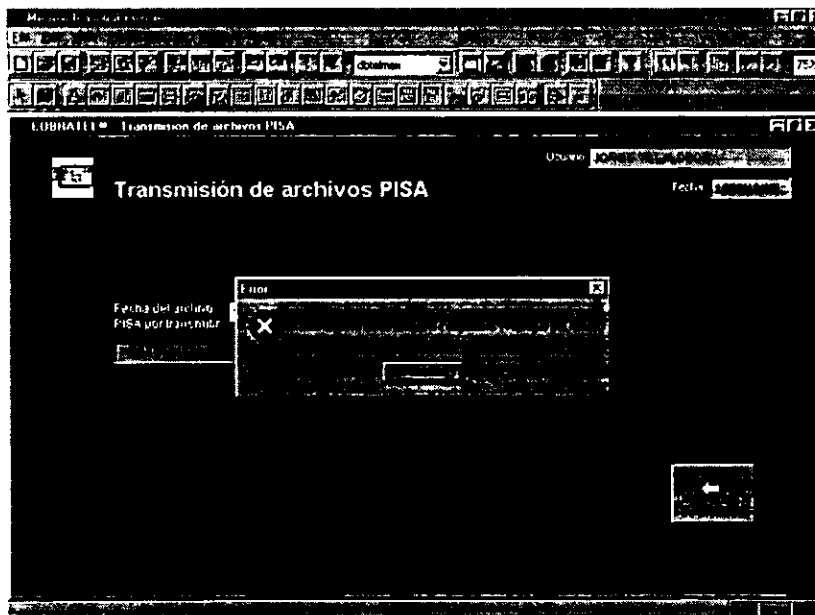
TRANSMISION DE ARCHIVOS PISA

Objetivo: Transferir el archivo PISA al MF de PISA



Despliega una pantalla donde nos solicita la fecha de proceso. Muestra una ventana que despliega el archivo que se va a transferir. El nombre del archivo de PISA será compuesto por la fecha.txt

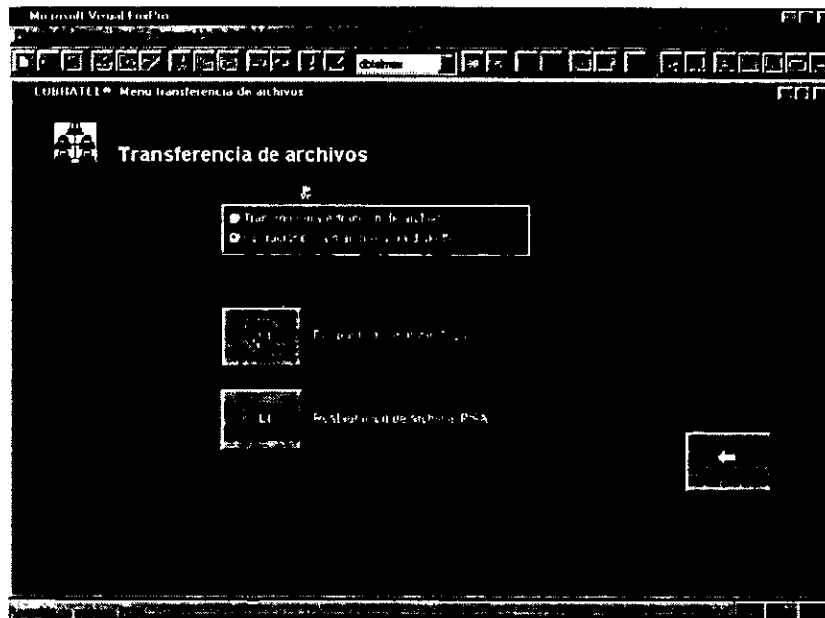
Si se escribe una fecha no existente, mostrara un mensaje que le dice "El archivo no se encuentra disponible ¡verifique por favor!".



Si el archivo se transmite correctamente, nos mostrara un mensaje que nos indicará que el archivo se transfirió exitosamente y nos regresara a la pantalla de “transmisión de Archivos PISA”. Si el archivo fue transmitido con anterioridad nos manda un mensaje donde indica que el archivo ya fue transmitido y deseamos enviarlo nuevamente.

OPERACIONES CON ARCHIVOS VÍA DISKETTE.

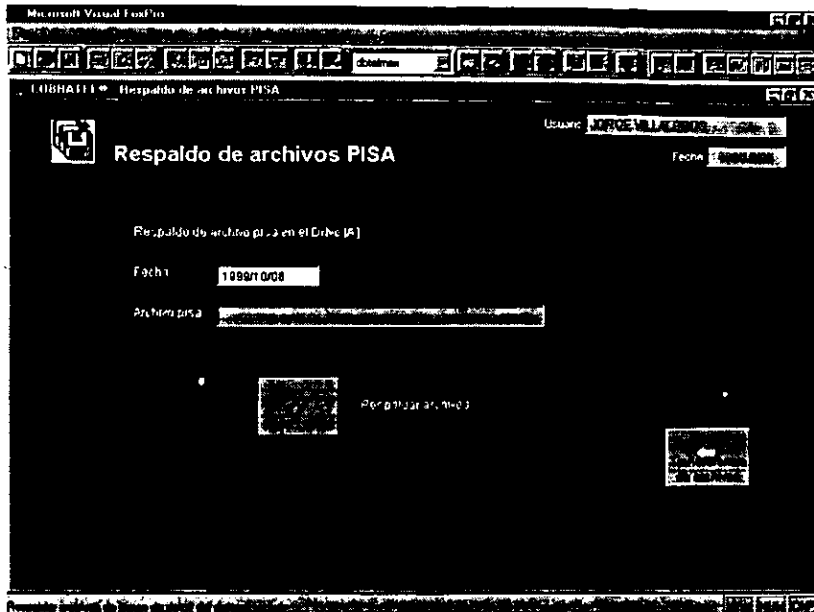
Aquí nos muestra una pantalla para elegir las opciones que son respaldo o restauración de archivos PISA



RESPALDO DE ARCHIVO PISA

Objetivo: respaldar el Archivo PISA en diskette.

Despliega una pantalla donde nos solicita la fecha de proceso.



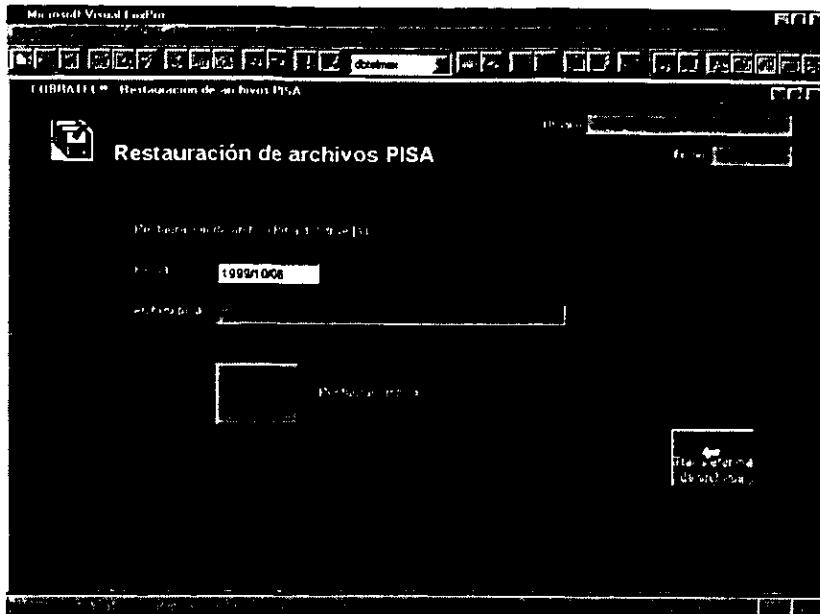
En caso de que no exista la fecha manda un mensaje que indicará "no existe el archivo PISA correspondiente."

Si el archivo existe, desplegara una ventana con la dirección del archivo que se va a respaldar. Manda un mensaje preguntando ¿"Esta preparada la unidad A?"; en virtud de que es donde se hacen estos respaldos. Una vez que se da respaldar archivo mostrara un mensaje que dice "el archivo fue respaldado."

En caso de que el diskette no este montado en la unidad, mostrará un mensaje: "unidad de disco no preparada, favor de verificar que este preparada".

RESTAURACIÓN DE ARCHIVOS PISA.

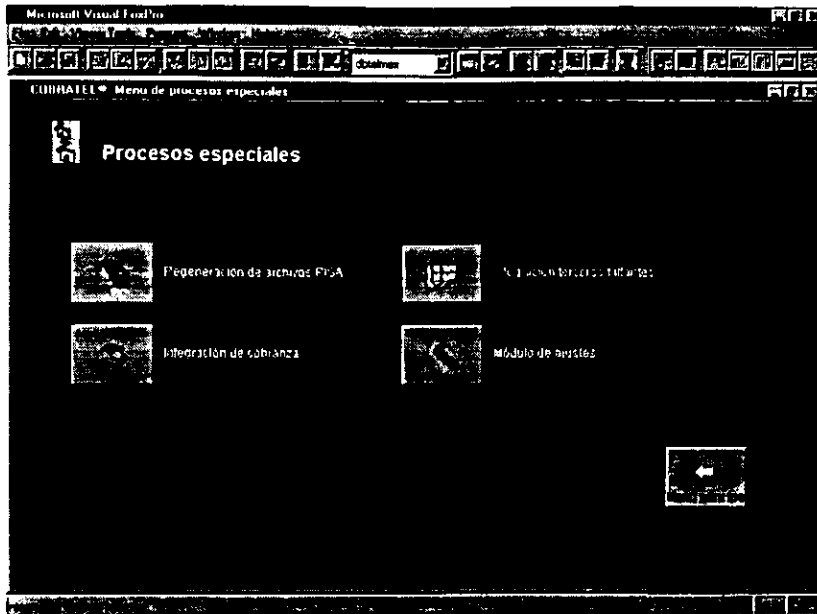
Objetivo: Restaurar de drive A, hacia disco duro donde se encuentran los archivos de cobranza para su procesamiento se mostrará una ventana donde indica el archivo a restaurar.



Se desplegará una ventana donde solicita la fecha del archivo a restaurar. Si encuentra el archivo con esa fecha procederá a restaurar el archivo en el directorio /COBRATEL/Recepción y mostrará un mensaje "archivo restaurado exitosamente." La PC donde se realice la restauración deberá tener un proceso corriendo, el cuál debe permitir el acceso a la unidad de disco A: para su lectura via FTP y considerar usuario, password y dirección IP.

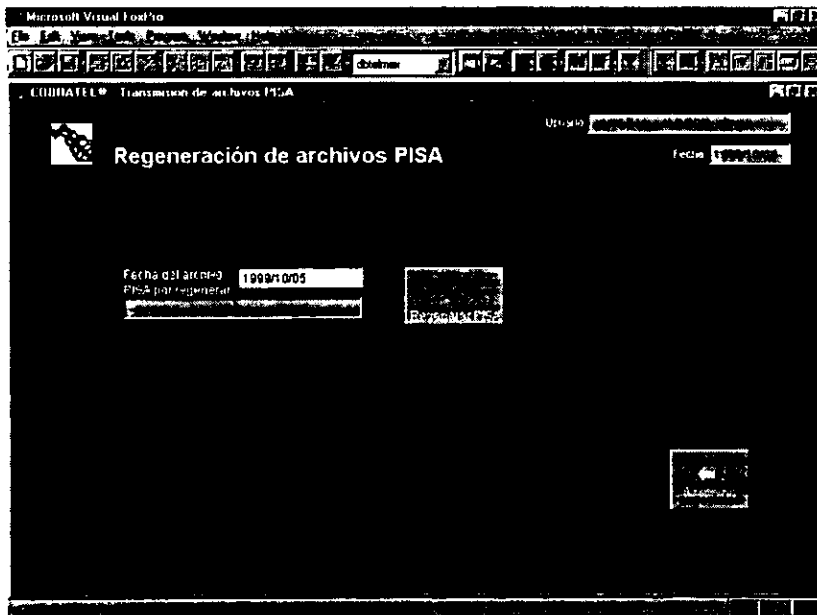
PROCESOS ESPECIALES.

Objetivo: Debido a su importancia, estos módulos se catalogaron como procesos especiales, ya que procesarán los datos de entrada con dos motivos, para transformarlos con otra estructura y así poder ser asimilados por otros dispositivos, y para ser cargados a la base de datos y estar listos para convertirse en información que el usuario requiera.



REGENERACIÓN DE ARCHIVO PISA.

Objetivo: Generar nuevamente el archivo PISA para su transmisión. Es importante hacer mención que la fecha que se desea procesar es la fecha de PROCESO, no la fecha de pago.

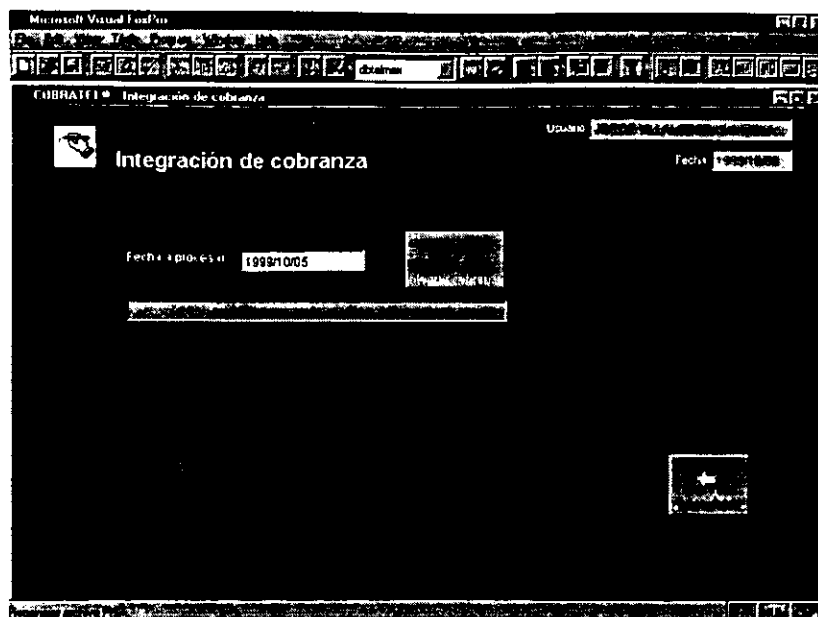


Esta opción es para los casos:

- Cuando haya integración de los terceros faltantes. El sistema procesa automáticamente la generación del archivo PISA por lo menos tres veces al día, pero en el caso de que hayan archivos de cobranza de terceros que hayan sido rechazados por el sistema se tiene la necesidad de verificarlos en el directorio rechazos e informar al responsable del envío de esos archivos para su corrección; por consiguiente el archivo PISA generado en el directorio Tmp. No incluye a los archivos rechazados, para que los incluya es necesario correr la opción del menú regeneración de archivos PISA después de haber sido recibidos e integrados con la opción de Integración de Terceros Faltantes. Y volver a enviar el archivo PISA antes haber borrado la transmisión hecha anteriormente por el proceso automático.
- En que ya no exista el archivo de transmisión PISA y se requiera transmisión nuevamente.
- Para generar el archivo PISA nuevamente. Para hacer una verificación de la generación del archivo PISA.

INTEGRACIÓN DE COBRANZA.

Una vez regenerados los los archivos deberá elegir esta opción, para integrar toda la cobranza en un solo archivo, a fin de poderlo transmitir.

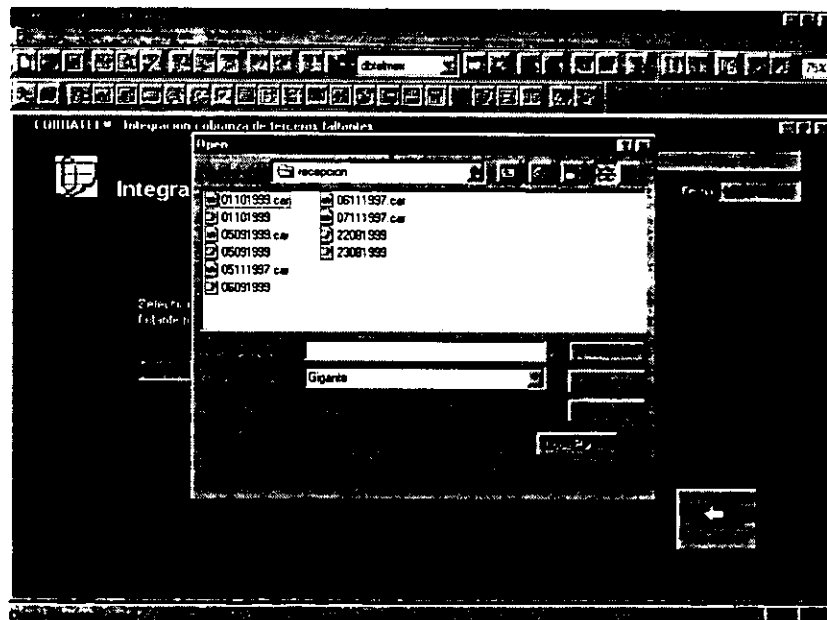
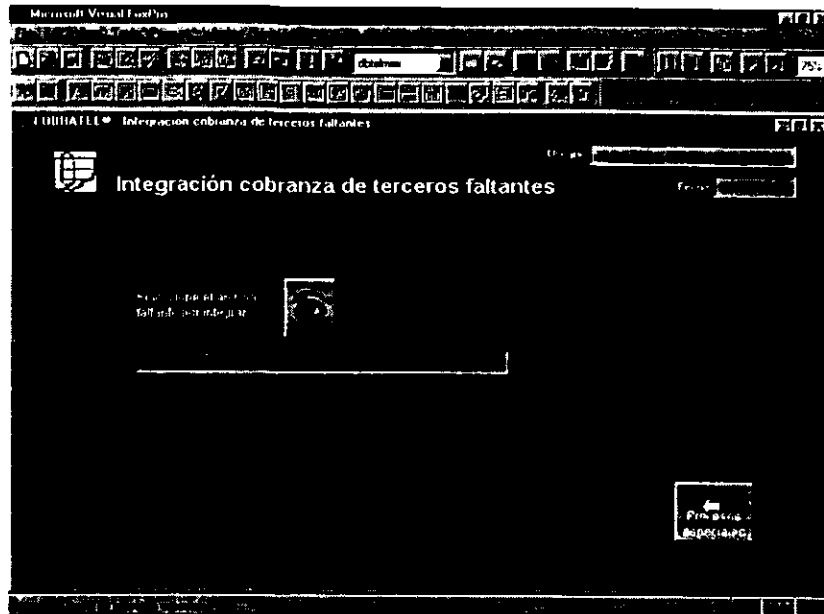


Esta opción integra todos los archivos de la fecha de proceso que se encuentren en el directorio /COBRATEL/Recepción y generará el archivo de transmisión PISA de la vía de cobro integrada en el día y posteriormente los transmitirá al servidor de PISA.

En el caso de que existan archivos rechazados es necesario corregir dichos archivos, moverlos al directorio recepción, correr la Integración de Terceros Faltantes de cada uno de los archivos rechazados, regenerar el archivo PISA y efectuar la transmisión del archivo PISA.

INTEGRACIÓN TERCEROS FALTANTES.

Esta opción integrará uno por uno los archivos y no generará el archivo de transmisión PISA, teniendo que regenerar el archivo opción; "Regeneración de Archivo PISA" de la vía de cobro que se haya generado, en el caso que ya se haya transmitido el archivo a PISA, solicitar al operador de PISA, que elimine la cobranza anteriormente transmitida de esa vía de cobro, para transmitirlo por la opción "Transferencia de Archivo PISA".



MÓDULO DE AJUSTES

Módulo que sirve para realizar ajustes en la tabla de cobros así como en la tabla de pagos.



AJUSTES A LA TABLA DE COBROS.

En esta opción del menú se podrán realizar altas, bajas, cambios, y consultas en la tabla de cobros.



Al seleccionar esta opción el sistema requerirá del número de teléfono de pago así como la fecha de cobro y el importe de cobro.

Al afectar el importe de cobro en cualquiera de los registros de estas tablas es necesario afectar la tabla de pagos y utilizar la opción de Ajuste a Tabla de Pagos

AJUSTE TABLA DE PAGOS

En esta opción del menú se podrán realizar cambios al registro de pagos, mediante la vía de cobro y la fecha inicial de pago, solo se podrá modificar el importe de cobro. Es importante mencionar que todo ajuste en la tabla de pagos deberá estar reflejado también con un ajuste en las tablas de cobros.

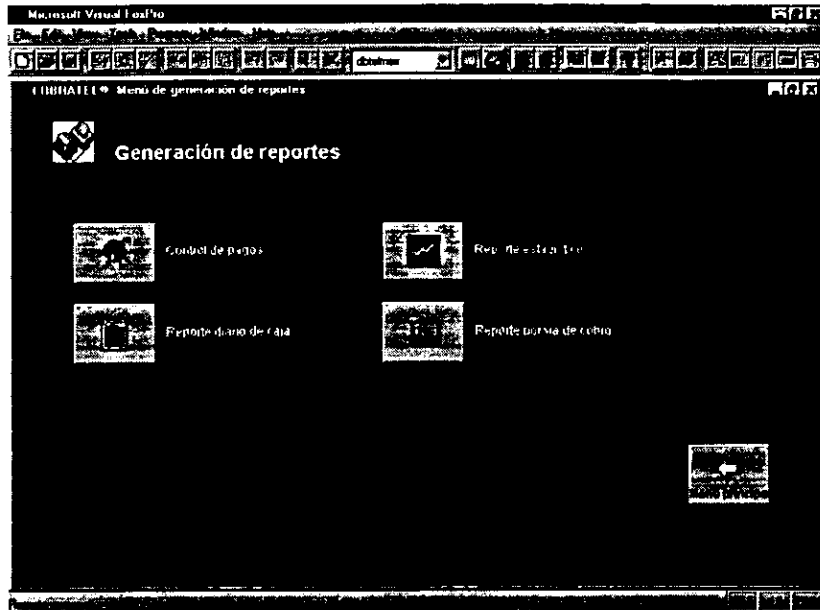
The screenshot shows a window titled "AJUSTES A PAGOS" within a Microsoft Visual Basic environment. The window has a title bar with "Microsoft Visual Basic" and "LIBRATERIA Ajustes a pagos". The main area contains the following fields:

- Un campo de texto para "U. Int." (Unidad de Intercambio) con un botón de selección de lista.
- Un campo de texto para "Fecha" con un botón de calendario.
- Un campo de texto para "Vía de cobro" con un botón de selección de lista.
- Un campo de texto para "Fecha de inicio de pago" con un botón de calendario.
- Un campo de texto para "Fecha final de pago" con un botón de calendario.
- Un campo de texto para "Importe del cobro (pagos)" con un botón de selección de lista.
- Un campo de texto para "Fecha de cobro" con un botón de calendario.
- Un campo de texto para "Límite" con un botón de selección de lista.

En la parte inferior del formulario, se encuentran tres botones de acción: "Aceptar", "Cancelar" y "Salir".

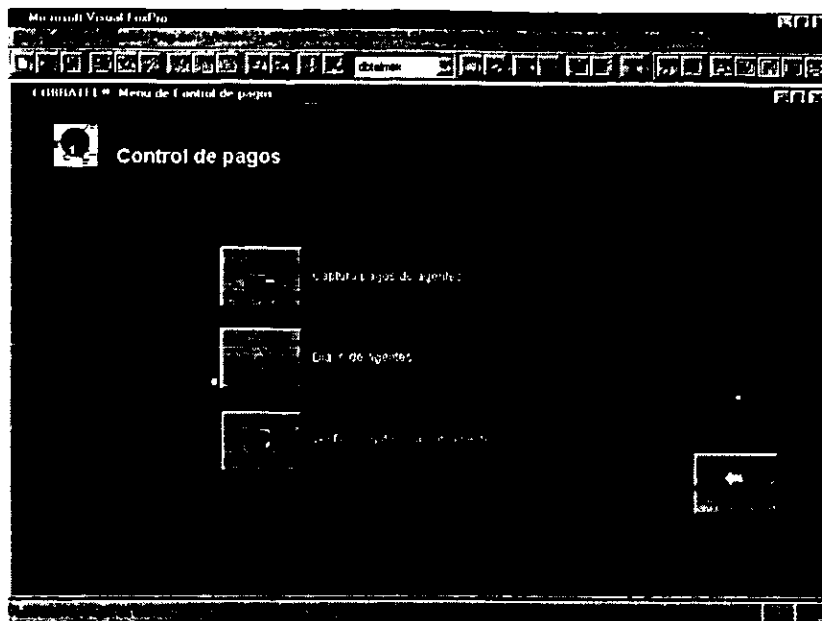
GENERACIÓN DE REPORTES.

Objetivo: Seleccionar los datos cargados previamente en la base de datos para transformarlos en información y desplegarlos ya sea por papel o pantalla proporcionando así la información solicitada por el usuario.



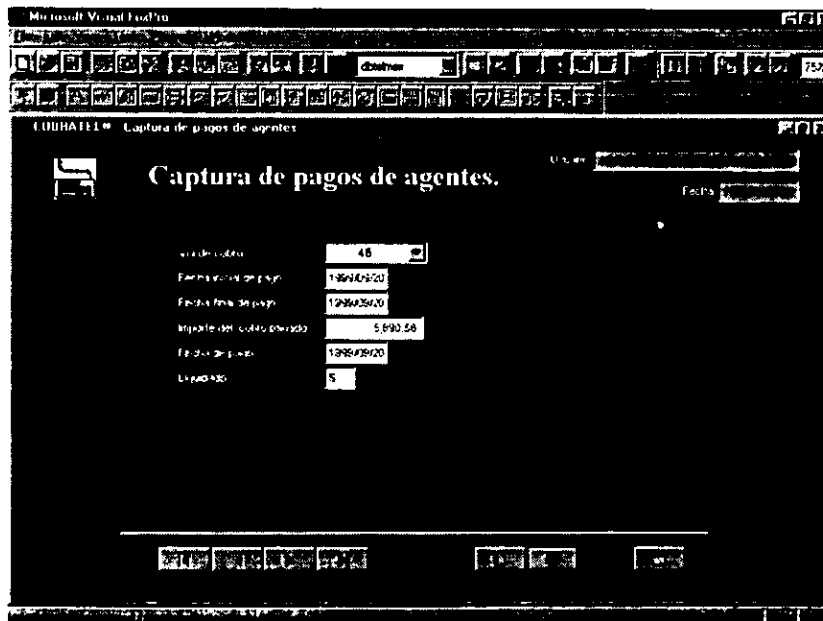
CONTROL DE PAGOS

Este módulo engloba la captura de pagos de agentes, el reporte diario de agente y la verificación de pagos de clientes.



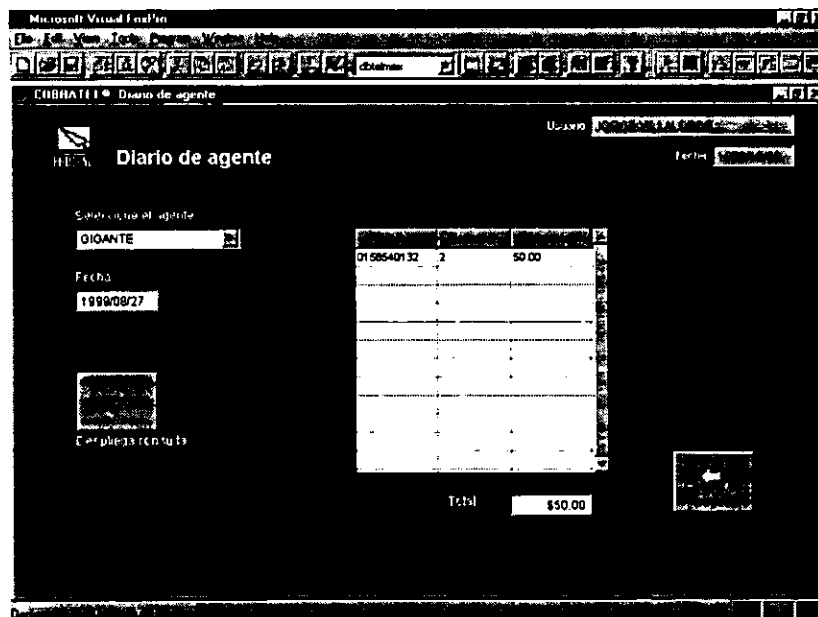
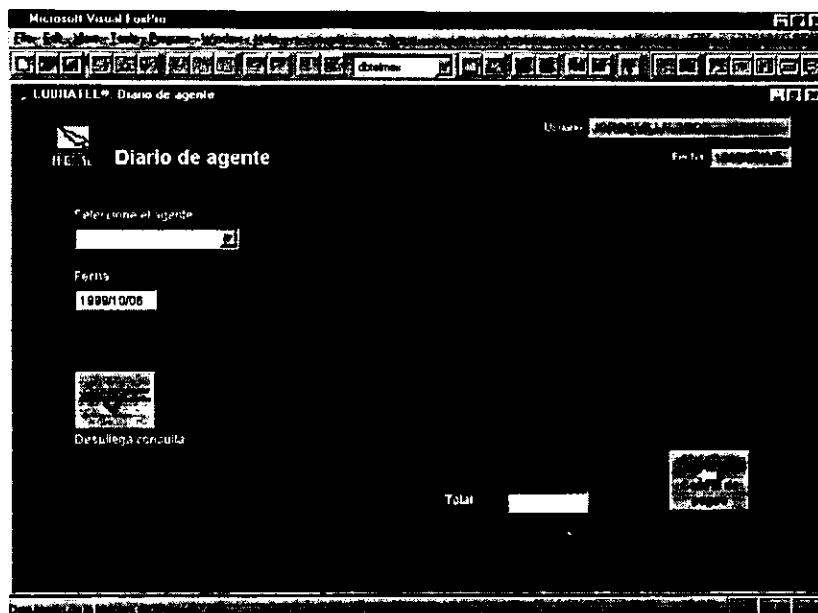
CAPTURA DE PAGOS DE AGENTES

Opción que sirve para capturar el importe del periodo cobrado por cada vía de cobro. En esta captura es necesario teclear la fecha de pago y el sistema consultará la vía de cobro, se capturará el importe del período, fecha de deposito en caso de que haya sido liquidado.



DIARIO DE AGENTE

Esta opción sirve para consultar el detalle de los cobros diarios de todas las estaciones de cobro del agente solicitado, mostrando la estación de cobro, teléfono e importe.

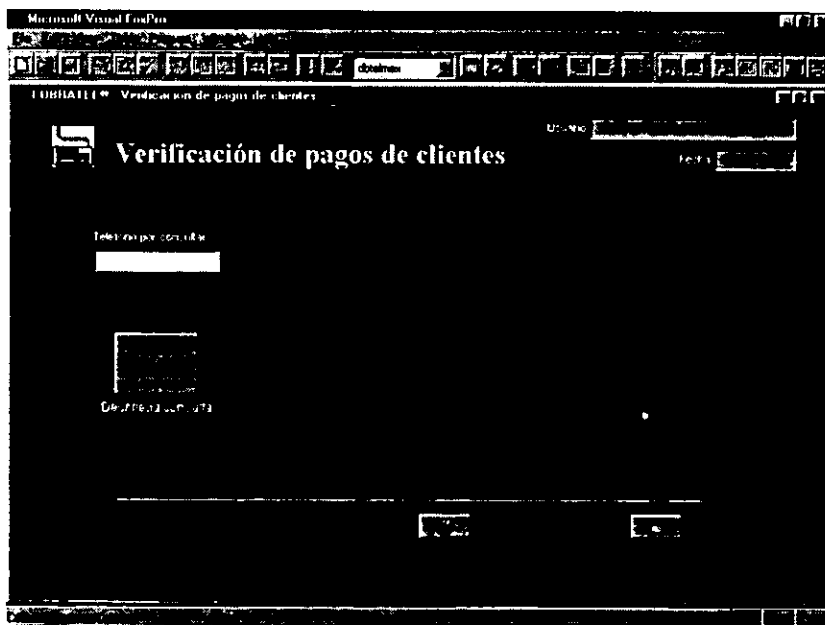


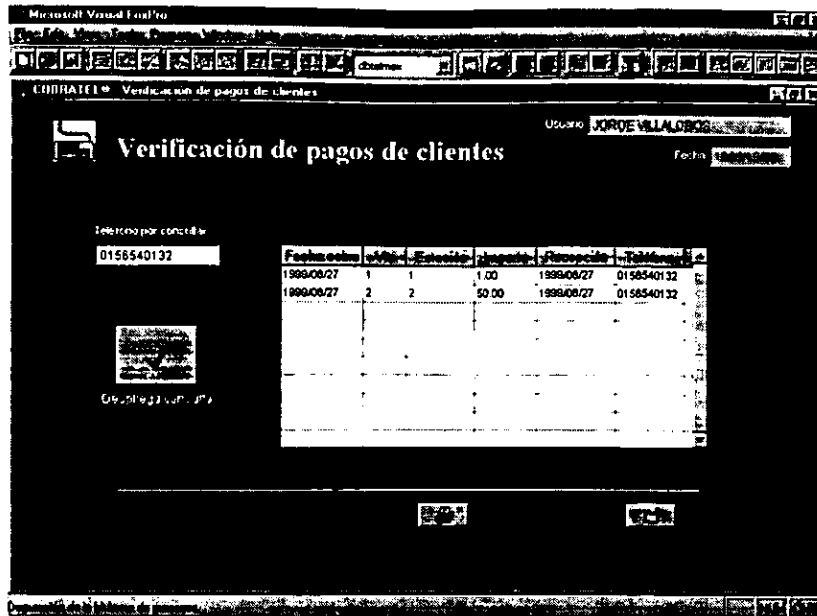
El sistema solicitará la fecha de cobro y la vía de cobro respectivamente, en el campo vía de cobro se podrá observar el catálogo de agentes.

Se puede utilizar la barra de herramientas que se muestra en la parte inferior para realizar las tareas mostradas.

VERIFICACIÓN DE PAGOS DE CLIENTES.

Consulta que sirve para verificar los pagos a detalle efectuados en los agentes por un número telefónico. En esta consulta se requiere teclear clave LADA y número telefónico seguido, y desplegará la fecha de pago, importe, fecha de transmisión a PISA, la vía de pago y la estación de pago. En la barra de herramientas que se muestra en la parte inferior se podrán realizar las tareas indicadas.





REPORTE DIARIO DE CAJA

Esta opción del menú se solicita la fecha de cobro y vía de cobro respectivamente, en el campo de vía de cobro. Se podrá observar el catálogo de agentes, esta consulta mostrará el total de recibos e importe total por cada una de las estaciones de cobro de dicho agente. En la parte inferior se muestra la barra de herramientas para realizar las diferentes opciones que se manejan.



Report Designer - estadística fin - Page 1 - Microsoft Visual Basic

Zoom

COMANDA		Reporte estadístico		Fin	
NO	DESCRIPCION	MON	DIA	MON	DIA
1	1.1.1.1	100	10/09/99	100	10/09/99
2	1.1.1.2	100	11/09/99	100	11/09/99
3	1.1.1.3	100	12/09/99	100	12/09/99
4	1.1.1.4	100	13/09/99	100	13/09/99
5	1.1.1.5	100	14/09/99	100	14/09/99
6	1.1.1.6	100	15/09/99	100	15/09/99
7	1.1.1.7	100	16/09/99	100	16/09/99
8	1.1.1.8	100	17/09/99	100	17/09/99
9	1.1.1.9	100	18/09/99	100	18/09/99
10	1.1.1.10	100	19/09/99	100	19/09/99
11	1.1.1.11	100	20/09/99	100	20/09/99
12	1.1.1.12	100	21/09/99	100	21/09/99
13	1.1.1.13	100	22/09/99	100	22/09/99
14	1.1.1.14	100	23/09/99	100	23/09/99
15	1.1.1.15	100	24/09/99	100	24/09/99
16	1.1.1.16	100	25/09/99	100	25/09/99
17	1.1.1.17	100	26/09/99	100	26/09/99
18	1.1.1.18	100	27/09/99	100	27/09/99
19	1.1.1.19	100	28/09/99	100	28/09/99
20	1.1.1.20	100	29/09/99	100	29/09/99
21	1.1.1.21	100	30/09/99	100	30/09/99
22	1.1.1.22	100	01/10/99	100	01/10/99
23	1.1.1.23	100	02/10/99	100	02/10/99
24	1.1.1.24	100	03/10/99	100	03/10/99
25	1.1.1.25	100	04/10/99	100	04/10/99
26	1.1.1.26	100	05/10/99	100	05/10/99
27	1.1.1.27	100	06/10/99	100	06/10/99
28	1.1.1.28	100	07/10/99	100	07/10/99
29	1.1.1.29	100	08/10/99	100	08/10/99
30	1.1.1.30	100	09/10/99	100	09/10/99
31	1.1.1.31	100	10/10/99	100	10/10/99
32	1.1.1.32	100	11/10/99	100	11/10/99
33	1.1.1.33	100	12/10/99	100	12/10/99
34	1.1.1.34	100	13/10/99	100	13/10/99
35	1.1.1.35	100	14/10/99	100	14/10/99
36	1.1.1.36	100	15/10/99	100	15/10/99
37	1.1.1.37	100	16/10/99	100	16/10/99
38	1.1.1.38	100	17/10/99	100	17/10/99
39	1.1.1.39	100	18/10/99	100	18/10/99
40	1.1.1.40	100	19/10/99	100	19/10/99
41	1.1.1.41	100	20/10/99	100	20/10/99
42	1.1.1.42	100	21/10/99	100	21/10/99
43	1.1.1.43	100	22/10/99	100	22/10/99
44	1.1.1.44	100	23/10/99	100	23/10/99
45	1.1.1.45	100	24/10/99	100	24/10/99
46	1.1.1.46	100	25/10/99	100	25/10/99
47	1.1.1.47	100	26/10/99	100	26/10/99
48	1.1.1.48	100	27/10/99	100	27/10/99
49	1.1.1.49	100	28/10/99	100	28/10/99
50	1.1.1.50	100	29/10/99	100	29/10/99
51	1.1.1.51	100	30/10/99	100	30/10/99
52	1.1.1.52	100	31/10/99	100	31/10/99

REPORTE ESTADÍSTICO.

Generar reporte de la cobranza para ser consultado en pantalla, escribiendo el mes año a consultar, se podrá observar el detalle de la cobranza de cada día, por vía de cobro.

Microsoft Visual Basic

Zoom

Reporte estadístico

Reporte estadístico

Uso: [] Factor: []

Mes: [SEPTIEMBRE] Año: [1999]

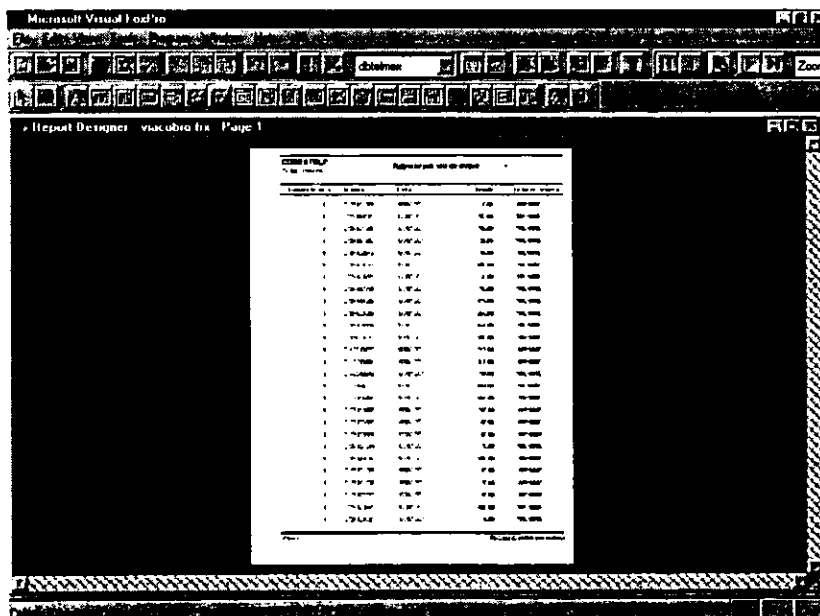
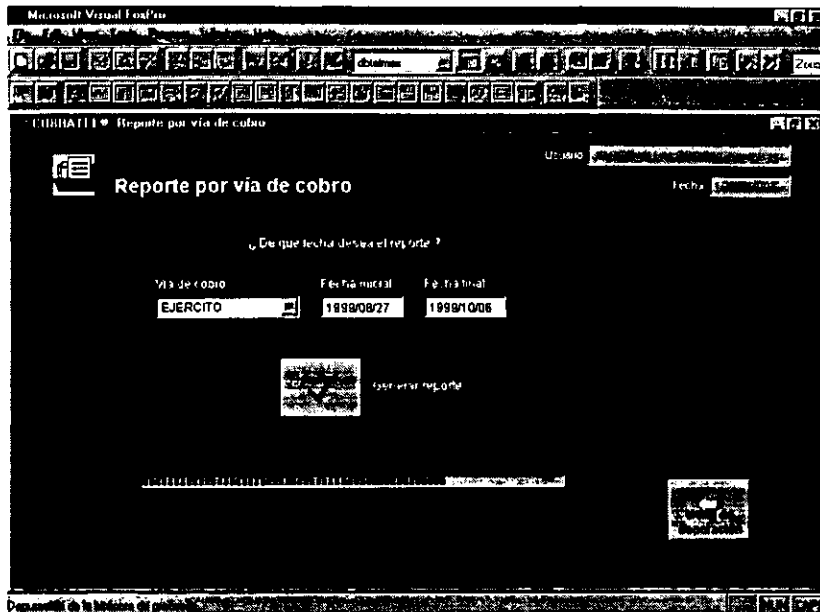
[]

[←]

REPORTE POR VÍA DE COBRO

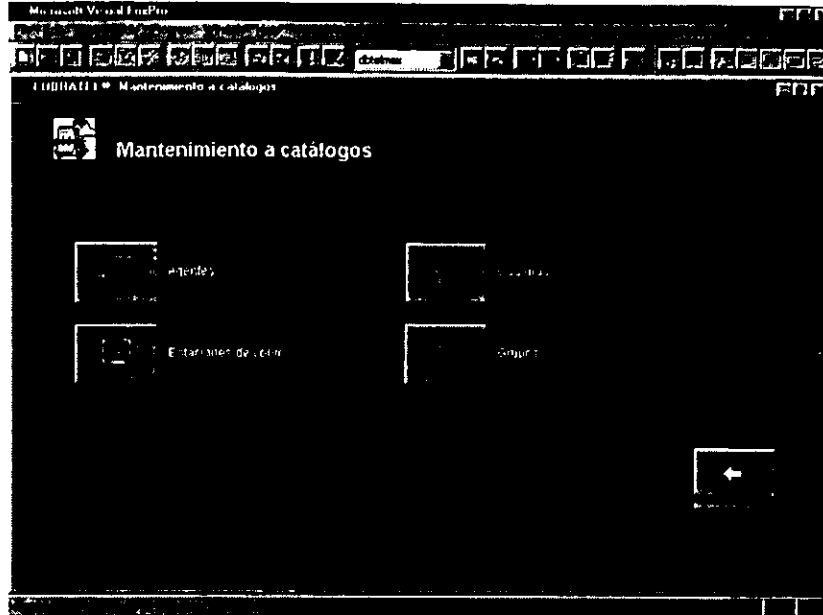
En esta opción del menú se solicita la vía de cobro y la fecha a consultar. Se mostrará estación de cobro, e importes por estación, así como sus totales.

Con la barra de herramientas mostrada en la parte inferior de la pantalla se podrán realizar las diferentes funciones.



MANTENIMIENTO DE CATÁLOGOS.

Objetivo: Tener actualizados los catálogos del sistema para tener registradas todas las entidades que conforman el Sistema de Cobranza por Terceros.



MANTENIMIENTO AL CATÁLOGO DE AGENTES.

Objetivo: Tener la opción de realizar altas, bajas, cambios consultas para tener actualizado el catálogo de agentes.



Solicitará los campos

Vía de cobro: Esta vía será proporcionado por la gerencia de cobros especiales el cual consta de tres dígitos numéricos

Clave de cobro: Es la clave que deberá traer el layout de los terceros en cual identifica a cada agente.

Nombre del agente, responsable y teléfono del responsable.

Al terminar de escribir el teléfono el sistema preguntará si el agente requiere de acuse de recibo. En las opciones de bajas y cambio que siempre estarán presentes consideran los mismos campos, en el caso de que hayas seleccionado consultas, el usuario podrá ver todos los agentes existentes así como su layout, nombre del agente, responsable, teléfono y si requiere acuse.

MANTENIMIENTO AL CATÁLOGO DE ESTACIONES DE COBRO.

Objetivo: Tener la opción de realizar altas, bajas, cambios consultar para tener actualizado el catálogo de agentes.

Visual Basic application window titled "Catálogo de estaciones de cobro". The window contains a form with the following fields:

Estación	
Numero	1
Dígito	12
Nombre de estación	CUERTO
Vía	CEJ
Dirección	CUERTO REAL 700
Responsable	JULIO APAMILLO
Tel. número	52502545
Tel. responsable	5750832

Al seleccionar la opción de mantenimiento al catálogo de estaciones de cobro se pedirá la vía de cobro, estación de cobro, nombre de estación de cobro, siglas de la estación de cobro, dirección, responsable de la estación de cobro, teléfono del responsable. El número de estación de cobro a mantener siendo un valor único para cada agente. Dentro de esta pantalla se podrá ver el catálogo de estaciones de cobro.

MANTENIMIENTO AL CATÁLOGO DE USUARIOS

Objetivo: Tener la opción de realizar altas, bajas, cambios, consultas para tener actualizado el catálogo de usuarios.

Microsoft Visual Basic

FOURATEL - Mantenimiento al Catálogo de usuarios

Catálogo de usuarios

Usuario

Nombre: D. ADRIEL

P. (Apellido): ALEXIA

Cargo: 11

Estación: ESTACION COBRO

Dirección: CALLE NO. 23

Teléfono: 85412267

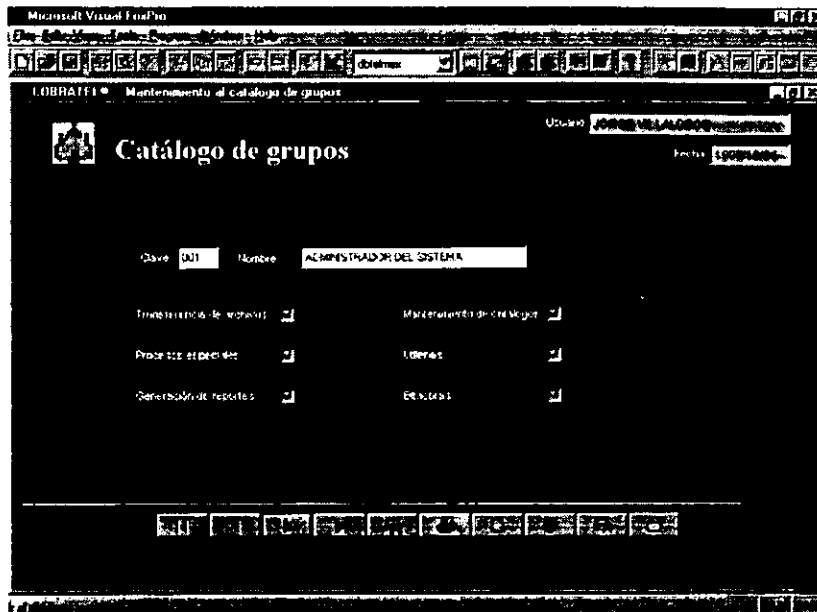
Vía de Cobro: 1

Al seleccionar la opción de Mtt. Al catálogo de usuarios, se pedirá la clave de usuario, el password del usuario, el grupo de usuario, la dirección, el teléfono, y en el caso de ser un usuario con el grupo consultar, pedirá vía de cobro.

MANTENIMIENTO AL CATÁLOGO DE GRUPOS

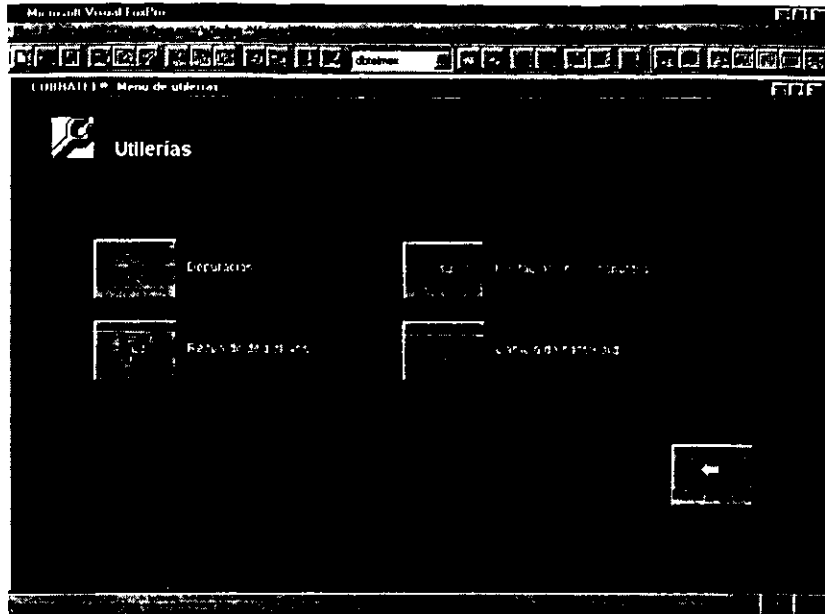
Objetivo: Tener la opción de realizar cambios consultas para tener actualizado el catálogo de usuarios.

En esta opción se podrá actualizar los derechos que tiene cada grupo de usuarios sobre cada una de las opciones con que cuenta este sistema.



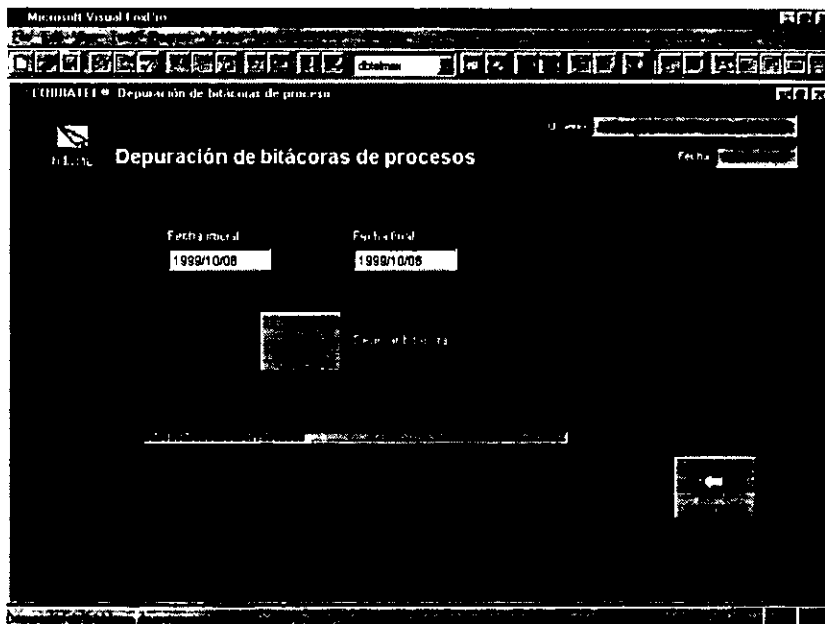
UTILERIAS.

Objetivo: Son programas que ayudan al administrador del sistema a mantener en condiciones optimas el sistema de cobranza por terceros y así poder dar un mejor servicio al usuario final



DEPURACIÓN DE BITÁCORA DE PROCESOS

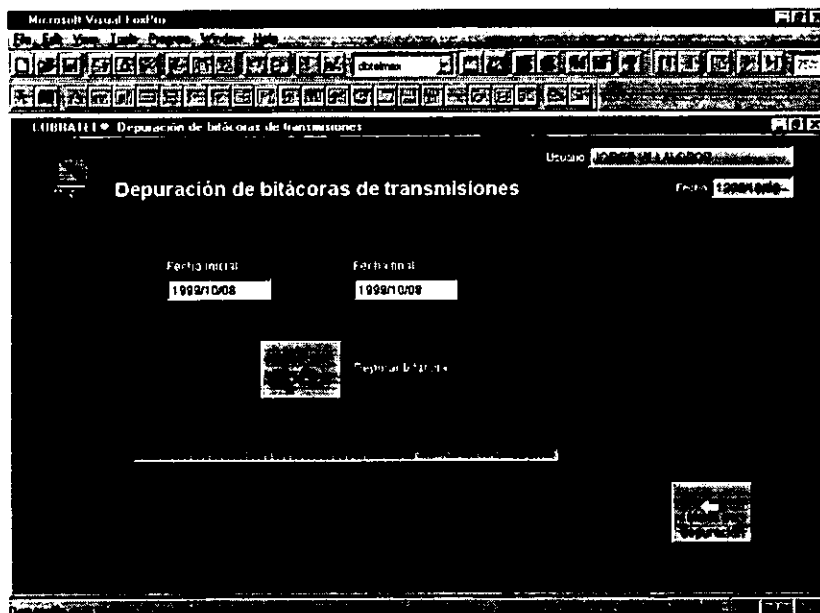
Objetivo: Tener la posibilidad de poder depurar la tabla b_procesos para mantener el performance del sistema, enviando la demora en los procesos por exceso de información, así como eliminar la información que ya no es de utilidad para el usuario.



Al seleccionar esta opción solicitará que teclee la fecha de inicio de depuración, así como la fecha final de depuración, este rango es el que será eliminado de la tabla b_procesos . es importante haber hecho el respaldo de la base de datos ya que en el caso de querer recuperar la información eliminada, solo tendremos oportunidad por medio de la restauración del respaldo de la base de datos.

DEPURACIÓN DE BITÁCORA DE TRANSMISIONES

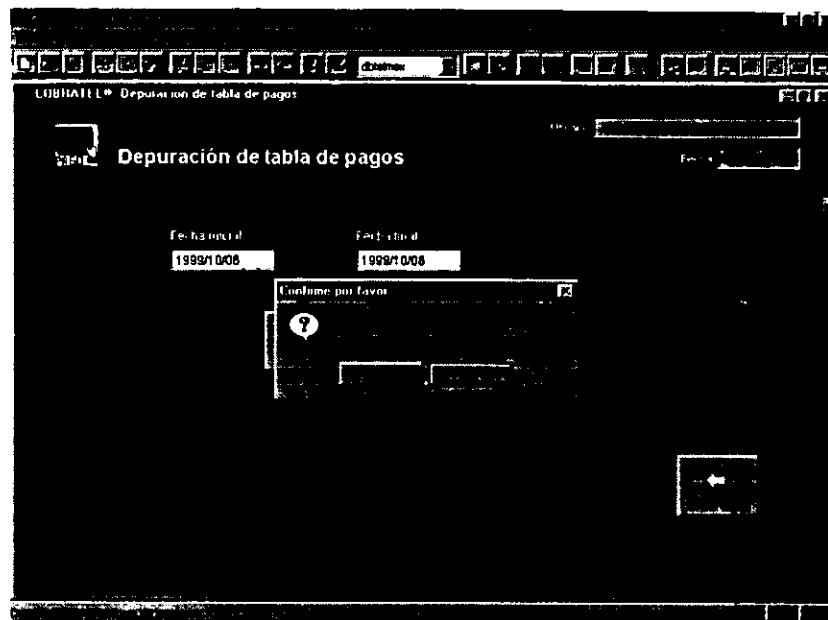
Objetivo: Tener la posibilidad de poder depurar la tabla b_transmisiones para mantener el performance del sistema, enviando la demora en los procesos por exceso de información, así como eliminar la información que ya no es de utilidad para el usuario.



Al seleccionar esta opción solicitará que teclee la fecha de inicio de depuración, así como la fecha final de depuración, este rango es el que será eliminado de la tabla b_transmisiones, es importante haber hecho el respaldo de la base de datos ya que en el caso de querer recuperar la información eliminada, solo tendremos oportunidad por medio de la restauración del respaldo de la base de datos.

DEPURACIÓN DE TABLA DE PAGOS

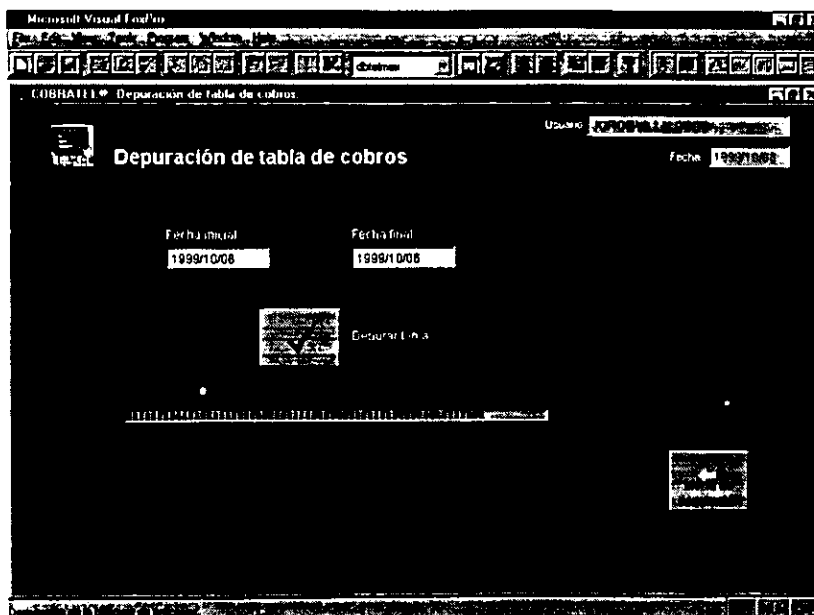
Objetivo: Tener la posibilidad de poder depurar la tabla de Pagos para mantener el performance del sistema, enviando la demora en los procesos por exceso de información, así como eliminar la información que ya no es de utilidad para el usuario.



Al seleccionar esta opción solicitará que teclee la fecha de inicio de depuración, así como la fecha final de depuración, este rango es el que será eliminado de la tabla de Pagos, es importante haber hecho el respaldo de la base de datos ya que en el caso de querer recuperar la información eliminada, solo tendremos oportunidad por medio de la restauración del respaldo de la base de datos.

DEPURACIÓN DE LA TABLA DE COBROS

Objetivo: Tener la posibilidad de poder depurar la tabla de Cobros para mantener el performance del sistema, enviando la demora en los procesos por exceso de información, así como eliminar la información que ya no es de utilidad para el usuario.



Al seleccionar esta opción solicitará que teclee la fecha de inicio de depuración, así como la fecha final de depuración, este rango es el que será eliminado de la tabla de Histórico de Cobros, es importante haber hecho el respaldo de la base de datos ya que en el caso de querer recuperar la información eliminada, solo tendremos oportunidad por medio de la restauración del respaldo de la base de datos.

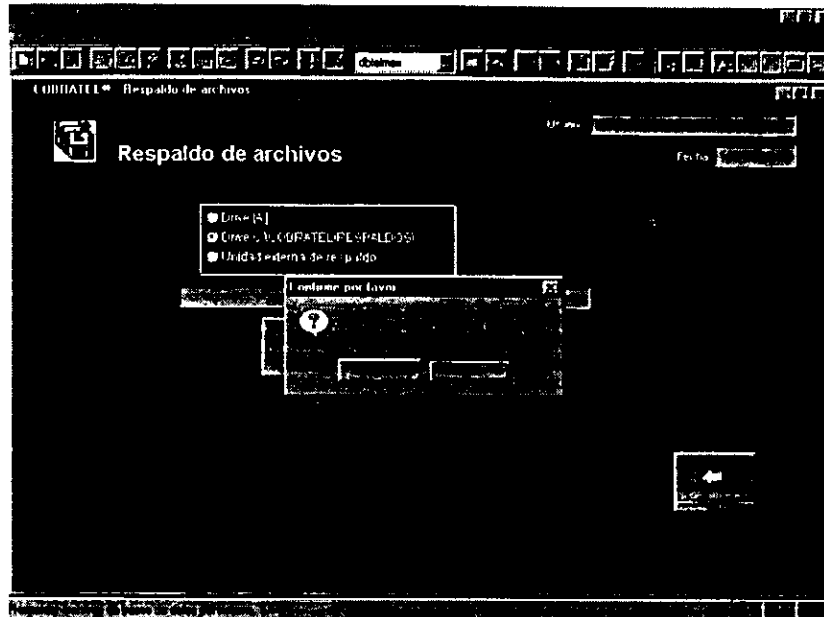
RESPALDO DE ARCHIVOS

Al seleccionar esta opción se presentará un menú para escoger entre respaldar la base de datos y respaldar los archivos de cobranza.

Si se hizo la elección de respaldo de la base de datos, es importante considerar que no debe existir ningún usuario ocupando el sistema. Es decir la base de datos no debe estar abierta por ningún usuario, en caso contrario el sistema desplegará un mensaje que indique que la base de datos esta abierta por otro usuario y preguntara si desea intentar de nuevo, en este lapso el supervisor del sistema tendrá que solicitar a los usuarios que salgan del sistema o en su defecto dar de baja los procesos que están involucrados a la base de datos. Una vez que no haya usuarios ocupando la base de datos dar respuesta afirmativa a la pregunta de intentar de nuevo o volver a correr la opción de respaldo de la base de datos.

Nota: Es importante etiquetar la fecha en la que se realizó el último respaldo, ya que para llevar a cabo la restauración es necesario saber este dato, de lo contrario tendrá que listar cuales son los respaldos que tiene el diskette y ver cual es la fecha del último respaldo, o buscar el respaldo de la fecha que se desee restaurar.

Al efectuar esta operación debe estar preparada un diskette nuevo en la unidad de disco.



Si se hizo la elección de respaldo de archivo de cobranza, el sistema pedirá la fecha de inicio de respaldo y la fecha final de respaldo, es decir el sistema respaldará en cualquiera de las opciones seleccionada por el usuario los archivos de cobranza comprendidos entre la fecha de inicio de respaldo y la fecha final de respaldo que se encuentre en el directorio /COBRATEL/cobranza.

RESTAURACIÓN DE RESPLADO.

Al seleccionar esta opción se permitirá un menú horizontal para escoger entre la acción de restaurar la base de datos y restaurar los archivos de cobranza.

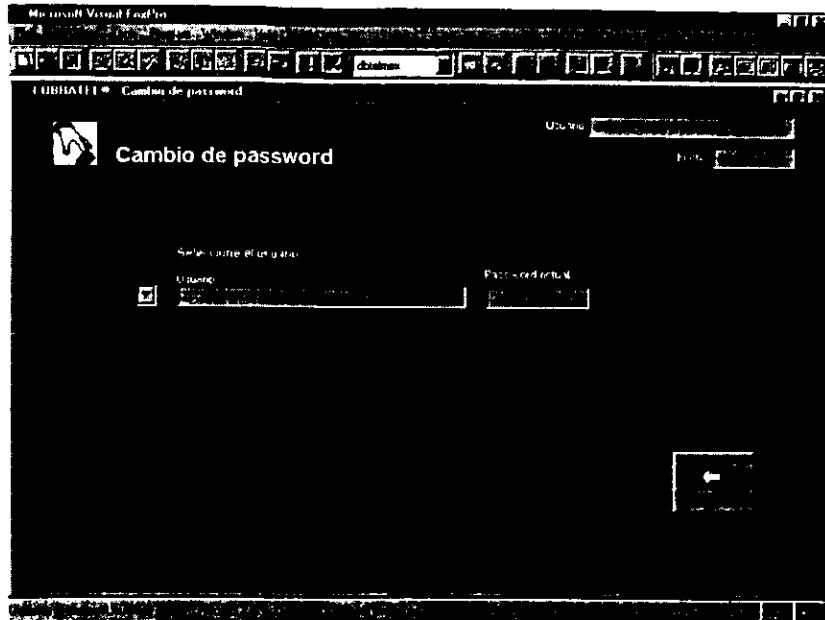


Si se elige la opción de restaurar la base de datos, es necesario que ningún usuario este ocupando la base de datos en ese momento, de lo contrario el sistema lo detectara. En esta acción de la base de datos actual será eliminada por el sistema, por lo cual se recomienda realizar un respaldo de la base de datos antes de hacer alguna restauración de la base de datos.

Si se elige la opción de restaurar los archivos de cobranza, el sistema tomará de la unidad donde se haya respaldado los archivos y los pondrá en el directorio /COBRATEL/recepción.

CAMBIO DE PASSWORD

Objetivo: Que el usuario de cualquier grupo tenga la opción de cambiar su password en intervalos de tiempo para evitar el acceso a personas y así mantener la integridad del sistema.



Para poderlo cambiar es necesario llamar al administrador del sistema a fin de que el pueda dar el acceso para poder cambiar el password. El administrador pondra el password como "nuevo", y dará la instrucción al usuario para que pueda entrar de esta forma al sistema y posteriormente podrá actualizar su password. Una vez actualizado lo mandara al menú principal.

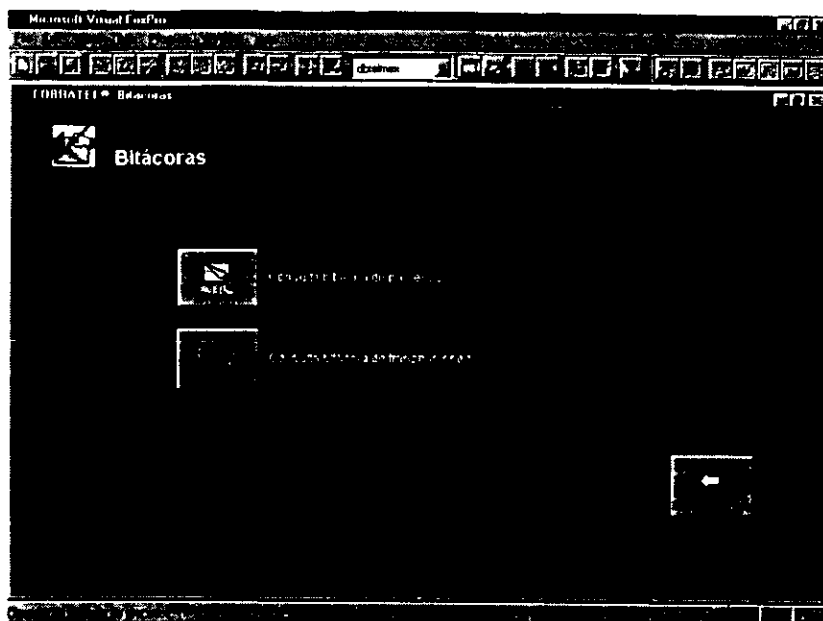


Cuando el usuario vuelva a entrar al sistema su password será "nuevo" tendrá que teclee la opción de cambio de password, se abrirá una pantalla solicitará el nuevo password así como la confirmación del mismo, al validar mostrará un mensaje que indica que el password fue actualizado exitosamente y podrá regresar a la pantalla de menú principal.



BITÁCORAS.

Objetivo: permiten tener un mejor control en todas las tareas realizadas por los usuarios del sistema, para determinar con que frecuencia se esta realizando cierta función además de indicar que usuarios efectuaron cierta acción. También ayuda a determinar que terceros transmitieron su cobranza, en que fecha a que hora y en que estado llego el archivo para así llevar el control diario de la cobranza recibida.



CONSULTA BITÁCORA DE PROCESOS.

Objetivo: Monitorear la frecuencia de las tareas en el sistema.

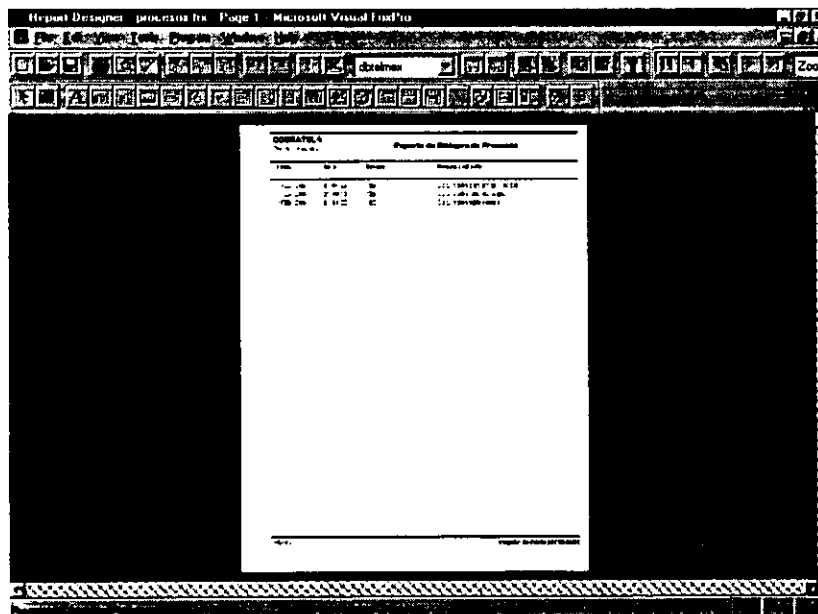
Al seleccionar esta opción despliega una pantalla en la cual nos pregunta la fecha inicial y final de la búsqueda de tareas a consultar, si el rango solicitado existe en la base de datos, desplegará los campos:

Operación: Indica que tarea se ejecuto

Usuario: Muestra que usuario fue el que realizó la tarea.

Fecha: Fecha de ejecución

Hora: Hora de inicio de ejecución.



CONSULTA BITÁCORA DE TRANSMISIÓN.

Objetivo: Consultar la bitácora del o los días solicitados para identificar las estaciones de cobro que se recibieron, que faltaron y que se encuentran con error.

De la misma forma que la bitácora de procesos, se deberá teclear la fecha inicial y la fecha final de consulta, en el caso que exista información en la base de datos, nos desplegará los siguientes campos:

V. cobro: Vía de cobro

Estación: Estación de cobro

Fecha: Fecha de recepción de la cobranza

Hora: Hora inicial de recepción

Estado: Faltante, Correcto o Incorrecto.



CONCLUSIONES:

Se dice que para poder acceder a una buena decisión y un buen proyecto en el momento preciso, se debe de tener una Planeación Estratégica.

Por tal motivo algunas de las conclusiones a las que hemos llegado son:

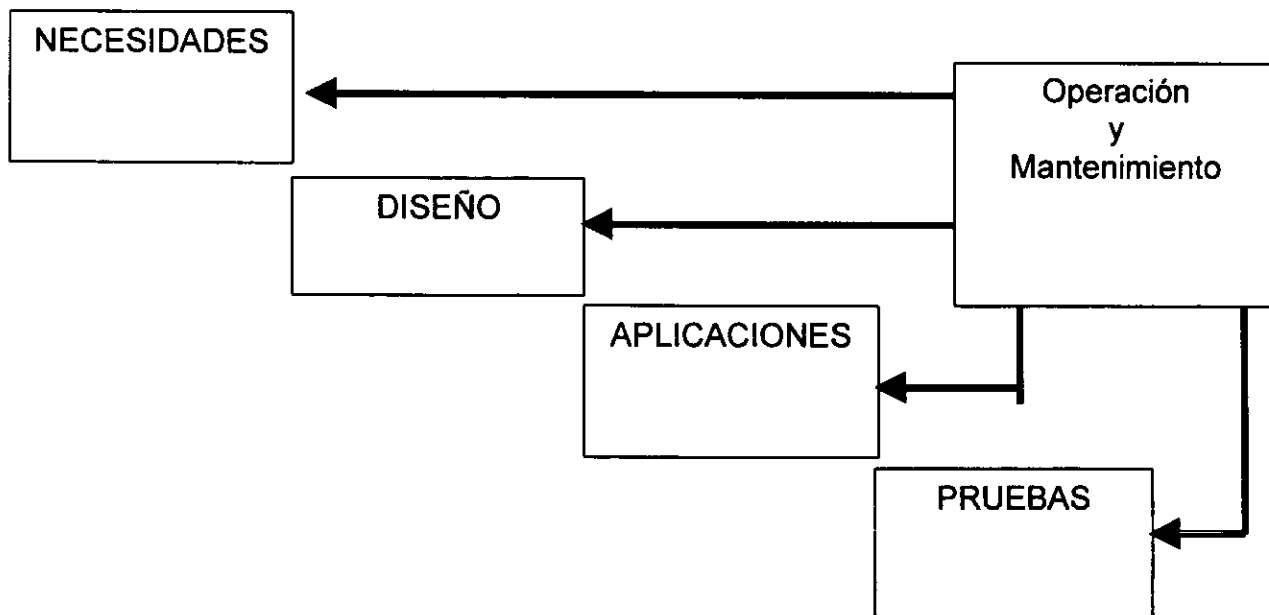
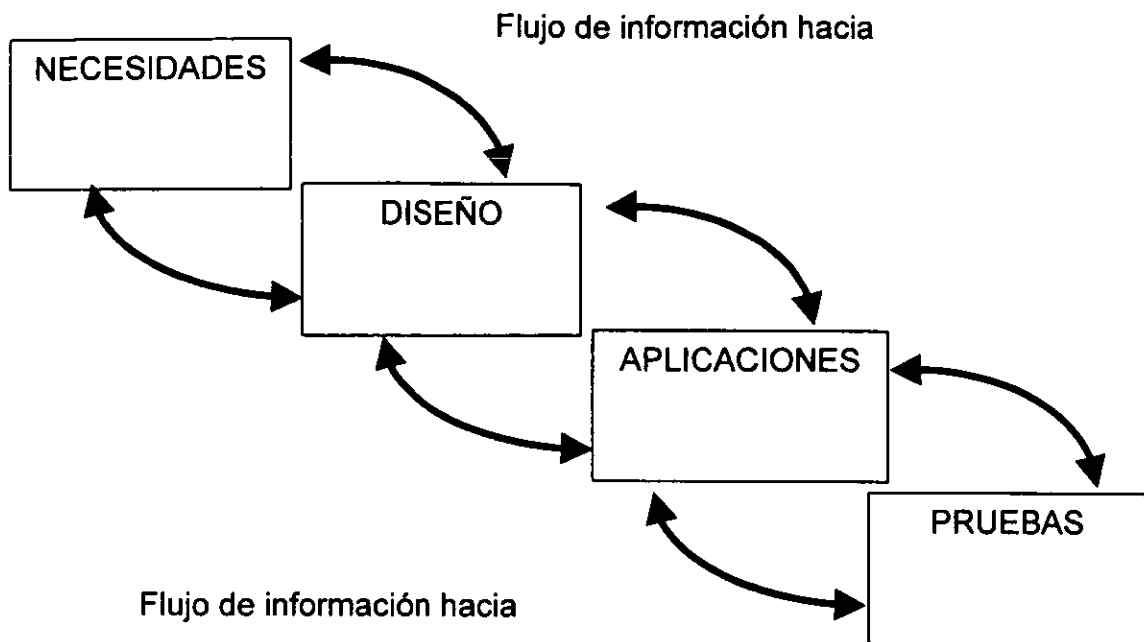
- De acuerdo a lo planteado en esta tesis hemos cumplido con el objetivo primordial.
- Buscamos un Perfil, es decir, ¿Qué es lo que sé esta pretendiendo hacer?.
- Un Objetivo ¿Qué quiero conseguir?.
- Una Estrategia ¿En qué me baso?.
- Una Táctica ¿Qué hay que hacer?.
- Un Recurso ¿Qué necesito y con que cuento?.

A todo lo anterior se le llama planeación estratégica.

Por otro lado nos hemos enfocado a los principios fundamentales de desarrollo de un sistema que son:

- Buscar las necesidades del cliente.
- Elaborar un diseño.
- Desarrollar sus aplicaciones.
- Realizar las pruebas.

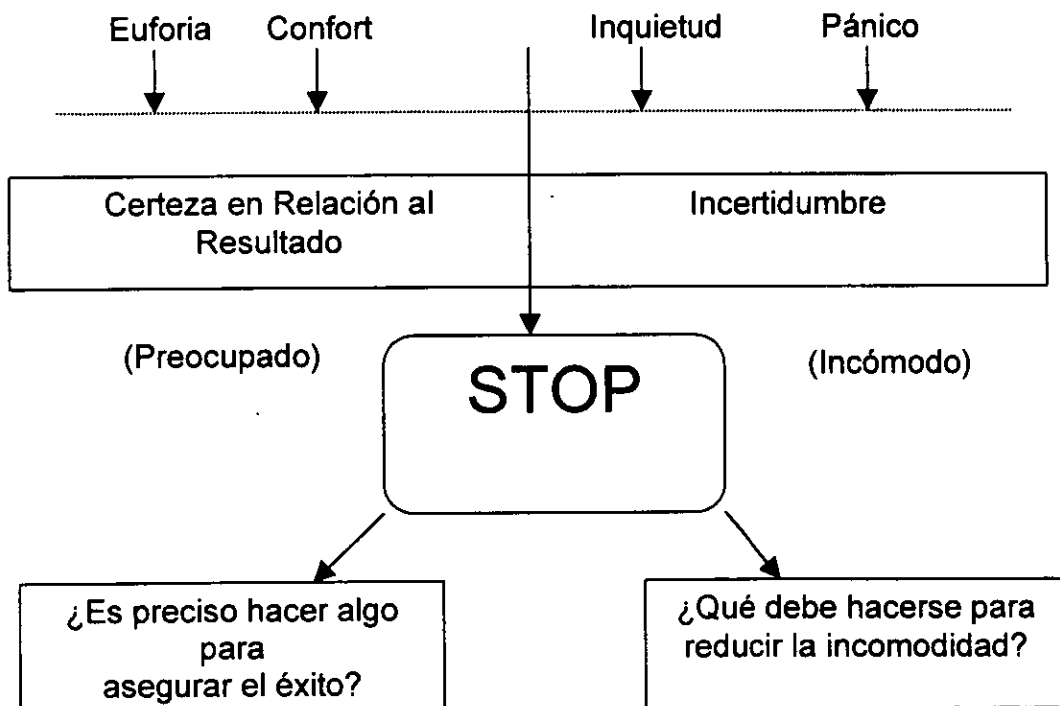
Así como se muestra en las siguientes figuras:



Así mismo podemos decir que el sistema tiene se puede comercializar de la siguiente forma.

- Por licencia teniendo un costo de \$650.00 dólares.
- Total con un costo de \$750,000.00 dólares.

Sabemos que el cliente tendrá cierta incertidumbre con el trabajo realizado y los costos planteados. Nos debemos preguntar ¿cómo se debe sentir el cliente con este producto?. La siguiente figura lo ilustra:



Las conclusiones para este punto serian:

- El cliente tiene un crecimiento en su infraestructura e incremento en sus ingresos.
- El resultado fue el esperado.
- Existe mayor cantidad y calidad en el servicio.
- El impacto que causó fue el esperado y la adaptación del sistema fue muy bien aceptada.

El sistema de Cobranza por Terceros se realizó mediante el software Visual FoxPro 5.0, por lo cual se aprovecharon sus características, entre otras, se presentan las siguientes:

- Entorno de programación visual.
- Programación orientada a objetos.
- El más rápido acceso a datos del mercado, mediante la Tecnología Rushmore.
- Desarrollo de aplicaciones en poco tiempo, R.A.D. (Rapid Application Development).
- Capacidad de programación en Internet.
- Arquitectura Cliente-Servidor.

Además se han utilizado, las ventajas de la programación orientada a objetos como son:

- Generación de clases propias.

- Reutilización de código fuente.
- Polimorfismo en formularios (pantallas de captura).
- Encapsulamiento de código de objetos.

El Sistema de Cobranza por Terceros cumple con los requerimientos planteados por el usuario como son:

- Un sistema modular
- Mantiene la integridad y constancia de la información.
- Permite el procesamiento concurrente de transacciones.
- Existe control de seguridad en los accesos a la información.
- Contiene una bitácora de todas las transferencias de archivos realizadas con éxito así como las acciones o procesos realizados en el sistema.
- Contiene un archivo histórico de movimientos.
- Existe un módulo para la depuración de archivos.
- Es un sistema sencillo, de gran facilidad de uso y totalmente visual.

Actualmente:

- La empresa recibe en tiempo y forma los cobros que realizan los terceros.
- Los clientes tienen mayores opciones para realizar su pago.

- Se han reducido los tiempos de procesamiento y recuperación de los cobros realizados a través de los terceros.

Agente	Antes de implantar el sistema	Después de implantar el sistema
Gigante	De 7 a 15 días	Diferencia de 1 día
Bancos	De 3 a 5 días	Diferencia de 1 día

- La calidad del servicio se ha mejorado.
- Con el Sistema de cobranza por terceros (COBRATEL), se aumento el porcentaje de cobranza de los agentes (Gigante, Bancos, etc.) en algunos casos hasta en un 100 %, es decir el cliente ha realizado con mayor continuidad sus pagos mediante estos agentes, con la confianza de que el pago no se realizará retrasado y proporcionándoles la facilidad de poder realizar estos pagos cualquier día de la semana y en cualquiera de los establecimientos autorizados.

Agente	Antes de implantar el sistema	Después de implantar el sistema	Porcentaje de aumento
Gigante	10 %	15 %	50 %
Bancos	31%	40 %	29 %
Sears	5 %	10 %	100 %
Total	46 %	65 %	41 %

Por ultimo mencionamos los objetivos que se cumplen con el sistema:

El sistema cumple con el objetivo de mejorar la calidad en la respuesta de los procesos, así como hacerlos más eficientes al poder atender a todas las entidades externas que

se vayan integrando sin tener limitantes en cuanto a los días que se puedan o no recibir los pagos y que se realizan de manera rápida, precisa y confiable.

Las tareas que anteriormente se realizaban en forma manual, con el sistema propuesto se han automatizado en su totalidad, facilitando la realización de las labores y reduciendo en gran medida la cantidad de trabajo, recursos humanos y tiempo a las áreas de cobranza de la empresa.

En resumen el Sistema de Cobranza por Terceros, cumple exitosamente el objetivo planteado, al alcanzar y de hecho sobrepasar los siguientes puntos:

- Minimizar costos de operación.
- Aumentar la rapidez en los tiempos de procesamiento y disponibilidad de la información de cobranza por terceros. El departamento de cobranza obtiene la información correcta y oportunamente.
- Obtención de resultados confiables producidos por el sistema.
- La captura de la cobranza por terceros ahora de una forma práctica y genera archivos para transferencia sin error.
- Mayor facilidad en la implantación de los puntos de cobro a nivel Nacional.
- Se tiene un medio de comunicación eficiente, lo cual permite tener la capacidad adecuada para atender a todos los centros de cobranza, así como tener una reducción del tiempo de transferencia de datos entre el sistema y todas las entidades externas.

- Se pueden procesar el volumen de transmisiones requeridas de todos los centros de cobranza sin tener problemas con la capacidad de procesamiento de información y en un tiempo mucho menor al actual.
- En cuanto a la funcionalidad, el sistema cumple con los requerimientos de los usuarios, es un sistema modular, amigable y sencillo. El acceso a la información, la generación de reportes y la transferencia de archivos se realiza de manera rápida y confiable.
- Las tareas o procesos que realizaban las áreas de cobranza de la empresa, estaban referidas a periodos preestablecidos, por ejemplo, reporte semanal de vía de cobro o reporte mensual, el sistema COBRATEL es un sistema flexible, es decir, podrán ejecutar estos reportes y o procesos en cualquier periodo deseado (diario, cada tercer día, por semana, por quincena, por mes, etc.)
- El cliente puede pagar su recibo con los mismos beneficios que ofrece una sucursal de la empresa, con la ventaja que aprovecha los horarios de los establecimientos comerciales o bancos.
- Mejorar el servicio al cliente en el rubro de cobros, proporcionándole más opciones para la realización de sus pagos de manera rápida, sencilla y confiable.
- La empresa recibe con oportunidad los pagos realizados para realizar las acciones necesarias, mejorando su imagen y calidad de servicio con los clientes.

GLOSARIO.

ABSTRACCIÓN. Esto es el resultado de eliminar ciertas distinciones entre los objetos de tal forma que se puedan observar los aspectos comunes. El resultado del proceso de abstracción es un concepto (o bien un tipo de objeto).

ASOCIACIÓN. Es un medio para poder enlazar los tipos de objeto significativamente. Dos ejemplos comunes de la asociación son la relación y la función.

ATRIBUTO. Es la información y/o las características que se encuentran tanto en los objetos como en las clases, y que tiene como función primordial, el otorgar un valor único a cada uno de los objetos o clases para poder ser identificados.

BASE DE DATOS. Conjunto de información almacenada y organizada de manera estructurada que es sistemáticamente buscada, actualizada y registrada.

CAMBIO DE ESTADO. Esto resulta de un cambio en la colección de todos los tipos de objeto que se aplican a un objeto determinado. Un cambio es el estado de un objeto (es equivalente a un evento).

CAMPO. Es una implantación de un tipo de atributo (sinónimo de una variable de instancia).

CARACTERÍSTICA. Es un campo de una estructuras de datos o de una operación permitida en un tipo de datos abstracto. La estructura de datos y las operaciones de un TDA son sus características.

CLASE. Esta es una implantación de un concepto o tipo de objeto. En los lenguajes de programación OO, los tipos de datos abstractos se llaman clases. En matemáticas, el significado de clase es similar al de conjunto. Su significado en los LPOO se deriva de su definición matemática.

CLASIFICACIÓN. Es el resultado de determinar si un concepto, o un tipo de objeto, es aplicado a un objeto específico.

COMPORTAMIENTO. Se ejemplifica como la forma en que los objetos cambian con el tiempo dentro de una estructura definida.

CONJUNTO. La colección de todos los objetos a los que se aplica un tipo de objeto. (es utilizado indistintamente con extensión).

DESARROLLO DE MODELOS. Es la técnica mediante la cual, el diseñador de sistemas puede crear, probar y aprobar, diferentes estructuras que trabajen dentro de un sistema en particular, con el fin de que funcione óptimamente.

DESCLASIFICACIÓN. Este consiste en determinar que un tipo de objeto ya no se aplica a un objeto en específico. Cuando se desclasifica, se elimina como miembro de un conjunto específico.

DIAGRAMA. Es una representación gráfica. Una representación formal se denomina esquema.

DIAGRAMA DE CONTEXTO. El primer nivel de un DFD (Diagrama de Flujo de Datos) es el diagrama de contexto, en donde a través de una sola burbuja se representa al sistema completo, con datos de entrada y de salida respectivamente.

ENCAPSULADO. Es una protección que oculta los detalles de una implantación de un objeto, provocando que el acceso a los datos solo se realice mediante operaciones colocadas ahí de manera intencional.

ENLACE DE UN MÉTODO. Es un termino de la implantación OO, el cual se refiere a la selección de un método para realizar una operación solicitada.

ESQUEMA. Es una representación diagramática formal. El diagrama es considerado formal cuando consta de un conjunto de símbolos definidos, junto con un conjunto de reglas que gobiernan la formación de los diagramas que utilizan estos símbolos.

ESTADO. Es la colección de tipos de objetos que se aplican a un objeto. **DE UN OBJETO.** Es la colección de asociaciones que tiene éste con otros objetos.

ESTRUCTURA. Es una ejemplificación que se refiere a una expresión estática de la manera en que los objetos se distribuyen en el espacio. Los esquemas de objetos son una forma de expresar la estructura. (Ver comportamiento).

EVENTO. Es un cambio significativo en el estado de un objeto. Es la conclusión con éxito del llamado de una operación. Así, una operación llamada es el potencial para un cambio de estado, el cambio de estado real es el evento. Este cambio se establece en términos de un pre-estado de un evento y un post-estado de un evento.

GENERALIZACIÓN. Es el acto o resultado de distinguir un tipo de objeto de una manera más general, o inclusive, que otro. La especificación de un supertipo. Por ejemplo, persona es una generalización de hombre. La generalización no es una forma de abstracción, es una comparación entre las diferentes abstracciones. Esta define una relación en dos tipos de objeto; uno más general que otro.

HERENCIA. La herencia de clase es una implantación de generalización. Nos permite que todas las características de una clase de LPOO estén disponibles o sean reutilizables para sus subclases, como si fueran las características de la subclase. La herencia de una clase a un objeto permite que las instancias de una clase hereden valores predefinidos. La herencia de un objeto a otro, también llamada delegación transfiere el estado de un objeto a otro.

HOST. Anfitrión En redes de computadoras y telecomunicaciones, computadora que realiza funciones centralizadas, como poner al alcance de las demás computadoras los programas y los archivos de datos disponibles.

INSTANCIA. Es el modelo del dominio del problema, que establece que un objeto necesita de otros objetos, para poder ejecutar sus responsabilidades.

INTERFACE. Conexión lógica o física que permite la comunicación interactiva entre dos o más partes.

INTERNET. Sistema de redes de computadora ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como entrada remota, transferencia de archivos, correo electrónico y grupo de noticias.

MÉTODO. Una especificación de los pasos mediante los cuales es realizada una operación. Es un guión de la implantación de una operación. Se pueden aplicar diferentes métodos para realizar la misma operación. Por ejemplo, el método para sumar difiere si se suman enteros o números de punto flotante. En los LPOO, es el código que se ejecuta.

MODELO. Un dispositivo de abstracción. Representa nuestros conceptos.

NORMALIZACIÓN. Es el proceso de agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones.

NORMALIZACIÓN PRIMERA FORMA NORMAL. Cuando no contiene grupos que se repiten.

NORMALIZACIÓN SEGUNDA FORMA NORMAL. Los campos clave dependen de la totalidad de la clave.

NORMALIZACIÓN TERCERA FORMA NORMAL. Los campos no clave son independientes entre sí, es decir, no hay un campo no clave que depende de otro no clave.

OBJETO. Todo aquello a lo que se le aplique un concepto o tipo de objeto; una instancia de un concepto o tipo de objeto. En los LPOO, es cualquier instancia de una clase.

OCULTAMIENTO DE LA INFORMACIÓN. La noción de ocultar la implementación de un objeto, a la vez que se permite una interfaz pública por medio de sus operaciones permitidas: Es indistintamente utilizado con encapsulado.

PING. Es un protocolo que se usa en el entorno del Protocolo de control de transmisión (TCP) para probar si un nodo o dispositivo remoto se comunica en la red.

PROCESO. Es una secuencia ordenada de actos. Es algo que se lleva a un resultado particular.

PROGRAMACIÓN ORIENTADA A OBJETOS (POO). Es la técnica de programación que utiliza la semántica del problema separándola en partes y que tiene como características principales: la representación uniforme del problema, la reutilización de programas y el mantenimiento adecuado de los programas.

PROTOCOLO. Conjunto de reglas y procedimientos predefinidos

RELACIONES. Constituyen el enlace entre las diferentes entidades del sistema.

RELACIÓN UNO A UNO. Una entidad en A esta relacionada únicamente con una entidad de B y una entidad de B con solo una de A.

RELACIÓN UNO A MUCHOS. Una entidad de A esta relacionada con cualquier número de entidades de B, pero una entidad de B solo esta relacionada con una de A.

RELACIÓN MUCHOS A UNO. Una entidad de A esta relacionada únicamente con una entidad de B, pero una entidad de B esta relacionada con cualquier número de entidades en A.

RELACIÓN MUCHOS A MUCHOS. Una entidad A esta asociada con cualquier número de entidades B, y una entidad de B esta vinculada con cualquier número de entidades de A.

RESTRICCIÓN DE CARDINALIDAD. Una restricción de cardinalidad de objetos que deben participar en la asociación de una función. Esta restricción se expresa por lo general como un numero máximo y un mínimo. La restricción de cardinalidad mínima indica el menor numero de objetos con los que debe asociarse. La restricción de cardinalidad máxima indica el mayor numero de objetos con los que un objeto dado debe asociarse.

RESTRICCIÓN DE UNICIDAD. Restricción en la que todos los objetos de cierta clase en particular están sujetos a un solo identificador específico.

REUTILIZACIÓN DEL CÓDIGO. Técnica mediante la cual cierta parte del código de un programa, puede ser utilizado en otro debido a que existe una relación en el funcionamiento de los dos programas.

SUBCLASE. Una clase que es un subtipo de una o más clases (llamadas también superclases). Como tal hereda todas las características de sus superclases (ver herencia). Es decir, todas las características de una clase son reutilizables por sus subclases.

SUBTIPO. Es un tipo de objeto particular. Todas las propiedades aplicadas a un tipo de objeto se aplican a sus subtipos. Sin embargo, un subtipo tiene propiedades adicionales. Por ejemplo, todas las propiedades de persona se aplican a hombre y mujer. Además, hombre y mujer tendrán sus propiedades adicionales a la persona. La extensión de un subtipo es un subconjunto del conjunto correspondiente a un tipo de objeto. (Un tipo de objeto más general se llama supertipo).

SUJETO. Es el mecanismo mediante el cual el lector es guiado (ya sea analista, usuario, administrador o líder de proyecto) a través del modelo representado. Los sujetos son una base muy útil para organizar los grupos de trabajo en proyectos muy grandes.

SUPERCLASE. Es una clase que es un supertipo de una o más clases (llamadas subclases). Como tal, es una clase cuyas características son heredadas por sus subclases (ver Herencia). Es decir, todas las características de una superclase son reutilizables por las clases que son subtipos suyos.

SUPERTIPO. Es un tipo de objeto generalizado. Un tipo de objeto con propiedades más generales que sus subtipos. Todas las instancias de un tipo de objeto son instancias de sus supertipo, pero no al revés.

TCP/IP. (Transmission Control Protocol) protocolo de control de transmisiones/protocolo Internet. Conjunto de protocolos para intercomunicar sistemas diferentes. Se ejecuta en un gran número de computadoras Vax y basadas en Unix, es utilizado por muchos fabricantes de hardware, desde los de computadoras personales hasta las macrocomputadoras.

TIPO. Equivale a tipo de objeto.

TIPO DE EVENTO. Es una clase de evento. El conjunto de todos los cambios de estado resultantes de una operación. Un tipo de evento pedido aceptado define el conjunto de todos los eventos cuando se aceptan los pedidos.

BIBLIOGRAFÍA

Visual FoxPro 5

Fundamentos y técnicas de programación

Rubén Iglesias Balbás

Edición original RA-MA, Madrid, España 1997

Alfaomega grupo editor 1998

Ingeniería de software: Un enfoque práctico

Roger S. Pressman

McGraw-Hill 1998

Programación orientada a objetos

Luis Joyanes Aguilar

McGraw-Hill 1998

Métodos orientados a objetos

Conceptos fundamentales

James Martin, James J. Odell

Prentice-Hall 1997

Análisis y diseño orientado a objetos

James Martin, James J. Odell

Prentice-Hall 1994

Sistemas de bases de datos orientadas a objetos

Conceptos y arquitecturas

Elisa Bertino, Lorenzo Martino

Adisson-Wesley/Díaz de Santos

Microsoft Windows 95, Kit de recursos

Traducción: Fernando Saenz y Rafael Moreno Vozmediano

Revisión técnica: Antonio Vaquero Sánchez y Luis Hernández Yañez

McGraw-Hill 1996

Aprendiendo Windows NT Server 4

Allen Wyatt

Prentice-Hall Hispanoamericana 1998

Fundamentos de Unix

Desarrollado por Rapport de Clarity Software

Publicado por la División Educacional de Silvatech S.A. de C.V.

AIX 6000 básico

Manual del estudiante

IBM Educación

IBM de México S. A.

AIX 6000

Administración del sistema

IBM Educación

IBM de México S. A.

Construya un site perfecto de internet con Linux

Dee-Ann Leblanc

Prentice-Hall 1996

Visual Basic 4 para Windows 95 Paso a paso

Michael Halvorson

McGraw-Hill 1995

Redes de ordenadores

Andrew S. Tanenbaum

Prentice-Hall 1996

Las finanzas en la empresa

Joaquín A. Moreno Fernández

Instituto Mexicano de Contadores Públicos 1996

Principios de Contabilidad.

Javier Romero López.

Mc Graw Hill, 1997

Enciclopedia de redes.

Tom Sheldon

Mc Graw Hill, 1997

Redes de computadoras

Uyless Black.

Macrobit 1995

Ingeniería de Software

Ian Sommerville

Addison-Wesley Iberoamerica

Sistema de Bases de Datos Orientadas a Objetos

Conceptos y Arquitecturas

Elisa Bertino y Lorenzo Martino

Addison-Wesley/Díaz de Satos

Diseño de Bases de datos

Wiederhold McGraw Hill

Introducción a los sistemas de bases de datos

DATE

Addison-Wesley

Administración de Operaciones

Joseph G. Monks

McGraw-Hill

Manual de Herramientas Profesionales de Ventas

Inttelmex

Junio 1997

Coad, P. Y Yourdon. E.

OOA: Object-Oriented Analysis

Prentice Hall Estados Unidos 1991

Coad, P. Y Yourdon. E.

OOD: Object-Oriented Design

Prentice Hall Estados Unidos 1991

<http://rotativo.com>

<http://home.microsoft.com>

<http://www.linux.com>

<http://www.sco.com>

<http://www.hp.com>

<http://www.compaq.com>

APÉNDICE A

GENERACIÓN DE CÓDIGO DEL BACK END

```
*****
C:\COBRATEL\COBRATEL.PRG
*****
*: Procedure File
C:\COBRATEL\COBRATEL.PRG
*:
*:      COBRATEL® Sistema de
cobranza por terceros
*:
*: Documented using Visual FoxPro
Formatting wizard version .05
*****
*: COBRATEL
* MODULO PRINCIPAL
Clear
CD c:\COBRATEL
SET date to ymd
SET century on
SET deleted on
SET talk off
SET SYSMENU Off

DO FORM PRINCIPAL
READ EVENTS
SET SYSMENU ON
Clear EVENTS
RETURN

***** * C:\ARCHIVOS DE
PROGRAMA\DEVSTUDIO\VF\WIZ
ARDS\WIZSTYLE.VCX
RecordSource = "customer"
FontName = "MS Sans Serif"
FontSize = 8
Height = 152
Width = 397
Name = "grid1"
ScaleMode = 3
Top = 6
Left = 6
DoCreate = .T.
wizbuttons = 0
wizbtnpos = 0
Name = "nobtns"
Height = 2
Left = 1
Top = 48
Width = 584
Name = "Shape1"
Height = 2
Left = 3
Top = 264
Width = 584
Name = "Shape2"
FontSize = 14
BackStyle = 0
Caption = "Label1"
Height = 22
Left = 27
Top = 12
Width = 62
AutoSize = .T.
Name = "Label1"
Height = 68
Width = 68
wizeffect = 0
Name = "wizshape"
Width = 517
Height = 205
TabIndex = 0
odimensions = Shape1
olabel1 = Shape2
olabel2 = Shape3
ocol2 = Shape4
Name = "layoutsty"
Height = 205
Left = 0
Top = 0
Width = 517
Name = "Shape1"
Height = 13
Left = 24
Top = 24
Width = 37
Name = "Shape2"
Height = 13
Left = 24
Top = 45
Width = 37
Name = "Shape3"
Height = 13
Left = 228
Top = 24
Width = 37
Name = "Shape4"
FontBold = .F.
FontName = "Arial"
FontSize = 8
BackStyle = 0
Caption = "Label1"
Height = 13
Width = 43
Name = "embossedlabel"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Height = 100
Width = 150
Visible = .T.
Sizable = .T.
Name = "embossedole"
FontName = "MS Sans Serif"
FontSize = 8
Alignment = 0
BackStyle = 0
Caption = "Label1"
Height = 15
Width = 39
AutoSize = .F.
Name = "standardlabel"
MS Sans Serif, 1, 8, 6, 13, 11, 12, 2,
0
Height = 16
Width = 68
FontName = "MS Sans Serif"
FontSize = 9
Alignment = 0
BackStyle = 0
Caption = "Check1"
SpecialEffect = 1
Name = "standardlogic"
MS Sans Serif, 1, 9, 6, 13, 11, 12, 2,
0
Height = 100
Width = 150
Name = "standardole"
Width = 94
Height = 276
Name = "vertxbtns"
cmdPrev.Top = 24
cmdPrev.Left = 0
cmdPrev.Name = "cmdPrev"
cmdNext.Top = 48
cmdNext.Left = 0
cmdNext.Name = "cmdNext"
cmdTop.Top = 0
cmdTop.Left = 0
cmdTop.Name = "cmdTop"
cmdEnd.Top = 72
cmdEnd.Left = 0
cmdEnd.Name = "cmdEnd"
cmdFind.Top = 108
cmdFind.Left = 0
cmdFind.Width = 55
cmdFind.Name = "cmdFind"
cmdPrint.Top = 132
cmdPrint.Left = 0
cmdPrint.Width = 55
cmdPrint.Name = "cmdPrint"
cmdExit.Top = 252
cmdExit.Left = 0
cmdExit.Height = 24
cmdExit.Width = 55
cmdExit.Name = "cmdExit"
cmdAdd.Top = 168
cmdAdd.Left = 0
cmdAdd.Name = "cmdAdd"
cmdDelete.Top = 216
cmdDelete.Left = 0
cmdDelete.Name = "cmdDelete"
cmdEdit.Top = 192
cmdEdit.Left = 0
cmdEdit.Name = "cmdEdit"
MS Sans Serif, 1, 8, 6, 13, 11, 12, 2,
0
ScaleMode = 3
Height = 293
Width = 401
DoCreate = .T.
AutoCenter = .T.
BackColor = 192,192,192
Caption = "Add Record"
WindowType = 1
Name = "gridaddform"
PROCEDURE Init
```

LPARAMETERS

```
cFldKey,cKeyValue,nBtnAction,nAdd  
Action,IChildPrimaryKey,IUpdatableP  
arentKey,IInoSendUpdates
```

```
IF PARAMETERS() # 7  
RETURN .F.
```

```
ENDIF
```

```
this.CommandGroup1.Value = 0  
this.Label3.Caption = m.cFldKey  
this.cKeyValue = m.cKeyValue
```

```
DO CASE
```

```
CASE m.IInoSendUpdates
```

```
* Send Parent Updates
```

```
this.Optiongroup1.Option1.
```

```
Enabled = .F.
```

```
this.Optiongroup1.Option3.
```

```
Enabled = .F.
```

```
CASE !m.IUpdatableParentKey
```

```
* Updatable parent key
```

```
this.Optiongroup1.Option3.
```

```
Enabled = .F.
```

```
ENDCASE
```

```
* Has a primary key
```

```
IF m.IChildPrimaryKey
```

```
this.Optiongroup1.Option2.
```

```
Enabled = .F.
```

```
ENDIF
```

```
DO CASE
```

```
CASE TYPE('THIS.cKeyValue') = "C"  
this.Text1.Value = ""
```

```
CASE
```

```
ATC(TYPE('THIS.cKeyValue'),"NYIB  
F") # 0
```

```
this.Text1.Value = 0
```

```
CASE
```

```
ATC(TYPE('THIS.cKeyValue'),"DT")  
# 0
```

```
this.Text1.Value = {/}
```

```
CASE TYPE('THIS.cKeyValue') = "L"  
this.Text1.Value = .T.
```

```
ENDCASE
```

```
ENDPROC
```

```
AutoSize = .T.
```

```
ButtonCount = 3
```

```
BackStyle = 0
```

```
BorderStyle = 0
```

```
Value = 2
```

```
ControlSource = "nAddAction"
```

```
Height = 68
```

```
Left = 36
```

```
SpecialEffect = 0
```

```
Top = 48
```

```
Width = 218
```

```
TabIndex = 1
```

```
Name = "Optiongroup1"
```

```
Option1.BackStyle = 0
```

```
Option1.Caption = "Add record to  
parent only"
```

```
Option1.Value = 0
```

```
Option1.Height = 18
```

```
Option1.Left = 5
```

```
Option1.Top = 5
```

```
Option1.Width = 181
```

```
Option1.AutoSize = .T.
```

```
Option1.Name = "Option1"
```

```
Option2.BackStyle = 0
```

```
Option2.Caption = "Add record to  
child (grid) only"
```

```
Option2.Value = 1
```

```
Option2.Height = 18
```

```
Option2.Left = 5
```

```
Option2.Top = 25
```

```
Option2.Width = 208
```

```
Option2.AutoSize = .T.
```

```
Option2.Name = "Option2"
```

```
Option3.BackStyle = 0
```

```
Option3.Caption = "Add record to  
both"
```

```
Option3.Value = 0
```

```
Option3.Height = 18
```

```
Option3.Left = 5
```

```
Option3.Top = 45
```

```
Option3.Width = 137
```

```
Option3.AutoSize = .T.
```

```
Option3.Name = "Option3"
```

```
PROCEDURE InteractiveChange
```

```
DO CASE
```

```
CASE THIS.Value = 2
```

```
thisform.Text1.Value =
```

```
THISFORM.cKeyValue
```

```
CASE
```

```
TYPE('THISFORM.cKeyValue') = "C"
```

```
thisform.Text1.Value = ""
```

```
CASE
```

```
ATC(TYPE('THISFORM.cKeyValue'),
```

```
"NYIBF") # 0
```

```
thisform.Text1.Value = 0
```

```
CASE
```

```
ATC(TYPE('THISFORM.cKeyValue'),
```

```
"DT") # 0
```

```
thisform.Text1.Value = {/}
```

```
CASE
```

```
TYPE('THISFORM.cKeyValue') = "L"
```

```
thisform.Text1.Value = .T.
```

```
ENDCASE
```

```
thisform.Text1.ReadOnly =
```

```
(THIS.Value = 2)
```

```
ENDPROC
```

```
AutoSize = .T.
```

```
ButtonCount = 2
```

```
BackStyle = 0
```

```
BorderStyle = 0
```

```
Value = 1.00
```

```
ControlSource = "nBtnAction"
```

```
Height = 39
```

```
Left = 120
```

```
Top = 248
```

```
Width = 163
```

```
Name = "Commandgroup1"
```

```
Command1.Top = 5
```

```
Command1.Left = 5
```

```
Command1.Height = 29
```

```
Command1.Width = 70
```

```
Command1.Caption = "Add"
```

```
Command1.TabIndex = 1
```

```
Command1.Name = "Command1"
```

```
Command2.Top = 5
```

```
Command2.Left = 88
```

```
Command2.Height = 29
```

```
Command2.Width = 70
```

```
Command2.Caption = "Cancel"
```

```
Command2.TabIndex = 2
```

```
Command2.Name = "Command2"
```

```
PROCEDURE Click
```

```
cKeyValue =
```

```
THIS.Parent.Text1.Value
```

```
nAddAction =
```

```
THIS.Parent.Optiongroup1.Value
```

```
nBtnAction = THIS.Value
```

```
thisform.Release()
```

```
ENDPROC
```

```
Alignment = 0
```

```
Height = 24
```

```
Left = 132
```

```
TabIndex = 3
```

```
Top = 192
```

```
Width = 218
```

```
Name = "Text1"
```

```
AutoSize = .T.
```

```
BackStyle = 0
```

```
Caption = "Key value:"
```

```
Height = 18
```

```
Left = 48
```

```
Top = 196
```

```
Width = 67
```

```
Name = "Label1"
```

```
AutoSize = .T.
```

```
BackStyle = 0
```

```
Caption = "A. Select option for
```

```
adding records:"
```

```
Height = 18
```

```
Left = 24
```

```
Top = 24
```

```
Width = 225
```

```
Name = "Label2"
```

```
FontItalic = .T.
```

```
BackStyle = 0
```

```
Caption = "Label3"
```

```
Height = 16
```

```
Left = 132
```

```
Top = 168
```

```
Width = 145
```

```
Name = "Label3"
```

```
AutoSize = .T.
```

```
BackStyle = 0
```

```
Caption = "Key field:"
```

```
Height = 18
```

```
Left = 48
```

```
Top = 168
```

```
Width = 60
```

```
Name = "Label4"
```

```
AutoSize = .T.
```

```
BackStyle = 0
```

```
Caption = "B: Enter key value linking
```

```
parent and child records:"
```

```
Height = 18
```

```
Left = 24
```

```
Top = 132
```

```
Width = 325
```

```
Name = "Label5"
```

```
Arial, 1, 10, 6, 16, 13, 15, 3, 0
```

```
Arial, 3, 10, 6, 16, 13, 17, 3, 0
```

```
Height = 22
```

```
Width = 39
```

```
Name = "gridaddress"
```

```
PROCEDURE runaddform
```

```
#DEFINE C_NOOBJ_LOC "Failed
```

```
to create the Add Record form class.
```

```
Check or reinstall the
```

```
WIZSTYLE.VCX file."
```

```
PRIVATE
cFldKey,cKeyValue,nBtnAction,nAdd
Action,oGridAddForm
```

```
IF TYPE("THIS.KeyField") # "C"
    this.KeyField = ""
ENDIF
```

```
cFldKey = THIS.KeyField
cKeyValue = THIS.KeyValue
nBtnAction = 1
nAddAction = 1
```

```
oGridAddForm =
CREATE("gridaddform",m.cFldKey,m
.cKeyValue,;
    m.nBtnAction,m.nAddActio
n,THIS.ChildPrimaryKey,THIS.Updat
ableParentKey,THIS.NoSendUpdate
s)
```

```
IF TYPE("m.oGridAddForm") # "O"
    =MESSAGEBOX(C_NOO
BJ_LOC)
    this.AddOption = 0
    RETURN
ENDIF
```

```
IF THIS.NoSendUpdates
    oGridAddForm.Optiongrou
p1.Value = 2
    oGridAddForm.Text1.Valu
e = m.cKeyValue
ENDIF
```

```
oGridAddForm.Show()
```

```
this.AddOption =
IIF(m.nBtnAction=1,m.nAddAction,0)
this.KeyValue = m.cKeyValue
```

```
ENDPROC
Width = 207
Height = 111
Name = "shadowole"
Shape1.Height = 100
Shape1.Left = 54
Shape1.Top = 7
Shape1.Width = 150
Shape1.ZOrderSet = 0
Shape1.Name = "Shape1"
Label1.Height = 16
Label1.Left = 0
Label1.Top = 3
Label1.Width = 41
Label1.ZOrderSet = 1
Label1.Name = "Label1"
Height = 100
Left = 48
Top = 2
Width = 151
ZOrderSet = 2
Name = "Shape2"
Top = 2
Left = 48
Height = 100
Width = 151
TabIndex = 0
ZOrderSet = 3
Name = "Oleboundcontrol1"
```

```
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Width = 148
Height = 24
Name = "chiselfield"
Shape1.Height = 2
Shape1.Left = 46
Shape1.Top = 20
Shape1.Width = 97
Shape1.Name = "Shape1"
Label1.Height = 13
Label1.Left = 0
Label1.Top = 5
Label1.Width = 45
Label1.Name = "Label1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
BackColor = 192,192,192
BackStyle = 0
BorderStyle = 0
Height = 17
Left = 46
SpecialEffect = 1
TabIndex = 0
Top = 3
Width = 97
DisabledForeColor = 0,0,128
SelectedForeColor = 255,255,255
SelectedBackColor = 0,0,0
Name = "Text1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Arial, 0, 8, 5, 14, 11, 20, 3, 0
Width = 198
Height = 52
Name = "chiselmemo"
Shape1.Height = 2
Shape1.Left = 46
Shape1.Top = 50
Shape1.Width = 150
Shape1.Name = "Shape1"
Label1.Height = 13
Label1.Left = 0
Label1.Top = 3
Label1.Width = 45
Label1.Name = "Label1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
BackColor = 192,192,192
BackStyle = 0
Height = 48
Left = 46
SpecialEffect = 0
TabIndex = 0
Top = -1
Width = 150
DisabledForeColor = 0,0,128
SelectedForeColor = 255,255,255
SelectedBackColor = 0,0,0
Name = "Edit1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Arial, 0, 8, 5, 14, 11, 20, 3, 0
Width = 200
Height = 106
Name = "chiselole"
Shape1.Height = 2
Shape1.Left = 46
Shape1.Top = 103
Shape1.Width = 150
Shape1.Name = "Shape1"
```

```
Label1.Height = 13
Label1.Left = 0
Label1.Top = 3
Label1.Width = 45
Label1.Name = "Label1"
Top = 0
Left = 46
Height = 100
Width = 150
TabIndex = 0
Name = "Oleboundcontrol1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Width = 94
Height = 18
Name = "chisellogic"
Shape1.Height = 2
Shape1.Left = 53
Shape1.Top = 15
Shape1.Width = 13
Shape1.Name = "Shape1"
Label1.Height = 13
Label1.Left = 0
Label1.Top = 1
Label1.Width = 45
Label1.Name = "Label1"
Top = 0
Left = 53
Height = 15
Width = 15
FontName = "MS Sans Serif"
FontSize = 8
BackStyle = 0
Caption = ""
SpecialEffect = 0
TabIndex = 0
Name = "Check1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
MS Sans Serif, 1, 8, 6, 13, 11, 12, 2,
0
Width = 148
Height = 24
BackColor = 192,192,192
BackStyle = 0
BorderWidth = 0
SpecialEffect = 2
Name = "chiselbase"
BackColor = 255,255,255
Height = 2
Left = 46
Top = 20
Width = 97
SpecialEffect = 0
Name = "Shape1"
FontName = "Arial"
FontSize = 8
Alignment = 1
BackStyle = 0
Caption = "Label1"
Height = 13
Left = 0
Top = 5
Width = 45
ColorSource = 3
TabIndex = 0
Name = "Label1"
Arial, 1, 8, 5, 14, 11, 14, 3, 0
Width = 94
Height = 43
Name = "boxfield"
Shape1.Height = 43
```

Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 94
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
FontSize = 9
BackColor = 255,255,255
BackStyle = 1
BorderStyle = 0
Height = 18
Left = 3
SpecialEffect = 1
TabIndex = 0
Top = 23
Width = 88
SelectedBackColor = 128,128,128
ColorSource = 0
Name = "Text1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 247
Height = 75
Name = "boxmemo"
Shape1.Height = 75
Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 246
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
FontSize = 9
BackColor = 255,255,255
BorderStyle = 0
Height = 50
Left = 3
ScrollBars = 0
SpecialEffect = 1
TabIndex = 0
Top = 23
Width = 241
ColorSource = 0
Name = "Edit1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 156
Height = 125
Name = "boxole"
Shape1.Height = 125
Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 156
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
Top = 23
Left = 3
Height = 100
Width = 150
TabIndex = 0
Name = "Oleboundcontrol1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 88

Height = 43
Name = "boxlogic"
Shape1.Height = 43
Shape1.Left = 0
Shape1.Top = 0
Shape1.Width = 88
Shape1.Name = "Shape1"
Label1.Height = 15
Label1.Left = 3
Label1.Top = 2
Label1.Width = 49
Label1.Name = "Label1"
Top = 22
Left = 2
Height = 17
Width = 13
FontSize = 9
BackStyle = 1
Caption = ""
SpecialEffect = 0
TabIndex = 0
ColorSource = 0
Name = "Check1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 94
Height = 43
BackStyle = 0
BorderWidth = 0
SpecialEffect = 2
Name = "boxbase"
BackColor = 255,255,255
BackStyle = 1
FillColor = 255,255,255
FillStyle = 0
Height = 43
Left = 0
Top = 0
Width = 94
SpecialEffect = 1
wizeffect = 1
Name = "Shape1"
FontSize = 9
BackColor = 0,0,255
BackStyle = 0
Caption = "Label1"
ForeColor = 0,0,255
Height = 15
Left = 3
Top = 2
Width = 49
TabIndex = 0
Name = "Label1"
Arial, 1, 9, 6, 15, 12, 15, 3, 0
Width = 204
Height = 55
Name = "shadowmemo"
Shape1.Top = 7
Shape1.Left = 52
Shape1.Height = 48
Shape1.Width = 152
Shape1.Name = "Shape1"
Label1.Height = 16
Label1.Left = 0
Label1.Top = 4
Label1.Width = 41
Label1.Name = "Label1"
FontName = "Arial"
FontSize = 8
Height = 48
Left = 48

SpecialEffect = 1
TabIndex = 0
Top = 2
Width = 150
ColorSource = 0
BackColor = 255,255,255
Name = "Edit1"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Width = 149
Height = 25
Name = "shadowfield"
Shape1.Top = 5
Shape1.Left = 51
Shape1.Height = 20
Shape1.Width = 97
Shape1.Name = "Shape1"
Label1.Height = 16
Label1.Left = 0
Label1.Top = 3
Label1.Width = 41
Label1.Name = "Label1"
FontName = "Arial"
FontSize = 8
Height = 22
Left = 46
SpecialEffect = 1
TabIndex = 0
Top = 0
Width = 97
BackColor = 255,255,255
DisabledBackColor = 255,255,255
SelectedBackColor = 128,128,128
Name = "Text1"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Width = 94
Height = 18
Name = "shadowlogic"
Shape1.Top = 7
Shape1.Left = 58
Shape1.Height = 11
Shape1.Width = 11
Shape1.Name = "Shape1"
Label1.Height = 16
Label1.Left = 0
Label1.Top = 0
Label1.Width = 41
Label1.Name = "Label1"
Top = 0
Left = 52
Height = 15
Width = 15
BackStyle = 0
Caption = ""
SpecialEffect = 1
TabIndex = 0
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "Check1"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
FontBold = F.
FontName = "Arial"
FontSize = 8
Height = 22
Width = 113
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "embossedfield"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Height = 15
Width = 72

```

FontBold = .F.
FontName = "Arial"
FontSize = 8
BackStyle = 0
Caption = "Check1"
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "embossedlogic"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
Height = 50
Width = 150
ColorSource = 0
Name = "embossedmemo"
Arial, 0, 8, 5, 14, 11, 12, 3, 0
Width = 513
Height = 29
wizbmppath =
(HOME()+"WIZARDS\WIZBMPS\")
Name = "picbtns"
cmdPrev.Top = 4
cmdPrev.Left = 51
cmdPrev.Width = 50
cmdPrev.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
BACK.BMP")
cmdPrev.Caption = ""
cmdPrev.ToolTipText = "Previous
record"
cmdPrev.Name = "cmdPrev"
cmdNext.Top = 4
cmdNext.Left = 102
cmdNext.Width = 50
cmdNext.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
NEXT.BMP")
cmdNext.Caption = ""
cmdNext.ToolTipText = "Next record"
cmdNext.ColorSource = 0
cmdNext.Name = "cmdNext"
cmdTop.Top = 4
cmdTop.Left = 0
cmdTop.Width = 50
cmdTop.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
TOP.BMP")
cmdTop.Caption = ""
cmdTop.ToolTipText = "Top record"
cmdTop.Name = "cmdTop"
cmdEnd.Top = 4
cmdEnd.Left = 153
cmdEnd.Width = 50
cmdEnd.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
END.BMP")
cmdEnd.Caption = ""
cmdEnd.ToolTipText = "Bottom
record"
cmdEnd.Name = "cmdEnd"
cmdFind.Top = 4
cmdFind.Left = 204
cmdFind.Width = 50
cmdFind.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
LOCATE.BMP")
cmdFind.Caption = ""

```

```

cmdFind.ToolTipText = "Find
records"
cmdFind.Name = "cmdFind"
cmdPrint.Top = 4
cmdPrint.Left = 255
cmdPrint.Width = 50
cmdPrint.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
PRINT.BMP")
cmdPrint.Caption = ""
cmdPrint.ToolTipText = "Print report"
cmdPrint.Name = "cmdPrint"
cmdExit.Top = 4
cmdExit.Left = 459
cmdExit.Width = 50
cmdExit.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
CLOSE.BMP")
cmdExit.Caption = ""
cmdExit.ToolTipText = "Exit form"
cmdExit.Name = "cmdExit"
cmdAdd.Top = 4
cmdAdd.Left = 306
cmdAdd.Width = 50
cmdAdd.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
NEW.BMP")
cmdAdd.Caption = ""
cmdAdd.ToolTipText = "Add / Save
record"
cmdAdd.Name = "cmdAdd"
cmdDelete.Top = 4
cmdDelete.Left = 408
cmdDelete.Width = 50
cmdDelete.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
DELETE.BMP")
cmdDelete.Caption = ""
cmdDelete.ToolTipText = "Delete
record"
cmdDelete.Name = "cmdDelete"
cmdEdit.Top = 4
cmdEdit.Left = 357
cmdEdit.Width = 50
cmdEdit.Picture =
(HOME()+"WIZARDS\WIZBMPS\WZ
EDIT.BMP")
cmdEdit.Caption = ""
cmdEdit.ToolTipText = "Edit / Revert
record"
cmdEdit.Name = "cmdEdit"
PROCEDURE Init

TxtBtns::Init()
thisform.ShowTips = .T.

ENDPROC
PROCEDURE setcaption
IF !THIS.EditMode
    this.cmdAdd.Picture =
THIS.wizbmppath+"wznew.bmp"
    this.cmdEdit.Picture =
THIS.wizbmppath+"wzedit.bmp"
    this.cmdAdd.DownPicture
= THIS.wizbmppath+"wznew.bmp"
    this.cmdEdit.DownPicture
= THIS.wizbmppath+"wzedit.bmp"
ELSE
    this.cmdAdd.Picture =
THIS.wizbmppath+"wzsave.bmp"

```

```

    this.cmdEdit.Picture =
THIS.wizbmppath+"wzundo.bmp"
    this.cmdAdd.DownPicture
= THIS.wizbmppath+"wzsave.bmp"
    this.cmdEdit.DownPicture
= THIS.wizbmppath+"wzundo.bmp"
ENDIF
ENDPROC
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
FontName = "MS Sans Serif"
FontSize = 8
Height = 43
SpecialEffect = 1
Width = 150
ColorSource = 0
DisabledBackColor = 255,255,255
Name = "standardmemo"
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
ScaleMode = 3
DoCreate = .T.
ColorSource = 0
BackColor = 255,255,255
wizlayout = layoutsty
wizfield = standardfield
wizmemo = standardmemo
wizlogic = standardlogic
wizole = standardole
wizmaxcharfld = 45
wizblsuffix = (":")
wizblcap = proper
wizformstretch = .T.
wizbldefwid = .T.
wiztitle = Label1
wizuser = 0
wizgrid = grid1
wizverify = .T.
wizbtnpos = 0
wizcodestyle = .T.
wizlabel = standardlabel
wizblspace = 10
wizcboxlbl = .T.
Name = "standardform"
Top = 6
Left = 9
Width = 561
Height = 253
TabIndex = 0
BackColor = 255,255,255
odimensions = Shape1
Name = "Layoutsty"
Shape1.Top = 0
Shape1.Left = 0
Shape1.Height = 252
Shape1.Width = 561
Shape1.BackColor = 255,255,255
Shape1.Name = "shape1"
Shape2.Top = 7
Shape2.Left = 16
Shape2.Height = 13
Shape2.Width = 37
Shape2.Name = "shape2"
Shape3.Top = 22
Shape3.Left = 16
Shape3.Height = 13
Shape3.Width = 37
Shape3.Name = "shape3"
Shape4.Top = 7
Shape4.Left = 56

```

```

Shape4.Height = 13
Shape4.Width = 37
Shape4.Name = "shape4"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
ErasePage = .T.
PageCount = 2
TabStretch = 1
TabStyle = 1
Width = 287
Height = 169
Name = "boxpages"
Page1.Caption = "Page1"
Page1.Name = "Page1"
Page2.Caption = "Page2"
Page2.Name = "Page2"
Width = 296
Height = 26
BackStyle = 0
BorderWidth = 0
Visible = .T.
TabIndex = 1
nworkarea = 0
previewmode = .F.
oldrefresh = 0
oldrec = 0
oldreprocess = 0
previewinit = .T.
usedataenv = .T.
Name = "gridbtns"
PROCEDURE initvars
#DEFINE C_NOUPDATEVIEW_LOC
    "Edits to one or more of the
Views may not be permanent. "+;
    "To remedy this, ensure
the View's Send SQL Updates
checkbox is checked in the View
Designer."
#DEFINE C_READONLY_LOC
    "The table is
Read-Only. You will not be able to
edit it."

LOCAL
aTablesUsed, nTablesUsed, i, aMems,
nTotMem, cWizFile, IShowedMess, cD
ataEnvRef
DIMENSION aTablesUsed[1]
DIMENSION aMems[1]

this.nworkarea = SELECT()
this.oldSetDelete =
SET("DELETED")
SET DELETED ON
this.oldreprocess =
SET("REPROCESS")
SET REPROCESS TO 0

* These properties should not be
used. They are reserved for use by
* the Preview button of the Form
Wizards.
this.previewmode =
IIF(TYPE("THIS.PreviewMode")# "L",.
F., THIS.previewmode)
this.previewinit =
IIF(TYPE("THIS.PreviewInit")# "L",.T.,
THIS.previewinit)

* Check for data environment
DO CASE

```

```

CASE
TYPE("THISFORM.DataEnvironment
") = "O"
    this.usedataenv = .T.
    nTotMem =
AMEMBERS(aMems, THISFORM.Da
taEnvironment, 2)
    cDataEnvRef =
"THISFORM.DataEnvironment"
CASE
TYPE("THISFORMSET.DataEnviron
ment") = "O"
    this.usedataenv = .T.
    nTotMem =
AMEMBERS(aMems, THISFORMSE
T.DataEnvironment, 2)
    cDataEnvRef =
"THISFORMSET.DataEnvironment"
    * let's also set the
IF
TYPE("THISFORM") = "O" AND
!THISFORM.VISIBLE
    thisform.VISIBLE
= .T.
ENDIF
OTHERWISE
    this.usedataenv = .F.
ENDCASE

IF THIS.usedataenv
WITH
EVAL(m.cDataEnvRef)
Views
    FOR i = 1 TO
m.nTotMem
        IF
UPPER(EVAL("." + aMems[m.i] + ".Bas
eClass")) = "CURSOR"
            WITH
EVAL("." + aMems[m.i])
                IF
CURSORGETPROP("SourceType",.
Alias)#3
                    *
                    Check if we need to requery for
deleted records.
                    * _
                    Note: Parameterized views are not
requeried so all records will be
brought over.
                    * To
                    remedy this situation you can place a
SET DELETED ON command in the
                    *
                    BeforeOpenTables event of the
DataEnvironment.
                    IF
THIS.oldSetDelete = "OFF" AND
ATC("?", CURSORGETPROP("SQL",
.Alias)) = 0

```

```

=REQUERY(.Alias)
ENDIF
*
Check if updates are made
IF
!CURSORGETPROP("SendUpdates
",.Alias) AND !m.IShowedMess
    =MESSAGEBOX(C_NOUP
DATEVIEW_LOC)
    IShowedMess = .T.
ENDIF
ENDIF
ENDWITH
ENDIF
ENDFOR
ENDWITH
ENDIF
IF ISREADONLY()
    WAIT WINDOW
C_READONLY_LOC TIMEOUT 2
ENDIF
* Disable appropriate buttons
this.cmdAdd.Enabled =
!ISREADONLY()
this.cmdDelete.Enabled =
!ISREADONLY()
IF THIS.previewmode
    RETURN
ENDIF
IF THIS.usedataenv
    RETURN
ENDIF
* The following code is here to
support forms not using a
DataEnvironment.
this.oldSetFields = SET("FIELDS")
SET FIELDS OFF
this.oldMultiLocks =
SET("MULTILOCKS")
SET MULTILOCKS ON
this.oldrefresh = SET("REFRESH")
SET REFRESH TO 5
IF IEMPTY(ALIAS())
    this.oldBuffering=CursorGe
tProp("buffering")
m.nTablesUsed =
AUSED(aTablesUsed)
FOR i = 1 TO
m.nTablesUsed
    IF
CursorGetProp('sourcetype', aTables
Used[m.i,1])#3 &&skip for views

```

```

=CursorSetProp("buffering"
,5,aTablesUsed[m,i,1])
&&optimistic table buffering
ENDIF
ENDFOR
ENDIF
GO TOP

ENDPROC
PROCEDURE updaterecords
#DEFINE E_FAIL_LOC
"Failed to update
table: "
#DEFINE E_TRIGGERFAIL_LOC
"Trigger failed."
#DEFINE E_FIELDNULL_LOC
"Field doesn't
accept NULL"
#DEFINE E_FIELDRULE_LOC
"Field rule
violated"
#DEFINE E_RECORDLOCK_LOC
"Record in use by another
user"
#DEFINE E_ROWRULE_LOC
"Row rule
violated"
#DEFINE E_UNIQUEINDEX_LOC
"Unique index violation"
#DEFINE E_DIRTYREC_LOC
"Data has been
changed by another user. Overwrite
changes with your edits?"
#DEFINE E_NOFORCE_LOC
"Could not force table
updates."
#DEFINE E_PROMPT_LOC
"Error: "
#DEFINE MSGBOX_YES
6

LOCAL
aErrors,cErrorMessage,aTablesUsed
,nTablesUsed,nTotErr
LOCAL
nFld,i,nOldArea,iSuccess,iInDBC,iOv
erwrite,iHadMessage

DIMENSION aTablesUsed[1]
DIMENSION aErrors[1]
m.cErrorMessage=""
m.iSuccess = .T.
m.nOldArea = SELECT()
m.nTablesUsed =
AUUSED(aTablesUsed)

* Can wrap everything in transaction
if using strictly DBCs

FOR i = 1 TO m.nTablesUsed

SELECT
(aTablesUsed[m,i,1])

m.iInDBC =
!EMPTY(CURSORGETPROP("Data
base"))
m.cErrorMessage = ""
m.iOverwrite = .F.

```

```

m.iHadMessage = .F.

DO CASE
CASE
CURSORGETPROP("Buffering") = 1
* Skip if buffering
not on
LOOP
CASE GetFldState(0) = 2
&&deleted record
* Only delete
current record and force it
m.iSuccess =
TableUpdate(.F.,.T.)
IF m.iSuccess
&&successful update
LOOP
ENDIF
CASE !m.iInDBC AND
(ATC("2",GetFldState(-1))#0 OR;
ATC("3",GetFldState(-1))#0)
* Field was
edited - in Free Table
* Since free
tables are not supported by
transactions,
* we must
process record by record
m.nModRecord =
GetNextMod(0)
DO WHILE
m.nModRecord # 0 &&loop locks all
records
GO
m.nModRecord
m.iSuccess = RLOCK()
&&try to lock record
IF
!m.iSuccess
&&failed to lock record
m.cErrorMessage =
E_RECORDLOCK_LOC
UNLOCK ALL
EXIT
ENDIF
IF
!m.iHadMessage &&so we don't
repeat alert
* See if record(s) modified
by another user
FOR m.nFld = 1 TO
FCOUNT()
IF
TYPE(FIELD(m.nFld)) = "G" &&skip
for General fields
LOOP
ENDIF
ENDIF

```

```

IF
OLDVAL(FIELD(m.nFld)) #
CURVAL(FIELD(m.nFld))
m.iHadMessage = .T.
IF
MESSAGEBOX(E_DIRTYREC_LOC
,4+48) = MSGBOX_YES
m.iOverwrite = .T.
ELSE
m.iSuccess = .F.
UNLOCK ALL
EXIT
ENDIF
ENDIF
ENDIF
ENDIF
m.nModRecord =
GetNextMod(m.nModRecord)
ENDDO
IF m.iSuccess
&&was able to lock all
records
m.iSuccess =
TableUpdate(.T.,m.iOverwrite)
IF
m.iSuccess &&was able to update all
records
LOOP
ENDIF
UNLOCK ALL
ENDIF
CASE m.iInDBC
BEGIN
TRANSACTION
* Try to update
all records in selected table
m.iSuccess =
TableUpdate(.T.,.F.)
&&successful update
IF m.iSuccess
END
TRANSACTION
LOOP
ENDIF
ROLLBACK
ENDCASE
* Handle errors
nTotErr
=AERROR(aErrors)

```



```

DO CASE
CASE nTotErr = 0
ROLLBACK

CASE aErrors[1,1] = 1539
"Please locate."

&& Trigger failed
=MESSAGEBOX(E_NOFO
RCE_LOC)
LOCAL
cWizHomePath,separator,cWizStyFil
e

m.cErrorMessage =
E_TRIGGERFAIL_LOC
CASE aErrors[1,1] = 1581
ENDIF
ENDIF

&& Field doesn't accept
NULL
OTHERWISE
IF
!EMPTY(m.cErrorMessage) &&for
free table handling above
ENDIF

m.cErrorMessage =
E_FIELDNULL_LOC
CASE aErrors[1,1] = 1582
&& Field rule violated
ENDIF

m.cErrorMessage =
E_FIELDRULE_LOC
CASE aErrors[1,1] = 1700
* Had an error we couldn't
handle
IF
&&revert all records
m.ISuccess = .F.
IF
!EMPTY(m.cErrorMessage)
=MESSAGEBOX(E_FAIL_
LOC+m.cErrorMessage)
ENDIF
ENDIF

&& Record in use by
another user
ATC(C_WIZSTYLE,SET("CLASSLIB
")) = 0
* Returns just the
pathname
cWizHomePath =
_WIZARD
IF '! $ cWizHomePath
cWizHomePath =
SUBSTR(m.cWizHomePath,1,RAT('!
,m.cWizHomePath))
IF
RIGHT(m.cWizHomePath,1) = '\
AND LEN(m.cWizHomePath) > 1 ;
AND
SUBSTR(m.cWizHomePath,LEN(m.c
WizHomePath)-1,1) <> '\'
cWizHomePath =
SUBSTR(m.cWizHomePath,1,LEN(
m.cWizHomePath)-1)
ENDIF
ELSE
cWizHomePath = "
ENDIF

&& Row rule violated
E_ROWRULE_LOC
CASE aErrors[1,1] = 1884
&& Unique index violation
SELECT (m.nOldArea)
RETURN m.ISuccess

m.cErrorMessage =
E_UNIQUEINDEX_LOC
CASE aErrors[1,1] = 1585
**** Special Preview Mode Handling
****
IF THIS.previewmode AND
THIS.previewinit
this.previewinit = .F.
this.cmdAdd.Enabled = .F.
this.cmdDelete.Enabled =
.F.
this.cmdFind.Enabled = .F.
this.cmdPrint.Enabled = .F.
this.cmdExit.Enabled = .F.
this.nworkarea =
ENDIF
ENDIFPROC
PROCEDURE Refresh
**** Special Preview Mode Handling
****
IF THIS.previewmode AND
THIS.previewinit
this.previewinit = .F.
this.cmdAdd.Enabled = .F.
this.cmdDelete.Enabled =
.F.
this.cmdFind.Enabled = .F.
this.cmdPrint.Enabled = .F.
this.cmdExit.Enabled = .F.
this.nworkarea =
ENDIF
ENDIFPROC
PROCEDURE Init
#DEFINE C_WIZSTYLE
"WIZSTYLE.VCX"
#DEFINE C_WIZDIR
"WIZARDSI"
#DEFINE C_PROMPT1_LOC
"Find: "
#DEFINE E_NOSTYLE_LOC
"The class library
(WIZSTYLE.VCX) needed by this
form could not be found. "+;

&& Record changed by
another user
IF m.lnDBC
&&handle free
tables above
*
Display conflict alert
IF
MESSAGEBOX(E_DIRTYREC_LOC
,4+48) = MSGBOX_YES
*Try to force update
BEGIN TRANSACTION
m.ISuccess =
TABLEUPDATE(T,,T.)
IF m.ISuccess
END
TRANSACTION
LOOP
ELSE

```

```

        cWizFile =
m.cWizHomePath+C_WIZDIR+C_WI
ZSTYLE
        CASE
FILE(HOME()+C_WIZSTYLE)
        cWizFile =
HOME()+C_WIZSTYLE
        CASE
FILE(HOME()+C_WIZDIR+C_WIZST
YLE)
        cWizFile =
HOME()+C_WIZDIR+C_WIZSTYLE
OTHERWISE

=MESSAGEBOX(E_NOST
YLE_LOC)
        cWizFile =
GETFILE("VCX",C_PROMPT1_LOC
+C_WIZSTYLE)
        ENDCASE

        IF
ATC(C_WIZSTYLE,m.cWizFile)#0
        SET CLASS TO
(m.cWizFile) ADDITIVE
        ELSE
        * Failed to get
WIZSTYLE.VCX file
        RETURN .F.
        ENDIF

ENDIF

this.initvars()

ENDPROC
PROCEDURE Error
PARAMETERS nError, cMethod,
nLine
LOCAL aFoxErr,nTotErr
DIMENSION aFoxErr[1]
nTotErr = AERROR(aFoxErr)

DO CASE
CASE INLIST(m.nError,1733,1734)
&&property not found - traps
SETALL()
        RETURN
CASE m.nError=1938 &&no parent
        RETURN
CASE nTotErr>0 AND aFoxErr[1,1] =
1420
        * Corrupt Ole object in
General field.
=MESSAGEBOX(aFoxErr{
1,2})
        RETURN
ENDCASE

**** Error Dialog *****
=MESSAGEBOX(MESSAGE(1)+CHR
R(13)+;
"Error: "+STR(nError)+CHR(13)+;
MESSAGE()+CHR(13)+;
"Method: "+cMethod+CHR(13)+;
"Line: "+STR(nLine))

RETURN TO MASTER

ENDPROC

```

```

PROCEDURE Destroy
* Restore various settings
LOCAL
nTablesUsed,aTablesUsed,i,nDECur
sors,aDECursors,cDataEnvRef
DIMENSION aTablesUsed[1]

IF TYPE('THIS.Parent') # "O"
        RETURN
ENDIF

IF TYPE("THIS.oldTalk") = "C" AND
THIS.oldTalk="ON"
        SET TALK ON
ENDIF

* OLE Servers can still send data
back to General fields
* even though they are not in Edit
Mode. We need to reset
* buffering to 1 so the buffer is not
updated by the OLE Server.
* Also, folks might exit out while
editing.
IF THIS.usedataenv
        DIMENSION
aDECursors[1]
        DO CASE
        CASE
TYPE("THISFORM.DataEnvironment
") = "O"
                nDECursors =
AMEMBERS(aDECursors,THISFOR
M.DataEnvironment,2)
                cDataEnvRef =
"THISFORM.DataEnvironment"
                CASE
TYPE("THISFORMSET.DataEnviron
ment") = "O"
                        nDECursors =
AMEMBERS(aDECursors,THISFOR
MSET.DataEnvironment,2)
                        cDataEnvRef =
"THISFORMSET.DataEnvironment"
                ENDCASE
                FOR i = 1 TO
m.nDECursors
                        WITH
EVAL(m.cDataEnvRef + "." +
aDECursors[m.i])
                                IF
USED(.ALIAS) AND
ATC("CURSOR",.BaseClass)#0 AND
;
CursorGetProp("sourcetype",.ALIAS)
=3 AND ;
CursorGetProp("buffering",.ALIAS)>1
                                =TableRevert(.T,.ALIAS)
                                =CursorSetProp("buffering"
,1,.ALIAS) &&optimistic
table buffering
                                ENDF
                                ENDFWITH
                                ENDFOR
                        ENDIF
                ENDIF
ENDIF

```

```

* Skip if using preview mode
IF THIS.previewmode
        RETURN
ENDIF

IF THIS.oldSetDelete = "OFF"
        SET DELETED OFF
ENDIF

SET REPROCESS TO
THIS.oldreprocess
SET MESSAGE TO
SELECT (THIS.nworkarea)

IF THIS.usedataenv
        RETURN
ENDIF

* The following code is here to
support
* forms not using a
DataEnvironment.
m.nTablesUsed =
AUSED(aTablesUsed)
FOR i = 1 TO m.nTablesUsed
        IF
USED(aTablesUsed[m.i,1]) AND
ATC(".TMP",DBF(aTablesUsed[m.i,1
]))=0 &&skip for views
                =CursorSetProp("buffering"
,THIS.oldBuffering,aTablesUsed[m,i,
1]) &&optimistic table buffering
                ENDF
        ENDFOR

IF THIS.oldMultiLocks = "OFF"
        SET MULTILOCKS OFF
ENDIF

IF THIS.oldSetFields = "ON"
        SET FIELDS ON
ENDIF

SET REFRESH TO THIS.oldrefresh

ENDPROC
Top = 2
Left = 0
Height = 24
Width = 49
FontName = "MS Sans Serif"
FontSize = 8
Caption = "<Find"
Enabled = .T.
StatusBarText = "Search for records"
TabIndex = 5
ToolTipText = ""
Name = "cmdFind"
PROCEDURE Click

LOCAL
IVisChange,IStateChange,oSearchDi
og

* Check if SDI Window
IF THISFORM.ShowWindow = 2
        IF !_VFP.Visible
                _VFP.Visible =
.T.
                IVisChange = .T.

```

```

        ENDIF
        IF !_SCREEN.WindowState
= 1
            _SCREEN.WindowState =
0
                IStateChange =
.T.
        ENDIF
    ENDIF

oSearchDlg =
CREATE("searchform")
oSearchDlg.SHOW()
thisform.REFRESH()

IF m.VisChange
    _VFP.Visible = .F.
ENDIF
IF m.IStateChange
    _SCREEN.WindowState =
1
ENDIF
IF THISFORM.ShowWindow = 2
    Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
Top = 2
Left = 58
Height = 24
Width = 49
FontName = "MS Sans Serif"
FontSize = 8
Caption = "Print"
Enabled = .T.
StatusBarText = "Print to report"
TabIndex = 6
ToolTipText = ""
Name = "cmdPrint"
PROCEDURE Click
#DEFINE C_MAKEREPO_LOC
"Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC
"Error opening
table. Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC
"Select a report to print:"

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

#IF 0

```

```

        * Handling for Private data
sessions
        IF m.nSaveSess # 1
            SET
DATASESSION TO 1
            SELECT 0
            IF
!EMPTY(m.cSaveData)
                OPEN
DATABASE (m.cSaveData)
            ENDIF
            IF
USED(m.cSaveAlias)
                SELECT (m.cSaveAlias)
                IF
CURSORGETPROP("SourceName")
#m.cSaveSource
                    cDiffSource =
CURSORGETPROP("SourceName")
                    USE IN (m.cSaveAlias)
                    SELECT 0
                    ENDIF
                    ENDIF
                    IF
EMPTY(ALIAS())
                        USE
(m.cSaveSource) AGAIN ALIAS
(m.cSaveAlias) SHARED
                    IF
EMPTY(ALIAS())
                        =MESSAGEBOX(C_NOO
PEN_LOC)
                        RETURN
                    ENDIF
                    ENDIF
                ENDIF
            #ENDIF
            IF FILE(m.cRepName)
                REPORT FORM
(m.cRepName) PREVIEW NOWAIT
            ELSE
                m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
                IF !EMPTY(m.cRepName)
                    IF
FILE(m.cRepName)
                        * User
pressed Open button
                        REPORT FORM
(m.cRepName) PREVIEW NOWAIT
                    ELSE
                        * User
pressed New button
                        DO
(_WIZARD) WITH "AUTOREPORT"
                    ENDIF
                ENDIF
            ENDIF
            #IF 0
                IF !EMPTY(cDiffSource)

```

```

                USE
(m.cDiffSource) IN 0
            ENDIF
            SET DATASESSION TO
m.nSaveSess
            SELECT (m.cSaveAlias)
#ENDIF
            IF THISFORM.ShowWindow = 2
                Activate Window
(THISFORM.Name)
            ENDIF
        ENDPROC
        Top = 2
        Left = 240
        Height = 24
        Width = 53
        FontName = "MS Sans Serif"
        FontSize = 8
        Caption = "Exit"
        Enabled = .T.
        StatusBarText = "Exit input form"
        TabIndex = 10
        ToolTipText = ""
        Name = "cmdExit"
        PROCEDURE Click
        thisform.Release

    ENDPROC
    Top = 2
    Left = 116
    Height = 24
    Width = 53
    FontName = "MS Sans Serif"
    FontSize = 8
    Caption = "\<Add"
    Enabled = .T.
    StatusBarText = "Add new record /
Save edits"
    TabIndex = 7
    ToolTipText = ""
    Name = "cmdAdd"
    PROCEDURE Click
    #DEFINE C_NOUPDATE_LOC
    "You cannot add a new
record because the view(s) selected
does not send updates."
    IF
CURSORGETPROP("SourceType")#
3 AND
!CURSORGETPROP("SendUpdates
")
        MESSAGEBOX(C_NOUP
DATE_LOC)
        RETURN
    ENDIF
    APPEND BLANK
    thisform.REFRESH()
    ENDPROC
    Top = 2
    Left = 178
    Height = 24
    Width = 53
    FontName = "MS Sans Serif"
    FontSize = 8
    Caption = "\<Delete"
    Enabled = .T.

```

```

StatusBarText = "Delete existing
record"
TabIndex = 9
ToolTipText = ""
Name = "cmdDelete"
PROCEDURE Click
#DEFINE MSGBOX_YES
6
#DEFINE C_MSGBOX1
36
#DEFINE C_DELETE_LOC "Do
you want to delete this record?"

IF
MESSAGEBOX(C_DELETE_LOC,C
_MSGBOX1) = MSGBOX_YES
Delete
IF
THIS.Parent.updaterows()
&&success
* Success
IF !EOF()
Skip 1
ENDIF
IF EOF() AND
!BOF()
Skip -1
ENDIF
ENDIF
thisform.REFRESH()
IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
Width = 220
Height = 120
Name = "fancyole"
Label1.Name = "Label1"
Top = 12
Left = 60
Height = 100
Width = 145
Name = "Oleboundcontrol1"
Width = 149
Height = 25
BackStyle = 0
BorderWidth = 0
Name = "shadowbase"
Top = 4
Left = 51
Height = 20
Width = 97
BackStyle = 0
BorderStyle = 0
BorderWidth = 0
FillStyle = 0
BackColor = 0,255,255
FillColor = 128,128,128
Name = "Shape1"
FontBold = .T.
FontName = "Arial"
FontSize = 8
Alignment = 1
BackStyle = 0
Caption = "Label1"

```

```

Height = 16
Left = 0
Top = 3
Width = 41
TabIndex = 0
Name = "Label1"
DoCreate = .T.
BackColor = 0,128,128
wizfield = fancyfield
wizlabel =
wizlbicap = lower
wizlbsuffix = (""")
wizlogic = fancylogic
wizmemo = fancymemo
wizole = fancyole
wizcboxlbl = .F.
wizlbspace =
wizmaxcharfd = 40
Name = "fancyform"
layoutsty.Shape1.Top = 0
layoutsty.Shape1.Left = 0
layoutsty.Shape1.Height = 228
layoutsty.Shape1.Width = 538
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Top = 7
layoutsty.Shape2.Left = 6
layoutsty.Shape2.Height = 13
layoutsty.Shape2.Width = 37
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Top = 27
layoutsty.Shape3.Left = 6
layoutsty.Shape3.Height = 13
layoutsty.Shape3.Width = 37
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Top = 6
layoutsty.Shape4.Left = 111
layoutsty.Shape4.Height = 13
layoutsty.Shape4.Width = 37
layoutsty.Shape4.Name = "Shape4"

layoutsty.Top = 48
layoutsty.Left = 21
layoutsty.Width = 538
layoutsty.Height = 228
layoutsty.Name = "layoutsty"
Label1.FontName = "Tahoma"
Label1.FontSize = 18
Label1.Height = 31
Label1.Left = 21
Label1.Top = 10
Label1.Width = 78
Label1.Name = "Label1"
Shape1.Top = 279
Shape1.Left = 5
Shape1.Height = 1
Shape1.Width = 569
Shape1.BackColor = 255,255,255
Shape1.BorderColor = 255,255,255
Shape1.Name = "Shape1"
Shape2.Top = 43
Shape2.Left = 5
Shape2.Height = 1
Shape2.Width = 569
Shape2.Name = "Shape2"
Shape3.Top = 314
Shape3.Left = 5
Shape3.Height = 1
Shape3.Width = 569
Shape3.BackColor = 255,255,255
Shape3.BorderColor = 255,255,255

```

```

Shape3.Name = "Shape3"
Shape4.Top = 5
Shape4.Left = 5
Shape4.Height = 0
Shape4.Width = 569
Shape4.BackStyle = 0
Shape4.SpecialEffect = 0
Shape4.Name = "Shape4"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
Width = 184
Height = 34
SpecialEffect = 0
Name = "fancybase"
FontName = "Tahoma"
FontSize = 9
Caption = "Label1"
Height = 18
Left = 8
Top = 9
Width = 43
Name = "Label1"
FontName = "MS Sans Serif"
FontSize = 8
Alignment = 3
Height = 22
SpecialEffect = 1
Width = 100
ColorSource = 0
Name = "standardfield"
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
ColumnCount = 1
AllowHeaderSizing = .F.
AllowRowSizing = .F.
DeleteMark = .F.
HeaderHeight = 0
Height = 178
Highlight = .F.
HighlightRow = .F.
RecordMark = .F.
RowHeight = 300
ScrollBars = 2
SplitBar = .F.
Width = 369
BackColor = 192,192,192
wizbtnstyle = gridbtns
Name = "scrollgrid"
Column1.Width = 500
Column1.Spase = .F.
Column1.BackColor = 192,192,192
Column1.Name = "Column1"
Caption = "Header1"
Name = "Header1"
Top = 23
Left = 15
Width = 200
Height = 200
BackStyle = 1
Name = "Container1"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
DataSession = 1
ScaleMode = 3
Height = 320
Width = 580
DoCreate = .T.
AutoCenter = .T.
Caption = "Form1"
Enabled = .T.
wizlayout =
wizfield =

```

```

wizmemo =
wizlogic = 0
wizole =
wizmaxcharfld = 0
wizblsuffix = 0
wizblcap = 0
wizformstretch = 0
wizbldefwid = 0
wiztitle =
wizuser = 0
wizgrid =
wizverify = 0
wizbuttons = 0
wizbtnpos = 0
wizcodestyle = .T.
wizbtnlayout = 0
wizcaptions = .T.
wizbuffering = 5
wizlabel =
wizblspace = 0
wizpages = 1
wizpagestyle =
Name = "baseform"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
ScaleMode = 3
DoCreate = .T.
ColorSource = 0
BackColor = 192,192,192
wizlayout = layoutsty
wizfield = boxfield
wizmemo = boxmemo
wizlogic = boxlogic
wizole = boxole
wizmaxcharfld = 45
wizblsuffix = (":")
wizblcap = proper
wizformstretch = .T.
wizbldefwid = .F.
wiztitle = Label1
wizuser = 0
wizgrid = grid1
wizverify = .T.
wizcodestyle = .T.
wizpages = 2
wizpagestyle = boxpages
Name = "boxform"
Top = 62
Left = 5
Width = 571
Height = 195
TabIndex = 0
Name = "Layoutsty"
Shape1.Top = 0
Shape1.Left = 0
Shape1.Height = 194
Shape1.Width = 570
Shape1.Name = "shape1"
Shape2.Top = 1
Shape2.Left = 16
Shape2.Height = 13
Shape2.Width = 37
Shape2.Name = "shape2"
Shape3.Top = 13
Shape3.Left = 16
Shape3.Height = 13
Shape3.Width = 37
Shape3.Name = "shape3"
Shape4.Top = 1
Shape4.Left = 52
Shape4.Height = 13

```

```

Shape4.Width = 37
Shape4.Name = "shape4"
Top = 0
Left = 0
Height = 61
Width = 580
FillStyle = 0
BackColor = 0,0,255
FillColor = 0,0,255
Name = "Shape1"
AutoSize = .T.
FontName = "Arial"
FontSize = 20
BackStyle = 0
Caption = "Label1"
Height = 35
Left = 21
Top = 17
Width = 83
ForeColor = 255,255,255
BackColor = 255,255,255
Name = "Label1"
Top = 261
Left = 0
Height = 61
Width = 580
FillStyle = 0
BackColor = 0,0,255
FillColor = 0,0,255
Name = "Shape2"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
ScaleMode = 3
DoCreate = .T.
ColorSource = 0
BackColor = 192,192,192
wizlayout = layoutsty
wizfield = chiselfield
wizmemo = chiselmemo
wizlogic = chisellogic
wizole = chiselole
wizmaxcharfld = 45
wizblsuffix = (":")
wizblcap = proper
wizformstretch = .T.
wizbldefwid = .T.
wiztitle = Label1
wizuser = 0
wizgrid = grid1
wizverify = .T.
wizbtnpos = 0
wizcodestyle = .T.
Name = "chiselform"
Top = 53
Left = 2
Width = 573
Height = 213
TabIndex = 0
ZOrderSet = 0
odimensions = Shape1
Name = "Layoutsty"
Shape1.Top = 0
Shape1.Left = 0
Shape1.Height = 212
Shape1.Width = 572
Shape1.Name = "shape1"
Shape2.Top = 7
Shape2.Left = 16
Shape2.Name = "shape2"
Shape3.Top = 22
Shape3.Left = 16

```

```

Shape3.Name = "shape3"
Shape4.Top = 7
Shape4.Left = 255
Shape4.Name = "shape4"
Top = 270
Left = 0
Height = 2
Width = 584
ZOrderSet = 1
Name = "Shape2"
Top = 0
Left = 0
Height = 50
Width = 580
FillStyle = 0
FillColor = 0,0,0
BorderColor = 0,0,0
ZOrderSet = 2
Name = "Shape3"
AutoSize = .T.
FontSize = 20
BackStyle = 0
Caption = "Label1"
Height = 35
Left = 19
Top = 10
Width = 83
ForeColor = 255,255,255
BackColor = 0,0,0
ZOrderSet = 3
Name = "Label1"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
ScaleMode = 3
DoCreate = .T.
ColorSource = 0
BackColor = 192,192,192
wizlayout = layoutsty
wizfield = shadowfield
wizmemo = shadowmemo
wizlogic = shadowlogic
wizole = shadowole
wizmaxcharfld = 45
wizblsuffix = (":")
wizblcap = proper
wizformstretch = .T.
wizbldefwid = .T.
wiztitle = Label1
wizuser = 0
wizgrid = grid1
wizverify = .T.
wizbtnpos = 0
wizcodestyle = .T.
Name = "shadowform"
Top = 54
Left = 7
Width = 564
Height = 205
TabIndex = 0
odimensions = Shape1
Name = "Layoutsty"
Shape1.Top = 0
Shape1.Left = 0
Shape1.Height = 205
Shape1.Width = 563
Shape1.Name = "shape1"
Shape2.Top = 7
Shape2.Left = 16
Shape2.Height = 13
Shape2.Width = 37
Shape2.Name = "shape2"

```

```

Shape3.Top = 22
Shape3.Left = 16
Shape3.Height = 13
Shape3.Width = 37
Shape3.Name = "shape3"
Shape4.Top = 7
Shape4.Left = 252
Shape4.Height = 13
Shape4.Width = 37
Shape4.Name = "shape4"
Top = 48
Left = 0
Height = 2
Width = 584
Name = "Shape1"
Top = 264
Left = 2
Height = 2
Width = 584
Name = "Shape2"
AutoSize = .T.
FontBold = .T.
FontSize = 14
BackColor = 0
Caption = "Label1"
Height = 25
Left = 24
Top = 12
Width = 64
Name = "Label1"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
ScaleMode = 3
DoCreate = .T.
wizlayout = layoutsty
wizfield = embossedfield
wizmemo = embossedmemo
wizlogic = embossedlogic
wizole = embossedole
wizmaxcharfld = 45
wizlblsuffix = (":")
wizlblcap = proper
wizformstretch = .T.
wizlbldefwid = .T.
wiztitle = Label1
wizuser = 0
wizgrid = grid1
wizverify = .T.
wizbtnpos = 0
wizcodestyle = .T.
wizcboxlbl = .T.
wizlabel = embossedlabel
Name = "embossedform"
Top = 54
Left = 4
Width = 571
Height = 205
TabIndex = 0
Name = "Layoutsty"
Shape1.Top = 0
Shape1.Left = 0
Shape1.Height = 205
Shape1.Width = 570
Shape1.Name = "shape1"
Shape2.Top = 7
Shape2.Left = 6
Shape2.Height = 13
Shape2.Width = 37
Shape2.Name = "shape2"
Shape3.Top = 22
Shape3.Left = 6

```

```

Shape3.Height = 13
Shape3.Width = 37
Shape3.Name = "shape3"
Shape4.Top = 7
Shape4.Left = 260
Shape4.Height = 13
Shape4.Width = 37
Shape4.Name = "shape4"
AutoSize = .T.
FontBold = .T.
FontName = "Times New Roman"
FontSize = 20
BackColor = 0
Caption = "Label1"
Height = 34
Left = 12
Top = 8
Width = 83
Name = "Label1"
Top = 42
Left = 5
Height = 1
Width = 568
BackColor = 128,128,128
BorderColor = 128,128,128
Name = "Shape1"
Top = 43
Left = 5
Height = 1
Width = 568
BackColor = 255,255,255
BorderColor = 255,255,255
Name = "Shape2"
Top = 264
Left = 6
Height = 1
Width = 568
BackColor = 128,128,128
BorderColor = 128,128,128
Name = "Shape3"
Top = 265
Left = 6
Height = 1
Width = 568
BackColor = 255,255,255
BorderColor = 255,255,255
Name = "Shape4"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
Width = 184
Height = 34
Name = "fancyfield"
Label1.Name = "Label1"
FontName = "Tahoma"
Height = 23
Left = 59
Top = 5
Width = 113
DisabledForeColor = 0,0,0
NullDisplay = ("n/a")
Name = "Text1"
Tahoma, 0, 9, 5, 14, 12, 17, 2, 0
Width = 88
Height = 34
Name = "fancylogic"
Label1.Name = "Label1"
Top = 8
Left = 56
Height = 18
Width = 17
Caption = ""

```

```

DisabledForeColor = 0,0,0
Name = "Check1"
Arial, 0, 9, 5, 15, 12, 13, 3, 0
Width = 240
Height = 90
Name = "fancymemo"
Label1.Name = "Label1"
FontName = "Tahoma"
Height = 78
Left = 54
Top = 5
Width = 177
ForeColor = 0,0,0
DisabledForeColor = 0,0,0
IntegralHeight = .T.
Name = "Edit1"
Tahoma, 0, 9, 5, 14, 12, 17, 2, 0
Width = 452
Height = 160
BackColor = 0
TabIndex = 1
BackColor = 192,192,192
remotedelimiter = (.F.)
Name = "searchclass"
PROCEDURE searchexpr
LOCAL
cGetExpr1,cGetExpr2,cJoin,cGetExp
r

m.cGetExpr1 =
THIS.SearchItem(THIS.cboFields1,T
HIS.cboOperators1,THIS.txtExpr1)
m.cGetExpr2 =
THIS.SearchItem(THIS.cboFields2,T
HIS.cboOperators2,THIS.txtExpr2)
m.cJoin =
IIF(THIS.optGrpAndOr.value = 2,"
OR "," AND ")

DO CASE
CASE EMPTY(m.cGetExpr1) AND
EMPTY(m.cGetExpr2)
m.cGetExpr = ""
CASE EMPTY(m.cGetExpr2)
m.cGetExpr =
m.cGetExpr1
CASE EMPTY(m.cGetExpr1)
m.cGetExpr =
m.cGetExpr2
OTHERWISE
m.cGetExpr =
m.cGetExpr1+m.cJoin+m.cGetExpr2
ENDCASE

RETURN m.cGetExpr
ENDPROC
PROCEDURE dataexpr
LPARAMETER cDataType,cFldExpr

LOCAL cTmpExpr

DO CASE
CASE
INLIST(m.cDataType,"M","G","P","O"
,"U")
RETURN ""

CASE m.cDataType = "C"
IF
TYPE(""Test'="+m.cFldExpr) # "L"

```

```

IF
THIS.remotedelimiter

    cTmpExpr =
""+m.cFldExpr+""
ELSE
    cTmpExpr =
"["+m.cFldExpr+"]"
ENDIF
ELSE
    cTmpExpr =
m.cFldExpr
ENDIF

* Check for case sensitive
IF
THIS.chkCaseSensitive.Value = 0
    m.cTmpExpr=
"UPPER("+m.cTmpExpr+")"
ENDIF
RETURN m.cTmpExpr

CASE
INLIST(m.cDataType,"N","F","I","Y","
B")
* Check for any commas
and remove
RETURN
ALLTRIM(STR(VAL(STRTRAN(m.cF
ldExpr,""),16,4))

CASE INLIST(m.cDataType,"D","T")
RETURN
"{+CHRTRAN(m.cFldExpr,"{}",",")+}"

OTHERWISE
RETURN ""

ENDCASE

ENDPROC
PROCEDURE SearchItem
LPARAMETERS oField,oOp,oExpr
LOCAL
cExpr,cDataType,cOp,cFldName,cFl
dExpr,cRetExpr,aExprs,nTotExprs,i

* Check to make sure proper
parameters passed
IF TYPE("m.oField")# "O" OR
TYPE("m.oOp")# "O" OR
TYPE("m.oExpr")# "O"
RETURN ""
ENDIF

m.cFldName =
ALLTRIM(m.oField.Value)
m.cFldExpr =
ALLTRIM(m.oExpr.Value)

* if empty expression return empty.
IF EMPTY(m.cFldExpr) AND
!INLIST(m.oOp.listitemid,5,6)
RETURN ""
ENDIF

* Get data type of field

```

```

m.cDataType = '
aWizFList(m.oField.listitemid,2)

IF m.cDataType = "C" AND
THIS.chkCaseSensitive.Value = 0
    m.cFldName =
"UPPER("+m.cFldName+")"
ENDIF

* Get the operator language
equivalent
DO CASE
CASE m.oOp.listitemid = 1 &&
equals
    m.cOp = "="
CASE m.oOp.listitemid = 2 && not
equals
    m.cOp = "<>"
CASE m.oOp.listitemid = 3 &&
more than
    m.cOp = ">"
CASE m.oOp.listitemid = 4 &&
less than
    m.cOp = "<"
CASE m.oOp.listitemid = 5 && is
blank
RETURN
"EMPTY("+m.cFldName+")"
CASE m.oOp.listitemid = 6 && is
NULL
RETURN
"ISNULL("+m.cFldName+")"
CASE m.cDataType = "L"
&& don't allow other
options for logical type
m.cOp = "="
CASE m.oOp.listitemid = 7 &&
contains
    m.cFldExpr =
THIS.dataexpr("C",m.cFldExpr)
DO CASE
CASE m.cDataType = "T"
RETURN
"AT("+m.cFldExpr+",TTOC("+m.cFld
Name+"))>0"
CASE m.cDataType = "D"
RETURN
"AT("+m.cFldExpr+",DTOC("+m.cFld
Name+"))>0"
CASE
INLIST(m.cDataType,"N","F","I","Y","
B")
RETURN
"AT("+m.cFldExpr+",ALLTRIM(STR(
+m.cFldName+"))>0"
OTHERWISE
RETURN
"AT("+m.cFldExpr+", "+m.cFldName+
">0"
ENDCASE
OTHERWISE
nTotExprs =
OCCURS(", ",m.cFldExpr)+1
DIMENSION
aExprs[m.nTotExprs]
FOR i = 1 TO
m.nTotExprs
DO CASE
CASE m.i =

```

```

aExprs[m.i] =
SUBSTR(m.cFldExpr,RAT(", ",m.cFld
Expr)+1)
CASE m.i = 1
aExprs[m.i] =
LEFT(m.cFldExpr,AT(", ",m.cFldExpr)
-1)
OTHERWISE
aExprs[m.i] =
SUBSTR(m.cFldExpr,AT(", ",m.cFldE
xpr,m.i-1)+1,;
AT(", ",m.cFldExpr,m.i)-
AT(", ",m.cFldExpr,m.i-1)-1)
ENDCASE
aExprs[m.i] =
THIS.dataexpr(m.cDataType,aExprs[
m.i])
ENDFOR
DO CASE
CASE m.oOp.listitemid = 8
&& in
    m.cFldExpr = ""
FOR i = 1 TO
m.nTotExprs
    m.cFldExpr = m.cFldExpr +
aExprs[m.i]
IF
m.i # m.nTotExprs
    m.cFldExpr = m.cFldExpr +
";"
ENDIF
ENDFOR
RETURN
"INLIST("+m.cFldName+", "+m.cFldE
xpr+)"
CASE m.oOp.listitemid = 9
&& between
IF
ALEN(aExprs)=1
DIMENSION aExprs[2]
aExprs[2] = aExprs[1]
ENDIF
IF
ALEN(aExprs)>2
DIMENSION aExprs[2]
ENDIF
RETURN
"BETWEEN("+m.cFldName+", "+aEx
prs[1]+", "+aExprs[2]+")"
OTHERWISE
RETURN ""
ENDCASE
ENDCASE
DO CASE
CASE
INLIST(m.cDataType,"M","G","P","O"
,"U")
RETURN ""

```

```

CASE m.cDataType = "L"
  IF
  TYPE(m.cFldName+m.cOp+m.cFldExpr) # "L"
    IF
    (AT(m.cFldExpr,"fFnN")#0 AND
    m.cOp # "<>") OR
    (AT(m.cFldExpr,"tTyY")#0 AND
    m.cOp = "<>")
      m.cFldName =
      "!" + m.cFldName
    ENDIF
    RETURN
  m.cFldName
  ENDIF
  OTHERWISE
  m.cFldExpr =
  THIS.dataexpr(m.cDataType,m.cFldExpr)
  ENDCASE

IF EMPTY(m.cFldExpr)
  RETURN ""
ELSE
  RETURN
  m.cFldName+m.cOp+m.cFldExpr
ENDIF

ENDPROC
PROCEDURE Destroy
Release aWizFList

ENDPROC
FontName = "MS Sans Serif"
FontSize = 8
ControlSource = ""
Height = 23
Left = 11
Style = 2
TabIndex = 1
Top = 25
Width = 141
Name = "cboFields1"
PROCEDURE Init
#DEFINE NUM_AFIELDS 16
LOCAL i
PUBLIC aWizFList
DIMENSION aWizFList[1]
=AFIELDS(aWizFList)
FOR m.i = FCOUNT() TO 1 STEP -1
  IF
  INLIST(aWizFList[m.i,2],"G","M","U")
    &&Memo field

    =ADEL(aWizFList,m.i)
    DIMENSION
    aWizFList[MAX(1,ALEN(aWizFList,1)
    -1),NUM_AFIELDS]
  ENDIF
ENDFOR
this.RowSourceType = 5
this.RowSource = "aWizFList"
this.VALUE = THIS.LIST[1]

ENDPROC
FontName = "MS Sans Serif"
FontSize = 8
Alignment = 0

```

```

Height = 23
Left = 158
Style = 2
TabIndex = 2
Top = 25
Width = 96
Name = "cboOperators1"
PROCEDURE InteractiveChange
IF INLIST(THIS.ListItemId,5,6)
  this.Parent.txtExpr1.Value
  = ""
ENDIF
this.Parent.txtExpr1.ENABLED =
!!INLIST(THIS.ListItemId,5,6)

ENDPROC
PROCEDURE Init
#DEFINE C_OPERATORS_LOC
"equals\;not equals\;more
than\;less than\;is blank\;is
NULL\;contains\;in\;between"
this.ADDITEM(C_OPERATORS_LO
C)
this.VALUE = THIS.LIST[1]

ENDPROC
FontName = "MS Sans Serif"
FontSize = 8
Height = 23
Left = 260
TabIndex = 3
Top = 25
Width = 176
Name = "txtExpr1"
FontName = "MS Sans Serif"
FontSize = 8
Height = 23
Left = 11
Style = 2
TabIndex = 4
Top = 117
Width = 141
Name = "cboFields2"
PROCEDURE Init
this.RowSourceType = 5
this.RowSource = "aWizFList"
this.VALUE = THIS.LIST[1]

ENDPROC
FontName = "MS Sans Serif"
FontSize = 8
Height = 23
Left = 160
Style = 2
TabIndex = 5
Top = 117
Width = 96
Name = "cboOperators2"
PROCEDURE InteractiveChange
IF INLIST(THIS.ListItemId,5,6)
  this.Parent.txtExpr2.Value
  = ""
ENDIF
this.Parent.txtExpr2.ENABLED =
!!INLIST(THIS.ListItemId,5,6)

ENDPROC
PROCEDURE Init
#DEFINE C_OPERATORS_LOC
"equals\;not equals\;more

```

```

than\;less than\;is blank\;is
NULL\;contains\;in\;between"
this.ADDITEM(C_OPERATORS_LO
C)
this.VALUE = THIS.LIST[1]

ENDPROC
FontName = "MS Sans Serif"
FontSize = 8
Height = 23
Left = 260
TabIndex = 6
Top = 118
Width = 176
Name = "txtExpr2"
AutoSize = .T.
ButtonCount = 2
BackStyle = 0
Value = 1
Height = 46
Left = 50
Top = 55
Width = 58
TabIndex = 7
Name = "optgrpAndOr"
Option1.FontName = "MS Sans
Serif"
Option1.FontSize = 8
Option1.BackStyle = 0
Option1.Caption = "\<And"
Option1.Value = 1
Option1.Height = 18
Option1.Left = 5
Option1.Top = 5
Option1.Width = 48
Option1.Name = "Option1"
Option2.FontName = "MS Sans
Serif"
Option2.FontSize = 8
Option2.BackStyle = 0
Option2.Caption = "\<Or"
Option2.Value = 0
Option2.Height = 18
Option2.Left = 5
Option2.Top = 23
Option2.Width = 39
Option2.Name = "Option2"
PROCEDURE Init
this.VALUE = 1
ENDPROC
FontName = "MS Sans Serif"
FontSize = 8
BackStyle = 0
Caption = "Field."
Height = 16
Left = 12
Top = 7
Width = 102
Name = "label1"
FontName = "MS Sans Serif"
FontSize = 8
BackStyle = 0
Caption = "Operator:"
Height = 16
Left = 158
Top = 7
Width = 93
Name = "label2"
FontName = "MS Sans Serif"
FontSize = 8

```



```

BackStyle = 0
Caption = "Value:"
Height = 16
Left = 262
Top = 7
Width = 165
Name = "label3"
FontName = "MS Sans Serif"
FontSize = 8
BackStyle = 0
Caption = "Field:"
Height = 16
Left = 12
Top = 99
Width = 102
Name = "label4"
FontName = "MS Sans Serif"
FontSize = 8
BackStyle = 0
Caption = "Operator:"
Height = 16
Left = 161
Top = 99
Width = 93
Name = "label5"
FontName = "MS Sans Serif"
FontSize = 8
BackStyle = 0
Caption = "Value:"
Height = 16
Left = 262
Top = 99
Width = 165
Name = "label6"
Top = 69
Left = 290
Height = 15
Width = 88
FontName = "MS Sans Serif"
FontSize = 8
AutoSize = .T.
BackStyle = 0
Caption = "Case Sensitive"
TabIndex = 8
Name = "chkCaseSensitive"
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
Height = 216
Width = 477
DoCreate = .T.
AutoCenter = .T.
BorderStyle = 1
Caption = "Search"
WindowType = 1
AlwaysOnTop = .T.
Name = "searchform"
PROCEDURE Destroy
LOCAL cFilterExpr
IF THIS.ChangedFilter
    IF
        EMPTY(THIS.SaveFilter)
            SET FILTER TO
        ELSE
            m.cFilterExpr =
THIS.SaveFilter
            SET FILTER TO
&cFilterExpr
ENDIF
IF RECCOUNT() # 0

```

```

GO
THIS.SaveRecord
ENDIF
ENDIF

ENDPROC
PROCEDURE Init
IF !EMPTY(ALIAS())
    this.SaveRecord =
RECNO()
    this.SaveFilter =
SET("FILTER")
    this.ChangedFilter = .F.
ENDIF
ENDPROC
Top = 3
Left = 12
Width = 452
Height = 160
BorderWidth = 0
TabIndex = 1
Name = "Searchclass1"
cboFields1.TabIndex = 1
cboFields1.Name = "cbofields1"
cboOperators1.TabIndex = 2
cboOperators1.Name =
"cbooperators1"
txtExpr1.TabIndex = 3
txtExpr1.Name = "txtexpr1"
cboFields2.TabIndex = 4
cboFields2.Name = "cbofields2"
cboOperators2.TabIndex = 5
cboOperators2.Name =
"cbooperators2"
txtExpr2.TabIndex = 6
txtExpr2.Name = "txtexpr2"
optGrpAndOr.Option1.Value = 1
optGrpAndOr.Option1.Left = 5
optGrpAndOr.Option1.Top = 5
optGrpAndOr.Option1.Name =
"option1"
optGrpAndOr.Option2.Value = 0
optGrpAndOr.Option2.Left = 5
optGrpAndOr.Option2.Top = 23
optGrpAndOr.Option2.Name =
"option2"
optGrpAndOr.Height = 46
optGrpAndOr.Width = 58
optGrpAndOr.TabIndex = 7
optGrpAndOr.Name = "optgrpandor"
Label1.Height = 16
Label1.Left = 12
Label1.Top = 7
Label1.Width = 102
Label1.Name = "label1"
Label2.Height = 16
Label2.Left = 158
Label2.Top = 7
Label2.Width = 93
Label2.Name = "label2"
Label3.Height = 16
Label3.Left = 262
Label3.Top = 7
Label3.Width = 165
Label3.Name = "label3"
Label4.Height = 16
Label4.Left = 12
Label4.Top = 99
Label4.Width = 102
Label4.Name = "label4"

```

```

Label5.Height = 16
Label5.Left = 161
Label5.Top = 99
Label5.Width = 93
Label5.Name = "label5"
label6.Height = 16
label6.Left = 262
label6.Top = 99
label6.Width = 165
label6.Name = "label6"
chkCaseSensitive.Top = 69
chkCaseSensitive.Left = 290
chkCaseSensitive.Height = 15
chkCaseSensitive.Width = 88
chkCaseSensitive.Name =
"chkCaseSensitive"
ButtonCount = 3
BackStyle = 0
BorderStyle = 0
Value = 1
Height = 31
Left = 118
Top = 175
Width = 241
Name = "Commandgroup1"
Command1.Top = 1
Command1.Left = 0
Command1.Height = 29
Command1.Width = 73
Command1.Caption = "\<Search"
Command1.TabIndex = 1
Command1.Name = "Command1"
Command2.Top = 1
Command2.Left = 84
Command2.Height = 29
Command2.Width = 73
Command2.Caption = "A<|I"
Command2.TabIndex = 2
Command2.Name = "Command2"
Command3.Top = 1
Command3.Left = 168
Command3.Height = 29
Command3.Width = 73
Command3.Cancel = .T.
Command3.Caption = "\<Cancel"
Command3.TabIndex = 3
Command3.Name = "Command3"
PROCEDURE Click
#DEFINE NORECSFOUND_LOC
    "No records were
found which meet your query. Please
try another query."

DO CASE
CASE THIS.VALUE = 1
    &&SET FILTER condition
    LOCAL
cGetExpr,cSavePoint
    SET FILTER TO
        m.cSavePoint =
SET("POINT")
    SET POINT TO "."
        m.cGetExpr =
THISFORM.Searchclass1.searchexp
r()
    SET POINT TO
&cSavePoint
    IF !EMPTY(m.cGetExpr)
        SET FILTER TO
&cGetExpr

```

```

LOCATE
* Check if no
records found
IF EOF()

=MESSAGEBOX(NOREC
SFOUND_LOC)

this.Parent.ChangedFilter
=.T.

RETURN
ENDIF

this.Parent.ChangedFilter
=.F.
ELSE
LOCATE
ENDIF
CASE THIS.VALUE = 2 &&SET
FILTER TO all
SET FILTER TO
LOCATE
this.Parent.ChangedFilter
=.F.
&&good query
OTHERWISE

ENDCASE

Release THISFORM
ENDPROC
Arial, 0, 9, 5, 15, 12, 13, 3, 0
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
Width = 542
Height = 25
BackStyle = 0
BorderWidth = 0
Visible = .T.
TabIndex = 1
nworkarea = 0
EditMode = 0
previewmode = .F.
topfile = 0
endfile = 0
oldrefresh = 0
oldrec = 0
oldreprocess = 0
previewinit = .T.
usedataenv = .T.
viewkey = ("")
Name = "txtbtns"
PROCEDURE buttonrefresh
* This is a generic routine which
refreshes the buttons
* for appropriate table environments.
IF SELECT()# THIS.nworkarea
SELECT (THIS.nworkarea)
ENDIF

this.SetAllProp()
this.cmdFind.Enabled =
!THIS.EditMode
this.cmdPrint.Enabled =
!THIS.EditMode
this.cmdExit.Enabled =
!THIS.EditMode

```

```

this.cmdDelete.Enabled =
!THIS.EditMode AND
!ISREADONLY()
this.setcaption()
ENDPROC
PROCEDURE initvars

#DEFINE C_NOUPDATEVIEW_LOC
"Edits to one or more of the
Views may not be permanent. "+;

"To remedy this, ensure
the View's Send SQL Updates
checkbox is checked in the View
Designer."
#DEFINE C_READONLY_LOC
"The table is
Read-Only. You will not be able to
edit it."

LOCAL
aTablesUsed,nTablesUsed,i,aMems,
nTotMem,cWizFile,IShowedMess,cD
ataEnvRef
DIMENSION aTablesUsed[1]
DIMENSION aMems[1]

* This routine sets the member
variables
this.viewkey = ""
this.ParentKey = ""
this.ViewType = 3
this.GridAlias = ""
this.topfile = .F.
this.endfile = .F.
this.AddMode = .F.

this.nworkarea = SELECT()
this.oldSetDelete =
SET("DELETED")
SET DELETED ON
this.oldreprocess =
SET("REPROCESS")
SET REPROCESS TO 0

* These properties should not be
used. They are reserved for use by
* the Preview button of the Form
Wizards.
this.previewmode =
IIF(TYPE("THIS.PreviewMode")#"L",.
F.,THIS.previewmode)
this.previewinit =
IIF(TYPE("THIS.PreviewInit")#"L",.T.,
THIS.previewinit)

* Check for data environment
DO CASE
CASE
TYPE("THISFORM.DataEnvironment
") = "O"
this.usedataenv = .T.
nTotMem =
AMEMBERS(aMems,THISFORM.Da
taEnvironment,2)
cDataEnvRef =
"THISFORM.DataEnvironment"

```

```

CASE
TYPE("THISFORMSET.DataEnviron
ment") = "O"
this.usedataenv = .T.
nTotMem =
AMEMBERS(aMems,THISFORMSE
T.DataEnvironment,2)
cDataEnvRef =
"THISFORMSET.DataEnvironment"
* let's also set the
IF
TYPE("THISFORM")="O" AND
!THISFORM.VISIBLE
thisform.VISIBLE
=.T.
ENDIF
OTHERWISE
this.usedataenv = .F.
ENDCASE

IF THIS.usedataenv
WITH
EVAL(m.cDataEnvRef)
* Check for
relation
FOR i = 1 TO
m.nTotMem
IF
UPPER(EVAL("."+aMems[m.i]+".Bas
eClass")) = "RELATION"
this.oDataRelation =
m.cDataEnvRef+"."+aMems[m.i]
EXIT
ENDIF
ENDFOR

* Check for
Views
FOR i = 1 TO
m.nTotMem
IF
UPPER(EVAL("."+aMems[m.i]+".Bas
eClass")) = "CURSOR"
WITH
EVAL("."+aMems[m.i])
IF
CURSORGETPROP("SourceType",.
Alias)#3 AND ;
|CURSORGETPROP("offli
ne",.Alias)
*
Check if we need to requery for
deleted records.
*
Note: Parameterized views are not
queried so all
records will be brought over. To
remedy this situation

```

* you

can place a SET DELETED ON command in the

```

*
BeforeOpenTables event of the
DataEnvironment.

IF
THIS.oldSetDelete = "OFF" AND
ATC("?",CURSORGETPROP("SQL",
.Alias))=0

=REQUERY(.Alias)
ENDIF

*
Check if updates are made

IF
!CURSORGETPROP("SendUpdates
",.Alias) AND !m.IShowedMess

=MESSAGEBOX(C_NOUP
DATEVIEW_LOC)

IShowedMess = .T.
ENDIF

ENDIF

ENDWITH
ENDIF
ENDFOR
ENDWITH
ENDIF

this.EditMode =
IF(TYPE("THIS.EditMode")# "L",.F.,T
HIS.EditMode)

IF ISREADONLY()
WAIT WINDOW
C_READONLY_LOC TIMEOUT 2
this.EditMode = .F.
ENDIF

* Disable appropriate buttons
this.cmdAdd.Enabled =
!ISREADONLY()
this.cmdEdit.Enabled =
!ISREADONLY()
this.cmdDelete.Enabled =
!ISREADONLY()

IF THIS.previewmode
RETURN
ENDIF

this.GetGridRef()

IF THIS.usedataenv
RETURN
ENDIF

```

```

* The following code is here to
support forms not using a
DataEnvironment.
this.oldSetFields = SET("FIELDS")
SET FIELDS OFF
this.oldMultiLocks =
SET("MULTILOCKS")
SET MULTILOCKS ON
this.oldrefresh = SET("REFRESH")
SET REFRESH TO 5
IF !EMPTY(ALIAS())
this.oldBuffering=CursorGe
tProp("buffering")
m.nTablesUsed =
AUSED(aTablesUsed)
FOR i = 1 TO
m.nTablesUsed
IF
CursorGetProp('sourcetype',aTables
Used[m.i,1])#3 &&skip for views

=CursorSetProp("buffering"
,5,aTablesUsed[m.i,1])
&&optimistic table buffering
ENDIF
ENDFOR
ENDIF
GO TOP

ENDPROC
PROCEDURE updaterecords
#DEFINE E_FAIL_LOC
"Failed to update
table: "
#DEFINE E_TRIGGERFAIL_LOC
"Trigger failed."
#DEFINE E_FIELDNULL_LOC
"Field doesn't
accept NULL"
#DEFINE E_FIELDRULE_LOC
"Field rule
violated"
#DEFINE E_RECORDLOCK_LOC
"Record in use by another
user"
#DEFINE E_ROWRULE_LOC
"Row rule
violated"
#DEFINE E_UNIQUEINDEX_LOC
"Unique index violation"
#DEFINE E_DIRTYREC_LOC
"Data has been
changed by another user. Overwrite
changes with your edits?"
#DEFINE E_NOFORCE_LOC
"Could not force table
updates."
#DEFINE E_PROMPT_LOC
"Error: "
#DEFINE MSGBOX_YES
6

LOCAL
aErrors,cErrorMessage,aTablesUsed
,nTablesUsed,nTotErr
LOCAL
nFld,i,nOldArea,ISuccess,lnDBC,IOv
erwrite,IHadMessage

DIMENSION aTablesUsed[1]

```

```

DIMENSION aErrors[1]
m.cErrorMessage=""
m.ISuccess = .T.
m.nOldArea = SELECT()
m.nTablesUsed =
AUSED(aTablesUsed)

* Can wrap everything in transaction
if using strictly DBCs

FOR i = 1 TO m.nTablesUsed

SELECT
(aTablesUsed[m.i,1])

m.lnDBC =
!EMPTY(CURSORGETPROP("Data
base"))
m.cErrorMessage = ""
m.IOverwrite = .F.
m.IHadMessage = .F.

DO CASE
CASE
CURSORGETPROP("Buffering") = 1
* Skip if buffering
not on
LOOP
CASE GetFldState(0) = 2
&&deleted record
* Only delete
current record and force it
m.ISuccess =
TableUpdate(.F.,.T.)
IF m.ISuccess
&&successful update
LOOP
ENDIF
CASE !m.lnDBC AND
(ATC("2",GetFldState(-1))#0 OR;
ATC("3",GetFldState(-1))#0)
* Field was
edited - in Free Table
* Since free
tables are not supported by
transactions,
* we must
process record by record
m.nModRecord =
GetNextMod(0)
DO WHILE
m.nModRecord # 0 &&loop locks all
records
GO
m.nModRecord

m.ISuccess = RLOCK()
&&try to lock record
IF
!m.ISuccess
&&failed to lock record

m.cErrorMessage =
E_RECORDLOCK_LOC

UNLOCK ALL

```

```

EXIT
                                ENDIF
                                IF
!m.lHadMessage &&so we don't
repeat alert
                                * See if record(s) modified
                                by another user
                                FOR m.nFld = 1 TO
                                FCOUNT()
                                IF
                                TYPE(FIELD(m.nFld)) = "G" &&skip
                                for General fields
                                LOOP
                                ENDIF
                                IF
                                OLDVAL(FIELD(m.nFld)) #
                                CURVAL(FIELD(m.nFld))
                                m.lHadMessage = .T.
                                IF
                                MESSAGEBOX(E_DIRTYREC_LOC
                                ,4+48) = MSGBOX_YES
                                m.lOverwrite = .T.
                                ELSE
                                m.lSuccess = .F.
                                UNLOCK ALL
                                EXIT
                                ENDIF
                                ENDIF
                                ENDFOR
                                ENDIF
                                m.nModRecord =
                                GetNextMod(m.nModRecord)
                                ENDDO
                                IF m.lSuccess
                                &&was able to lock all
                                records
                                m.lSuccess =
                                TableUpdate(.T.,m.lOverwrite)
                                IF
                                m.lSuccess &&was able to update all
                                records
                                LOOP
                                ENDIF
                                UNLOCK ALL

```

```

                                ENDIF
                                CASE m.lInDBC
                                BEGIN
                                TRANSACTION
                                * Try to update
                                all records in selected table
                                m.lSuccess =
                                TableUpdate(.T...F.)
                                &&successful update
                                IF m.lSuccess
                                END
                                TRANSACTION
                                LOOP
                                ENDIF
                                ROLLBACK
                                ENDCASE
                                * Handle errors
                                nTotErr
                                =AERROR(aErrors)
                                DO CASE
                                CASE nTotErr = 0
                                CASE aErrors[1,1] = 1539
                                && Trigger failed
                                m.cErrorMessage =
                                E_TRIGGERFAIL_LOC
                                CASE aErrors[1,1] = 1581
                                && Field doesn't accept
                                NULL
                                m.cErrorMessage =
                                E_FIELDNULL_LOC
                                CASE aErrors[1,1] = 1582
                                && Field rule violated
                                m.cErrorMessage =
                                E_FIELDRULE_LOC
                                CASE aErrors[1,1] = 1700
                                && Record in use by
                                another user
                                m.cErrorMessage =
                                E_RECORDLOCK_LOC
                                CASE aErrors[1,1] = 1583
                                && Row rule violated
                                m.cErrorMessage =
                                E_ROWRULE_LOC
                                CASE aErrors[1,1] = 1884
                                && Unique index violation
                                m.cErrorMessage =
                                E_UNIQUEINDEX_LOC
                                CASE aErrors[1,1] = 1585
                                && Record changed by
                                another user
                                IF m.lInDBC
                                &&handle free
                                tables above

```

```

Display conflict alert
                                IF
                                MESSAGEBOX(E_DIRTYREC_LOC
                                ,4+48) = MSGBOX_YES
                                *Try to force update
                                BEGIN TRANSACTION
                                m.lSuccess =
                                TABLEUPDATE(.T.,T.)
                                IF m.lSuccess
                                END
                                TRANSACTION
                                LOOP
                                ELSE
                                ROLLBACK
                                =MESSAGEBOX(E_NOFO
                                RCE_LOC)
                                ENDIF
                                ENDIF
                                OTHERWISE
                                IF
                                !EMPTY(m.cErrorMessage) &&for
                                free table handling above
                                m.cErrorMessage =
                                E_PROMPT_LOC+aErrors[1,2]
                                ENDIF
                                ENDCASE
                                * Had an error we couldn't
                                handle
                                =TABLEREVERT(.T.)
                                &&revert all records
                                m.lSuccess = .F.
                                IF
                                !EMPTY(m.cErrorMessage)
                                =MESSAGEBOX(E_FAIL_
                                LOC+m.cErrorMessage)
                                ENDIF
                                ENDFOR
                                SELECT (m.nOldArea)
                                RETURN m.lSuccess
                                ENDPROC
                                PROCEDURE setcaption
                                #DEFINE ADD_CAPTION_LOC
                                "\<Add"
                                #DEFINE EDIT_CAPTION_LOC
                                "\<Edit"
                                #DEFINE REV_CAPTION_LOC
                                "\<Revert"
                                #DEFINE SAVE_CAPTION_LOC
                                "\<Save"

```

```

IF THIS.EditMode
    this.cmdAdd.Caption =
SAVE_CAPTION_LOC
    this.cmdEdit.Caption =
REV_CAPTION_LOC
ELSE
    this.cmdAdd.Caption =
ADD_CAPTION_LOC
    this.cmdEdit.Caption =
EDIT_CAPTION_LOC
ENDIF

ENDPROC
PROCEDURE SetAllProp
LPARAMETER oContainer

* Checks for General fields
LOCAL i,oControlParent,nCtrlCount

IF PARAMETERS() = 0
    m.oControlParent =
THISFORM
ELSE
    m.oControlParent =
m.oContainer
ENDIF

DO CASE
CASE
ATC("Pageframe",m.oControlParent.
BaseClass)#0
    nCtrlCount =
oControlParent.PageCount
CASE
ATC(m.oControlParent.BaseClass,"
Optiongroup,Commandgroup")#0
    nCtrlCount =
oControlParent.ButtonCount
OTHERWISE
    nCtrlCount =
oControlParent.ControlCount
ENDCASE

FOR i = 1 TO m.nCtrlCount
    DO CASE
    CASE
ATC("Pageframe",m.oControlParent.
BaseClass)#0

        this.SetAllProp(m.oControl
Parent.Pages[m.i])

        CASE
ATC(m.oControlParent.BaseClass,"
Optiongroup,Commandgroup")#0
AND ;

            THIS.UserControlMode

            m.oControlParent.Buttons[
m.i].Enabled = THIS.EditMode

        CASE
ATC(m.oControlParent.Controls[m.i].
BaseClass,"Optiongroup,Commandg
roup")#0 ;

            AND
THIS.UserControlMode

```

```

        this.SetAllProp(m.oControl
Parent.Controls[m.i])

        CASE
ATC("Container",m.oControlParent.C
ontrols[m.i].BaseClass) # 0 OR;

            ATC("Page",m.oControlPar
ent.Controls[m.i].BaseClass) # 0

            this.SetAllProp(m.oControl
Parent.Controls[m.i])

            CASE
ATC(m.oControlParent.Controls[m.i].
BaseClass,"ListBox,ComboBox,Spin
ner") # 0 AND;

                THIS.UserControlMode

                m.oControlParent.Controls[
m.i].Enabled = THIS.EditMode

            CASE
ATC(m.oControlParent.Controls[m.i].
BaseClass,"CheckBox,TextBox,OleB
oundControl") # 0

                m.oControlParent.Controls[
m.i].Enabled = THIS.EditMode

            CASE
ATC(m.oControlParent.Controls[m.i].
BaseClass,"EditBox") # 0

                m.oControlParent.Controls[
m.i].ReadOnly = !THIS.EditMode
                IF
!THIS.HasMemo
                    WITH
m.oControlParent.Controls[m.i]

                        this.EditForeColor =
.ForeColor

                        this.EditDisForeColor =
.DisabledForeColor

                        this.EditBackColor =
.BackColor

                        this.EditDisBackColor =
.DisabledBackColor

                        this.HasMemo = .T.

                    ENDWITH
                ENDIF

                m.oControlParent.Controls[
m.i].ForeColor =
IIF(THIS.EditMode,THIS.EditForeCol
or,THIS.EditDisForeColor)

                m.oControlParent.Controls[
m.i].BackColor =
IIF(THIS.EditMode,THIS.EditBackCol
or,THIS.EditDisBackColor)

```

```

CASE
ATC(m.oControlParent.Controls[m.i].
BaseClass,"Grid") # 0

    m.oControlParent.Controls[
m.i].ReadOnly = !THIS.EditMode

    m.oControlParent.Controls[
m.i].DeleteMark = THIS.EditMode

ENDCASE
ENDFOR

ENDPROC
PROCEDURE navrefresh
**** Navigational Button Handling ****
LOCAL
OldLockScreen,KeyValue,cFitExpr
m.OldLockScreen =
THISFORM.LockScreen
thisform.LockScreen = .T.

IF SELECT()#THIS.nworkarea
    SELECT (THIS.nworkarea)
ENDIF

IF !THIS.EditMode

    * Check for bottom of file
this.endfile = EOF() OR
THIS.endfile

    * Test to see we are on last
record
IF !THIS.endfile
    Skip
    this.endfile =
EOF()
    Skip -1
ELSE
    GO BOTTOM
ENDIF

    * Check for top of file
this.topfile = BOF() OR
EOF() OR THIS.topfile

    * Test to see if we are on
first record
IF !THIS.topfile
    Skip -1
    this.topfile =
BOF()
    IF !THIS.topfile
        Skip
    ENDIF
ENDIF

    IF THIS.topfile
        GO TOP
    ENDIF

ENDIF

this.cmdTop.Enabled = !THIS.topfile
AND !THIS.EditMode
this.cmdPrev.Enabled = !THIS.topfile
AND !THIS.EditMode

```

```

this.cmdNext.Enabled =
!THIS.endfile AND !THIS.EditMode
this.cmdEnd.Enabled = !THIS.endfile
AND !THIS.EditMode

```

```

* Check if no records in query set
DO CASE
CASE THIS.previewmode OR
ISREADONLY()
    * Nothing
CASE THIS.EditMode AND
CURSORGETPROP("BUFFERING")
=1
    this.cmdEdit.Enabled = .F.
CASE THIS.EditMode
    this.cmdEdit.Enabled = .T.
CASE RECCOUNT()=0 OR BOF()
OR EOF()
    this.cmdEdit.Enabled = .F.
    this.cmdDelete.Enabled =
.F.
CASE !THIS.cmdEdit.Enabled
    this.cmdEdit.Enabled = .T.
    this.cmdDelete.Enabled =
.T.
ENDCASE

```

```

* Update Grid for Views
IF !THIS.EditMode AND
!EMPTY(THIS.viewkey)
    KeyValue =
EVAL(THIS.ParentKey)
    DO CASE
    CASE
TYPE(THIS.ParentKey) = "C"
    cFiltExpr =
THIS.viewkey + "=" +
["+m.KeyValue+"]
    CASE
TYPE(THIS.ParentKey) = "L"
    cFiltExpr =
THIS.viewkey
    CASE
TYPE(THIS.ParentKey) = "D"
    cFiltExpr =
THIS.viewkey + "=" +
["+DTC(m.KeyValue)+"]
    CASE
TYPE(THIS.ParentKey) = "T"
    cFiltExpr =
THIS.viewkey + "=" +
["+TTC(m.KeyValue)+"]
    OTHERWISE
        * Numeric
        cFiltExpr =
THIS.viewkey + "=" +
ALLTRIM(STR(m.KeyValue,20,18))
    ENDCASE

```

```

SELECT (THIS.GridAlias)
DO CASE
CASE .F.
&&parameterized query
    * set parameter
here
    * =query()
CASE THIS.ViewType = 1
&&local views

```

```

SET FILTER TO
&cFiltExpr
CASE THIS.ViewType = 2
&&remote views
ENDCASE
SELECT (THIS.nworkarea)
ENDIF

```

```

thisform.Refresh()
thisform.LockScreen =
m.OldLockScreen
ENDPROC
PROCEDURE GetGridRef
* Check if we have a grid
LOCAL aMems,nTotMem,i
this.GridRef = ""
IF TYPE("THISFORM") = "O"
    DIMENSION aMems[1]
    nTotMem =
AMEMBERS(aMems,THISFORM,2)
    WITH THISFORM
        FOR i = 1 TO
m.nTotMem
            IF
UPPER(EVAL("."+aMems[m.i]+".Bas
eClass")) = "GRID"

```

```

                this.GridRef = aMems[m.i]
            WITH
EVAL("."+THIS.GridRef)
                * Check if we
have a view and get Tag property
                this.ViewType =
CURSORGETPROP('sourcetype',.R
ecordSource)
                this.GridAlias =
.RecordSource
            IF
THIS.ViewType # 3

```

```

                this.viewkey = .Tag
                this.ParentKey =
.Comment
            ENDFOR
        ENDFOR
    ENDFOR
ENDWITH
EXIT
ENDIF
ENDPROC
PROCEDURE Refresh
**** Special Preview Mode Handling
****
IF THIS.previewmode AND
THIS.previewinit

```

```


```

```

this.previewinit = .F.
this.cmdAdd.Enabled = .F.
this.cmdEdit.Enabled = .F.
this.cmdDelete.Enabled =
.F.
this.cmdFind.Enabled = .F.
this.cmdPrint.Enabled = .F.
this.cmdExit.Enabled = .F.
this.nworkarea =
SELECT()
this.GetGridRef()
this.SetAllProp()
this.navrefresh()
ENDIF

```

```

ENDPROC
PROCEDURE Init
#DEFINE C_WIZSTYLE
    "WIZSTYLE.VCX"
#DEFINE C_WIZDIR
    "WIZARDS"
#DEFINE C_PROMPT1_LOC
    "Find: "
#DEFINE E_NOSTYLE_LOC
    "The class library
(WIZSTYLE.VCX) needed by this
form could not be found. "+;

```

```

    "Please locate."
LOCAL
cGridRef,cWizHomePath,separator,c
WizStyFile
IF TYPE('THIS.Parent') # "O"
    RETURN
ENDIF
IF SET("TALK") = "ON"
    SET TALK OFF
    this.oldTalk = "ON"
ELSE
    this.oldTalk = "OFF"
ENDIF

```

```

IF
ATC(C_WIZSTYLE,SET("CLASSLIB
")) = 0
    * Returns just the
pathname
    cWizHomePath =
_WIZARD
    IF '\ $ cWizHomePath
    cWizHomePath =
SUBSTR(m.cWizHomePath,1,RAT('\
,m.cWizHomePath))
    IF
RIGHT(m.cWizHomePath,1) = '\'
AND LEN(m.cWizHomePath) > 1 ;
        AND
SUBSTR(m.cWizHomePath,LEN(m.c
WizHomePath)-1,1) <> ':'
        cWizHomePath =
SUBSTR(m.cWizHomePath,1,LEN(
m.cWizHomePath)-1)

```

```

ENDIF
ELSE
  cWizHomePath = "
ENDIF

* Add a backslash unless
there is one already there.
Separator =
IIF(_MAC, "\", "\")
IF
  !(RIGHT(m.cWizHomePath, 1) $ '\:')
AND !EMPTY(m.cWizHomePath)
  m.cWizHomePath=
m.cWizHomePath+ m.separator
ENDIF

DO CASE
CASE FILE(C_WIZSTYLE)
  cWizFile =
C_WIZSTYLE
CASE
FILE(m.cWizHomePath+C_WIZSTY
LE)
  cWizFile =
m.cWizHomePath+C_WIZSTYLE
CASE
FILE(m.cWizHomePath+C_WIZDIR+
C_WIZSTYLE)
  cWizFile =
m.cWizHomePath+C_WIZDIR+C_WI
ZSTYLE
CASE
FILE(HOME()+C_WIZSTYLE)
  cWizFile =
HOME()+C_WIZSTYLE
CASE
FILE(HOME()+C_WIZDIR+C_WIZST
YLE)
  cWizFile =
HOME()+C_WIZDIR+C_WIZSTYLE
OTHERWISE

=MESSAGEBOX(E_NOST
YLE_LOC)
  cWizFile =
GETFILE("VCX", C_PROMPT1_LOC
+C_WIZSTYLE)
ENDCASE

IF
ATC(C_WIZSTYLE, m.cWizFile)#0
SET CLASS TO
(m.cWizFile) ADDITIVE
ELSE
  * Failed to get
WIZSTYLE.VCX file
RETURN .F.
ENDIF

ENDIF

this.initvars()
this.buttonrefresh()
this.navrefresh()
cGridRef=THIS.GridRef
IF !EMPTY(m.cGridRef)
  * Change this if you desire
to have the grid initially selected.
THISFORM.&cGridRef..SetFocus()

```

```

ENDIF
ENDPROC
PROCEDURE Error
PARAMETERS nError, cMethod,
nLine
LOCAL aFoxErr, nTotErr
DIMENSION aFoxErr[1]
nTotErr = AERROR(aFoxErr)

DO CASE
CASE INLIST(m.nError, 1733, 1734)
&&property not found -- traps
SETALL()
RETURN
CASE m.nError=1938 &&no parent
RETURN
CASE nTotErr>0 AND aFoxErr[1, 1] =
1420
  * Corrupt Ole object in
General field.
=MESSAGEBOX(aFoxErr[
1, 2])
RETURN
CASE nTotErr>0 AND aFoxErr[1, 1] =
1884
  * Uniqueness ID error
=MESSAGEBOX(MESSA
GE()+CHR(13))
RETURN
ENDCASE

**** Error Dialog ****
=MESSAGEBOX(MESSAGE(1)+CH
R(13)+;
"Error: "+STR(nError)+CHR(13)+;
MESSAGE()+CHR(13)+;
"Method: "+cMethod+CHR(13)+;
"Line: "+STR(nLine))

RETURN TO MASTER

ENDPROC
PROCEDURE Destroy
* Restore various settings
LOCAL
nTablesUsed, aTablesUsed, i, nDECur
sors, aDECursors, cDataEnvRef
DIMENSION aTablesUsed[1]

IF TYPE("THIS.Parent") # "O"
RETURN
ENDIF

IF TYPE("THIS.oldTalk") = "C" AND
THIS.oldTalk="ON"
SET TALK ON
ENDIF

* OLE Servers can still send data
back to General fields
* even though they are not in Edit
Mode. We need to reset
* buffering to 1 so the buffer is not
updated by the OLE Server.
* Also, folks might exit out while
editing.
IF THIS.usedataenv
DIMENSION
aDECursors[1]
DO CASE

```

```

CASE
TYPE("THISFORM.DataEnvironment") = "O"
  nDECursors =
AMEMBERS(aDECursors, THISFOR
M.DataEnvironment, 2)
  cDataEnvRef =
"THISFORM.DataEnvironment"
CASE
TYPE("THISFORMSET.DataEnviron
ment") = "O"
  nDECursors =
AMEMBERS(aDECursors, THISFOR
MSET.DataEnvironment, 2)
  cDataEnvRef =
"THISFORMSET.DataEnvironment"
ENDCASE
FOR i = 1 TO
m.nDECursors
WITH
EVAL(m.cDataEnvRef + "." +
aDECursors[m.i])
IF
USED(.ALIAS) AND
ATC("CURSOR", .BaseClass)#0 AND
;
CursorGetProp("sourcetype", .ALIAS)
=3 AND ;
CursorGetProp("buffering", .ALIAS)>1
=TableRevert(.T., .ALIAS)
=CursorSetProp("buffering"
, 1, .ALIAS) &&optimistic
table buffering
ENDIF
ENDWITH
ENDFOR
ENDIF

* Skip if using preview mode
IF THIS.previewmode
RETURN
ENDIF

IF THIS.oldSetDelete = "OFF"
SET DELETED OFF
ENDIF
SET REPROCESS TO
THIS.oldreprocess
SET MESSAGE TO
SELECT (THIS.nworkarea)

IF THIS.usedataenv
RETURN
ENDIF

* The following code is here to
support
* forms not using a
DataEnvironment.
m.nTablesUsed =
AUUSED(aTablesUsed)
FOR i = 1 TO m.nTablesUsed
IF
USED(aTablesUsed[m.i, 1]) AND
ATC(".TMP", DBF(aTablesUsed[m.i, 1
]))=0 &&skip for views

```

```

        =CursorSetProp("buffering"
,THIS.oldBuffering,aTablesUsed(m.i
1))      &&optimistic table buffering
        ENDIF
    ENDFOR

    IF THIS.oldMultiLocks = "OFF"
        SET MULTILOCKS OFF
    ENDIF

    IF THIS.oldSetFields = "ON"
        SET FIELDS ON
    ENDIF

    SET REFRESH TO THIS.oldrefresh

    ENDPROC
    Top = 0
    Left = 56
    Height = 24
    Width = 55
    FontName = "MS Sans Serif"
    FontSize = 8
    Caption = "\<Prev"
    StatusBarText = "Skip to previous
record"
    TabIndex = 2
    ToolTipText = ""
    Name = "cmdPrev"
    PROCEDURE Click
    SELECT (THIS.parent.nworkarea)
    IF !BOF()
        Skip -1
    ENDIF

    this.Parent.topfile = BOF()
    this.Parent.endfile = EOF()

    this.Parent.navrefresh()
    ENDPROC
    Top = 0
    Left = 112
    Height = 24
    Width = 55
    FontName = "MS Sans Serif"
    FontSize = 8
    Caption = "\<Next"
    StatusBarText = "Skip to next record"
    TabIndex = 3
    ToolTipText = ""
    Name = "cmdNext"
    PROCEDURE Click
    SELECT (THIS.parent.nworkarea)
    IF !EOF()
        Skip 1
    ENDIF

    this.Parent.endfile = EOF()
    this.Parent.topfile = BOF()

    this.Parent.navrefresh()

    ENDPROC
    Top = 0
    Left = 0
    Height = 24
    Width = 55
    FontName = "MS Sans Serif"
    FontSize = 8

```

```

    Caption = "\<Top"
    StatusBarText = "Go to top of table"
    TabIndex = 1
    ToolTipText = ""
    Name = "cmdTop"
    PROCEDURE Click
    SELECT (THIS.parent.nworkarea)
    LOCATE
    this.Parent.topfile = .T.
    this.Parent.endfile = EOF()
    this.Parent.navrefresh()
    ENDPROC
    Top = 0
    Left = 168
    Height = 24
    Width = 55
    FontName = "MS Sans Serif"
    FontSize = 8
    Caption = "\<Bottom"
    StatusBarText = "Go to bottom of
table"
    TabIndex = 4
    ToolTipText = ""
    Name = "cmdEnd"
    PROCEDURE Click
    SELECT (THIS.parent.nworkarea)
    GO BOTTOM
    this.Parent.topfile = .F.
    this.Parent.endfile = .T.
    this.Parent.navrefresh()
    ENDPROC
    Top = 0
    Left = 224
    Height = 24
    Width = 49
    FontName = "MS Sans Serif"
    FontSize = 8
    Caption = "\<Find"
    Enabled = .T.
    StatusBarText = "Search for records"
    TabIndex = 5
    ToolTipText = ""
    Name = "cmdFind"
    PROCEDURE Click
    LOCAL oSearchDlg
    LOCAL !VisChange,!StateChange

    * Check if SDI Window
    IF THISFORM.ShowWindow = 2
        IF !_VFP.Visible
            !_VFP.Visible =
.T.
            !VisChange = .T.
        ENDIF
        IF !_SCREEN.WindowState
            = 1
        ELSE
            !_SCREEN.WindowState =
0
            !StateChange =
.T.
        ENDIF
    ENDIF

    oSearchDlg =
CREATE("searchform")
    oSearchDlg.SHOW()

    IF m.VisChange
        !_VFP.Visible = .F.

```

```

    ENDIF
    IF m.IStateChange
        !_SCREEN.WindowState =
1
    ENDIF

    IF THISFORM.ShowWindow = 2
        Activate Window
        (THISFORM.Name)
    ENDIF

    * Reset from prior
    this.Parent.topfile = .F.
    this.Parent.endfile = .F.
    this.Parent.navrefresh()
    ENDPROC
    Top = 0
    Left = 274
    Height = 24
    Width = 49
    FontName = "MS Sans Serif"
    FontSize = 8
    Caption = "P\<rint"
    Enabled = .T.
    StatusBarText = "Print to report"
    TabIndex = 6
    ToolTipText = ""
    Name = "cmdPrint"
    PROCEDURE Click
    #DEFINE C_MAKEREPO_LOC
    "Could
not locate a report to print. Create
new one?"
    #DEFINE C_NOOPEN_LOC

    "Error opening table.
Unable to print report."
    #DEFINE
C_GETFILEPROMPT_LOC
    "Pick report:"

    LOCAL
    cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

    cSaveAlias = ALIAS()
    cSaveSource =
CURSORGETPROP("SourceName")
    cSaveData =
CURSORGETPROP("Database")
    cDiffSource = ""
    cRepName =
LEFT(ALIAS(),8)+".FRX"
    nSaveSess =
SET("DATASESSION")

    * Handling for Private data sessions
    #IF 0
        IF m.nSaveSess # 1
            SET
DATASESSION TO 1
            SELECT 0
            IF
!EMPTY(m.cSaveData)
                OPEN
DATABASE (m.cSaveData)
            ENDIF
            IF
USED(m.cSaveAlias)

```



```

        SELECT (m.cSaveAlias)
            IF
CURSORGETPROP("SourceName")
#m.cSaveSource

        cDiffSource =
CURSORGETPROP("SourceName")

        USE IN (m.cSaveAlias)

        SELECT 0
            ENDIF
        ENDIF
        IF
EMPTY(ALIAS())
            USE
(m.cSaveSource) AGAIN ALIAS
(m.cSaveAlias) SHARED
        IF
EMPTY(ALIAS())
            =MESSAGEBOX(C_NOO
PEN_LOC)

            RETURN
        ENDIF
        ENDIF
#ENDIF

IF FILE(m.cRepName)
    REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
    m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
    IF !EMPTY(m.cRepName)
        IF
FILE(m.cRepName)
            * User
pressed Open button

            REPORT FORM
(m.cRepName) PREVIEW NOWAIT
            ELSE
                * User
pressed New button

            DO
HOME()+ "wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
            ENDIF
        ENDIF
    ENDIF

* Private data session code
#IF 0
        IF !EMPTY(cDiffSource)
            USE
(m.cDiffSource) IN 0
        ENDIF
        SET DATASESSION TO
m.nSaveSess
        SELECT (m.cSaveAlias)
#ENDIF

IF THISFORM.ShowWindow = 2
    Activate Window
(THISFORM.Name)

```

```

ENDIF

ENDPROC
Top = 0
Left = 486
Height = 24
Width = 53
FontName = "MS Sans Serif"
FontSize = 8
Caption = "E\<xit"
Enabled = .T.
StatusBarText = "Exit input form"
TabIndex = 10
ToolTipText = ""
Name = "cmdExit"
PROCEDURE Click
thisform.Release

ENDPROC
Top = 0
Left = 324
Height = 24
Width = 53
FontName = "MS Sans Serif"
FontSize = 8
Caption = "\<Add"
Enabled = .T.
StatusBarText = "Add new record /
Save edits"
TabIndex = 7
ToolTipText = ""
Name = "cmdAdd"
PROCEDURE Error
LPARAMETERS nError, cMethod,
nLine
this.parent.error(m.nError,
m.cMethod, m.nLine)
ENDPROC
PROCEDURE Click
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
#DEFINE MB_A_YES
6

#DEFINE
C_KEYFLDNOUPDATE_LOC
"The field relating the grid's
view to the parent data source is not
updatable. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
"The fields
relating the parent and child tables
are not the same data type. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
"The child data source is a
view and does not send updates. "+;
"Do you just want to add a
new record to the parent table?"

```

```

#DEFINE C_NOOBJ_LOC
"Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
"You cannot add a new
record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
"You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."

LOCAL
oSearchDlg,oAddRec,cChildAlias,c
PapaAlias,i,IPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,IBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
IBadChildKey,IUpdatableParentKey,I
NoSendParentUpdates,I NoSendChil
dUpdates

DO CASE
CASE THIS.Parent.EditMode

    ** Code for saving record
this.Parent.updaterows()

CASE EMPTY(THIS.Parent.GridRef)
    && not using
Grid object

    ** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IF
CURSORGETPROP("SourceType")#
3 AND ;

!CURSORGETPROP("offline") AND ;

!CURSORGETPROP("Sen
dUpdates")

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
APPEND BLANK

OTHERWISE

    ** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IPrimeKey = .F.

```

```

    &&is child key primary
index
    IF
!EMPTY(THIS.Parent.oDataRelation)
    oRel =
    EVAL(THIS.Parent.oDataRelation)
    ENDIF

    DO CASE
    CASE TYPE("m.oRel") =
"O"
    &&data environment
    WITH oRel

    cPapaAlias = .ParentAlias

    cPapaKey =
    .RelationalExpr

    cChildAlias = .ChildAlias

    cChildKey = .ChildOrder
    ENDWITH
    oRel = .NULL.

    &&reset
    CASE
!EMPTY(THIS.Parent.viewkey)
    &&using views
    cPapaAlias =
    ALIAS()
    cPapaKey =
    THIS.Parent.ParentKey
    cChildAlias =
    THIS.Parent.GridAlias
    cChildKey =
    THIS.Parent.viewkey

    * Need to check
    if Key field is updatable
    IF
    CURSORGETPROP("SourceType",
    m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

    ATC(", "+m.cChildKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0

    IBadViewKey = .T.
    ENDIF
    OTHERWISE
    cPapaAlias =
    ALIAS()
    cPapaKey =
    RELATION(1)
    cChildAlias =
    THIS.Parent.GridAlias
    cChildKey =
    ORDER(m.cChildAlias)
    ENDCASE

    * Test to see if both keys
    are of the same data type
    SELECT
    (THIS.Parent.GridAlias)
    cGridKeyType = ""

    IF
    EMPTY(THIS.Parent.viewkey) AND

```

```

!EMPTY(m.cChildKey)
    &&regular table used
    * Get the Child
    Key field since only index tag is here
    * if these is an
    expression, then default to blank.
    cTagName = ""
    FOR i = 1 TO
    TagCount("")
    IF
    UPPER(TAG(m.i)) ==
    UPPER(m.cChildKey)

    cTagName = KEY(m.i)

    IPrimeKey =
    PRIMARY(m.i)

    EXIT
    ENDIF
    ENDFOR

    * Check if we
    have an index expression here.
    IF
    ATC(", ",m.cTagName)#0 OR
    ATC(",+",m.cTagName)#0

    cChildKey = ""
    ELSE
    cChildKey = m.cTagName

    cGridKeyType =
    TYPE(m.cChildKey)
    ENDIF
    ENDIF

    IF
    CURSORGETPROP("SourceType",
    m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cPapaAlias)

    INoSendParentUpdates =
    .T.
    ENDIF
    IF
    CURSORGETPROP("SourceType",
    m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cChildAlias)

    INoSendChildUpdates = .T.
    ENDIF

    * Check if have two views
    and both do not send updates
    IF
    m.INoSendParentUpdates AND
    m.INoSendChildUpdates

    =MESSAGEBOX(C_NOUP
    DATE_LOC)

```

```

    RETURN
    ENDIF

    SELECT
    (THIS.Parent.OldAlias )
    IF
    !EMPTY(m.cGridKeyType) AND
    m.cGridKeyType #
    TYPE(m.cPapaKey)
    IBadChildKey =
    .T.
    ENDIF

    DO CASE
    CASE
    m.INoSendParentUpdates AND
    (m.IBadViewKey OR
    m.IBadChildKey)

    =MESSAGEBOX(C_NOUP
    DATE_LOC)
    RETURN
    CASE m.IPrimeKey AND
    m.INoSendParentUpdates

    =MESSAGEBOX(C_NOUP
    DATE2_LOC)
    RETURN
    CASE
    m.INoSendChildUpdates
    &&child view does not
    send updates
    IF
    MESSAGEBOX(C_NOCHILDUPDAT
    E_LOC,MB_Q_YESNO) #
    MB_A_YES

    RETURN
    ENDIF
    APPEND BLANK
    CASE m.IBadViewKey
    &&view key in not
    updatable
    IF
    MESSAGEBOX(C_KEYFLDNOUPD
    ATE_LOC,MB_Q_YESNO) #
    MB_A_YES

    RETURN
    ENDIF
    APPEND BLANK
    CASE m.IBadChildKey
    &&bad child key -- different
    data type from parent
    IF
    MESSAGEBOX(C_BADCHILDKEY_
    LOC,MB_Q_YESNO) # MB_A_YES

    RETURN
    ENDIF
    APPEND BLANK
    OTHERWISE
    * Need to check
    if Key field is updatable

    IUpdatableParentKey = .T.
    IF
    CURSORGETPROP("SourceType",
    m.cPapaAlias)#3 AND

```

```

!CURSORGETPROP("offline",m.cPapaAlias) AND;

      ATC(", "+m.cPapaKey+",",",",
"+CURSORGETPROP("UpdatableFieldList",m.cPapaAlias)+",")=0
*
cPapaKey = ""

      !UpdatableParentKey = .F.
      ENDIF

      * Check for
      expression in cPapaKey if so, then
      do not update record with keyvalue
      IF
      ATC(", "+m.cPapaKey)+"#0 OR
      ATC(", "+m.cPapaKey)+"#0

      cPapaKey = ""
      ENDIF

      nSaveSess =
      SET("DATASESSION")
      oAddRec =
      CREATE("GridAddRec")
      IF
      TYPE("m.oAddRec") # "O"

      =MESSAGEBOX(C_NOO
      BJ_LOC)

      RETURN
      ENDIF

      oAddRec.ChildPrimaryKey
      = m.IPrimeKey

      oAddRec.UpdatableParent
      Key = m.IUpdatableParentKey

      oAddRec.NoSendUpdates
      = m.INoSendParentUpdates

      oAddRec.KeyField =
      m.cPapaKey

      oAddRec.KeyValue =
      IIF(IEEMPTY(m.cPapaKey),
      EVAL(m.cPapaKey), "")

      oAddRec.runaddform()
      IF
      oAddRec.AddOption =
      OPT_CANCEL

      RETURN
      ENDIF

      SET
      DATASESSION TO nSaveSess
      SELECT
      (THIS.Parent.OldAlias)

      IF
      TYPE("oAddRec.KeyValue") = "C"

      oAddRec.KeyValue =
      TRIM(oAddRec.KeyValue)
      ENDIF

```

```

      * Add record to
      parent table
      nSaveRec =
      RECNO()
      IF
      INLIST(oAddRec.AddOption,OPT_A
      DD_PARENT,OPT_ADD_BOTH)
      IF
      EMPTY(m.cPapaKey)

      APPEND BLANK IN
      (m.cPapaAlias)

      ELSE

      INSERT INTO
      (m.cPapaAlias)
      ((oAddRec.KeyField))
      VALUES(oAddRec.KeyValue)
      ENDIF
      ENDIF
      nSaveRec2 =
      RECNO()

      * Add child
      record
      IF
      INLIST(oAddRec.AddOption,OPT_A
      DD_CHILD,OPT_ADD_BOTH)
      * Need
      to check
      GO
      m.nSaveRec

      EMPTY(m.cChildKey) OR
      TYPE("m.cChildKey")#TYPE('oAddR
      ec.KeyValue')

      APPEND BLANK IN
      (m.cChildAlias)

      ELSE

      INSERT INTO
      (m.cChildAlias) ((m.cChildKey))
      VALUES(oAddRec.KeyValue)
      ENDIF
      GO
      m.nSaveRec2

      ENDIF
      ENDCASE

      this.Parent.EditMode =
      !THIS.Parent.EditMode
      this.Parent.AddMode =
      THIS.Parent.EditMode
      this.Parent.topfile = .F.
      thisform.LockScreen = .T.
      this.Parent.buttonrefresh()
      this.Parent.navrefresh()
      thisform.LockScreen = .F.
      ENDPROC
      Top = 0
      Left = 432
      Height = 24
      Width = 53
      FontName = "MS Sans Serif"
      FontSize = 8
      Caption = "\<Delete"
      Enabled = .T.

```

```

      StatusBarText = "Delete existing
      record"
      TabIndex = 9
      ToolTipText = ""
      Name = "cmdDelete"
      PROCEDURE Error
      LPARAMETERS nError, cMethod,
      nLine
      this.parent.error(m.nError,
      m.cMethod, m.nLine)
      ENDPROC
      PROCEDURE Click
      #DEFINE MSGBOX_YES
      6
      #DEFINE C_MSGBOX1
      36
      #DEFINE C_DELETE_LOC "Do
      you want to delete this record?"
      #DEFINE C_NOLOCK_LOC
      "Record could not be
      deleted because it is being used by
      someone else."

      * Note: Cascading deletes should be
      handled via RI triggers in DBC!

      IF
      MESSAGEBOX(C_DELETE_LOC,C
      _MSGBOX1) = MSGBOX_YES
      Delete
      IF
      THIS.Parent.updaterows()
      &&success

      * Success
      IF !EOF()
      Skip 1
      ENDIF
      IF EOF() AND
      Skip -1
      ENDIF
      ENDIF
      thisform.LockScreen = .T.
      this.Parent.buttonrefresh()
      this.Parent.navrefresh()
      thisform.LockScreen = .F.
      ENDIF

      IF THISFORM.ShowWindow = 2
      Activate Window
      (THISFORM.Name)
      ENDIF

      ENDPROC
      Top = 0
      Left = 378
      Height = 24
      Width = 53
      FontName = "MS Sans Serif"
      FontSize = 8
      Caption = "\<Edit"
      Enabled = .T.
      StatusBarText = "Edit existing record
      / Revert edits"
      TabIndex = 8
      ToolTipText = ""
      Name = "cmdEdit"
      PROCEDURE Error
      LPARAMETERS nError, cMethod,
      nLine

```

```

this.parent.error(m.nError,
m.cMethod, m.nLine)
ENDPROC
PROCEDURE Click
#DEFINE C_NOUPDATE_LOC
"You cannot edit because
the view(s) selected does not send
updates."

LOCAL
INoSendParentUpdates, INoSendChildUpdates
LOCAL aTablesUsed, nTablesUsed, i

** Reverting record
IF THIS.Parent.EditMode

    IF THIS.Parent.usedataenv
        SELECT
        (THIS.Parent.OldAlias)
        IF
        CURSORGETPROP("Buffering")>1

            =TableRevert(.T.)
            ENDIF
            IF
            !EMPTY(THIS.Parent.GridAlias)

                SELECT
                (THIS.Parent.GridAlias)
                IF
                CURSORGETPROP("Buffering")>1

                    =TableRevert(.T.)
                    ENDIF
                    ENDIF
                    ELSE
                    DIMENSION
                    aTablesUsed[1]
                    m.nTablesUsed
                    = AUUSED(aTablesUsed)
                    FOR i = 1 TO
                    m.nTablesUsed
                    IF
                    CURSORGETPROP("Buffering", aTablesUsed[m.i, 1])>1

                        =TableRevert(.T., aTablesUsed[m.i, 1])
                        ENDIF
                        ENDIF
                        ENDIF
                        ENDIF
                        * Go back to original place
                        SELECT
                        (THIS.Parent.OldAlias)
                        IF RECCOUNT() <
                        THIS.Parent.oldrec &&added record
                        at EOF()

                            GO TOP
                            ELSE
                            GO
                            THIS.Parent.oldrec
                            ENDIF
                            ELSE
                            * Check to see if view
                            allows updates
                            IF
                            CURSORGETPROP("SourceType")#

```

```

3 AND
!CURSORGETPROP("offline") AND;

!CURSORGETPROP("SendUpdates")

    INoSendParentUpdates =
.T.
    ENDIF
    IF
    !EMPTY(THIS.Parent.GridAlias)
    AND ;

    CURSORGETPROP("SourceType", THIS.Parent.GridAlias)#3 AND ;

!CURSORGETPROP("offline", THIS.Parent.GridAlias) AND ;

!CURSORGETPROP("SendUpdates", THIS.Parent.GridAlias)

    INoSendChildUpdates= .T.
    ENDIF

    IF
    (m.INoSendChildUpdates AND
    m.INoSendParentUpdates) OR;

    (EMPTY(THIS.Parent.GridAlias) AND
    m.INoSendParentUpdates)

        =MESSAGEBOX(C_NOUPDATE_LOC)
        RETURN
        ENDIF
        ENDIF
        this.Parent.OldAlias =
        ALIAS() &&save alias in case
        reverting
        this.Parent.oldrec =
        RECNO() &&save record in case
        reverting
        ENDIF

** Editing record
this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.
ENDPROC
MS Sans Serif, 0, 8, 5, 13, 11, 11, 2,
0
*****
C:\COBRATEL\AGENTES.FRX
*****
C:\COBRATEL\ARENA.BMP
*****
C:\COBRATEL\ARW09LT.ICO
*****
C:\COBRATEL\BCKUP.ICO
*****
C:\COBRATEL\CHECKMRK.ICO
*****
C:\COBRATEL\CLIP07.ICO
*****
C:\COBRATEL\COBROS.FRX

```

```

*: File List

*: COBRATEL® Sistema de
cobranza por terceros

FILE
COBRATEL.PRG
..ARCHIVOS DE
PROGRAMA\DEVSTUDIO\FWPWIZ
ARDSWIZSTYLE.VCX
AGENTES.FRX
ARENA.BMP
ARW09LT.ICO
BCKUP.ICO
CHECKMRK.ICO
CLIP07.ICO
COBROS.FRX
CPT0100.SCX
CPT0111.SCX
CPT0112.SCX
CPT0121.SCX
CPT0122.SCX
CPT0200.SCX
CPT0200B.SCX
CPT0210.SCX
CPT0220.SCX
CPT0230.SCX
CPT0241.SCX
CPT0242.SCX
CPT0300.SCX
CPT0300B.SCX
CPT0311.SCX
CPT0312.SCX
CPT0313.SCX
CPT0320.SCX
CPT0330.SCX
CPT0340.SCX
CPT0400.SCX
CPT0410.SCX
CPT0420.SCX
CPT0430.SCX
CPT0440.SCX
CPT0500.SCX
CPT0500B.SCX
CPT0510.SCX
CPT0520.SCX
CPT0530.SCX
CPT0540.SCX
CPT0550.SCX
CPT0560.SCX
CPT0570.SCX
CPT0580.SCX
CPT0600.SCX
CPT0610.SCX
CPT0620.SCX
CRDFLE11.ICO
DBTELMEX.DBC
DEPU1.ICO
DEPU2.ICO
DEPU3.ICO
DEPU4.ICO
DEPURA2.ICO
DIACAJA.FRX
ERASE02.ICO
ESTACION.FRX
ESTADISTICO.FRX
EYE.ICO
FILES09.ICO
FINALIZA.SCX
FOLDER04.ICO

```

FONDO.BMP
 GRAPH03.ICO
 GRAPH11.ICO
 GRUPOS.FRX
 KEY04.ICO
 KEYBRD02.ICO
 KEYS03.ICO
 MAIL03.ICO
 MAIL14.ICO
 MENUPRIN.SCX
 MISC25.ICO
 MISC27.ICO
 MISC29.ICO
 MONITR01.ICO
 NET01.ICO
 NET06.ICO
 NOTE02.ICO
 NOTE03.ICO
 NOTE14.ICO
 PEN07.ICO
 PENCIL05.ICO
 PHONE11.ICO
 PRINCIPAL.SCX
 PROCESOS.FRX
 RESTO.ICO
 TRXS.FRX
 USUARIOS.FRX
 VALIDAR.SCX
 VB416MOM.BMP
 VIACOBRO.FRX
 W95MBX01.ICO
 WRENCH.ICO
 WZBACK.BMP
 WZCLOSE.BMP
 WZCLOSE.MSK
 WZDELETE.BMP
 WZDELETE.MSK
 WZEDIT.BMP
 WZEDIT.MSK
 WZEND.BMP
 WZLOCATE.BMP
 WZLOCATE.MSK
 WZNEW.BMP
 WZNEW.MSK
 WZNEXT.BMP
 WZPRINT.BMP
 WZPRINT.MSK
 WZTOP.BMP

Database Database
 Version 10
 Database TransactionLog
 Database StoredProceduresSource
 Database StoredProceduresObject
 Database
 StoredProceduresDependencies
 Table agentes
 Path
 agentes.dbf
 Field vi_co_ag
 VI_CO_AG I 4 0
 Field nom_agen
 NOM_AGEN C 20 0
 Field vi_lo_ag
 VI_LO_AG C 2 0
 Field resp_agen
 RESP_AGEN C 20 0
 Field te_re_ag
 TE_RE_AG C 10 0

Field baja_ag
 BAJA_AG C 1 0
 Index vi_co_ag
 Unique 0
 Table arch_cobro
 Path
 arch_cobro.dbf
 Field archivos
 ARCHIVOS C 35 0
 Table b_transmision
 Path
 b_transmision.dbf
 Field es_co_bt
 ES_CO_BT I 4 0
 Field via_co_bt
 VIA_CO_BT I 4 0
 Field fech_bt
 FECH_BT D 8 0
 Field hora_bt
 HORA_BT C 8 0
 Field estado_bt
 ESTADO_BT C 10 0
 Table cobros
 Path
 cobros.dbf
 Field telef_co
 TELEF_CO C 10 0
 Field fech_cob
 FECH_COB D 8 0
 Field via_cobr
 VIA_COBR I 4 0
 Field est_cobr
 EST_COBR I 4 0
 Field imp_cob
 IMP_COB N 11 2
 Field fe_re_co
 FE_RE_CO D 8 0
 Index fech_cob
 Unique 0
 Table est_cobro
 Path
 est_cobro.dbf
 Field num_ec
 NUM_EC I 4 0
 Field via_ec
 VIA_EC I 4 0
 Field nomb_ec
 NOMB_EC C 20 0
 Field sigla_ec
 SIGLA_EC C 3 0
 Field direc_ec
 DIREC_EC C 20 0
 Field resp_ec
 RESP_EC C 20 0
 Field tel_mod_ec
 TEL_MOD_EC C 10 0
 Field tel_re_ec
 TEL_RE_EC C 10 0
 Table grupos
 Path
 grupos.dbf
 Index cve_gru
 Unique 0
 Field cve_gru
 CVE_GRU C 3 0
 Field nom_gru
 NOM_GRU C 30 0
 Field m01
 L 1 0
 M01
 Caption

Field m02
 L 1 0
 M02
 Field m03
 L 1 0
 M03
 Field m04
 L 1 0
 M04
 Field m05
 L 1 0
 M05
 Field m06
 L 1 0
 M06
 Table h_cobros
 Path
 h_cobros.dbf
 Field telef_hc
 TELEF_HC C 10 0
 Field fech_hc
 FECH_HC D 8 0
 Field via_co_hc
 VIA_CO_HC I 4 0
 Field es_co_hc
 ES_CO_HC I 4 0
 Field imp_hc
 IMP_HC N 9 2
 Field fe_rec_hc
 FE_REC_HC D 8 0
 Table pagos
 Path
 pagos.dbf
 Field via_co_pa
 VIA_CO_PA I 4 0
 Field fe_in_pa
 FE_IN_PA D 8 0
 Field fe_fi_pa
 FE_FI_PA D 8 0
 Field imp_cob_pa
 IMP_COB_PA N 11 2
 Field fech_pag
 FECH_PAG D 8 0
 Field liqui_pa
 LIQUI_PA C 2 0
 Index wizard_1
 Unique 0
 Table parametros
 Path
 parametros.dbf
 Field cve_para
 CVE_PARA C 3 0
 Field val_par1
 VAL_PAR1 D 8 0
 Field val_par2
 VAL_PAR2 C 20 0
 Field val_par3
 VAL_PAR3 C 25 0
 Field val_par4
 VAL_PAR4 C 30 0
 Field val_par5
 VAL_PAR5 C 30 0
 Field val_par6
 VAL_PAR6 C 30 0
 Field conc_par
 CONC_PAR C 50 0
 Index cve_para
 Unique 0
 Table procesos
 Path
 procesos.dbf
 Field tarea_bp
 TAREA_BP C 30 0
 Field usua_bp
 USUA_BP C 7 0

```

Field fecha_bp
FECHA_BP D 8 0
Field hora_bp
HORA_BP C 8 0
Table todo
Path
todo.dbf
Field registro
REGISTRO C 120 0
Table usuarios
Path
usuarios.dbf
Field cve_usua
CVE_USUA C 7 0
Field pass_usu
PASS_USU C 5 0
Field gpo_usu
GPO_USU C 3 0
Field nom_usu
NOM_USU C 20 0
Field dir_usu
DIR_USU C 20 0
Field tel_usu
TEL_USU C 10 0
Field vi_co_us
VI_CO_US I 4 0
Index cve_usua
Unique 0
Table meses
Path
meses.dbf
Field mes MES
C 15 0
Field nummes
NUMMES I 4 0

```

```

* AAAAForm1
CPT0311.SCX
* AAAAForm1
CPT0313.SCX
* AAAAForm1
CPT0410.SCX
* AAAAForm1
CPT0420.SCX
* AAAAForm1
CPT0430.SCX
* AAAAForm1
CPT0440.SCX
* AAAAForm1
CPT0510.SCX
AAAAsearchform
WIZSTYLE.VCX
AAAAForm1
CPT0100.SCX
AAAAForm1
CPT0111.SCX
AAAAForm1
CPT0112.SCX
AAAAForm1
CPT0121.SCX
AAAAForm1
CPT0122.SCX
AAAAForm1
CPT0200.SCX
AAAAForm1
CPT0200B.SCX
AAAAForm1
CPT0210.SCX
AAAAForm1
CPT0220.SCX
AAAAForm1
CPT0230.SCX
AAAAForm1
CPT0300.SCX
AAAAForm1
CPT0300B.SCX
AAAAForm1
CPT0312.SCX
AAAAForm1
CPT0320.SCX
AAAAForm1
CPT0330.SCX
AAAAForm1
CPT0340.SCX
AAAAForm1
CPT0400.SCX
AAAAForm1
CPT0500.SCX
AAAAForm1
CPT0500B.SCX
AAAAForm1
CPT0520.SCX
AAAAForm1
CPT0530.SCX
AAAAForm1
CPT0540.SCX
AAAAForm1
CPT0550.SCX
AAAAForm1
CPT0560.SCX
AAAAForm1
CPT0570.SCX
AAAAForm1
CPT0580.SCX
AAAAForm1
CPT0600.SCX

```

```

AAAAForm1
CPT0610.SCX
AAAAForm1
CPT0620.SCX
AAAAForm1
FINALIZA.SCX
AAAAForm1
MENUPRIN.SCX
AAAAPRINCIPAL
PRINCIPAL.SCX
AAAAvalida
VALIDAR.SCX

container
AAAAlayoutsty
WIZSTYLE.VCX
AAAAchiselbase
WIZSTYLE.VCX
* AAAAchiselfield
WIZSTYLE.VCX
* AAAAchiselmemo
WIZSTYLE.VCX
* AAAAchiselole
WIZSTYLE.VCX
* AAAAchisellogic
WIZSTYLE.VCX
AAAAaboxbase
WIZSTYLE.VCX
* AAAAaboxfield
WIZSTYLE.VCX
* AAAAaboxmemo
WIZSTYLE.VCX
* AAAAaboxole
WIZSTYLE.VCX
* AAAAaboxlogic
WIZSTYLE.VCX
AAAAagridbtns
WIZSTYLE.VCX
AAAAashadowbase
WIZSTYLE.VCX
* AAAAashadowole
WIZSTYLE.VCX
* AAAAashadowmemo
WIZSTYLE.VCX
* AAAAashadowfield
WIZSTYLE.VCX
* AAAAashadowlogic
WIZSTYLE.VCX
AAAAfancybase
WIZSTYLE.VCX
* AAAAfancyole
WIZSTYLE.VCX
* AAAAfancyfield
WIZSTYLE.VCX
* AAAAfancylogic
WIZSTYLE.VCX
* AAAAfancymemo
WIZSTYLE.VCX
AAAAContainer1
WIZSTYLE.VCX
AAAAsearchclass
WIZSTYLE.VCX
* AAAASearchclass1
WIZSTYLE.VCX
AAAATxtBtns
WIZSTYLE.VCX
AAAAvertxtbtns
WIZSTYLE.VCX
AAAApicbtns
WIZSTYLE.VCX

```

*: Tree Diagram

*: COBRATEL® Sistema de cobranza por terceros

COBRATEL

.....

```

Class Hierarchy
form
AAAAagridaddform
WIZSTYLE.VCX
AAAAbaseform
WIZSTYLE.VCX
* AAAAnobtns
WIZSTYLE.VCX
* AAAAstandardform
WIZSTYLE.VCX
* AAAAaboxform
WIZSTYLE.VCX
* AAAAchiselform
WIZSTYLE.VCX
* AAAAshadowform
WIZSTYLE.VCX
* AAAAembossedform
WIZSTYLE.VCX
* AAAAfancyform
WIZSTYLE.VCX
* AAAAForm1
CPT0241.SCX
* AAAAForm1
CPT0242.SCX

```

AAAA Navegar
CPT0241.SCX
AAAA Navegar
CPT0242.SCX
AAAA Navegar
CPT0311.SCX
AAAA Navegar
CPT0313.SCX
AAAA Navegar
CPT0410.SCX
AAAA Navegar
CPT0420.SCX
AAAA Navegar
CPT0430.SCX
AAAA Navegar
CPT0440.SCX
AAAA Navegar
CPT0510.SCX

checkbox
AAAA standardlogic
WIZSTYLE.VCX
AAAA Check1
WIZSTYLE.VCX
AAAA Check1
WIZSTYLE.VCX
AAAA Check1
WIZSTYLE.VCX
AAAA embossedlogic
WIZSTYLE.VCX
³ AAAAm011
CPT0440.SCX
³ AAAAm021
CPT0440.SCX
³ AAAAm031
CPT0440.SCX
³ AAAAm041
CPT0440.SCX
³ AAAAm051
CPT0440.SCX
³ AAAAm061
CPT0440.SCX
AAAA Check1
WIZSTYLE.VCX
AAAA chkCaseSensitive
WIZSTYLE.VCX

combobox
AAAA cboFields1
WIZSTYLE.VCX
AAAA cboOperators1
WIZSTYLE.VCX
AAAA cboFields2
WIZSTYLE.VCX
AAAA cboOperators2
WIZSTYLE.VCX
AAAA Combo2
CPT0111.SCX
AAAA Combo1
CPT0111.SCX
AAAA Combo2
CPT0241.SCX
AAAA Combo1
CPT0241.SCX
AAAA Combo1
CPT0311.SCX
AAAA busca
CPT0311.SCX
AAAA Combo1
CPT0312.SCX

AAAA mes
CPT0330.SCX
AAAA Combo1
CPT0340.SCX
AAAA busca
CPT0410.SCX
AAAA busca
CPT0420.SCX
AAAA busca
CPT0430.SCX
AAAA Agpo_usu1
CPT0430.SCX
AAAA busca
CPT0440.SCX
AAAA busca
CPT0510.SCX
AAAA Combo1
CPT0580.SCX

commandbutton
AAAA cmdFind
WIZSTYLE.VCX
AAAA cmdPrint
WIZSTYLE.VCX
AAAA cmdExit
WIZSTYLE.VCX
AAAA cmdAdd
WIZSTYLE.VCX
AAAA cmdDelete
WIZSTYLE.VCX
AAAA cmdPrev
WIZSTYLE.VCX
AAAA cmdNext
WIZSTYLE.VCX
AAAA cmdTop
WIZSTYLE.VCX
AAAA cmdEnd
WIZSTYLE.VCX
AAAA cmdFind
WIZSTYLE.VCX
AAAA cmdPrint
WIZSTYLE.VCX
AAAA cmdExit
WIZSTYLE.VCX
AAAA cmdAdd
WIZSTYLE.VCX
AAAA cmdDelete
WIZSTYLE.VCX
AAAA cmdEdit
WIZSTYLE.VCX
AAAA Command10
CPT0100.SCX
AAAA Command1
CPT0100.SCX
AAAA Command2
CPT0100.SCX
AAAA Command3
CPT0100.SCX
AAAA Command4
CPT0100.SCX
AAAA Command10
CPT0111.SCX
AAAA Command1
CPT0111.SCX
AAAA Command10
CPT0112.SCX
AAAA Command1
CPT0112.SCX
AAAA Command10
CPT0121.SCX

AAAA Command1
CPT0121.SCX
AAAA Command10
CPT0122.SCX
AAAA Command1
CPT0122.SCX
AAAA Command10
CPT0200.SCX
AAAA Command1
CPT0200.SCX
AAAA Command2
CPT0200.SCX
AAAA Command3
CPT0200.SCX
AAAA Command4
CPT0200.SCX
AAAA Command10
CPT0200B.SCX
AAAA Command1
CPT0200B.SCX
AAAA Command2
CPT0200B.SCX
AAAA Command10
CPT0210.SCX
AAAA Command1
CPT0210.SCX
AAAA Command10
CPT0220.SCX
AAAA Command1
CPT0220.SCX
AAAA Command10
CPT0230.SCX
AAAA Command1
CPT0230.SCX
AAAA validar
CPT0241.SCX
AAAA Command10
CPT0300.SCX
AAAA Command1
CPT0300.SCX
AAAA Command4
CPT0300.SCX
AAAA Command5
CPT0300.SCX
AAAA Command6
CPT0300.SCX
AAAA Command10
CPT0300B.SCX
AAAA Command1
CPT0300B.SCX
AAAA Command2
CPT0300B.SCX
AAAA Command3
CPT0300B.SCX
AAAA Command10
CPT0312.SCX
AAAA Command1
CPT0312.SCX
AAAA Command1
CPT0313.SCX
AAAA Command10
CPT0320.SCX
AAAA Command1
CPT0320.SCX
AAAA Command10
CPT0330.SCX
AAAA Command1
CPT0330.SCX
AAAA Command10
CPT0340.SCX

AAAACommand1
CPT0340.SCX
AAAACommand10
CPT0400.SCX
AAAACommand1
CPT0400.SCX
AAAACommand2
CPT0400.SCX
AAAACommand3
CPT0400.SCX
AAAAcommand4
CPT0400.SCX
AAAACommand10
CPT0500.SCX
AAAACommand1
CPT0500.SCX
AAAACommand2
CPT0500.SCX
AAAACommand6
CPT0500.SCX
AAAACommand7
CPT0500.SCX
AAAACommand8
CPT0500.SCX
AAAACommand10
CPT0500B.SCX
AAAACommand2
CPT0500B.SCX
AAAACommand3
CPT0500B.SCX
AAAAcommand4
CPT0500B.SCX
AAAACommand5
CPT0500B.SCX
AAAACommand10
CPT0520.SCX
AAAACommand1
CPT0520.SCX
AAAACommand10
CPT0530.SCX
AAAACommand1
CPT0530.SCX
AAAACommand10
CPT0540.SCX
AAAACommand1
CPT0540.SCX
AAAACommand10
CPT0550.SCX
AAAACommand1
CPT0550.SCX
AAAACommand10
CPT0560.SCX
AAAACommand1
CPT0560.SCX
AAAACommand10
CPT0570.SCX
AAAACommand1
CPT0570.SCX
AAAACommand10
CPT0580.SCX
AAAACommand1
CPT0580.SCX
AAAACommand10
CPT0600.SCX
AAAACommand1
CPT0600.SCX
AAAACommand2
CPT0600.SCX
AAAACommand10
CPT0610.SCX

AAAACommand1
CPT0610.SCX
AAAACommand10
CPT0620.SCX
AAAACommand1
CPT0620.SCX
AAAACommand1
FINALIZA.SCX
AAAACommand10
MENUPRIN.SCX
AAAACommand1
MENUPRIN.SCX
AAAACommand2
MENUPRIN.SCX
AAAACommand3
MENUPRIN.SCX
AAAAcommand4
MENUPRIN.SCX
AAAACommand5
MENUPRIN.SCX
AAAACommand6
MENUPRIN.SCX
AAAACommand1
VALIDAR.SCX
AAAACommand2
VALIDAR.SCX

commandgroup
AAAACommandGroup1
WIZSTYLE.VCX
AAAACommandGroup1
WIZSTYLE.VCX

cursor
AAAACursor1
CPT0111.SCX
AAAACursor2
CPT0111.SCX
AAAACursor3
CPT0111.SCX
AAAACursor4
CPT0111.SCX
AAAACursor3
CPT0112.SCX
AAAACursor3
CPT0210.SCX
AAAACursor1
CPT0210.SCX
AAAACursor3
CPT0220.SCX
AAAACursor1
CPT0220.SCX
AAAACursor2
CPT0220.SCX
AAAACursor3
CPT0230.SCX
AAAACursor1
CPT0230.SCX
AAAACursor2
CPT0230.SCX
AAAACursor1
CPT0241.SCX
AAAACursor2
CPT0241.SCX
AAAACursor3
CPT0241.SCX
AAAACursor4
CPT0241.SCX
AAAACURSOR5
CPT0241.SCX

AAAACursor1
CPT0242.SCX
AAAACursor2
CPT0242.SCX
AAAACursor1
CPT0311.SCX
AAAACursor2
CPT0311.SCX
AAAACursor3
CPT0311.SCX
AAAACursor2
CPT0312.SCX
AAAACursor1
CPT0312.SCX
AAAACursor3
CPT0312.SCX
AAAACursor1
CPT0313.SCX
AAAACursor2
CPT0313.SCX
AAAACursor2
CPT0320.SCX
AAAACursor1
CPT0320.SCX
AAAACursor2
CPT0330.SCX
AAAACursor1
CPT0330.SCX
AAAACursor3
CPT0330.SCX
AAAACursor2
CPT0340.SCX
AAAACursor1
CPT0340.SCX
AAAACursor3
CPT0340.SCX
AAAACursor1
CPT0410.SCX
AAAACursor2
CPT0410.SCX
AAAACursor1
CPT0420.SCX
AAAACursor2
CPT0420.SCX
AAAACursor1
CPT0430.SCX
AAAACursor2
CPT0430.SCX
AAAACursor3
CPT0430.SCX
AAAACursor1
CPT0440.SCX
AAAACursor2
CPT0440.SCX
AAAACursor1
CPT0510.SCX
AAAACursor2
CPT0510.SCX
AAAACursor1
CPT0520.SCX
AAAACursor2
CPT0530.SCX
AAAACursor1
CPT0540.SCX
AAAACursor2
CPT0550.SCX
AAAACursor1
CPT0550.SCX
AAAACursor1
CPT0580.SCX

AAAACursor1
MENUPRIN.SCX
AAAACursor1
VALIDAR.SCX

custom
AAAAGridaddrec
WIZSTYLE.VCX

editbox
AAAEdit1
WIZSTYLE.VCX
AAAEdit1
WIZSTYLE.VCX
AAAEdit1
WIZSTYLE.VCX
AAAembossedmemo
WIZSTYLE.VCX
* AAAACONC_PAR1
CPT0510.SCX
AAAstandardmemo
WIZSTYLE.VCX
AAAEdit1
WIZSTYLE.VCX

textbox
AAAAText1
WIZSTYLE.VCX
AAAAText1
WIZSTYLE.VCX
AAAAText1
WIZSTYLE.VCX
AAAAText1
WIZSTYLE.VCX
AAAembossedfield
WIZSTYLE.VCX
* AAAATELEF_CO1
CPT0241.SCX
* AAAAFECH_COB1
CPT0241.SCX
* AAAAIMP_COB1
CPT0241.SCX
* AAAAfe_re_co1
CPT0241.SCX
* AAAAVIA_CO_PA1
CPT0242.SCX
* AAAAFE_IN_PA1
CPT0242.SCX
* AAAAFE_FI_PA1
CPT0242.SCX
* AAAAIMP_COB_PA1
CPT0242.SCX
* AAAAFECH_PAG1
CPT0242.SCX
* AAAALIQUI_PA1
CPT0242.SCX
* AAAAFE_IN_PA1
CPT0311.SCX
* AAAAFE_FI_PA1
CPT0311.SCX
* AAAAIMP_COB_PA1
CPT0311.SCX
* AAAAFECH_PAG1
CPT0311.SCX
* AAAALIQUI_PA1
CPT0311.SCX
* AAAAVI_CO_AG1
CPT0410.SCX
* AAAAnom_agen1
CPT0410.SCX

* AAAAavi_lo_ag1
CPT0410.SCX
* AAAAresp_agen1
CPT0410.SCX
* AAAAte_re_ag1
CPT0410.SCX
* AAAAbaja_ag1
CPT0410.SCX
* AAAANUM_EC1
CPT0420.SCX
* AAAAvia_ec1
CPT0420.SCX
* AAAAnomb_ec1
CPT0420.SCX
* AAAAsigla_ec1
CPT0420.SCX
* AAAAdirec_ec1
CPT0420.SCX
* AAAAresp_ec1
CPT0420.SCX
* AAAAtel_mod_ec1
CPT0420.SCX
* AAAAtel_re_ec1
CPT0420.SCX
* AAAACVE_USUA1
CPT0430.SCX
* AAAApass_usu1
CPT0430.SCX
* AAAAnom_usu1
CPT0430.SCX
* AAAAdir_usu1
CPT0430.SCX
* AAAAtel_usu1
CPT0430.SCX
* AAAAavi_co_us1
CPT0430.SCX
* AAAACVE_GRU1
CPT0440.SCX
* AAAANOM_GRU1
CPT0440.SCX
* AAAACVE_PARA1
CPT0510.SCX
* AAAAVAL_PAR11
CPT0510.SCX
* AAAAVAL_PAR21
CPT0510.SCX
* AAAAVAL_PAR31
CPT0510.SCX
* AAAAVAL_PAR41
CPT0510.SCX
* AAAAVAL_PAR51
CPT0510.SCX
* AAAAVAL_PAR61
CPT0510.SCX
AAAstandardfield
WIZSTYLE.VCX
AAAAText1
WIZSTYLE.VCX
AAAAbtExpr1
WIZSTYLE.VCX
AAAAbtExpr2
WIZSTYLE.VCX
AAAAText1
CPT0111.SCX
AAAAtext2
CPT0111.SCX
AAAAtext3
CPT0111.SCX
AAAAText4
CPT0111.SCX
AAAAText5
CPT0111.SCX

AAAAtext5
CPT0111.SCX
AAAAText6
CPT0111.SCX
AAAAText7
CPT0111.SCX
AAAAText8
CPT0111.SCX
AAAAtext9
CPT0111.SCX
AAAAtext10
CPT0111.SCX
AAAAText1
CPT0112.SCX
AAAAtext2
CPT0112.SCX
AAAAtext3
CPT0112.SCX
AAAAtext4
CPT0112.SCX
AAAAtext5
CPT0112.SCX
AAAAtext9
CPT0112.SCX
AAAAText6
CPT0112.SCX
AAAAText1
CPT0121.SCX
AAAAtext2
CPT0121.SCX
AAAAtext3
CPT0121.SCX
AAAAtext4
CPT0121.SCX
AAAAtext5
CPT0121.SCX
AAAAtext6
CPT0121.SCX
AAAAtext7
CPT0121.SCX
AAAAText1
CPT0122.SCX
AAAAtext2
CPT0122.SCX
AAAAtext3
CPT0122.SCX
AAAAtext4
CPT0122.SCX
AAAAtext5
CPT0122.SCX
AAAAText6
CPT0122.SCX
AAAAText7
CPT0122.SCX
AAAAText1
CPT0210.SCX
AAAAtext2
CPT0210.SCX
AAAAtext3
CPT0210.SCX
AAAAText4
CPT0210.SCX
AAAAtext5
CPT0210.SCX
AAAAtext9
CPT0210.SCX
AAAAText6
CPT0210.SCX
AAAAText1
CPT0220.SCX

AAAAtext2
CPT0220.SCX
AAAAtext3
CPT0220.SCX
AAAAText4
CPT0220.SCX
AAAAtext5
CPT0220.SCX
AAAAtext9
CPT0220.SCX
AAAAText6
CPT0220.SCX
AAAAText1
CPT0230.SCX
AAAAtext2
CPT0230.SCX
AAAAtext3
CPT0230.SCX
AAAAText4
CPT0230.SCX
AAAAtext5
CPT0230.SCX
AAAAText9
CPT0230.SCX
AAAAText6
CPT0230.SCX
AAAAText1
CPT0241.SCX
AAAAtext2
CPT0241.SCX
AAAAtext3
CPT0241.SCX
AAAAText4
CPT0241.SCX
AAAAtext5
CPT0241.SCX
AAAAText1
CPT0242.SCX
AAAAtext2
CPT0242.SCX
AAAAtext3
CPT0242.SCX
AAAAText4
CPT0242.SCX
AAAAtext5
CPT0242.SCX
AAAAText1
CPT0311.SCX
AAAAtext2
CPT0311.SCX
AAAAtext3
CPT0311.SCX
AAAAText4
CPT0311.SCX
AAAAtext5
CPT0311.SCX
AAAAText1
CPT0312.SCX
AAAAtext2
CPT0312.SCX
AAAAtext3
CPT0312.SCX
AAAAText4
CPT0312.SCX
AAAAtext5
CPT0312.SCX
AAAAText6
CPT0312.SCX
AAAAText1
CPT0312.SCX

AAAAText1
CPT0312.SCX
AAAAText1
CPT0312.SCX
AAAAText7
CPT0312.SCX
AAAAText8
CPT0312.SCX
AAAAText1
CPT0313.SCX
AAAAtext2
CPT0313.SCX
AAAAtext3
CPT0313.SCX
AAAAText4
CPT0313.SCX
AAAAtext5
CPT0313.SCX
AAAAText1
CPT0313.SCX
AAAAText1
CPT0313.SCX
AAAAText1
CPT0313.SCX
AAAAText1
CPT0313.SCX
AAAAText1
CPT0313.SCX
AAAAText1
CPT0313.SCX
AAAAText7
CPT0313.SCX
AAAAtext11
CPT0320.SCX
AAAAText1
CPT0320.SCX
AAAAtext2
CPT0320.SCX
AAAAtext3
CPT0320.SCX
AAAAText4
CPT0320.SCX
AAAAtext5
CPT0320.SCX
AAAAText1
CPT0330.SCX
AAAAtext2
CPT0330.SCX
AAAAtext3
CPT0330.SCX
AAAAText4
CPT0330.SCX
AAAAtext5
CPT0330.SCX
AAAAText6
CPT0330.SCX
AAAAText1
CPT0340.SCX
AAAAtext2
CPT0340.SCX
AAAAtext3
CPT0340.SCX
AAAAText4
CPT0340.SCX
AAAAtext5
CPT0340.SCX
AAAAText6
CPT0340.SCX
AAAAText7
CPT0340.SCX

AAAAText8
CPT0340.SCX
AAAAText1
CPT0410.SCX
AAAAtext2
CPT0410.SCX
AAAAtext3
CPT0410.SCX
AAAAText4
CPT0410.SCX
AAAAtext5
CPT0410.SCX
AAAAText1
CPT0420.SCX
AAAAtext2
CPT0420.SCX
AAAAtext3
CPT0420.SCX
AAAAText4
CPT0420.SCX
AAAAtext5
CPT0420.SCX
AAAAText1
CPT0430.SCX
AAAAtext2
CPT0430.SCX
AAAAtext3
CPT0430.SCX
AAAAText4
CPT0430.SCX
AAAAtext5
CPT0430.SCX
AAAAText1
CPT0440.SCX
AAAAtext2
CPT0440.SCX
AAAAtext3
CPT0440.SCX
AAAAText4
CPT0440.SCX
AAAAtext5
CPT0440.SCX
AAAAText1
CPT0510.SCX
AAAAtext2
CPT0510.SCX
AAAAtext3
CPT0510.SCX
AAAAText4
CPT0510.SCX
AAAAtext5
CPT0510.SCX
AAAAtext11
CPT0520.SCX
AAAAtext12
CPT0520.SCX
AAAAText1
CPT0520.SCX
AAAAtext2
CPT0520.SCX
AAAAtext3
CPT0520.SCX
AAAAText4
CPT0520.SCX
AAAAtext5
CPT0520.SCX
AAAAText11
CPT0530.SCX
AAAAtext12
CPT0530.SCX

AAAAText1
CPT0530.SCX
AAAAtext2
CPT0530.SCX
AAAAtext3
CPT0530.SCX
AAAAText4
CPT0530.SCX
AAAAtext5
CPT0530.SCX
AAAAtext11
CPT0540.SCX
AAAAtext12
CPT0540.SCX
AAAAText1
CPT0540.SCX
AAAAtext2
CPT0540.SCX
AAAAtext3
CPT0540.SCX
AAAAText4
CPT0540.SCX
AAAAtext5
CPT0540.SCX
AAAAtext11
CPT0550.SCX
AAAAtext12
CPT0550.SCX
AAAAText1
CPT0550.SCX
AAAAtext2
CPT0550.SCX
AAAAtext3
CPT0550.SCX
AAAAText4
CPT0550.SCX
AAAAtext5
CPT0550.SCX
AAAAText1
CPT0560.SCX
AAAAtext2
CPT0560.SCX
AAAAtext3
CPT0560.SCX
AAAAText4
CPT0560.SCX
AAAAtext5
CPT0560.SCX
AAAAText6
CPT0560.SCX
AAAAText1
CPT0570.SCX
AAAAtext2
CPT0570.SCX
AAAAtext3
CPT0570.SCX
AAAAText4
CPT0570.SCX
AAAAtext5
CPT0570.SCX
AAAAText1
CPT0580.SCX
AAAAtext2
CPT0580.SCX
AAAAtext3
CPT0580.SCX
AAAAText4
CPT0580.SCX
AAAAtext5
CPT0580.SCX

AAAAText6
CPT0580.SCX
AAAAText7
CPT0580.SCX
AAAAText8
CPT0580.SCX
AAAAtext9
CPT0580.SCX
AAAAText1
CPT0610.SCX
AAAAtext2
CPT0610.SCX
AAAAText1
CPT0620.SCX
AAAAtext2
CPT0620.SCX
AAAAText1
VALIDAR.SCX
AAAAtext2
VALIDAR.SCX

label
AAAALabel1
WIZSTYLE.VCX
AAAAembossedlabel
WIZSTYLE.VCX
* AAAAEmbossedlabel1
CPT0111.SCX
* AAAAEmbossedlabel2
CPT0111.SCX
* AAAAEmbossedlabel3
CPT0111.SCX
* AAAAEmbossedlabel4
CPT0111.SCX
* AAAAEmbossedlabel1
CPT0112.SCX
* AAAAEmbossedlabel2
CPT0112.SCX
* AAAAEmbossedlabel3
CPT0112.SCX
* AAAAEmbossedlabel4
CPT0112.SCX
* AAAAEmbossedlabel1
CPT0121.SCX
* AAAAEmbossedlabel2
CPT0121.SCX
* AAAAEmbossedlabel3
CPT0121.SCX
* AAAAEmbossedlabel4
CPT0121.SCX
* AAAAEmbossedlabel1
CPT0122.SCX
* AAAAEmbossedlabel2
CPT0122.SCX
* AAAAEmbossedlabel3
CPT0122.SCX
* AAAAEmbossedlabel4
CPT0122.SCX
* AAAAEmbossedlabel1
CPT0210.SCX
* AAAAEmbossedlabel2
CPT0210.SCX
* AAAAEmbossedlabel3
CPT0210.SCX
* AAAAEmbossedlabel4
CPT0210.SCX
* AAAAEmbossedlabel1
CPT0220.SCX
* AAAAEmbossedlabel2
CPT0220.SCX

* AAAAEmbossedlabel3
CPT0220.SCX
* AAAAEmbossedlabel4
CPT0220.SCX
* AAAAEmbossedlabel1
CPT0230.SCX
* AAAAEmbossedlabel2
CPT0230.SCX
* AAAAEmbossedlabel3
CPT0230.SCX
* AAAAEmbossedlabel4
CPT0230.SCX
* AAAALBLTELEF_CO1
CPT0241.SCX
* AAAALBLFECH_COB1
CPT0241.SCX
* AAAALBLVIA_COBR1
CPT0241.SCX
* AAAALBLEST_COBR1
CPT0241.SCX
* AAAALBLIMP_COB1
CPT0241.SCX
* AAAALBLFE_RE_CO1
CPT0241.SCX
* AAAAEmbossedlabel1
CPT0241.SCX
* AAAAEmbossedlabel2
CPT0241.SCX
* AAAAEmbossedlabel3
CPT0241.SCX
* AAAAEmbossedlabel4
CPT0241.SCX
* AAAALBLVIA_CO_PA1
CPT0242.SCX
* AAAALBLFE_IN_PA1
CPT0242.SCX
* AAAALBLFE_FI_PA1
CPT0242.SCX
* AAAALBLIMP_COB_PA1
CPT0242.SCX
* AAAALBLFECH_PAG1
CPT0242.SCX
* AAAALBLLIQUI_PA1
CPT0242.SCX
* AAAAEmbossedlabel1
CPT0242.SCX
* AAAAEmbossedlabel2
CPT0242.SCX
* AAAAEmbossedlabel3
CPT0242.SCX
* AAAAEmbossedlabel4
CPT0242.SCX
* AAAALBLVIA_CO_PA1
CPT0311.SCX
* AAAALBLFE_IN_PA1
CPT0311.SCX
* AAAALBLFE_FI_PA1
CPT0311.SCX
* AAAALBLIMP_COB_PA1
CPT0311.SCX
* AAAALBLFECH_PAG1
CPT0311.SCX
* AAAALBLLIQUI_PA1
CPT0311.SCX
* AAAAEmbossedlabel1
CPT0311.SCX
* AAAAEmbossedlabel2
CPT0311.SCX
* AAAAEmbossedlabel3
CPT0311.SCX

* AAAAEmbossedlabel4
CPT0311.SCX
* AAAAEmbossedlabel1
CPT0312.SCX
* AAAAEmbossedlabel2
CPT0312.SCX
* AAAAEmbossedlabel3
CPT0312.SCX
* AAAAEmbossedlabel4
CPT0312.SCX
* AAAAEmbossedlabel1
CPT0313.SCX
* AAAAEmbossedlabel2
CPT0313.SCX
* AAAAEmbossedlabel3
CPT0313.SCX
* AAAAEmbossedlabel4
CPT0313.SCX
* AAAAEmbossedlabel5
CPT0313.SCX
* AAAAEmbossedlabel1
CPT0320.SCX
* AAAAEmbossedlabel2
CPT0320.SCX
* AAAAEmbossedlabel3
CPT0320.SCX
* AAAAEmbossedlabel4
CPT0320.SCX
* AAAAEmbossedlabel1
CPT0330.SCX
* AAAAEmbossedlabel2
CPT0330.SCX
* AAAAEmbossedlabel3
CPT0330.SCX
* AAAAEmbossedlabel4
CPT0330.SCX
* AAAAEmbossedlabel1
CPT0330.SCX
* AAAAEmbossedlabel2
CPT0330.SCX
* AAAAEmbossedlabel3
CPT0330.SCX
* AAAAEmbossedlabel4
CPT0330.SCX
* AAAAEmbossedlabel1
CPT0340.SCX
* AAAAEmbossedlabel2
CPT0340.SCX
* AAAAEmbossedlabel3
CPT0340.SCX
* AAAAEmbossedlabel4
CPT0340.SCX
* AAAAEmbossedlabel1
CPT0340.SCX
* AAAAEmbossedlabel2
CPT0340.SCX
* AAAAEmbossedlabel3
CPT0340.SCX
* AAAAEmbossedlabel4
CPT0340.SCX
* AAAALBLVI_CO_AG1
CPT0410.SCX
* AAAALBLNOM_AGEN1
CPT0410.SCX
* AAAALBLVI_LO_AG1
CPT0410.SCX
* AAAALBLRESP_AGEN1
CPT0410.SCX
* AAAALBLTE_RE_AG1
CPT0410.SCX
* AAAALBLBAJA_AG1
CPT0410.SCX
* AAAAEmbossedlabel1
CPT0410.SCX
* AAAAEmbossedlabel2
CPT0410.SCX
* AAAAEmbossedlabel3
CPT0410.SCX
* AAAAEmbossedlabel4
CPT0410.SCX
* AAAALBLNUM_EC1
CPT0420.SCX
* AAAALBLVIA_EC1
CPT0420.SCX
* AAAALBLNOMB_EC1
CPT0420.SCX

* AAAALBLSIGLA_EC1
CPT0420.SCX
* AAAALBLDIREC_EC1
CPT0420.SCX
* AAAALBLRESP_EC1
CPT0420.SCX
* AAAALBLTEL_MOD_EC1
CPT0420.SCX
* AAAALBLTEL_RE_EC1
CPT0420.SCX
* AAAAEmbossedlabel1
CPT0420.SCX
* AAAAEmbossedlabel2
CPT0420.SCX
* AAAAEmbossedlabel3
CPT0420.SCX
* AAAAEmbossedlabel4
CPT0420.SCX
* AAAALBLCVE_USUA1
CPT0430.SCX
* AAAALBLPASS_USU1
CPT0430.SCX
* AAAALBLGPO_USU1
CPT0430.SCX
* AAAALBLNOM_USU1
CPT0430.SCX
* AAAALBLDIR_USU1
CPT0430.SCX
* AAAALBLTEL_USU1
CPT0430.SCX
* AAAALBLVI_CO_US1
CPT0430.SCX
* AAAAEmbossedlabel1
CPT0430.SCX
* AAAAEmbossedlabel2
CPT0430.SCX
* AAAAEmbossedlabel3
CPT0430.SCX
* AAAAEmbossedlabel4
CPT0430.SCX
* AAAAEmbossedlabel1
CPT0430.SCX
* AAAAEmbossedlabel2
CPT0430.SCX
* AAAAEmbossedlabel3
CPT0430.SCX
* AAAAEmbossedlabel4
CPT0430.SCX
* AAAALBLCVE_GRU1
CPT0440.SCX
* AAAALBLM011
CPT0440.SCX
* AAAALBLM021
CPT0440.SCX
* AAAALBLM031
CPT0440.SCX
* AAAALBLM041
CPT0440.SCX
* AAAALBLM051
CPT0440.SCX
* AAAALBLM061
CPT0440.SCX
* AAAAEmbossedlabel1
CPT0440.SCX
* AAAAEmbossedlabel2
CPT0440.SCX
* AAAAEmbossedlabel3
CPT0440.SCX
* AAAAEmbossedlabel4
CPT0440.SCX
* AAAAEmbossedlabel5
CPT0440.SCX
* AAAALBLCVE_PARA1
CPT0510.SCX
* AAAALBLVAL_PAR11
CPT0510.SCX
* AAAALBLVAL_PAR21
CPT0510.SCX

* AAAALBLVAL_PAR31
CPT0510.SCX
* AAAALBLVAL_PAR41
CPT0510.SCX
* AAAALBLVAL_PAR51
CPT0510.SCX
* AAAALBLVAL_PAR61
CPT0510.SCX
* AAAALBLCONC_PAR1
CPT0510.SCX
* AAAAEmbossedlabel1
CPT0510.SCX
* AAAAEmbossedlabel2
CPT0510.SCX
* AAAAEmbossedlabel3
CPT0510.SCX
* AAAAEmbossedlabel4
CPT0510.SCX
* AAAAEmbossedlabel1
CPT0520.SCX
* AAAAEmbossedlabel2
CPT0520.SCX
* AAAAEmbossedlabel3
CPT0520.SCX
* AAAAEmbossedlabel4
CPT0520.SCX
* AAAAEmbossedlabel1
CPT0530.SCX
* AAAAEmbossedlabel2
CPT0530.SCX
* AAAAEmbossedlabel3
CPT0530.SCX
* AAAAEmbossedlabel4
CPT0530.SCX
* AAAAEmbossedlabel1
CPT0540.SCX
* AAAAEmbossedlabel2
CPT0540.SCX
* AAAAEmbossedlabel3
CPT0540.SCX
* AAAAEmbossedlabel4
CPT0540.SCX
* AAAAEmbossedlabel1
CPT0550.SCX
* AAAAEmbossedlabel2
CPT0550.SCX
* AAAAEmbossedlabel3
CPT0550.SCX
* AAAAEmbossedlabel4
CPT0550.SCX
* AAAAEmbossedlabel1
CPT0560.SCX
* AAAAEmbossedlabel2
CPT0560.SCX
* AAAAEmbossedlabel3
CPT0560.SCX
* AAAAEmbossedlabel4
CPT0560.SCX
* AAAAEmbossedlabel1
CPT0570.SCX
* AAAAEmbossedlabel2
CPT0570.SCX
* AAAAEmbossedlabel3
CPT0570.SCX
* AAAAEmbossedlabel4
CPT0570.SCX
* AAAAEmbossedlabel1
CPT0580.SCX
* AAAAEmbossedlabel2
CPT0580.SCX

AAAALabel8
CPT0500.SCX
AAAALabel10
CPT0500B.SCX
AAAALabel2
CPT0500B.SCX
AAAALabel3
CPT0500B.SCX
AAAALabel4
CPT0500B.SCX
AAAALabel5
CPT0500B.SCX
AAAALabel2
CPT0510.SCX
AAAALabel10
CPT0520.SCX
AAAALabel1
CPT0520.SCX
AAAALabel2
CPT0520.SCX
AAAALabel3
CPT0520.SCX
AAAALabel10
CPT0530.SCX
AAAALabel1
CPT0530.SCX
AAAALabel2
CPT0530.SCX
AAAALabel3
CPT0530.SCX
AAAALabel10
CPT0540.SCX
AAAALabel1
CPT0540.SCX
AAAALabel2
CPT0540.SCX
AAAALabel3
CPT0540.SCX
AAAALabel10
CPT0550.SCX
AAAALabel1
CPT0550.SCX
AAAALabel2
CPT0550.SCX
AAAALabel3
CPT0550.SCX
AAAALabel1
CPT0560.SCX
AAAALabel3
CPT0560.SCX
AAAALabel2
CPT0560.SCX
AAAALabel1
CPT0570.SCX
AAAALabel3
CPT0570.SCX
AAAALabel2
CPT0570.SCX
AAAALabel1
CPT0580.SCX
AAAALabel2
CPT0580.SCX
AAAALabel10
CPT0600.SCX
AAAALabel1
CPT0600.SCX
AAAALabel2
CPT0600.SCX
AAAALabel10
CPT0610.SCX

AAAALabel1
CPT0610.SCX
AAAALabel2
CPT0610.SCX
AAAALabel3
CPT0610.SCX
AAAALabel4
CPT0610.SCX
AAAALabel10
CPT0620.SCX
AAAALabel1
CPT0620.SCX
AAAALabel2
CPT0620.SCX
AAAALabel3
CPT0620.SCX
AAAALabel4
CPT0620.SCX
AAAALabel1
FINALIZA.SCX
AAAALabel2
FINALIZA.SCX
AAAALabel10
MENUPRIN.SCX
AAAALabel20
MENUPRIN.SCX
AAAALabel1
MENUPRIN.SCX
AAAALabel2
MENUPRIN.SCX
AAAALabel3
MENUPRIN.SCX
AAAALabel4
MENUPRIN.SCX
AAAALabel5
MENUPRIN.SCX
AAAALabel6
MENUPRIN.SCX
AAAALabel1
PRINCIPAL.SCX
AAAALabel2
PRINCIPAL.SCX
AAAALabel1
VALIDAR.SCX
AAAALabel2
VALIDAR.SCX
AAAALabel3
VALIDAR.SCX

grid
AAAAgrid1
WIZSTYLE.VCX
AAAAscrollgrid
WIZSTYLE.VCX
AAAAgrid1
CPT0312.SCX
AAAAgrid1
CPT0313.SCX

header
AAAAHeader1
WIZSTYLE.VCX
AAAAHeader1
CPT0312.SCX
AAAAHeader1
CPT0312.SCX
AAAAHeader1
CPT0312.SCX
AAAAHeader1
CPT0312.SCX
AAAAHeader1
CPT0313.SCX

AAAAHeader1
CPT0313.SCX
AAAAHeader1
CPT0313.SCX
AAAAHeader1
CPT0313.SCX
AAAAHeader1
CPT0313.SCX
AAAAHeader1
CPT0313.SCX
image
AAAAImage1
CPT0100.SCX
AAAAImage1
CPT0111.SCX
AAAAImage1
CPT0112.SCX
AAAAImage1
CPT0121.SCX
AAAAImage1
CPT0122.SCX
AAAAImage1
CPT0200.SCX
AAAAImage1
CPT0200B.SCX
AAAAImage1
CPT0210.SCX
AAAAImage1
CPT0220.SCX
AAAAImage1
CPT0230.SCX
AAAAImage1
CPT0241.SCX
AAAAImage1
CPT0242.SCX
AAAAImage1
CPT0300.SCX
AAAAImage1
CPT0300B.SCX
AAAAImage1
CPT0311.SCX
AAAAImage1
CPT0312.SCX
AAAAImage1
CPT0313.SCX
AAAAImage1
CPT0320.SCX
AAAAImage1
CPT0330.SCX
AAAAImage1
CPT0340.SCX
AAAAImage1
CPT0400.SCX
AAAAImage1
CPT0410.SCX
AAAAImage1
CPT0420.SCX
AAAAImage1
CPT0430.SCX
AAAAImage1
CPT0440.SCX
AAAAImage1
CPT0500.SCX
AAAAImage1
CPT0500B.SCX
AAAAImage1
CPT0520.SCX
AAAAImage1
CPT0530.SCX

ÄÄÄÄImage1
CPT0540.SCX
ÄÄÄÄImage1
CPT0550.SCX
ÄÄÄÄImage1
CPT0560.SCX
ÄÄÄÄImage1
CPT0570.SCX
ÄÄÄÄImage1
CPT0580.SCX
ÄÄÄÄImage1
CPT0600.SCX
ÄÄÄÄImage1
CPT0610.SCX
ÄÄÄÄImage1
CPT0620.SCX
ÄÄÄÄImage1
PRINCIPAL.SCX
ÄÄÄÄImage1
VALIDAR.SCX

oleboundcontrol
ÄÄÄÄembossedole
WIZSTYLE.VCX
ÄÄÄÄstandardole
WIZSTYLE.VCX
ÄÄÄÄOleboundcontrol1
WIZSTYLE.VCX
ÄÄÄÄOleboundcontrol1
WIZSTYLE.VCX
ÄÄÄÄOleboundcontrol1
WIZSTYLE.VCX
ÄÄÄÄOleboundcontrol1
WIZSTYLE.VCX

olecontrol
ÄÄÄÄprogreso
CPT0121.SCX
ÄÄÄÄprogreso
CPT0122.SCX
ÄÄÄÄprogreso
CPT0210.SCX
ÄÄÄÄprogreso
CPT0220.SCX
ÄÄÄÄprogreso
CPT0230.SCX
ÄÄÄÄprogreso
CPT0320.SCX
ÄÄÄÄprogreso
CPT0330.SCX
ÄÄÄÄprogreso
CPT0340.SCX
ÄÄÄÄprogreso
CPT0520.SCX
ÄÄÄÄprogreso
CPT0530.SCX
ÄÄÄÄprogreso
CPT0540.SCX
ÄÄÄÄprogreso
CPT0550.SCX
ÄÄÄÄprogreso
CPT0560.SCX
ÄÄÄÄprogreso
CPT0570.SCX

optiongroup
ÄÄÄÄOptiongroup1
WIZSTYLE.VCX
ÄÄÄÄoptGrpAndOr
WIZSTYLE.VCX

ÄÄÄÄtxs
CPT0100.SCX
ÄÄÄÄopciones
CPT0560.SCX
ÄÄÄÄopciones
CPT0570.SCX

pageframe
ÄÄÄÄboxpages
WIZSTYLE.VCX

shape
ÄÄÄÄShape1
WIZSTYLE.VCX
ÄÄÄÄShape2
WIZSTYLE.VCX
ÄÄÄÄwizshape
WIZSTYLE.VCX
ÄÄÄÄShape1
WIZSTYLE.VCX
ÄÄÄÄShape1
WIZSTYLE.VCX
ÄÄÄÄShape2
WIZSTYLE.VCX
ÄÄÄÄShape3
WIZSTYLE.VCX
ÄÄÄÄShape4
WIZSTYLE.VCX
ÄÄÄÄShape2
WIZSTYLE.VCX
ÄÄÄÄShape1
WIZSTYLE.VCX
ÄÄÄÄShape1
WIZSTYLE.VCX
ÄÄÄÄShape1
WIZSTYLE.VCX
ÄÄÄÄShape2
WIZSTYLE.VCX
ÄÄÄÄShape2
WIZSTYLE.VCX
ÄÄÄÄShape3
WIZSTYLE.VCX
ÄÄÄÄShape1
WIZSTYLE.VCX
ÄÄÄÄShape2
WIZSTYLE.VCX
ÄÄÄÄShape2
WIZSTYLE.VCX
ÄÄÄÄShape3
WIZSTYLE.VCX
ÄÄÄÄShape4
WIZSTYLE.VCX

spinner
ÄÄÄÄanyo
CPT0330.SCX

APÉNDICE B

GENERACIÓN DE CÓDIGO DEL FRONT END

```
*****
C:\COBRATEL\CPT0100.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú
transferencia de archivos."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializar
thisform.txs.Option1.value=valor1
thisform.txs.Option2.value=valor2
DO case
CASE valor1=1
thisform.Command1.visible=.T.
thisform.Command2.visible=.T.
thisform.Command3.visible=.F.
thisform.Command4.visible=.F.
CASE valor2=1
thisform.Command1.visible=.F.
thisform.Command2.visible=.F.
thisform.Command3.visible=.T.
thisform.Command4.visible=.T.
ENDCASE
thisform.Label1.visible=.T.
thisform.Label2.visible=.T.
thisform.Label3.visible=.F.
thisform.Label4.visible=.F.
thisform.refresh

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Transferencia de
archivos"
Height = 21
Left = 100
Top = 40
Width = 290
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .T.
Top = 320
Left = 650
Height = 59
Width = 84
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú principal"
StatusBarText = "Regresar al menú
principal."
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
* Boton Salir
thisform.release
DO form menuprin
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
ButtonCount = 2
BackStyle = 0
Value = 1
Height = 50
Left = 200
Top = 100
Width = 300
Name = "txs"
Option1.BackStyle = 0
Option1.Caption = "Transmisión y
extracción de archivos"
Option1.Value = 1
Option1.Height = 17
Option1.Left = 5
Option1.Top = 5
Option1.Width = 255
Option1.ForeColor = 255,255,255
Option1.Name = "Option1"
Option2.BackStyle = 0
Option2.Caption = "Operaciones con
archivos via diskette"
Option2.Height = 17
Option2.Left = 5
Option2.Top = 24
Option2.Width = 245
Option2.ForeColor = 255,255,255
Option2.Name = "Option2"
PROCEDURE Option1.Click
* Via modem
IF this.value=1 then
thisform.Command1.visible=.T.
thisform.Command2.visible=.T.
thisform.Command3.visible=.F.
thisform.Command4.visible=.F.
thisform.Label1.visible=.T.
thisform.Label2.visible=.T.
thisform.Label3.visible=.F.
thisform.Label4.visible=.F.
ENDIF

ENDPROC
PROCEDURE Option2.Click
* Via modem
IF this.value=1 then
thisform.Command1.visible=.F.
thisform.Command2.visible=.F.
thisform.Command3.visible=.T.
thisform.Command4.visible=.T.
thisform.Label1.visible=.F.
thisform.Label2.visible=.F.
thisform.Label3.visible=.T.
thisform.Label4.visible=.T.
ENDIF

thisform.Label2.visible=.F.
thisform.Label3.visible=.T.
thisform.Label4.visible=.T.
ENDIF

ENDPROC
AutoSize = .F.
Top = 200
Left = 200
Height = 59
Width = 80
WordWrap = .T.
Picture = files09.ico
Caption = ""
StatusBarText = "Extracción de
archivos."
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton Transferencia de archivos
thisform.release
valor1=1
valor2=0
DO form cpt0111

ENDPROC
Top = 280
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = mail03.ico
Caption = ""
StatusBarText = "Transmisión de
archivos PISA."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Procesos especiales
```



```

thisform.release
valor1=1
valor2=0
DO form cpt0112
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
Top = 200
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = bckup.ico
DownPicture =
Caption = ""
StatusBarText = "Respaldo de
archivos PISA."
TabIndex = 5
ToolTipText = ""
Name = "Command3"
PROCEDURE LostFocus
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton respaldo de archivos PISSA
thisform.release
valor1=0
valor2=1
DO form cpt0121
ENDPROC
Top = 280
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = resto.ico
Caption = ""
StatusBarText = "Cobranza por
terceros."
TabIndex = 6
ToolTipText = ""
Name = "Command4"
PROCEDURE LostFocus
thisform.Label4.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label4.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Mantenimiento de catálogos
thisform.release
valor1=0

```

```

valor2=1
DO form cpt0122
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
BackStyle = 0
Caption = "Extracción de archivos"
Height = 21
Left = 290
Top = 220
Width = 160
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Transmisión de archivos
PISA"
Height = 21
Left = 290
Top = 300
Width = 180
ForeColor = 255,255,255
Name = "Label2"
BackStyle = 0
Caption = "Respaldo de archivos
PISA"
Height = 21
Left = 290
Top = 220
Width = 170
ForeColor = 255,255,255
Name = "Label3"
BackStyle = 0
Caption = "Restauración de archivos
PISA"
Height = 21
Left = 290
Top = 300
Width = 275
ForeColor = 255,255,255
Name = "Label4"
Picture = net01.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

*****
C:\COBRATEL\CPT0111.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 10
Top = 20
Width = 90
Height = 90
Alias = "agentes"
Database = dbCOBRATEL.dbc
CursorSource = "agentes"
Name = "Cursor1"
Left = 150

```

```

Top = 20
Width = 90
Height = 90
Alias = "est_cobro"
Database = dbCOBRATEL.dbc
CursorSource = "est_cobro"
Name = "Cursor2"
Left = 290
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor3"
Left = 10
Top = 140
Width = 90
Height = 90
Alias = "b_transmision"
Database = dbCOBRATEL.dbc
CursorSource = "b_transmision"
Name = "Cursor4"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL® Extracción
de archivos de terceros al servidor."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
thisform.Command1.visible=.T.
thisform.refresh()

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Extracción de archivos de
terceros"
Height = 31
Left = 100
Top = 40
Width = 372
TabIndex = 6
ForeColor = 255,255,255
Name = "Label1"
Enabled = .F.
Height = 21
Left = 340
TabIndex = 11
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 530
Top = 13
Visible = .T.

```

```

Left = 440
TabIndex = 12
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 440
Top = 68
Visible = .F.
Width = 40
TabIndex = 7
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 690
TabIndex = 13
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 657
Top = 44
Visible = .T.
Width = 40
TabIndex = 10
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 540
TabIndex = 14
Top = 90
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 570
TabIndex = 15
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 340
Top = 68
Visible = .F.
Width = 90
TabIndex = 8
Name = "Embossedlabel4"
AutoSize = .T.
Top = 310
Left = 640
Height = 59
Width = 84
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Transferencia de
archivos"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 5
Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord

```

```

this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0100
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Vía de cobro"
Height = 21
Left = 170
Top = 150
Width = 71
TabIndex = 16
ForeColor = 255,255,255
Name = "Label6"
BackStyle = 0
Caption = "Estación de cobro"
Height = 21
Left = 170
Top = 190
Width = 100
TabIndex = 17
ForeColor = 255,255,255
Name = "Label2"
ColumnCount = 2
ColumnWidths = "122,59"
RowSourceType = 6
RowSource =
"est_cobro.nomb_ec,num_ec"
FirstElement = 1
Height = 21
Left = 280
NumberOfElements = 0
Style = 2
TabIndex = 2
Top = 190
Width = 150
Name = "Combo2"
Enabled = .F.
Height = 21
Left = 440
TabIndex = 18
Top = 150
Visible = .F.
Width = 41
Name = "Text6"
Enabled = .F.
Height = 21
Left = 440
TabIndex = 19
Top = 190
Visible = .F.
Width = 41
Name = "Text7"
Format = "!"
Height = 21
InputMask = "XXXXXXXX"
Left = 280
TabIndex = 3
Top = 230
Width = 90

```

```

Name = "Text8"
PROCEDURE Valid
*VALIDAR
IF len(trim(THIS.VALUE))=8
RETURN .t.
ELSE
MESSAGEBOX('El nombre del
archivo es incorrecto. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
BackStyle = 0
Caption = "Archivo"
Height = 21
Left = 170
Top = 230
Width = 100
TabIndex = 20
ForeColor = 255,255,255
Name = "Label3"
AutoSize = .F.
Top = 170
Left = 510
Height = 60
Width = 90
WordWrap = .T.
Picture = checkmrk.ico
Caption = "Extraer archivo"
StatusBarText = "Extraer archivos de
terceros."
TabIndex = 4
Name = "Command1"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Extraer
IF empty(thisform.Text8.value)
thisform.Text8.setfocus
ELSE

thisform.Text6.value=agentes.vi_co_
ag

thisform.Text7.value=est_cobro.num_
ec

thisform.text10.value=est_cobro.sigla_
ec

thisform.text9.value=alltrim(thisform.
Text8.value)+"."+thisform.text10.valu
e
nomarch=thisform.text9.value
ruta="c:\COBRATEL\repcion\
archivo=ruta+nomarch

newnomarch=substr(nomarch,10,3)+
substr(nomarch,1,8)+".TXT"
newruta="c:\COBRATEL\datos\
newarchivo=newruta+newnomarch
**Buscar el archivo

```

```

IF file(archivo)
COPY file &archivo to &newarchivo
MESSAGEBOX('El archivo ha sido
recuperado con
éxito',0+64+0,'Recuperación
exitosa.')
*****REGISTRO EN BITACORA de
procesos
SELECT procesos
APPEND blank
REPLACE tarea_bp with "Extracc
arch "+newnomarch
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
* SELECT AGENTES
***** CERRAR BITACORA DE
PROCESOS
*****REGISTRO EN BITACORA de
transmisiones
SELECT b_transmision
APPEND blank
REPLACE es_co_bt with
thisform.Text7.value
REPLACE via_co_bt with
thisform.Text6.value
REPLACE fech_bt with date()
REPLACE hora_bt with time()
REPLACE estado_bt with usuario
SELECT agentes
***** CERRAR BITACORA DE
PROCESOS
IF MESSAGEBOX('¿Desea
recuperar otro archivo de terceros?',
4+32+256, 'Por favor confirme') = 6
thisform.Text8.value=""
thisform.Text8.setfocus
ELSE
thisform.release
DO form cpt0100
ENDIF
ELSE
MESSAGEBOX('Este archivo no
se encuentra disponible. ¡Verifique
por favor!',0+16+0,'Error.')
ENDIF
ENDIF

ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
Enabled = .F.
Format = ""
Height = 21
InputMask = ""
Left = 390
TabIndex = 21
Top = 230
Visible = .F.
Width = 220
Name = "Text9"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 22
Top = 190
Visible = .F.

```

```

Width = 41
Name = "Text10"
ColumnCount = 2
ColumnWidths = "113,62"
RowSourceType = 6
RowSource =
"agentes.nom_agen,vi_co_ag"
FirstElement = 1
Height = 21
Left = 280
NumberOfElements = 0
SelectOnEntry = .F.
Style = 2
TabIndex = 1
Top = 150
Width = 150
Name = "Combo1"
Picture = files09.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

***** *
C:\COBRATEL\CPT0112.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 12
Top = 8
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor3"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®
Transmisión de archivos PISA."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
thisform.Text6.value=date()-1

ENDPROC
FontBold = .T.
FontSize = 16
FontUnderline = .F.
BackStyle = 0
Caption = "Transmisión de archivos
PISA"
Height = 31

```

```

Left = 100
Top = 40
Width = 332
TabIndex = 6
ForeColor = 255,255,255
Name = "Label1"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 11
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
FontUnderline = .F.
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 9
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 12
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.
Width = 40
TabIndex = 7
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 13
Top = 40
Width = 80
Name = "Text3"
FontUnderline = .F.
Caption = "Fecha"
Height = 13
Left = 644
Top = 43
Visible = .T.
Width = 40
TabIndex = 10
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 14
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 15
Top = 10

```

```

Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 68
Visible = .F.
Width = 90
TabIndex = 8
Name = "Embossedlabel4"
AutoSize = .T.
Top = 320
Left = 640
Height = 59
Width = 84
WordWrap = .T.
Picture = arw09it.ico
Caption = "Transferencia de
archivos"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 5
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0100
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
FontUnderline = .F.
WordWrap = .T.
BackStyle = 0
Caption = "Fecha del archivo PISA
por transmitir"
Height = 40
Left = 100
Top = 160
Width = 110
TabIndex = 16
ForeColor = 255,255,255
Name = "Label6"
AutoSize = .F.
Top = 160
Left = 390
Height = 60
Width = 100
WordWrap = .T.
Picture = checkmrk.ico
Caption = "Transmitir PISA"
StatusBarText = "Extraer archivos de
terceros."
TabIndex = 4
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton Extraer
IF thisform.Text6.value >= date()
MESSAGEBOX('La fecha
proporcionada no es válida.
¡Verifique la por
favor!',0+16+0,'Error.')
```

```

ELSE
dato=dtoc(thisform.Text6.value)
anyo=substr(dato,1,4)
mes=substr(dato,6,2)
dia=substr(dato,9,2)

thisform.text9.value="PISA"+anyo+m
es+dia+".TXT"
```

```

nomarch=alltrim(thisform.text9.value)
ruta="c:\COBRATEL\archpisa"
archivo=ruta+nomarch
newruta="c:\COBRATEL\bckupisa"
newarchivo=newruta+nomarch
```

```

SELECT procesos
GO top
LOCATE for tarea_bp =
"Transmision "+nomarch
IF found()
MESSAGEBOX('El archivo ya fué
transmitido con
anterioridad.',0+64+0,'Aviso.')
```

```

IF MESSAGEBOX('¿Desea
enviarlo nuevamente?', 4+32+256,
'Por favor confirme') = 6
IF file(archivo)
COPY file &archivo to
&newarchivo
MESSAGEBOX('Transmisión
de archivo PISA
finalizada.',0+64+0,'Transmisión
exitosa.')
```

```

* REGISTRO EN BITACORA de
procesos
SELECT procesos
APPEND blank
REPLACE tarea_bp with
"Transmision "+nomarch
REPLACE usua_bp with
usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT procesos
* CERRAR BITACORA DE
PROCESOS
ELSE
MESSAGEBOX('Este archivo
no se encuentra disponible.
¡Verifique por favor!',0+16+0,'Error.')
```

```

ENDIF
ENDIF
ELSE
* Buscar el archivo
IF file(archivo)
COPY file &archivo to
&newarchivo
MESSAGEBOX('Transmisión de
archivo PISA
finalizada.',0+64+0,'Transmisión
exitosa.')
```

```

*****REGISTRO EN BITACORA de
procesos
SELECT procesos
APPEND blank
REPLACE tarea_bp with
"Transmision "+nomarch
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT procesos
*****CERRAR BITACORA DE
PROCESOS
ELSE
MESSAGEBOX('Este archivo no
se encuentra disponible. ¡Verifique
por favor!',0+16+0,'Error.')
```

```

ENDIF
ENDIF
ENDIF
```

```

ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
```

```

Enabled = .F.
Format = ""
Height = 21
InputMask = ""
Left = 100
TabIndex = 21
Top = 200
Visible = .T.
Width = 230
Name = "Text9"
Height = 21
Left = 210
Top = 160
Width = 121
Name = "Text6"
```

```

PROCEDURE Valid
* Validar la fecha
IF this.value>=date()
MESSAGEBOX('La fecha
proporcionada no es válida.
¡Verifique la por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ELSE
RETURN .t.
ENDIF

ENDPROC
Picture = mail03.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
```

```

*****
C:\COBRATEL\CPT0121.SCX
Left = 1
Top = 220
```

```

Width = 520
Height = 200
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Respaldo
de archivos PISA."
WindowState = 2
AlwaysOnTop = .F.
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.Text6.value=date()

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Respaldo de archivos
PISA"
Height = 31
Left = 100
Top = 40
Width = 290
ForeColor = 255,255,255
Name = "Label1"
AutoSize = T
Top = 320
Left = 640
Height = 59
Width = 84
WordWrap = .T.
Picture = arw09flt.ico
Caption = "Transferencia de
archivos"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0100
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
Top = 270
Left = 240
Height = 60

```

```

Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Respaldar archivos
de bases de datos del sistema."
TabIndex = 9
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton depurar
mifecha=dtoc(thisform.Text6.value)
yr=substr(mifecha,1,4)
mes=substr(mifecha,6,2)
dia=substr(mifecha,9,2)

ruta="a:\\"
ruta2="c:\COBRATEL\archpisa\"
arch2="PISA"+yr+mes+dia+".TXT"
nomarch2=ruta2+arch2
thisform.Text7.value=nomarch2

newarch=ruta+arch2

IF file(nomarch2)
* messagebox('Si existe
archivo.'0+64+0.'OK')
IF MESSAGEBOX('¿ Está
preparada su unidad de respaldo [A:]
?', 4+32+256, 'Confirme por favor...')
= 6 then
&& si está lista la unidad de respaldo
thisform.progreso.visible=.T.
COPY file &nomarch2 to
&newarch
FOR i=1 to 100 step 2
thisform.progreso.value=i
WAIT " timeout 0.1
NEXT i

USE procesos
APPEND blank
REPLACE tarea_bp with
"Respaldo "+arch2
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
USE && cierra
procesos.....
..... cerrar procesos
MESSAGEBOX('Archivo PISA
respaldo con
éxito'.0+64+0.'Operación finalizada')
thisform.progreso.visible=.f.

ELSE
&& no está lista la unidad de
resplado
MESSAGEBOX('Prepare su
unidad de respaldo e inténtelo
nuevamente.'0+64+0.'Respaldo
cancelado')
ENDIF
ELSE

```

```

MESSAGEBOX('NO existe el
archivo PISA
correspondiente.'0+64+0.'Error!')
ENDIF

ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Respaldo archivos"
Height = 21
Left = 340
Top = 290
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Top = 360
Left = 120
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"
Enabled = .F.
Height = 21
Left = 480
TabIndex = 23
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 521
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 580
TabIndex = 24
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 580
Top = 68
Visible = F
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25

```

```

Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 648
Top = 44
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 16
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 148
Top = 68
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
BackStyle = 0
Caption = ""
Height = 21
Left = 120
Top = 320
Width = 400
Name = "Label2"
Height = 21
Left = 200
Top = 160
Width = 101
Name = "Text6"
PROCEDURE Valid
* Validar que la fecha sea menor a la
actual
IF this.value<=date()
RETURN .t.
ELSE
MESSAGEBOX('Esta fecha no es
válida. ¡Verifique por
favor!',0+16+0,'Error.')
RETURN .f.
ENDIF
ENDPROC
BackStyle = 0
Caption = "Fecha"
Height = 21
Left = 120
Top = 160
Width = 50
ForeColor = 255,255,255
Name = "Label4"
Enabled = .F.
Height = 21

```

```

Left = 200
Top = 200
Width = 320
Name = "Text7"
BackStyle = 0
Caption = "Archivo pisa"
Height = 21
Left = 120
Top = 200
Width = 80
ForeColor = 255,255,255
Name = "Label5"
Picture = bckup.ico
Stretch = 1
Height = 42
Left = 40
Top = 20
Width = 42
Name = "Image1"
BackStyle = 0
Caption = "Respaldo de archivo pisa
en el Drive [A:]"
Height = 21
Left = 120
Top = 120
Width = 230
ForeColor = 255,255,255
Name = "Label6"

*****
C:\COBRATEL\CPT0122.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®.
Restauración de archivos PISA."
WindowState = 2
AlwaysOnTop = .F.
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.Text6.value=date()

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Restauración de archivos
PISA"
Height = 31
Left = 100
Top = 40
Width = 320
ForeColor = 255,255,255
Name = "Label1"

```

```

AutoSize = .T.
Top = 320
Left = 640
Height = 59
Width = 84
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Transferencia de
archivos"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0100
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
Top = 260
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Restaurar archivo
PISA."
TabIndex = 9
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton RESTAURAR

ruta="A:\"

IF MESSAGEBOX('¿ Tiene lista la
unidad de respaldo [A:] ?',
4+32+256, 'Confirme por favor...') = 6
then
mifecha=dtoc(thisform.Text6.value)
yr=substr(mifecha,1,4)
mes=substr(mifecha,6,2)
dia=substr(mifecha,9,2)
*ruta="c:\COBRATEL\bckup\pisa\"
arch="PISA"+yr+mes+dia+".TXT"
nomarch=ruta+arch
thisform.Text7.value=nomarch

IF file(nomarch)
* messagebox('Si existe
archivo.',0+64+0,'Ok')
IF MESSAGEBOX('¿ Está seguro
de restaurar este archivo PISA?',

```

```

4+32+256, 'Confirme por favor...') = 6
then
&& si existe el respaldo del archivo
  thisform.progreso.visible=.T.
  COPY file &nomarch to
  "c:\COBRATEL\archpisa\"+arch
  FOR j=1 to 100 step 2
    thisform.progreso.value=j
    WAIT " timeout 0.1
  NEXT j

  USE procesos
  APPEND blank
  REPLACE tarea_bp with
  "Restauración "+arch
  REPLACE usua_bp with usuario
  REPLACE fecha_bp with date()
  REPLACE hora_bp with time()
  USE && cierra
procesos.....
..... cerrar procesos
  MESSAGEBOX('Archivo PISA
restaurado con
éxito',0+64+0,'Operación finalizada')
  thisform.progreso.visible=.f.
  ENDIF
  ELSE
  MESSAGEBOX('NO existe el
archivo PISA
correspondiente.',0+64+0,'Error!')
  ENDIF
  ELSE
  MESSAGEBOX('Por favor
prepárela y e inténtelo
nuevamente',0+64+0,'Error,')
  ENDIF

ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Restaurar archivo"
Height = 21
Left = 300
Top = 280
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Top = 360
Left = 120
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"
Enabled = .F
Height = 21
Left = 490
TabIndex = 23
Top = 70

```

```

Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 15
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 600
TabIndex = 24
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 600
Top = 58
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 646
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 690
TabIndex = 26
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
BackStyle = 0
Caption = ""

```

```

Height = 21
Left = 120
Top = 320
Width = 400
Name = "Label2"
Height = 21
Left = 200
Top = 160
Width = 101
Name = "Text6"
PROCEDURE Valid
* Validar que la fecha sea menor a la
actual
IF this.value<=date()
  RETURN .t.
ELSE
  MESSAGEBOX('Esta fecha no es
válida. ¡Verifique por
favor!',0+16+0,'Error!')
  RETURN .f.
ENDIF
ENDPROC
BackStyle = 0
Caption = "Fecha"
Height = 21
Left = 120
Top = 160
Width = 50
ForeColor = 255,255,255
Name = "Label4"
Enabled = .F.
Height = 21
Left = 200
Top = 200
Width = 340
Name = "Text7"
BackStyle = 0
Caption = "Archivo pisa"
Height = 21
Left = 120
Top = 200
Width = 80
ForeColor = 255,255,255
Name = "Label5"
Picture = resto.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
AutoSize = .T.
BackStyle = 0
Caption = "Restauración de archivo
Pisa del drive [A:]"
Height = 17
Left = 120
Top = 120
Width = 231
ForeColor = 255,255,255
Name = "Label6"

*****
C:\COBRATEL\CPT0200.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793

```

```

DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú de
procesos especiales."
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Procesos especiales"
Height = 30
Left = 90
Top = 40
Width = 210
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09ft.ico
Caption = "Menú principal"
StatusBarText = "Regresar al menú
principal."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form menuprin
ENDPROC
Top = 140
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = pen07.ico
Caption = ""
StatusBarText = "Regeneración de
archivos PISA."
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE Click
* Boton regeneracion de archivos
pisa
thisform.release
DO form cpt0210
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
Top = 220
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = pencil05.ico
Caption = ""
StatusBarText = "Integración de
cobranza manual."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton integracion de cobranza
manual
thisform.release
DO form cpt0220
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
Top = 140
Left = 380
Height = 60
Width = 81
WordWrap = .T.
Picture = clip07.ico
DownPicture =
Caption = ""
StatusBarText = "Integración
terceros faltantes."
TabIndex = 5
ToolTipText = ""
Name = "Command3"
PROCEDURE Click
* Boton integracion terceros faltantes
thisform.release
DO form cpt0230
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE GotFocus
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC

```

```

PROCEDURE LostFocus
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
Top = 220
Left = 380
Height = 60
Width = 81
WordWrap = .T.
Picture = erase02.ico
Caption = ""
StatusBarText = "Módulo de ajustes."
TabIndex = 6
ToolTipText = ""
Name = "Command4"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton modulo de ajustes
thisform.release
DO form cpt0200b
ENDPROC
PROCEDURE GotFocus
thisform.Label4.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label4.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Regeneración de archivos
PISA"
Height = 21
Left = 150
Top = 160
Width = 180
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Integración de cobranza"
Height = 21
Left = 150
Top = 240
Width = 180
ForeColor = 255,255,255
Name = "Label2"
BackStyle = 0
Caption = "Integración terceros
faltantes"
Height = 21
Left = 470
Top = 160
Width = 160
ForeColor = 255,255,255
Name = "Label3"
BackStyle = 0
Caption = "Módulo de ajustes"
Height = 21
Left = 470
Top = 240
Width = 160
ForeColor = 255,255,255
Name = "Label4"
Picture = vb416mom.bmp
Stretch = 1

```



```

BackStyle = 1
Height = 42
Left = 35
Top = 20
Width = 40
Name = "Image1"

*****
C:\COBRATEL\CPT0200B.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú de
Ajustes."
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Ajustes"
Height = 30
Left = 100
Top = 40
Width = 80
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lit ico
Caption = "Menú de procesos"
StatusBarText = "Regresar al menú
principal."
TabIndex = 9
Name = "Command10"
PROCEDURE Click
* Boton regresar al menu de ajustes
thisform.release
DO form cpt0200
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 140
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = note02.ico
Caption = ""

```

```

StatusBarText = "Ajustes tabla de
cobros"
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton ajuste a la tabla de cobros
thisform.release
DO form cpt0241
ENDPROC
Top = 220
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = note03.ico
Caption = ""
StatusBarText = "Ajustes tabla de
pagos"
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton ajuste a la tabla de pagos
thisform.release
DO form cpt0242
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
BackStyle = 0
Caption = "Ajustes a la tabla de
cobros"
Height = 21
Left = 290
Top = 160
Width = 180
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Ajustes a la tabla de
pagos"
Height = 21
Left = 290

```

```

Top = 240
Width = 180
ForeColor = 255,255,255
Name = "Label2"
Picture = erase02.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

*****
C:\COBRATEL\CPT0210.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 129
Top = 13
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor3"
Left = 13
Top = 12
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®
Transmisión de archivos PISA."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
thisform.Text6.value=date()-1

SELECT cobros

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Regeneración de archivos
PISA"
Height = 31
Left = 100
Top = 40
Width = 332
TabIndex = 6
ForeColor = 255,255,255
Name = "Label1"

```

```

Enabled = .F.
Height = 21
Left = 490
TabIndex = 11
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 9
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 12
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.
Width = 40
TabIndex = 7
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 13
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 646
Top = 44
Visible = .T.
Width = 40
TabIndex = 10
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 14
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 15
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 68
Visible = .F.

```

```

Width = 90
TabIndex = 8
Name = "Embossedlabel4"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 79
WordWrap = .T.
Picture = arw09lit.ico
Caption = "Procesos especiales"
StatusBarText = "Regresar al menú
de procesos especiales"
TabIndex = 5
Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0200
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
WordWrap = .T.
BackStyle = 0
Caption = "Fecha del archivo PISA
por regenerar"
Height = 40
Left = 110
Top = 160
Width = 110
TabIndex = 16
ForeColor = 255,255,255
Name = "Label6"
AutoSize = .F.
Top = 160
Left = 380
Height = 60
Width = 100
WordWrap = .T.
Picture = checkmrk.ico
Caption = "Regenerar PISA"
StatusBarText = "Regenerar archivo
PISA."
TabIndex = 4
Name = "Command1"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton regenerar archivo PISA
IF thisform.Text6.value >= date()
MESSAGEBOX('La fecha
proporcionada no es válida.

```

```

¡Verifique! por
favor!',0+16+0,'Error.')
```

```

ELSE
dato=dtoc(thisform.Text6.value)
anyo=substr(dato,1,4)
mes =substr(dato,6,2)
dia =substr(dato,9,2)

thisform.text9.value="PISA"+anyo+m
es+dia+".TXT"

nomarch=alltrim(thisform.text9.value)
ruta="c:\COBRATEL\archpisa\
archivo=ruta+nomarch

SELECT cobros
SET filter to
fech_cob=thisform.Text6.value
COUNT to hay
IF hay >0
** si hay datos para generar PISA
IF MESSAGEBOX('¿Está seguro
de regenerar este archivo PISA?',
4+32+256, 'Por favor confirme') = 6
COPY to &archivo delimited
thisform.progreso.visible=.T.
FOR i=1 to 100
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i
thisform.progreso.visible=.F.
MESSAGEBOX('Archivo PISA
regenerado.',0+16+0,'Regeneración
exitosa.')
```

```

***
SELECT procesos
APPEND blank
REPLACE tarea_bp with
"Regeneración "+nomarch
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT cobros
***
ENDIF
ELSE
** no hay datos para genrar PISA
MESSAGEBOX('No hay datos
para poder generar el archivo PISA
correspondiente a esta fecha.
¡Verifique por
favor!',0+16+0,'Regeneración
cancelada.')
```

```

ENDIF
SET filter to

ENDIF

ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
Enabled = .F.
Format = "!"
Height = 21
InputMask = ""
Left = 110

```

```

TabIndex = 21
Top = 190
Visible = .T.
Width = 230
Name = "Text9"
Height = 21
Left = 220
Top = 160
Width = 121
Name = "Text6"
PROCEDURE Valid
* Validar la fecha
IF this.value>=date()
MESSAGEBOX('La fecha
proporcionada no es válida.
;Verifique la por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ELSE
RETURN .t.
ENDIF

ENDPROC
Picture = pen07.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 338
Left = 128
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"
```

```

*****
C:\COBRATEL\CPT0220.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 129
Top = 13
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor3"
Left = 13
Top = 12
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor1"
Left = 250
Top = 15
Width = 90
Height = 90
Alias = "h_cobros"
Database = dbCOBRATEL.dbc
CursorSource = "h_cobros"
Name = "Cursor2"
```

```

Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®
Integración de cobranza."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
thisform.Text6.value=date()-1
thisform.Text6.setfocus
```

```

SELECT cobros

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Integración de cobranza"
Height = 31
Left = 100
Top = 40
Width = 332
TabIndex = 4
ForeColor = 255,255,255
Name = "Label1"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 9
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 7
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 10
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.
Width = 40
TabIndex = 5
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 11
```

```

Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 646
Top = 44
Visible = .T.
Width = 40
TabIndex = 8
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 12
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 13
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 68
Visible = .F.
Width = 90
TabIndex = 6
Name = "Embossedlabel4"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 79
WordWrap = .T.
Picture = arw09flt.ico
Caption = "Procesos especiales"
StatusBarText = "Regresar al menú
de procesos especiales"
TabIndex = 3
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0200
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
WordWrap = .T.
BackStyle = 0
Caption = "Fecha a procesar"
Height = 40
```

```

Left = 110
Top = 160
Width = 110
TabIndex = 14
ForeColor = 255,255,255
Name = "Label6"
AutoSize = .F.
Top = 160
Left = 380
Height = 60
Width = 100
WordWrap = .T.
Picture = checkmrk.ico
Caption = "Integrar cobranza"
StatusBarText = "Regenerar archivo PISA."
TabIndex = 2
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton regenerar archivo PISA

SELECT cobros
IF thisform.Text6.value >= date()
MESSAGEBOX("La fecha proporcionada no es válida. ¡Verifiquela por favor!",0+16+0,'Error.')
```

```

ELSE
LOCATE for
fe_re_co=thisform.Text6.value
IF found()
*si está
ELSE
* buscar en histórico de cobros
SELECT h_cobros
LOCATE for
fe_rec_hc=thisform.Text6.value
IF found()
*si está
IF messagebox("La fecha proporcionada ya ha sido procesada. ¿Desea procesarla nuevamente?",4+32+256,'Atención.')
```

```

=6
*si desea procesarla nuevamente
*
ruta="c:\COBRATEL\recepcion"

ELSE
* no desea procesarla nuevamente
MESSAGEBOX("Integración de cobranza cancelada.",0+16+0,'Atención.')
```

```

ENDIF
ELSE
* no está
* messagebox("No hay cobranza por integrar en esta fecha.",0+16+0,'Atención.')
```

```

ENDIF
ENDIF
```

```

ENDIF
SELECT cobros

* dato=dtoc(thisform.text6.value)
* anyo=substr(dato,1,4)
* mes =substr(dato,6,2)
* dia =substr(dato,9,2)
*
thisform.text9.value="PISA"+anyo+mes+dia+".TXT"
*
nomarch=alltrim(thisform.text9.value)
* ruta="c:\COBRATEL\archpisa"
* archivo=ruta+nomarch

* select cobros
* set filter to
fech_cob=thisform.text6.value
* count to hay
* if hay >0
** si hay datos para generar PISA
* if MESSAGEBOX('¿Está seguro de regenerar este archivo PISA?',4+32+256,'Por favor confirme') = 6
* copy to &archivo delimited
* thisform.progreso.visible=.T.
* for i=1 to 100
* thisform.progreso.value=i
* wait "" timeout .1
* next i
* thisform.progreso.visible=.F.
* messagebox('Archivo PISA regenerado.',0+16+0,'Regeneración exitosa.')
```

```

***
* SELECT procesos
* append blank
* replace tarea_bp with "Regeneración "+nomarch
* replace usua_bp with usuario
* replace fecha_bp with date()
* replace hora_bp with time()
* SELECT cobros
***
* endif
* else
* ** no hay datos para genrar PISA
* messagebox("No hay datos para poder generar el archivo PISA correspondiente a esta fecha. ¡Verifique por favor!",0+16+0,'Regeneración cancelada.')
```

```

* endif
* set filter to
*endif

ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(255,255,255)
ENDPROC
Enabled = .F.
Format = ""
```

```

Height = 21
InputMask = ""
Left = 110
TabIndex = 15
Top = 190
Visible = .T.
Width = 230
Name = "Text9"
Height = 21
Left = 220
TabIndex = 1
Top = 160
Width = 121
Name = "Text6"
PROCEDURE Valid
* Validar la fecha
IF this.value>=date()
MESSAGEBOX("La fecha proporcionada no es válida. ¡Verifiquela por favor!",0+16+0,'Error.')
```

```

RETURN .f.
ELSE
RETURN .t.
ENDIF

ENDPROC
Picture = pencil05.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 338
Left = 128
Height = 11
Width = 411
Visible = .F.
TabIndex = 16
Name = "Progreso"

*****
C:\COBRATEL\CPT0230.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 129
Top = 13
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor3"
Left = 13
Top = 12
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor1"
Left = 250
Top = 15
Width = 90
```

```

Height = 90
Alias = "h_cobros"
Database = dbCOBRATEL.dbc
CursorSource = "h_cobros"
Name = "Cursor2"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®
Integración cobranza de terceros
faltantes."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
thisform.Text6.value=date()-1
thisform.Text6.setfocus

SELECT cobros

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Integración cobranza de
terceros faltantes"
Height = 31
Left = 100
Top = 40
Width = 450
TabIndex = 4
ForeColor = 255,255,255
Name = "Label1"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 9
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 7
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = F
Height = 21
Left = 590
TabIndex = 10
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.

```

```

Width = 40
TabIndex = 5
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 11
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 646
Top = 44
Visible = .T.
Width = 40
TabIndex = 8
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 12
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 13
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 68
Visible = .F.
Width = 90
TabIndex = 6
Name = "Embossedlabel4"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 79
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Procesos especiales"
StatusBarText = "Regresar al menú
de procesos especiales"
TabIndex = 3
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0200
ENDPROC
PROCEDURE MouseMove

```

```

LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
WordWrap = .T.
BackStyle = 0
Caption = "Fecha a procesar"
Height = 40
Left = 110
Top = 160
Width = 110
TabIndex = 14
ForeColor = 255,255,255
Name = "Label6"
AutoSize = .F.
Top = 160
Left = 380
Height = 60
Width = 100
WordWrap = .T.
Picture = checkmrk.ico
Caption = "Integrar cobranza"
StatusBarText = "Regenerar archivo
PISA."
TabIndex = 2
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton regenerar archivo PISA
SELECT cobros
IF thisform.Text6.value >= date()
MESSAGEBOX('La fecha
proporcionada no es válida.
¡Verifique!a por
favor!',0+16+0,'Error.')
```

```

ELSE
LOCATE for
fe_re_co=thisform.Text6.value
IF found()
*si está
ELSE
* buscar en histórico de cobros
- SELECT h_cobros
LOCATE for
fe_rec_hc=thisform.Text6.value
IF found()
*si está
IF messagebox('La fecha
proporcionada ya ha sido procesada.
¿Desea procesarla
nuevamente?',4+32+256,'Atención.')
```

```

=6
*si desea procesarla
nuevamente
*
ruta="c:\COBRATEL\repcion"

ELSE
* no desea procesarla
nuevamente
MESSAGEBOX('Integración de
cobranza
cancelada.',0+16+0,'Atención.')
```

```

ENDIF

```

```

ELSE
  * no está
  * messagebox('No hay cobranza
por integrar en esta
fecha.',0+16+0,'Atención.')
ENDIF
ENDIF
ENDIF
SELECT cobros

* dato=dtoc(thisform.text6.value)
* anyo=substr(dato,1,4)
* mes =substr(dato,6,2)
* dia =substr(dato,9,2)
*
thisform.text9.value="PISA"+anyo+m
es+dia+".TXT"
*
nomarch=alltrim(thisform.text9.value)
* ruta="c:\COBRATEL\archpisa"
* archivo=ruta+nomarch

* select cobros
* set filter to
fech_cob=thisform.text6.value
* count to hay
* if hay >0
  ** si hay datos para generar PISA
  * if MESSAGEBOX('¿Está seguro
de regenerar este archivo PISA?',
4+32+256, 'Por favor confirme') = 6
  * copy to &archivo delimited
  * thisform.progreso.visible=.T.
  * for i=1 to 100
  *   thisform.progreso.value=i
  *   wait "" timeout .1
  *   next i
  * thisform.progreso.visible=.F.
  * messagebox('Archivo PISA
regenerado.',0+16+0,'Regeneración
exitosa.')
  ***
  * SELECT procesos
  * append blank
  * replace tarea_bp with
"Regeneración "+nomarch
  * replace usua_bp with usuario
  * replace fecha_bp with date()
  * replace hora_bp with time()
  * SELECT cobros
  ***
  * endif
  * else
  * ** no hay datos para genrar PISA
  * messagebox('No hay datos para
poder generar el archivo PISA
correspondiente a esta fecha.
¡Verifique por
favor!',0+16+0,'Regeneración
cancelada.')
  * endif
  * set filter to
*endif

ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)

```

```

ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(255,255,255)
ENDPROC
Enabled = .F.
Format = "!"
Height = 21
InputMask = ""
Left = 110
TabIndex = 15
Top = 190
Visible = .T.
Width = 230
Name = "Text9"
Height = 21
Left = 220
TabIndex = 1
Top = 160
Width = 121
Name = "Text6"
PROCEDURE Valid
* Validar la fecha
IF this.value>=date()
  MESSAGEBOX('La fecha
proporcionada no es válida.
¡Verifique la
fecha por
favor!',0+16+0,'Error.')
  RETURN .f.
ELSE
  RETURN .t.
ENDIF

ENDPROC
Picture = clip07.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 338
Left = 128
Height = 11
Width = 411
Visible = .F.
TabIndex = 16
Name = "Progreso"

***** *
C:\COBRATEL\CPT0241.SCX
Left = 4
Top = 170
Width = 520
Height = 220
Name = "Dataenvironment"
Alias = "cobros"
BufferModeOverride = 5
ORDER = "fech_cob"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor1"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "h_cobros"
Database = dbCOBRATEL.dbc

```

```

CursorSource = "h_cobros"
Name = "Cursor2"
Left = 290
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor3"
Left = 10
Top = 140
Width = 90
Height = 90
Alias = "est_cobro"
Database = dbCOBRATEL.dbc
CursorSource = "est_cobro"
Name = "Cursor4"
Alias = "agentes"
Database = dbCOBRATEL.dbc
CursorSource = "agentes"
Name = "CURSOR5"
ScaleMode = 3
Height = 574
Width = 793
DoCreate = .T.
ShowTips = .T.
AutoCenter = .T.
Picture = fondo.bmp
Caption = "COBRATEL® Ajustes a
cobros."
Closable = .T.
Visible = .F.
WindowType = 0
WindowState = 2
LockScreen = .F.
Name = "Form1"
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Name = "Shape4"
layoutsty.Left = 4004
layoutsty.Visible = .F.
layoutsty.TabIndex = 7
layoutsty.Name = "layoutsty"
Label1.Caption = "Ajustes a cobros"
Label1.Left = 100
Label1.Top = 30
Label1.Width = 194
Label1.TabIndex = 8
Label1.ForeColor = 255,255,255
Label1.Name = "Label1"
Shape1.Top = -1
Shape1.Left = 57
Shape1.Name = "Shape1"
Shape2.Top = 0
Shape2.Left = 57
Shape2.Name = "Shape2"
Shape3.Top = 390
Shape3.Left = 43
Shape3.Width = 700
Shape3.Name = "Shape3"
Shape4.Top = 391
Shape4.Left = 43
Shape4.Width = 700
Shape4.Name = "Shape4"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo

```

```

thisform.text3.value=date()
thisform.text5.value=nombre
GO top
regactual=recno()
*thisform.via_cobr1.enabled=.F.
*thisform.est_cobr1.enabled= F.
thisform.fe_re_co1.enabled=.F.

ENDPROC
Caption = "Teléfono"
Left = 99
Top = 154
Visible = .T.
Width = 51
TabIndex = 9
ForeColor = 255,255,255
Name = "LBLTELEF_CO1"
ControlSource = "cobros.telef_co"
Enabled = .F.
Format = "9"
InputMask = "XXXXXXXXXX"
Left = 190
TabIndex = 1
Top = 150
Visible = .T.
Width = 90
Name = "TELEF_CO1"
PROCEDURE Valid
* Validar que no esté vacío
IF !EMPTY(THIS.VALUE) .and.
len(alltrim(THIS.VALUE))=10
RETURN .t.
ELSE
MESSAGEBOX('Este dato no
puede estar vacío ó está incompleto.
¡Verifique por favor!',0+16+0,'Error.')
RETURN .f.
ENDIF

ENDPROC
Caption = "Fecha de cobro"
Height = 13
Left = 99
Top = 178
Visible = .T.
Width = 81
TabIndex = 10
ForeColor = 255,255,255
Name = "LBLFECH_COB1"
Alignment = 3
ControlSource = "cobros.fech_cob"
Enabled = .F.
Left = 190
TabIndex = 2
Top = 174
Visible = .T.
Width = 62
Name = "FECH_COB1"
PROCEDURE Valid
* Validar que no esté vacío
IF !EMPTY(THIS.VALUE) .and.
(this.value<=date())
RETURN .t.
ELSE
MESSAGEBOX('Este dato es
incorrecto. ¡Verifique por
favor!',0+16+0,'Error.')
RETURN .f.
ENDIF

```

```

ENDPROC
Caption = "Via de cobro"
Height = 13
Left = 99
Top = 244
Visible = .T.
Width = 71
TabIndex = 11
ForeColor = 255,255,255
Name = "LBLVIA_COBR1"
Caption = "Estación de cobro"
Height = 13
Left = 99
Top = 268
Visible = .T.
Width = 91
TabIndex = 12
ForeColor = 255,255,255
Name = "LBLEST_COBR1"
Caption = "Importe"
Left = 99
Top = 204
Visible = .T.
Width = 51
TabIndex = 13
ForeColor = 255,255,255
Name = "LBLIMP_COB1"
Alignment = 1
ControlSource = "cobros.imp_cob"
Enabled = .F.
Format = "K"
InputMask = "99,999,999.99"
Left = 190
TabIndex = 3
Top = 200
Visible = .T.
Width = 96
Name = "IMP_COB1"
PROCEDURE Valid
* Validar que no esté vacío
IF !EMPTY(THIS.VALUE) .and.
(this.value>=1.00)
RETURN .t.
ELSE
MESSAGEBOX('Este es incorrecto.
¡Verifique por favor!',0+16+0,'Error.')
RETURN .f.
ENDIF

ENDPROC
Caption = "Fecha de ajuste"
Height = 13
Left = 99
Top = 294
Visible = .T.
Width = 81
TabIndex = 14
ForeColor = 255,255,255
Name = "LBLFE_RE_CO1"
Alignment = 3
ControlSource = "cobros.fe_re_co"
Enabled = .F.
Left = 190
TabIndex = 6
Top = 290
Visible = .T.
Width = 62
Name = "FE_RE_CO1"
Top = 400
Left = 130

```

```

Visible = .T.
TabIndex = 19
oldTalk = OFF
viewkey = ("")
ParentKey =
ViewType = 3
GridAlias =
topfile = .T.
endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
oldreprocess = 0
previewmode = .F.
previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Name = "cmdPrev"
cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros específicos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Visible = .F.
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclose.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = ..archivos
de

```

```

programaldevstudio\vfplwizards\wizb
mps\wznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programaldevstudio\vfplwizards\wizb
mps\wzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
    Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDIF
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
    Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDIF
PROCEDURE cmdTop.Click
*boton primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDIF
PROCEDURE cmdEnd.Click
*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

```

```

ENDPROC
PROCEDURE cmdFind.Click
*boton búsqueda
*thisform.navegar.visible=.F.
*thisform.busca.visible=.T.

ENDIF
ENDIF
ENDIF
#ENDIF
IF FILE(m.cRepName)
    REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
    m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
    IF !EMPTY(m.cRepName)
        IF
FILE(m.cRepName)
            * User
pressed Open button

            REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
            * User
pressed New button
            DO
HOME()+"wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
            ENDIF
        ENDIF
    ENDIF
    * Private data session code
    #IF 0
        IF !EMPTY(cDiffSource)
            USE
(m.cDiffSource) IN 0
        ENDIF
        SET DATASESSION TO
m.nSaveSess
        SELECT (m.cSaveAlias)
    #ENDIF
    IF THISFORM.ShowWindow = 2
        Activate Window
(THISFORM.Name)
    ENDIF
ENDIF
PROCEDURE cmdExit.Click
*botón salir
thisform.Release
DO form cpt0200b &&menu de
procesos especiales
ENDIF
PROCEDURE cmdAdd.Click
*botón agrega
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData
cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")
* Handling for Private data sessions
#IF 0
    IF m.nSaveSess # 1
        SET
DATASESSION TO 1
        SELECT 0
        IF
!EMPTY(m.cSaveData)
            OPEN
DATABASE (m.cSaveData)
        ENDIF
        IF
USED(m.cSaveAlias)
            SELECT (m.cSaveAlias)
            IF
CURSORGETPROP("SourceName")
#m.cSaveSource
                cDiffSource =
CURSORGETPROP("SourceName")
            USE IN (m.cSaveAlias)
            SELECT 0
            ENDIF
        ENDIF
        IF
EMPTY(ALIAS())
            USE
(m.cSaveSource) AGAIN ALIAS
(m.cSaveAlias) SHARED
        ENDIF
        IF
EMPTY(ALIAS())

```

```

=MESSAGEBOX(C_NOO
PEN_LOC)
RETURN
ENDIF
ENDIF
ENDIF
#ENDIF
IF FILE(m.cRepName)
    REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
    m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
    IF !EMPTY(m.cRepName)
        IF
FILE(m.cRepName)
            * User
pressed Open button

            REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
            * User
pressed New button
            DO
HOME()+"wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
            ENDIF
        ENDIF
    ENDIF
    * Private data session code
    #IF 0
        IF !EMPTY(cDiffSource)
            USE
(m.cDiffSource) IN 0
        ENDIF
        SET DATASESSION TO
m.nSaveSess
        SELECT (m.cSaveAlias)
    #ENDIF
    IF THISFORM.ShowWindow = 2
        Activate Window
(THISFORM.Name)
    ENDIF
ENDIF
PROCEDURE cmdExit.Click
*botón salir
thisform.Release
DO form cpt0200b &&menu de
procesos especiales
ENDIF
PROCEDURE cmdAdd.Click
*botón agrega
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36

```



```

* Check if we
have an index expression here.
IF
ATC("(" ,m.cTagName)#0 OR
ATC("+",m.cTagName)#0
    cChildKey = ""
    ELSE
    cChildKey = m.cTagName
    cGridKeyType =
TYPE(m.cChildKey)
    ENDIF
ENDIF
IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;
!CURSORGETPROP("SendUpdates
",m.cPapaAlias)
.T.
    INoSendParentUpdates =
.T.
    ENDIF
    IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;
!CURSORGETPROP("SendUpdates
",m.cChildAlias)
    INoSendChildUpdates = .T.
    ENDIF
    * Check if have two views
and both do not send updates
    IF
m.!NoSendParentUpdates AND
m.!NoSendChildUpdates
        =MESSAGEBOX(C_NOUP
DATE_LOC)
        RETURN
    ENDIF
    SELECT
(THIS.Parent.OldAlias )
    IF
!EMPTY(m.cGridKeyType) AND
m.cGridKeyType #
TYPE(m.cPapaKey)
        IBadChildKey =
.T.
    ENDIF
    DO CASE
CASE
m.!NoSendParentUpdates AND
(m.IBadViewKey OR
m.IBadChildKey)
        =MESSAGEBOX(C_NOUP
DATE_LOC)
        RETURN

```

```

CASE m.IPrimeKey AND
m.!NoSendParentUpdates
        =MESSAGEBOX(C_NOUP
DATE2_LOC)
        RETURN
    CASE
m.!NoSendChildUpdates
&&child view does not
send updates
        IF
MESSAGEBOX(C_NOCHILDUPDAT
E_LOC,MB_Q_YESNO) #
MB_A_YES
            RETURN
        ENDIF
        APPEND BLANK
        CASE m.IBadViewKey
&&view key in not
updatable
            IF
MESSAGEBOX(C_KEYFLDNOUPD
ATE_LOC,MB_Q_YESNO) #
MB_A_YES
                RETURN
            ENDIF
            APPEND BLANK
            CASE m.IBadChildKey
&&bad child key -- different
data type from parent
                IF
MESSAGEBOX(C_BADCHILDKEY_
LOC,MB_Q_YESNO) # MB_A_YES
                    RETURN
                ENDIF
                APPEND BLANK
            OTHERWISE
                * Need to check
if Key field is updatable
                    IUpdatableParentKey = .T.
                    IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND;
                        ATC("(" ,m.cPapaKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0
                            *
                            cPapaKey = ""
                            IUpdatableParentKey = .F.
                            ENDIF
                            * Check for
expression in cPapaKey if so, then
do not update record with keyvalue
                            IF
ATC("(" ,m.cPapaKey)#0 OR
ATC("+",m.cPapaKey)#0
                                cPapaKey = ""
                                ENDIF

```

```

nSaveSess =
SET("DATASESSION")
oAddRec =
CREATE("GridAddRec")
IF
TYPE("m.oAddRec") # "O"
        =MESSAGEBOX(C_NOO
BJ_LOC)
        RETURN
    ENDIF
    oAddRec.ChildPrimaryKey
= m.IPrimeKey
    oAddRec.UpdatableParent
Key = m.IUpdatableParentKey
    oAddRec.NoSendUpdates
= m.NoSendParentUpdates
    oAddRec.KeyField =
m.cPapaKey
    oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey), "")
    oAddRec.runaddform() .
    IF
oAddRec.AddOption =
OPT_CANCEL
        RETURN
    ENDIF
    SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)
    IF
TYPE("oAddRec.KeyValue") = "C"
        oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
        ENDIF
        * Add record to
parent table
        nSaveRec =
RECNO()
        IF
INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
            IF
EMPTY(m.cPapaKey)
                APPEND BLANK IN
(m.cPapaAlias)
            ELSE
                INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
            ENDIF
        ENDIF

```



```
!CURSORGETPROP("SendUpdates",THIS.Parent.GridAlias)
```

```
INoSendChildUpdates=.T.  
ENDIF
```

```
IF
```

```
(m.INoSendChildUpdates AND  
m.INoSendParentUpdates) OR;
```

```
(EMPTY(THIS.Parent.Grid  
Alias) AND  
m.INoSendParentUpdates)
```

```
=MESSAGEBOX(C_NOUP  
DATE_LOC)
```

```
RETURN  
ENDIF
```

```
this.Parent.OldAlias =  
ALIAS() &&save alias in case  
reverting
```

```
this.Parent.oldrec =  
RECNO() &&save record in case  
reverting  
ENDIF
```

```
** Editing record  
this.Parent.EditMode =  
!THIS.Parent.EditMode  
this.Parent.AddMode = .F.  
thisform.LockScreen = .T.  
this.Parent.buttonrefresh()  
this.Parent.navrefresh()  
thisform.LockScreen = .F.
```

```
thisform.Text1.enabled=.F.  
thisform.text2.enabled=.F.  
thisform.text3.enabled=.F.  
thisform.Text4.enabled=.F.  
thisform.text5.enabled=.F.  
*thisform.telef_co1.enabled=.F.  
*thisform.fech_cob1.enabled=.F.  
*thisform.imp_cob1.enabled=.F.  
*thisform.via_cobr1.enabled=.F.  
*thisform.est_cobr1.enabled=.F.  
*thisform.fe_re_co1.enabled=.F.
```

```
ENDPROC
```

```
Enabled = .F.  
Height = 21  
Left = 490  
TabIndex = 20  
Top = 120  
Visible = .F.  
Width = 80  
Name = "Text1"  
Caption = "Usuario"  
Height = 13  
Left = 520  
Top = 14  
Visible = .T.  
Width = 40  
TabIndex = 17  
ForeColor = 255,255,255  
Name = "Embossedlabel1"  
Enabled = .F.
```

```
Height = 21  
Left = 590  
TabIndex = 21  
Top = 120  
Visible = .F.  
Width = 80  
Name = "Text2"  
Caption = "Grupo"  
Height = 13  
Left = 590  
Top = 110  
Visible = .F.  
Width = 40  
TabIndex = 15  
Name = "Embossedlabel2"  
Enabled = .F.  
Height = 21  
Left = 680  
TabIndex = 22  
Top = 40  
Width = 80  
Name = "Text3"  
Caption = "Fecha"  
Height = 13  
Left = 649  
Top = 44  
Visible = .T.  
Width = 40  
TabIndex = 18  
ForeColor = 255,255,255  
Name = "Embossedlabel3"  
Enabled = .F.  
Height = 21  
Left = 680  
TabIndex = 23  
Top = 80  
Visible = .F.  
Width = 80  
Name = "Text4"  
Caption = "clave de usuario"  
Height = 13  
Left = 489  
Top = 110  
Visible = .F.  
Width = 40  
TabIndex = 16  
Name = "Embossedlabel4"  
Enabled = .F.  
Height = 21  
Left = 560  
TabIndex = 24  
Top = 10  
Visible = .T.  
Width = 200  
Name = "Text5"  
AutoSize = .F.  
Top = 160  
Left = 360  
Height = 60  
Width = 80  
WordWrap = .T.  
Picture = checkmrk.ico  
Caption = "Validar cobro"  
Visible = .F.  
Name = "validar"  
PROCEDURE Click  
* Validar que no esté vacío  
IF  
!EMPTY(THISform.TELEF_CO1.VA  
LUE)
```

```
IF  
!EMPTY(THISform.FECH_COB1.VA  
LUE)  
IF  
!EMPTY(THISform.IMP_COB1.VA  
LUE)  
* los datos están completos =>  
hay que verificar si ya fue registrado  
anteriormente  
****busca en COBROS  
myposicion=recno()  
GO top  
COUNT for  
(telef_co=thisform.TELEF_CO1.valu  
e) and  
(fech_cob=thisform.FECH_COB1.val  
ue) and  
(imp_cob=thisform.IMP_COB1.value  
) to hay  
IF hay>1  
MESSAGEBOX('EI COBRO ya  
está registrado en la Base de  
Datos',0+16+0,'Cobro registrado.')
```

```
SELECT cobros  
GO record myposicion  
ELSE  
*****busca en HISTORICO DE  
COBROS  
SELECT h_cobros  
GO top  
COUNT for  
(telef_hc=thisform.TELEF_CO1.valu  
e) and  
(fech_hc=thisform.FECH_COB1.valu  
e) and  
(imp_hc=thisform.IMP_COB1.value)  
to hay
```

```
IF hay>1  
MESSAGEBOX('EI COBRO  
ya está registrado en la Base de  
Datos',0+16+0,'Cobro registrado.')
```

```
SELECT cobros  
GO record myposicion  
MESSAGEBOX('Deberá  
actualizar la información de  
Pagos',0+16+0,'Aviso importante.')
```

```
thisform.via_cobr1.enabled=.T.  
thisform.est_cobr1.enabled=.T.  
*  
thisform.fe_re_co1.enabled=.T.  
thisform.fe_re_co1.value=date()  
thisform.via_cobr1.setfocus  
* this.visible=.F.  
ENDIF  
ENDIF  
ELSE  
MESSAGEBOX('Proporcione el  
Importe del Cobro por  
favor!',0+16+0,'Error.')
```

```

MESSAGEBOX('Proporcione la
Fecha de Cobro por
favor!',0+16+0,'Error.')
thisform.FECH_COB1.setfocus
ENDIF
ELSE
MESSAGEBOX('Proporcione el No.
de Teléfono por
favor!',0+16+0,'Error.')
thisform.TELEF_CO1.setfocus
ENDIF

```

```

ENDPROC
ColumnCount = 2
ColumnWidths = "45,131"
RowSourceType = 6
RowSource =
"est_cobro.num_ec,nomb_ec"
ControlSource = "Cobros.Est_cobr"
FirstElement = 1
Height = 21
Left = 190
NumberOfElements = 0
Style = 2
Top = 263
Width = 141
Name = "Combo2"
ColumnCount = 2
ColumnWidths = "45,140"
RowSourceType = 6
RowSource =
"agentes.vi_co_ag,nom_agen"
ControlSource = "Cobros.Via_cobr"
FirstElement = 1
Height = 21
Left = 190
NumberOfElements = 0
Style = 2
Top = 240
Width = 141
Name = "Combo1"
Picture = note02.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

```

```

*****
C:\COBRATEL\CPT0242.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Alias = "pagos"
BufferModeOverride = 5
ORDER = "wizard_1"
Database = dbCOBRATEL.dbc
CursorSource = "pagos"
Name = "Cursor1"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "procesos"

```

```

Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor2"
ScaleMode = 3
Height = 574
Width = 793
DoCreate = .T.
ShowTips = .T.
AutoCenter = .T.
Picture = fondo.bmp
Caption = "COBRATEL@. Ajustes a
pagos"
Closable = .T.
Visible = .F.
WindowType = 0
WindowState = 2
LockScreen = .F.
Name = "Form1"
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Name = "Shape4"
layoutsty.Left = 4004
layoutsty.Visible = .F.
layoutsty.Name = "layoutsty"
Label1.Caption = "Ajustes a pagos"
Label1.Left = 150
Label1.Top = 30
Label1.Width = 183
Label1.ForeColor = 255,255,255
Label1.Name = "Label1"
Shape1.Top = 369
Shape1.Left = 64
Shape1.Name = "Shape1"
Shape2.Top = 370
Shape2.Left = 64
Shape2.Name = "Shape2"
Shape3.Top = 369
Shape3.Left = 63
Shape3.Name = "Shape3"
Shape4.Top = 370
Shape4.Left = 63
Shape4.Name = "Shape4"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
SELECT pagos
GO top
regactual=recno()
*thisform.via_cobr1.enabled=.F.
*thisform.est_cobr1.enabled=.F.
*thisform.fe_re_co1.enabled=.F.

ENDPROC
Caption = "Via de cobro"
Left = 160
Top = 146
Visible = .T.
Width = 61
ForeColor = 255,255,255
Name = "LBLVIA_CO_PA1"
Alignment = 1
ControlSource = "pagos.via_co_pa"
Enabled = .F.
Left = 290
Top = 140
Visible = .T.

```

```

Width = 48
Name = "VIA_CO_PA1"
Caption = "Fecha inicial de pago"
Height = 13
Left = 160
Top = 170
Visible = .T.
Width = 110
ForeColor = 255,255,255
Name = "LBLFE_IN_PA1"
Alignment = 3
ControlSource = "pagos.fe_in_pa"
Enabled = .F.
Left = 290
Top = 164
Visible = .T.
Width = 62
Name = "FE_IN_PA1"
Caption = "Fecha final de pago"
Height = 13
Left = 160
Top = 194
Visible = .T.
Width = 100
ForeColor = 255,255,255
Name = "LBLFE_FI_PA1"
Alignment = 3
ControlSource = "pagos.fe_fi_pa"
Enabled = .F.
Left = 290
Top = 188
Visible = .T.
Width = 62
Name = "FE_FI_PA1"
Caption = "Importe del cobro
pagado"
Height = 13
Left = 160
Top = 218
Visible = .T.
Width = 130
ForeColor = 255,255,255
Name = "LBLIMP_COB_PA1"
Alignment = 1
ControlSource =
"pagos.imp_cob_pa"
Enabled = .F.
InputMask = "99,999,999.99"
Left = 290
Top = 212
Visible = .T.
Width = 96
Name = "IMP_COB_PA1"
Caption = "Fecha de pago"
Left = 160
Top = 242
Visible = .T.
Width = 61
ForeColor = 255,255,255
Name = "LBLFECH_PAG1"
Alignment = 3
ControlSource = "pagos.fech_pag"
Enabled = .F.
Left = 290
Top = 236
Visible = .T.
Width = 62
Name = "FECH_PAG1"
Caption = "Liquidado"
Left = 160

```

```

Top = 266
Visible = .T.
Width = 61
ForeColor = 255,255,255
Name = "LBLLIQUI_PA1"
ControlSource = "pagos.liqui_pa"
Enabled = .F.
InputMask = "XX"
Left = 290
Top = 260
Visible = .T.
Width = 30
Name = "LIQUI_PA1"
Enabled = .F.
Height = 21
Left = 500
TabIndex = 20
Top = 90
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 528
Top = 22
Visible = .T.
Width = 40
TabIndex = 17
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 600
TabIndex = 21
Top = 90
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 600
Top = 80
Visible = .F.
Width = 40
TabIndex = 15
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 688
TabIndex = 22
Top = 48
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 657
Top = 52
Visible = .T.
Width = 40
TabIndex = 18
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 688
TabIndex = 23
Top = 88
Visible = .F.
Width = 80
Name = "Text4"

```

```

Caption = "clave de usuario"
Height = 13
Left = 499
Top = 80
Visible = .F.
Width = 40
TabIndex = 16
Name = "Embossedlabel4"
Enabled = .F.
Height = 21
Left = 568
TabIndex = 24
Top = 18
Visible = .T.
Width = 200
Name = "Text5"
Top = 380
Left = 120
Visible = .T.
TabIndex = 19
oldTalk = OFF
viewkey = ("")
ParentKey =
ViewType = 3
GridAlias =
topfile = .T.
endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
oldreprocess = 0
previewmode = .F.
previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Name = "cmdPrev"
cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros especificos."
cmdFind.ToolTipText = "Buscar"

```

```

cmdFind.Visible = .F.
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclose.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = ..\archivos
de
programa\devstudio\vfpl\wizards\wizb
mps\wznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Visible = .F.
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programa\devstudio\vfpl\wizards\wizb
mps\wzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC

```

```

PROCEDURE cmdTop.Click
*boton primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click
*boton registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*boton búsqueda
*thisform.navegar.visible=.F.
*thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*boton imprimir
#DEFINE C_MAKEREPO_LOC
"Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC
"Error opening table.
Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC
"Pick report."

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

* Handling for Private data sessions
#IF 0
IF m.nSaveSess # 1
SET
DATASESSION TO 1
SELECT 0
IF
!EMPTY(m.cSaveData)
OPEN
DATABASE (m.cSaveData)
ENDIF
IF
USED(m.cSaveAlias)

SELECT (m.cSaveAlias)

```

```

IF
CURSORGETPROP("SourceName")
#m.cSaveSource

cDiffSource =
CURSORGETPROP("SourceName")

USE IN (m.cSaveAlias)

SELECT 0
ENDIF
ENDIF
IF
EMPTY(ALIAS())
USE
(m.cSaveSource) AGAIN ALIAS
(m.cSaveAlias) SHARED
IF
EMPTY(ALIAS())

=MESSAGEBOX(C_NOO
PEN_LOC)

RETURN
ENDIF
ENDIF
#ENDIF

IF FILE(m.cRepName)
REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
IF !EMPTY(m.cRepName)
IF
FILE(m.cRepName)
* User
pressed Open button

REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
* User
pressed New button
DO
HOME()+"wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
ENDIF
ENDIF
ENDIF

* Private data session code
#IF 0
IF !EMPTY(cDiffSource)
USE
(m.cDiffSource) IN 0
ENDIF
SET DATASESSION TO
m.nSaveSess
SELECT (m.cSaveAlias)
#ENDIF

IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

```

```

ENDPROC
PROCEDURE cmdExit.Click
*boton salir
thisform.Release
DO form cpt0200b &&menu de
procesos especiales
ENDPROC
PROCEDURE cmdAdd.Click
*boton agrega
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
#DEFINE MB_A_YES
6

#DEFINE
C_KEYFLDNOUPDATE_LOC
"The field relating the grid's
view to the parent data source is not
updatable. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
"The fields
relating the parent and child tables
are not the same data type. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
"The child data source is a
view and does not send updates. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_NOOBJ_LOC
"Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
"You cannot add a new
record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
"You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."

LOCAL
oSearchDlog,oAddRec,cChildAlias,c
PapaAlias,i,lPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,lBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
lBadChildKey,lUpdatableParentKey,l
NoSendParentUpdates,lNoSendChil
dUpdates

DO CASE
CASE THIS.Parent.EditMode
&&.....
..... registro en bitácora

```

```

SELECT procesos
APPEND blank
IF
ALLTRIM(THISFORM.Text4.VALUE)
="MODIFICAR"
REPLACE tarea_bp with
"Cambio tabla : Pagos"
ELSE
REPLACE tarea_bp with "Alta
tabla : Pagos"
ENDIF
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT pagos
&&.....
.. cerrar procesos
thisform.Text4.VALUE=SP
ACE(7)
&&*****<<<<<<
<<<<<<

** Code for saving record
this.Parent.updaterows()

CASE EMPTY(THIS.Parent.GridRef)
&& not using
Grid object
thisform.Text4.VALUE="A
GREGAR"
&&*****<<<<<<
<<<<<<

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IF
CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;

!CURSORGETPROP("SendUpdates")
)

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
APPEND BLANK

OTHERWISE

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
iPrimeKey = .F.

&&is child key primary
index

```

```

IF
!EMPTY(THIS.Parent.oDataRelation)
oRel =
EVAL(THIS.Parent.oDataRelation)
ENDIF

DO CASE
CASE TYPE("m.oRel") =
"O"
&&data environment
WITH oRel

cPapaAlias = .ParentAlias

cPapaKey =
.RelationalExpr

cChildAlias = .ChildAlias

cChildKey = .ChildOrder
ENDWITH
oRel = .NULL.

&&reset
CASE
!EMPTY(THIS.Parent.viewkey)
&&using views
cPapaAlias =
ALIAS()
cPapaKey =
THIS.Parent.ParentKey
cChildAlias =
THIS.Parent.GridAlias
cChildKey =
THIS.Parent.viewkey

* Need to check
if Key field is updatable
IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

ATC(", "+m.cChildKey+", ",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0

IBadViewKey = .T.
ENDIF
OTHERWISE
cPapaAlias =
ALIAS()
cPapaKey =
RELATION(1)
cChildAlias =
THIS.Parent.GridAlias
cChildKey =
ORDER(m.cChildAlias)
ENDCASE

* Test to see if both keys
are of the same data type
SELECT
(THIS.Parent.GridAlias)
cGridKeyType = ""

IF
EMPTY(THIS.Parent.viewkey) AND
!EMPTY(m.cChildKey)
&&regular table used

```

```

* Get the Child
Key field since only index tag is here
* if these is an
expression, then default to blank.
cTagName = ""
FOR i = 1 TO
TagCount("")
IF
UPPER(TAG(m.i)) ==
UPPER(m.cChildKey)

cTagName = KEY(m.i)

iPrimeKey =
PRIMARY(m.i)

EXIT
ENDIF
ENDIF

* Check if we
have an index expression here.
IF
ATC(", "+m.cTagName)#0 OR
ATC(", "+m.cTagName)#0

cChildKey = ""
ELSE

cChildKey = m.cTagName

cGridKeyType =
TYPE(m.cChildKey)
ENDIF
ENDIF

IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cPapaAlias)

INoSendParentUpdates =
.T.
ENDIF
IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cChildAlias)

INoSendChildUpdates = .T.
ENDIF

* Check if have two views
and both do not send updates
IF
m.INoSendParentUpdates AND
m.INoSendChildUpdates

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF

```



```

SELECT
(THIS.Parent.OldAlias)
IF
!EMPTY(m.cGridKeyType) AND
m.cGridKeyType #
TYPE(m.cPapaKey)
.IBadChildKey =
.T.
ENDIF

DO CASE
CASE
m.INoSendParentUpdates AND
(m.IBadViewKey OR
m.IBadChildKey)

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
CASE m.IPrimeKey AND
m.INoSendParentUpdates

=MESSAGEBOX(C_NOUP
DATE2_LOC)
RETURN
CASE
m.INoSendChildUpdates
&&child view does not
send updates
IF
MESSAGEBOX(C_NOCHILDUPDAT
E_LOC,MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadViewKey
&&view key in not
updatable
IF
MESSAGEBOX(C_KEYFLDNOUPD
ATE_LOC,MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadChildKey
&&bad child key -- different
data type from parent
IF
MESSAGEBOX(C_BADCHILDKEY_
LOC,MB_Q_YESNO) # MB_A_YES

RETURN
ENDIF
APPEND BLANK
OTHERWISE
* Need to check
if Key field is updatable

IUpdatableParentKey = .T.
IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND;

```

```

ATC(", "+m.cPapaKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0
*
cPapaKey = ""

IUpdatableParentKey = .F.
ENDIF

* Check for
expression in cPapaKey if so, then
do not update record with keyvalue
IF
ATC("{",m.cPapaKey)#0 OR
ATC("+",m.cPapaKey)#0

cPapaKey = ""
ENDIF

nSaveSess =
SET("DATASESSION")
oAddRec =
CREATE("GridAddRec")
IF
TYPE("m.oAddRec") # "O"

=MESSAGEBOX(C_NOO
BJ_LOC)
RETURN
ENDIF

oAddRec.ChildPrimaryKey
= m.IPrimeKey

oAddRec.UpdatableParent
Key = m.IUpdatableParentKey

oAddRec.NoSendUpdates
= m.INoSendParentUpdates

oAddRec.KeyField =
m.cPapaKey

oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey),"")

oAddRec.runaddform()
IF
oAddRec.AddOption =
OPT_CANCEL

RETURN
ENDIF

SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)

IF
TYPE("oAddRec.KeyValue") = "C"

oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
ENDIF

```

```

* Add record to
parent table
nSaveRec =
RECNO()
IF
INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
IF
EMPTY(m.cPapaKey)

APPEND BLANK IN
(m.cPapaAlias)
ELSE

INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
ENDIF
ENDIF
nSaveRec2 =
RECNO()

* Add child
record
IF
INLIST(oAddRec.AddOption,OPT_A
DD_CHILD,OPT_ADD_BOTH)
* Need
to check
GO
m.nSaveRec
IF
EMPTY(m.cChildKey) OR
TYPE("m.cChildKey")#TYPE('oAddR
ec.KeyValue')

APPEND BLANK IN
(m.cChildAlias)
ELSE

INSERT INTO
(m.cChildAlias) ((m.cChildKey))
VALUES(oAddRec.KeyValue)
ENDIF
GO
m.nSaveRec2
ENDIF
ENDCASE
ENDCASE

this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode =
THIS.Parent.EditMode
this.Parent.topfile = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
*Thisform.telef_co1.enabled=.F.
*Thisform.fech_cob1.enabled=.F.

```

```

*Thisform.imp_cob1.enabled=.F.
*Thisform.via_cobr1.enabled=.F.
**Thisform.est_cobr1.enabled=.F.
*Thisform.fe_re_co1.enabled=.F.
*thisform.validar.visible=!thisform.validar.visible

```

```

ENDPROC
PROCEDURE cmdDelete.Click
*botón borra

```

```

* Tipo de cuadro de diálogo
* 4 + 32 + 256
* 4 = Botones SI y No
* 32 = Icono de PREGUNTA
* 256 = el segundo botón es el de default

```

```

* Respuestas
* 6 = SI
* 7 = NO

```

```

IF MESSAGEBOX('¿ Desea eliminar este registro?', 4+32+256, 'Confirme por favor...') = 6

```

```

Delete
IF
THIS.Parent.updaterows()
&&success

```

```

* Success
IF !EOF()
Skip 1
ENDIF
IF EOF() AND
IBOF()
Skip -1
ENDIF
regactual=recno()
ENDIF
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

```

```

&&.....
..... registro en bitácora
SELECT procesos
APPEND blank
REPLACE tarea_bp with "Baja tabla : Pagos"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT pagos
&&.....
.. cerrar procesos
ENDIF

```

```

IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

```

```

ENDPROC
PROCEDURE cmdEdit.Click
*botón edita
#DEFINE C_NOUPDATE_LOC
"You cannot edit because the view(s) selected does not send updates."

```

```

LOCAL
!NoSendParentUpdates,!NoSendChildUpdates
LOCAL aTablesUsed,nTablesUsed,i

```

```

** Reverting record
IF THIS.Parent.EditMode
thisform.Text4.VALUE=SP
ACE(7)
&&*****<<<<<<
<<<<<<

```

```

IF THIS.Parent.usedataenv
SELECT
(THIS.Parent.OldAlias)
IF
CURSORGETPROP("Buffering")>1

```

```

=TableRevert(.T.)
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)

```

```

SELECT
(THIS.Parent.GridAlias)
IF
CURSORGETPROP("Buffering")>1

```

```

=TableRevert(.T.)
ENDIF
ENDIF

```

```

ELSE
DIMENSION
aTablesUsed[1]
m.nTablesUsed
= AUUSED(aTablesUsed)
FOR i = 1 TO
m.nTablesUsed
IF
CURSORGETPROP("Buffering",aTablesUsed[m,i,1])>1

```

```

=TableRevert(.T.,aTablesUsed[m,i,1])
ENDIF
ENDIF
ENDIF

```

```

* Go back to original place
SELECT
(THIS.Parent.OldAlias)
IF RECCOUNT() <
THIS.Parent.oldrec &&added record at EOF()

```

```

GO TOP
ELSE
GO
THIS.Parent.oldrec
ENDIF
ELSE
thisform.Text4.VALUE="MODIFICAR"
&&*****<<<<<<
<<<<<<

```

```

* Check to see if view allows updates
IF
CURSORGETPROP("SourceType")#

```

```

3 AND
!CURSORGETPROP("offline") AND;
!CURSORGETPROP("SendUpdates")

```

```

.NoSendParentUpdates = .T.
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias) AND;

```

```

CURSORGETPROP("SourceType",THIS.Parent.GridAlias)#3 AND;

```

```

!CURSORGETPROP("offline",THIS.Parent.GridAlias) AND;

```

```

!CURSORGETPROP("SendUpdates",THIS.Parent.GridAlias)

```

```

.NoSendChildUpdates = .T.
ENDIF

```

```

IF
(m.NoSendChildUpdates AND m.NoSendParentUpdates) OR;

```

```

(EMPTY(THIS.Parent.GridAlias) AND m.NoSendParentUpdates)

```

```

=MESSAGEBOX(C_NOUP DATE_LOC)

```

```

RETURN
ENDIF

```

```

this.Parent.OldAlias = ALIAS() &&save alias in case reverting
this.Parent.oldrec = RECNO() &&save record in case reverting
ENDIF

```

```

** Editing record
this.Parent.EditMode = !THIS.Parent.EditMode
this.Parent.AddMode = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

```

```

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
*Thisform.telef_co1.enabled=.F.
*Thisform.fech_cob1.enabled=.F.
*Thisform.imp_cob1.enabled=.F.
*Thisform.via_cobr1.enabled=.F.
*Thisform.est_cobr1.enabled=.F.
*Thisform.fe_re_co1.enabled=.F.

```

```

ENDPROC

```

```

Picture = note03.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
*****
C:\COBRATEL\CPT0300.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú de
generación de reportes."
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Generación de reportes"
Height = 30
Left = 100
Top = 40
Width = 240
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 660
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lit.ico
Caption = "Menú principal"
StatusBarText = "Regresar al menú
principal."
TabIndex = 9
Name = "Command10"
PROCEDURE Click
* Boton Salir
thisform.release
DO form menuprin
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 120
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = graph11.ico
Caption = ""
StatusBarText = "Control de pagos"
TabIndex = 3

```

```

ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton captura pagos de agentes
thisform.release
DO form cpt0300b
ENDPROC
Top = 200
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = mail14.ico
Caption = ""
StatusBarText = "Reporte diario de
caja."
TabIndex = 6
ToolTipText = ""
Name = "Command4"
PROCEDURE LostFocus
thisform.Label4.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label4.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton reporte diario de caja
thisform.release
DO form cpt0320
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
BackStyle = 0
Caption = "Control de pagos"
Height = 21
Left = 150
Top = 140
Width = 130
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Reporte diario de caja"
Height = 21
Left = 150
Top = 220
Width = 160
ForeColor = 255,255,255
Name = "Label4"
Top = 120
Left = 360

```

```

Height = 60
Width = 81
WordWrap = .T.
Picture = graph03.ico
Caption = ""
StatusBarText = "Reporte mensual."
TabIndex = 3
ToolTipText = ""
Name = "Command5"
PROCEDURE Click
* Boton reporte mensual
thisform.release
DO form cpt0330
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE GotFocus
thisform.Label5.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label5.forecolor=rgb(255,25
5,255)
ENDPROC
Top = 200
Left = 360
Height = 60
Width = 81
WordWrap = .T.
Picture = folder04.ico
Caption = ""
StatusBarText = "Reporte semanal
por via de cobro."
TabIndex = 4
ToolTipText = ""
Name = "Command6"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton reporte semanal por via de
cobro
thisform.release
DO form cpt0340
ENDPROC
PROCEDURE GotFocus
thisform.Label6.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label6.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Reporte estadístico"
Height = 21
Left = 450
Top = 140
Width = 180
ForeColor = 255,255,255
Name = "Label5"
BackStyle = 0

```

```

Caption = "Reporte por vía de cobro"
Height = 21
Left = 450
Top = 220
Width = 200
ForeColor = 255,255,255
Name = "Label6"
Picture = net06.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

```

```

C:\COBRATEL\CPT0300B.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú de
Control de pagos."
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Control de pagos"
Height = 30
Left = 100
Top = 40
Width = 180
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 660
Height = 60
Width = 82
WordWrap = .T.
Picture = anw09lt.ico
Caption = "Generación de reportes"
StatusBarText = "Regresar al menú
principal."
TabIndex = 9
Name = "Command10"
PROCEDURE Click
* Boton regresar al menu de ajustes
thisform.release
DO form cpt0300
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC

```

```

PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 140
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = keybrd02.ico
Caption = ""
StatusBarText = "Captura pagos de
agentes"
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton ajuste a la tabla de cobros
thisform.release
DO form cpt0311
ENDPROC
Top = 220
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = key04.ico
Caption = ""
StatusBarText = "Diario de agentes."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton ajuste a la tabla de pagos
thisform.release
DO form cpt0312
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
BackStyle = 0
Caption = "Captura pagos de
agentes"
Height = 21

```

```

Left = 290
Top = 160
Width = 180
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Diario de agentes"
Height = 21
Left = 290
Top = 240
Width = 180
ForeColor = 255,255,255
Name = "Label2"
Top = 300
Left = 200
Height = 60
Width = 81
WordWrap = .T.
Picture = eye.ico
Caption = ""
StatusBarText = "Verificación de
pagos de clientes."
TabIndex = 4
ToolTipText = ""
Name = "Command3"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton ajuste a la tabla de pagos
thisform.release
DO form cpt0313
ENDPROC
PROCEDURE GotFocus
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Verificación de pagos de
clientes"
Height = 21
Left = 290
Top = 320
Width = 180
ForeColor = 255,255,255
Name = "Label3"
Picture = graph11.ico
Stretch = 1
BackStyle = 1
Height = 42
Left = 40
Top = 20
Width = 42
Name = "Image1"

```

```

C:\COBRATEL\CPT0311.SCX
Left = 1
Top = 220
Width = 520
Height = 200

```

Name = "Dataenvironment"
 Alias = "pagos"
 BufferModeOverride = 5
 ORDER = "wizard_1"
 Database = dbCOBRATEL.dbc
 CursorSource = "pagos"
 Name = "Cursor1"
 Alias = "agentes"
 Database = dbCOBRATEL.dbc
 CursorSource = "agentes"
 Name = "Cursor2"
 Left = 290
 Top = 20
 Width = 90
 Height = 90
 Alias = "procesos"
 Database = dbCOBRATEL.dbc
 CursorSource = "procesos"
 Name = "Cursor3"
 ScaleMode = 3
 Height = 574
 Width = 793
 DoCreate = .T.
 ShowTips = .T.
 AutoCenter = .T.
 Picture = fondo.bmp
 Caption = "COBRATEL®. Captura
 de pagos de agentes."
 Closable = .T.
 Visible = .F.
 WindowType = 0
 WindowState = 2
 LockScreen = .F.
 Name = "Form1"
 layoutsty.Shape1.Name = "Shape1"
 layoutsty.Shape2.Name = "Shape2"
 layoutsty.Shape3.Name = "Shape3"
 layoutsty.Shape4.Name = "Shape4"
 layoutsty.Left = 4004
 layoutsty.Visible = F
 layoutsty.Name = "layoutsty"
 Label1.Caption = "Captura de pagos
 de agentes."
 Label1.Left = 130
 Label1.Top = 20
 Label1.Width = 343
 Label1.ForeColor = 255,255,255
 Label1.Name = "Label1"
 Shape1.Top = 379
 Shape1.Left = 103
 Shape1.Name = "Shape1"
 Shape2.Top = 380
 Shape2.Left = 103
 Shape2.Name = "Shape2"
 Shape3.Top = 379
 Shape3.Left = 103
 Shape3.Name = "Shape3"
 Shape4.Top = 380
 Shape4.Left = 103
 Shape4.Name = "Shape4"
 PROCEDURE Activate
 * Al iniciar...
 thisform.Text1.value=usuario
 thisform.text2.value=grupo
 thisform.text3.value=date()
 thisform.text5.value=nombre
 SELECT pagos
 GO top
 regactual=recno()

ENDPROC
 Caption = "Vía de cobro"
 Left = 159
 Top = 104
 Visible = .T.
 Width = 61
 ForeColor = 255,255,255
 Name = "LBLVIA_CO_PA1"
 Caption = "Fecha inicial de pago"
 Height = 13
 Left = 159
 Top = 128
 Visible = .T.
 Width = 111
 ForeColor = 255,255,255
 Name = "LBLFE_IN_PA1"
 Alignment = 3
 ControlSource = "pagos.fe_in_pa"
 Enabled = .F.
 Left = 300
 Top = 124
 Visible = .T.
 Width = 62
 Name = "FE_IN_PA1"
 Caption = "Fecha final de pago"
 Height = 13
 Left = 159
 Top = 152
 Visible = .T.
 Width = 121
 ForeColor = 255,255,255
 Name = "LBLFE_FI_PA1"
 Alignment = 3
 ControlSource = "pagos.fe_fi_pa"
 Enabled = .F.
 Left = 300
 Top = 148
 Visible = .T.
 Width = 62
 Name = "FE_FI_PA1"
 Caption = "Importe del cobro
 pagado"
 Height = 13
 Left = 159
 Top = 176
 Visible = .T.
 Width = 131
 ForeColor = 255,255,255
 Name = "LBLIMP_COB_PA1"
 Alignment = 1
 ControlSource =
 "pagos.imp_cob_pa"
 Enabled = .F.
 InputMask = "99,999,999.99"
 Left = 300
 Top = 172
 Visible = .T.
 Width = 96
 Name = "IMP_COB_PA1"
 Caption = "Fecha de pago"
 Height = 13
 Left = 159
 Top = 200
 Visible = .T.
 Width = 101
 ForeColor = 255,255,255
 Name = "LBLFECH_PAG1"
 Alignment = 3
 ControlSource = "pagos.fech_pag"
 Enabled = .F.

Left = 300
 Top = 196
 Visible = .T.
 Width = 62
 Name = "FECH_PAG1"
 Caption = "Liquidado"
 Left = 159
 Top = 224
 Visible = .T.
 Width = 61
 ForeColor = 255,255,255
 Name = "LBLLIQUI_PA1"
 ControlSource = "pagos.liqui_pa"
 Enabled = .F.
 InputMask = "XX"
 Left = 300
 Top = 220
 Visible = .T.
 Width = 30
 Name = "LIQUI_PA1"
 ColumnCount = 2
 ColumnWidths = "60,105"
 RowSourceType = 6
 RowSource =
 "agentes.vi_co_ag.nom_agen"
 ControlSource = "Pagos.Via_co_pa"
 FirstElement = 1
 Height = 21
 Left = 300
 NumberOfElements = 0
 Style = 2
 Top = 100
 Width = 100
 Name = "Combo1"
 Top = 390
 Left = 140
 Visible = .T.
 TabIndex = 18
 oldTalk = OFF
 viewkey = ("")
 ParentKey =
 ViewType = 3
 GridAlias =
 topfile = .T.
 endfile = .T.
 AddMode = .F.
 nworkarea = 3
 oldSetDelete = OFF
 oldreprocess = 0
 previewmode = .F.
 previewinit = .F.
 usedataenv = .F.
 EditMode = .F.
 GridRef =
 oldSetFields = OFF
 oldMultiLocks = ON
 oldrefresh = 0
 oldBuffering = 1
 Name = "Navegar"
 cmdPrev.Picture = wzback.bmp
 cmdPrev.Enabled = .T.
 cmdPrev.StatusBarText = "Ir al
 registro anterior."
 cmdPrev.ToolTipText = "Anterior"
 cmdPrev.Visible = .T.
 cmdPrev.Name = "cmdPrev"
 cmdNext.Picture = wznext.bmp
 cmdNext.Enabled = .T.
 cmdNext.StatusBarText = "Ir al
 siguiente registro."

```

cmdNext.ToolTipText = "Siguiente"
cmdNext.Visible = .T.
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Visible = .T.
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Visible = .T.
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros especificos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Visible = .F.
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Visible = .F.
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclos.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = ..\archivos
de
programa\devstudio\vp\wizards\wizb
mps\wznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Visible = .F.
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programa\devstudio\vp\wizards\wizb
mps\wzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click

```

```

*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
    Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
    Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdTop.Click
*boton primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click
*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*boton búsqueda
thisform.Navegar.visible=.F.
thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*botón imprime
#DEFINE C_MAKEREPO_LOC
"Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC
"Error opening table.
Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC
"Pick report:"

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")

```

```

cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

* Handling for Private data sessions
#IF 0
    IF m.nSaveSess # 1
        SET
        DATASESSION TO 1
        SELECT 0
        IF
        !EMPTY(m.cSaveData)
            OPEN
        DATABASE (m.cSaveData)
        ENDIF
        IF
        USED(m.cSaveAlias)
            SELECT (m.cSaveAlias)
            IF
            CURSORGETPROP("SourceName")
            #m.cSaveSource
                cDiffSource =
                CURSORGETPROP("SourceName")
                USE IN (m.cSaveAlias)

            SELECT 0
            ENDIF
            ENDIF
            IF
            EMPTY(ALIAS())
                USE
            (m.cSaveSource) AGAIN ALIAS
            (m.cSaveAlias) SHARED
            IF
            EMPTY(ALIAS())
                =MESSAGEBOX(C_NOO
                PEN_LOC)

                RETURN
            ENDIF
            ENDIF
            #ENDIF
            IF FILE(m.cRepName)
                REPORT FORM
                (m.cRepName) PREVIEW NOWAIT
            ELSE
                m.cRepName =
                GETFILE("frx",C_GETFILEPROMPT
                _LOC,"",1)
                IF !EMPTY(m.cRepName)
                    IF
                    FILE(m.cRepName)
                        * User
                        pressed Open button

                    REPORT FORM
                    (m.cRepName) PREVIEW NOWAIT
                ELSE
                    * User
                    pressed New button
                
```

```

DO
HOME()+ "wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
ENDIF
ENDIF
ENDIF
* Private data session code
#IF 0
IF !EMPTY(cDiffSource)
USE
(m.cDiffSource) IN 0
ENDIF
SET DATASESSION TO
m.nSaveSess
SELECT (m.cSaveAlias)
#ENDIF
IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF
ENDPROC
PROCEDURE cmdExit.Click
*botón salir
thisform.Release
DO form cpt0300b &&menu de
control de pagos
ENDPROC
PROCEDURE cmdAdd.Click
*botón agrega
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
#DEFINE MB_A_YES
6
#DEFINE
C_KEYFLDNOUPDATE_LOC
"The field relating the grid's
view to the parent data source is not
updatable. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
"The fields
relating the parent and child tables
are not the same data type. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
"The child data source is a
view and does not send updates. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_NOOBJ_LOC
"Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
"You cannot add a new

```

```

record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
"You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."
LOCAL
oSearchDlog,oAddRec,cChildAlias,c
PapaAlias,i,lPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,lBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
lBadChildKey,lUpdatableParentKey,l
NoSendParentUpdates,lNoSendChil
dUpdates
DO CASE
CASE THIS.Parent.EditMode
&&.....
..... registro en bitácora
SELECT procesos
APPEND blank
IF
ALLTRIM(THISFORM.Text4.VALUE)
="MODIFICAR"
REPLACE tarea_bp with
"modifica pago de agentes"
ELSE
REPLACE tarea_bp with
"Captura pago de agentes"
ENDIF
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT pagos
&&.....
. cerrar procesos
thisform.Text4.VALUE=SP
ACE(7)
&&.....<<<<<<
<<<<<<<
** Code for saving record
this.Parent.updaterows()
CASE EMPTY(THIS.Parent.GridRef)
&& not using
Grid object
thisform.Text4.VALUE="A
GREGAR"
&&.....<<<<<<
<<<<<<<
** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IF
CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;

```

```

!CURSORGETPROP("SendUpdates")
=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
APPEND BLANK
OTHERWISE
** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
lPrimeKey = .F.
&&is child key primary
index
IF
!EMPTY(THIS.Parent.oDataRelation)
oRel =
EVAL(THIS.Parent.oDataRelation)
ENDIF
DO CASE
CASE TYPE("m.oRel") =
"O"
&&data environment
WITH oRel
cPapaAlias = .ParentAlias
cPapaKey =
.RelationalExpr
cChildAlias = .ChildAlias
cChildKey = .ChildOrder
ENDWITH
oRel = .NULL.
&&reset
CASE
!EMPTY(THIS.Parent.viewkey)
&&using views
cPapaAlias =
ALIAS()
cPapaKey =
THIS.Parent.ParentKey
cChildAlias =
THIS.Parent.GridAlias
cChildKey =
THIS.Parent.viewkey
* Need to check
if Key field is updatable
IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;
ATC(", "+m.cChildKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0

```

```

        IBadViewKey = .T.
            ENDIF
        OTHERWISE
            cPapaAlias =
ALIAS()
            cPapaKey =
RELATION(1)
            cChildAlias =
THIS.Parent.GridAlias
            cChildKey =
ORDER(m.cChildAlias)
        ENDCASE

        * Test to see if both keys
are of the same data type
        SELECT
        (THIS.Parent.GridAlias)
        cGridKeyType = ""

        IF
        EMPTY(THIS.Parent.viewkey) AND
!EMPTY(m.cChildKey)
            &&regular table used
            * Get the Child
Key field since only index tag is here
            * if these is an
expression, then default to blank.
            cTagName = ""
            FOR i = 1 TO
TagCount("")
                IF
                UPPER(TAG(m.i)) ==
                UPPER(m.cChildKey)

                    cTagName = KEY(m.i)

                IPrimeKey =
PRIMARY(m.i)

                EXIT
            ENDIF
        ENDFOR

        * Check if we
have an index expression here.
        IF
        ATC("",m.cTagName)#0 OR
        ATC("+",m.cTagName)#0

            cChildKey = ""
        ELSE

            cChildKey = m.cTagName

            cGridKeyType =
TYPE(m.cChildKey)
        ENDIF
    ENDIF

    IF
    CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cPapaAlias)

```

```

        INoSendParentUpdates =
.T.
            ENDIF
        CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cChildAlias)

        INoSendChildUpdates = .T.
            ENDIF

        * Check if have two views
and both do not send updates
        IF
        m.INoSendParentUpdates AND
        m.INoSendChildUpdates

            =MESSAGEBOX(C_NOUP
DATE_LOC)
            RETURN
        ENDIF

        SELECT
        (THIS.Parent.OldAlias )
        IF
        !EMPTY(m.cGridKeyType) AND
        m.cGridKeyType #
        TYPE(m.cPapaKey)

            IBadChildKey =
.T.
                ENDIF

            DO CASE
            CASE
            m.INoSendParentUpdates AND
            (m.IBadViewKey OR
            m.IBadChildKey)

                =MESSAGEBOX(C_NOUP
DATE_LOC)
                RETURN
            CASE m.IPrimeKey AND
            m.INoSendParentUpdates

                =MESSAGEBOX(C_NOUP
DATE2_LOC)
                RETURN
            CASE
            m.INoSendChildUpdates
            &&child view does not
send updates
                IF
                MESSAGEBOX(C_NOCHILDUPDAT
E_LOC,MB_Q_YESNO) #
                MB_A_YES

                    RETURN
                ENDIF
                APPEND BLANK
            CASE m.IBadViewKey
            &&view key in not
updatable
                IF
                MESSAGEBOX(C_KEYFLDNOUPD

```

```

ATE_LOC,MB_Q_YESNO) #
MB_A_YES

        RETURN
            ENDIF
        APPEND BLANK
        CASE m.IBadChildKey
        &&bad child key -- different
data type from parent
            IF
            MESSAGEBOX(C_BADCHILDKEY_
LOC,MB_Q_YESNO) # MB_A_YES

                RETURN
            ENDIF
        APPEND BLANK
        OTHERWISE
            * Need to check
if Key field is updatable

            IUpdatableParentKey = .T.
            IF
            CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND;

                ATC(""+m.cPapaKey+"",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0

                cPapaKey = ""

            IUpdatableParentKey = .F.
            ENDIF

            * Check for
expression in cPapaKey if so, then
do not update record with keyvalue
            IF
            ATC("(",m.cPapaKey)#0 OR
            ATC("+",m.cPapaKey)#0

                cPapaKey = ""
            ENDIF

            nSaveSess =
SET("DATASESSION")
            oAddRec =
CREATE("GridAddRec")
            IF
            TYPE("m.oAddRec") # "O"

                =MESSAGEBOX(C_NOO
BJ_LOC)

                RETURN
            ENDIF

            oAddRec.ChildPrimaryKey
= m.IPrimeKey

            oAddRec.UpdatableParent
Key = m.IUpdatableParentKey

            oAddRec.NoSendUpdates
= m.INoSendParentUpdates

            oAddRec.KeyField =
m.cPapaKey

```



```

                oAddRec.KeyValue =
                IIF(!EMPTY(m.cPapaKey),
                EVAL(m.cPapaKey), "")

                oAddRec.runaddform()
                IF
                oAddRec.AddOption =
                OPT_CANCEL

                RETURN
                ENDIF

                SET
                DATASESSION TO nSaveSess
                SELECT
                (THIS.Parent.OldAlias)

                IF
                TYPE("oAddRec.KeyValue") = "C"

                oAddRec.KeyValue =
                TRIM(oAddRec.KeyValue)
                ENDIF

                * Add record to
                parent table
                nSaveRec =
                RECNO()
                IF
                INLIST(oAddRec.AddOption,OPT_A
                DD_PARENT,OPT_ADD_BOTH)
                IF
                EMPTY(m.cPapaKey)

                APPEND BLANK IN
                (m.cPapaAlias)

                ELSE

                INSERT INTO
                (m.cPapaAlias)
                ((oAddRec.KeyField))
                VALUES(oAddRec.KeyValue)
                ENDIF
                ENDIF
                nSaveRec2 =
                RECNO()

                * Add child
                record
                IF
                INLIST(oAddRec.AddOption,OPT_A
                DD_CHILD,OPT_ADD_BOTH)
                * Need
                to check
                GO
                m.nSaveRec

                IF
                EMPTY(m.cChildKey) OR
                TYPE("m.cChildKey")#TYPE('oAddR
                ec.KeyValue')

                APPEND BLANK IN
                (m.cChildAlias)

                ELSE

                INSERT INTO
                (m.cChildAlias) ((m.cChildKey))
                VALUES(oAddRec.KeyValue)
                ENDIF

                m.nSaveRec2
                ENDCASE
                ENDCASE

                this.Parent.EditMode =
                !THIS.Parent.EditMode
                this.Parent.AddMode =
                THIS.Parent.EditMode
                this.Parent.topfile = .F.
                thisform.LockScreen = .T.
                this.Parent.buttonrefresh()
                this.Parent.navrefresh()
                thisform.LockScreen = .F.

                thisform.Text1.enabled=.F.
                thisform.text2.enabled=.F.
                thisform.text3.enabled=.F.
                thisform.Text4.enabled=.F.
                thisform.text5.enabled=.F.

                *****
                *thisform.COMBO1.ENABLED=!THI
                S.Parent.EditMode
                thisform.FE_IN_PA1.value=date()
                thisform.FE_FI_PA1.value=date()
                thisform.IMP_COB_PA1.value=0.00
                thisform.FECH_PAG1.value=date()

                ENDCASE
                PROCEDURE cmdDelete.Click
                *botón borra

                * Tipo de cuadro de diálogo
                * 4 + 32 + 256
                * 4 = Botones SI y No
                * 32 = Icono de PREGUNTA
                * 256 = el segundo botón es el de
                default

                * Respuestas
                * 6 = SI
                * 7 = NO

                IF MESSAGEBOX('¿ Desea eliminar
                este registro?', 4+32+256, 'Confirme
                por favor...') = 6
                Delete
                IF
                THIS.Parent.updaterows()
                &&success

                * Success
                IF !EOF()
                Skip 1
                ENDIF
                IF EOF() AND
                !BOF()
                Skip -1
                ENDIF
                regactual=recno()
                ENDCASE
                thisform.LockScreen = .T.
                this.Parent.buttonrefresh()
                GO
                this.Parent.navrefresh()
                thisform.LockScreen = .F.

                &&.....
                .. registro en bitácora
                SELECT procesos
                APPEND blank
                REPLACE tarea_bp with "Eliminar
                de pagos de agentes"
                REPLACE usua_bp with usuario
                REPLACE fecha_bp with date()
                REPLACE hora_bp with time()
                SELECT pagos
                &&.....
                .. cerrar procesos
                ENDCASE

                IF THISFORM.ShowWindow = 2
                Activate Window
                (THISFORM.Name)
                ENDCASE

                ENDCASE
                PROCEDURE cmdEdit.Click
                *botón edita
                #DEFINE C_NOUPDATE_LOC
                "You cannot edit because
                the view(s) selected does not send
                updates."

                LOCAL
                INoSendParentUpdates,INoSendChil
                dUpdates
                LOCAL aTablesUsed,nTablesUsed,i

                ** Reverting record
                IF THIS.Parent.EditMode
                thisform.Text4.VALUE=SP
                ACE(7)
                &&*****<<<<<<<<
                <<<<<<<<

                IF THIS.Parent.usedataenv
                SELECT
                (THIS.Parent.OldAlias)
                IF
                CURSORGETPROP("Buffering")>1
                =TableRevert(.T.)
                ENDCASE
                IF
                !EMPTY(THIS.Parent.GridAlias)

                SELECT
                (THIS.Parent.GridAlias)
                IF
                CURSORGETPROP("Buffering")>1
                =TableRevert(.T.)
                ENDCASE
                ENDCASE
                ELSE
                DIMENSION
                aTablesUsed[1]
                m.nTablesUsed
                = AUUSED(aTablesUsed)
                FOR i = 1 TO
                m.nTablesUsed
                IF
                CURSORGETPROP("Buffering",aTab
                lesUsed[m.i,1])>1

```

```

        =TableRevert(.T.,aTablesU
sed[m,i,1])
                ENDIF
        ENDFOR
    ENDIF

    * Go back to original place
    SELECT
    (THIS.Parent.OldAlias)
    IF RECCOUNT() <
    THIS.Parent.oldrec &&added record
    at EOF()
        GO TOP
    ELSE
        GO
    THIS.Parent.oldrec
    ENDIF
    ELSE
        thisform.Text4.VALUE="M
ODIFICAR"
&&*****<<<<<<
<<<<<<<

        * Check to see if view
        allows updates
        IF
        CURSORGETPROP("SourceType")#
        3 AND
        !CURSORGETPROP("offline") AND;

            !CURSORGETPROP("Sen
dUpdates")

            INoSendParentUpdates =
.T.
        ENDIF
        IF
        !EMPTY(THIS.Parent.GridAlias)
        AND ;

        CURSORGETPROP("SourceType",T
HIS.Parent.GridAlias)#3 AND ;

        !CURSORGETPROP("offline",THIS.
Parent.GridAlias) AND ;

        !CURSORGETPROP("SendUpdates
",THIS.Parent.GridAlias)

            INoSendChildUpdates= .T.
        ENDIF

        IF
        (m.INoSendChildUpdates AND
m.INoSendParentUpdates) OR;

            (EMPTY(THIS.Parent.Grid
Alias) AND
m.INoSendParentUpdates)

            =MESSAGEBOX(C_NOUP
DATE_LOC)
        RETURN
    ENDIF

    this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting

```

```

        this.Parent.oldrec =
RECNO() &&save record in case
reverting
    ENDIF

    ** Editing record
    this.Parent.EditMode =
!THIS.Parent.EditMode
    this.Parent.AddMode = .F.
    thisform.LockScreen = .T.
    this.Parent.buttonrefresh()
    this.Parent.navrefresh()
    thisform.LockScreen = .F.

    thisform.Text1.enabled=.F.
    thisform.text2.enabled=.F.
    thisform.text3.enabled=.F.
    thisform.Text4.enabled=.F.
    thisform.Text5.enabled=.F.

    thisform.Combo1.enabled=!THIS.PA
RENT.EditMode
    thisform.FE_IN_PA1.enabled=.F.
    thisform.FE_FI_PA1.enabled=.F.
    thisform.IMP_COB_PA1.enabled=.F.
    thisform.FECH_PAG1.enabled=.F.
    thisform.LIQUI_PA1.enabled=.F.

    ENDPROC
    Enabled = .F.
    Height = 21
    Left = 490
    TabIndex = 23
    Top = 90

    Visible = .F.
    Width = 80
    Name = "Text1"
    Caption = "Usuario"
    Height = 13
    Left = 520
    Top = 13
    Visible = .T.
    Width = 40
    TabIndex = 15
    ForeColor = 255,255,255
    Name = "Embossedlabel1"
    Enabled = .F.
    Height = 21
    Left = 580
    TabIndex = 24
    Top = 90
    Visible = .F.
    Width = 80
    Name = "Text2"
    Caption = "Grupo"
    Height = 13
    Left = 580
    Top = 80
    Visible = .F.
    Width = 40
    TabIndex = 14
    Name = "Embossedlabel2"
    Enabled = .F.
    Height = 21
    Left = 680
    TabIndex = 25
    Top = 40
    Width = 80

```

```

    Name = "Text3"
    Caption = "Fecha"
    Height = 13
    Left = 647
    Top = 43
    Visible = .T.
    Width = 40
    TabIndex = 16
    ForeColor = 255,255,255
    Name = "Embossedlabel3"
    Enabled = .F.
    Height = 21
    Left = 680
    TabIndex = 26
    Top = 90
    Visible = .F.
    Width = 80
    Name = "Text4"
    Caption = "clave de usuario"
    Height = 13
    Left = 489
    Top = 80
    Visible = .F.
    Width = 40
    TabIndex = 14
    Name = "Embossedlabel4"
    Enabled = .F.
    Height = 21
    Left = 560
    TabIndex = 26
    Top = 10
    Visible = .T.
    Width = 200
    Name = "Text5"
    ColumnCount = 3
    ColumnWidths = "67,71,88"
    RowSourceType = 6
    RowSource =
"pagos.via_co_pa,fech_pag,imp_cob
_pa"
    FirstElement = 1
    Height = 21
    Left = 120
    NumberOfElements = 0
    Style = 2
    Top = 100
    Visible = .F.
    Width = 20
    Name = "busca"
    PROCEDURE InteractiveChange
    * Al encontrar el elemento deseado
    GO record recno()
    this.visible=.F.
    thisform.Navegar.visible=.t.
    thisform.refresh
    ENDPROC
    Picture = keybrd02.ico
    Stretch = 1
    Height = 40
    Left = 40
    Top = 20
    Width = 40
    Name = "Image1"

    *****
    C:\COBRATELICPT0312.SCX
    Left = 76
    Top = 253
    Width = 520

```

```

Height = 200
Name = "Dataenvironment"
Left = 29
Top = 12
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor2"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor1"
Alias = "agentes"
Database = dbCOBRATEL.dbc
CursorSource = "agentes"
Name = "CURSOR3"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Diario de agente."
WindowState = 2
LockScreen = .F.
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
thisform.Text7.value=date()
thisform.grid1.VISIBLE=.f.

SELECT cobros

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Diario de agente"
Height = 30
Left = 100
Top = 40
Width = 370
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = F
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = T.
Picture = arw09lt.ico
Caption = "Control de pagos"
StatusBarText = "Regresar al menú de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar

```

```

this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0300b
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Seleccione el agente"
Height = 21
Left = 60
Top = 100
Width = 120
ForeColor = 255,255,255
Name = "Label1"
Top = 260
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Depuración de la bitácora de procesos."
TabIndex = 9
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Genera reporte

SELEC cobros
SET filter to
via_cobr=thisform.Text6.value .and.
fech_cob=thisform.Text7.value
COUNT to hay
IF hay <>0
&& si hay datos
SUM imp_cob to mytotal
thisform.grid1.VISIBLE=.t.
thisform.Text8.value= mytotal
ELSE
&& no hay datos
MESSAGEBOX('No hay datos solicitados. ¡Verifique por favor!',0+16+0,'Error.')
ENDIF
thisform.refresh

ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus

```

```

* Señalar
thisform.Label3.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Despliega consulta"
Height = 21
Left = 60
Top = 320
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 70
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 519
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 58
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21

```

```

Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
Picture = debu1.ico
Stretch = 1
BackStyle = 0
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
ColumnCount = 2
ColumnWidths = "105,69"
RowSourceType = 6
RowSource =
"agentes.nom_agen,vi_co_ag"
FirstElement = 1
Height = 21
Left = 60
NumberOfElements = 0
Style = 2
Top = 120
Width = 160
Name = "Combo1"
PROCEDURE InteractiveChange
* Al seleccionar...
thisform.Text6.value=agentes.vi_co_
ag

```

```

ENDPROC
PROCEDURE Click
thisform.grid1.VISIBLE=.f.
ENDPROC
Enabled = .F.
Height = 21
Left = 60
TabIndex = 26
Top = 140
Visible = .F.
Width = 80
Name = "Text6"
ColumnCount = 3
DeleteMark = .F.
GridLines = 3
GridLineWidth = 1
HeaderHeight = 20
Height = 241
Left = 330
RecordMark = .F.
RecordSource = "cobros"
ScrollBars = 2
Top = 120
Width = 250
ForeColor = 0,0,0
BackColor = 255,255,255
GridLineColor = 192,192,192
Name = "Grid1"
Column1.FontBold = .F.
Column1.FontName = "Arial"

```

```

Column1.FontSize = 8
Column1.Alignment = 0
Column1.ControlSource =
"cobros.telef_co"
Column1.Width = 79
Column1.DynamicBackColor = ""
Column1.ForeColor = 0,0,0
Column1.BackColor = 255,255,255
Column1.Name = "Column1"
Column2.FontBold = .F.
Column2.FontName = "Arial"
Column2.FontSize = 8
Column2.Alignment = 0
Column2.ControlSource =
"cobros.est_cobr"
Column2.Width = 71
Column2.DynamicBackColor = ""
Column2.ForeColor = 0,0,0
Column2.BackColor = 255,255,255
Column2.Name = "Column2"
Column3.FontBold = .F.
Column3.FontName = "Arial"
Column3.FontSize = 8
Column3.Alignment = 0
Column3.ControlSource =
"cobros.imp_cob"
Column3.Width = 77
Column3.DynamicBackColor = ""
Column3.ForeColor = 0,0,0
Column3.BackColor = 255,255,255
Column3.Name = "Column3"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Telefono"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Estación"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Importe"

```

```

ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
Enabled = .T.
Height = 21
Left = 60
TabIndex = 26
Top = 180
Visible = .T.
Width = 80
Name = "Text7"
PROCEDURE Valid
* Validar fecha correcta
IF this.value>date()
RETURN .F.
ELSE
RETURN .T.
ENDIF

```

```

ENDPROC
BackStyle = 0
Caption = "Fecha"
Height = 21
Left = 60
Top = 160
Width = 40
ForeColor = 255,255,255
Name = "Label2"
Enabled = .T.
Format = "$"
Height = 21
Left = 480
TabIndex = 26
Top = 380
Visible = .T.
Width = 80
Name = "Text8"
BackStyle = 0
Caption = "Total"
Height = 21
Left = 420
Top = 380
Width = 40
ForeColor = 255,255,255
Name = "Label4"

```

```

*****
C:\COBRATEL\CPT0313.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 10
Top = 20
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"

```

```

Name = "Cursor1"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor2"
ScaleMode = 3
Height = 574
Width = 793
DoCreate = .T.
ShowTips = .T.
AutoCenter = .T.
Picture = fondo.bmp
Caption = "COBRATEL®.
Verificación de pagos de clientes."
Closable = .T.
Visible = .F.
WindowType = 0
WindowState = 2
LockScreen = .F.
Name = "Form1"
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Name = "Shape4"
layoutsty.Left = 4004
layoutsty.Visible = .F.
layoutsty.TabIndex = 4
layoutsty.Name = "layoutsty"
Label1.Caption = "Verificación de
pagos de clientes"
Label1.Left = 100
Label1.Top = 30
Label1.Width = 381
Label1.TabIndex = 5
Label1.ForeColor = 255,255,255
Label1.Name = "Label1"
Shape1.Top = 379
Shape1.Left = 103
Shape1.Name = "Shape1"
Shape2.Top = 380
Shape2.Left = 103
Shape2.Name = "Shape2"
Shape3.Top = 379
Shape3.Left = 103
Shape3.Name = "Shape3"
Shape4.Top = 380
Shape4.Left = 103
Shape4.Name = "Shape4"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

SELECT cobros
GO top
regactual=recno()

thisform.Text7.Enabled=.T.
thisform.Text7.setfocus
thisform.grid1.visible=.F.

ENDPROC
Top = 390
Left = 140

```

```

Visible = .T.
TabIndex = 3
oldTalk = OFF
viewkey = ("")
ParentKey = .T.
ViewType = 3
GridAlias =
topfile = .T.
endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
oldreprocess = 0
previewmode = .F.
previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Visible = .F.
cmdPrev.Name = "cmdPrev"
cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Visible = .F.
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Visible = .F.
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Visible = .F.
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros específicos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Visible = .F.
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Visible = .T.
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclos.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."

```

```

cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = ..\archivos
de
programa\devstudio\wfp\wizards\wizb
mps\wznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Visible = .F.
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Visible = .F.
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programa\devstudio\wfp\wizards\wizb
mps\wzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Visible = .F.
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdTop.Click
*botón primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click

```

```

*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*boton búsqueda
*thisform.navegar.visible=.F.
*thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*botón imprime
#DEFINE C_MAKEREPO_LOC "Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC
"Error opening table.
Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC
"Pick report:"

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

* Handling for Private data sessions
#IF 0
IF m.nSaveSess # 1
SET
DATASESSION TO 1
SELECT 0
IF
!EMPTY(m.cSaveData)
OPEN
DATABASE (m.cSaveData)
ENDIF
IF
USED(m.cSaveAlias)

SELECT (m.cSaveAlias)
IF
CURSORGETPROP("SourceName")
#m.cSaveSource

cDiffSource =
CURSORGETPROP("SourceName")

USE IN (m.cSaveAlias)

SELECT 0
ENDIF

```

```

ENDIF
IF
EMPTY(ALIAS())
USE
(m.cSaveSource) AGAIN ALIAS
(m.cSaveAlias) SHARED
IF
EMPTY(ALIAS())
=MESSAGEBOX(C_NOO
PEN_LOC)
RETURN
ENDIF
ENDIF
#ENDIF
IF FILE(m.cRepName)
REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
IF !EMPTY(m.cRepName)
IF
FILE(m.cRepName)
* User
pressed Open button

REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
* User
pressed New button
DO
HOME()+"\wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
ENDIF
ENDIF
ENDIF

* Private data session code
#IF 0
IF !EMPTY(cDiffSource)
USE
(m.cDiffSource) IN 0
ENDIF
SET DATASESSION TO
m.nSaveSess
SELECT (m.cSaveAlias)
#ENDIF

IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
PROCEDURE cmdExit.Click
*botón salir
thisform.Release
DO form cpt0300b &&menu de
control de pagos
ENDPROC
PROCEDURE cmdAdd.Click
*botón agrega
#DEFINE OPT_CANCEL
0

```

```

#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
#DEFINE MB_A_YES
6
#DEFINE
C_KEYFLDNOUPDATE_LOC
"The field relating the grid's
view to the parent data source is not
updatable. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
"The fields
relating the parent and child tables
are not the same data type. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
"The child data source is a
view and does not send updates. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_NOOBJ_LOC
"Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
"You cannot add a new
record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
"You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."

LOCAL
oSearchDlg,oAddRec,cChildAlias,c
PapaAlias,i,iPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,iBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
iBadChildKey,iUpdatableParentKey,i
NoSendParentUpdates,iNoSendChil
dUpdates

DO CASE
CASE THIS.Parent.EditMode
&&.....
registro en bitácora
SELECT procesos
APPEND blank
IF
ALLTRIM(THISFORM.Text4.VALUE)
="MODIFICAR"
REPLACE tarea_bp with
"modifica pago de agentes"
ELSE
REPLACE tarea_bp with
"Captura pago de agentes"
ENDIF

```

```

REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT pagos
&&.....
.. cerrar procesos
   thisform.Text4.VALUE=SP
ACE(7)
&&.....<<<<<<
<<<<<<<

** Code for saving record
this.Parent.updaterows()

CASE EMPTY(THIS.Parent.GridRef)
  && not using
Grid object
  thisform.Text4.VALUE="A
GREGAR"
&&.....<<<<<<
<<<<<<<

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
  this.Parent.oldrec =
RECNO() &&save record in case
reverting
  IF
  !CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;
  !CURSORGETPROP("Send
Updates")
  =MESSAGEBOX(C_NOUP
DATE_LOC)
  RETURN
  ENDIF
  APPEND BLANK

OTHERWISE

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
  this.Parent.oldrec =
RECNO() &&save record in case
reverting
  !PrimeKey = .F.

  &&is child key primary
index
  IF
  !EMPTY(THIS.Parent.oDataRelation)
  oRel =
EVAL(THIS.Parent.oDataRelation)
  ENDIF

  DO CASE
  CASE TYPE("m.oRel") =
"O"
    &&data environment
    WITH oRel

```

```

cPapaAlias = .ParentAlias

cPapaKey =
.RelationalExpr

cChildAlias = .ChildAlias

cChildKey = .ChildOrder
ENDWITH
oRel = .NULL.

&&reset
CASE
!EMPTY(THIS.Parent.viewkey)
  &&using views
  cPapaAlias =
ALIAS()
  cPapaKey =
THIS.Parent.ParentKey
  cChildAlias =
THIS.Parent.GridAlias
  cChildKey =
THIS.Parent.viewkey

  * Need to check
  if Key field is updatable
  IF
  CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;
    ATC(" "+m.cChildKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0
  !BadViewKey = .T.
  ENDIF
  OTHERWISE
  cPapaAlias =
ALIAS()
  cPapaKey =
RELATION(1)
  cChildAlias =
THIS.Parent.GridAlias
  cChildKey =
ORDER(m.cChildAlias)
  ENDCASE

  * Test to see if both keys
  are of the same data type
  SELECT
  (THIS.Parent.GridAlias)
  cGridKeyType = ""

  IF
  EMPTY(THIS.Parent.viewkey) AND
!EMPTY(m.cChildKey)
    &&regular table used
    * Get the Child
    Key field since only index tag is here
    * if these is an
    expression, then default to blank.
    cTagName = ""
    FOR i = 1 TO
    TagCount("")
      IF
      UPPER(TAG(m.i)) ==
      UPPER(m.cChildKey)

```

```

cTagName = KEY(m.i)

!PrimeKey =
PRIMARY(m.i)

EXIT
ENDIF

* Check if we
have an index expression here.
IF
ATC("("+m.cTagName)#0 OR
ATC(" "+m.cTagName)#0
  cChildKey = ""
  ELSE
  cChildKey = m.cTagName

  cGridKeyType =
TYPE(m.cChildKey)
  ENDIF

ENDIF

IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;
  !CURSORGETPROP("SendUpdates
",m.cPapaAlias)

  !NoSendParentUpdates =
.T.
  ENDIF
  IF
  CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;
  !CURSORGETPROP("SendUpdates
",m.cChildAlias)

  !NoSendChildUpdates = .T.
  ENDIF

  * Check if have two views
  and both do not send updates
  IF
  m.!NoSendParentUpdates AND
  m.!NoSendChildUpdates

  =MESSAGEBOX(C_NOUP
DATE_LOC)
  RETURN
  ENDIF

  SELECT
  (THIS.Parent.OldAlias )
  IF
  !EMPTY(m.cGridKeyType) AND
  m.cGridKeyType #
TYPE(m.cPapaKey)
  !BadChildKey =
.T.
  ENDIF

```

```

DO CASE
CASE
m.INoSendParentUpdates AND
(m.IBadViewKey OR
m.IBadChildKey)

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
CASE m.IPrimeKey AND
m.INoSendParentUpdates

=MESSAGEBOX(C_NOUP
DATE2_LOC)
RETURN
CASE
m.INoSendChildUpdates
&&child view does not
send updates
IF
MESSAGEBOX(C_NOCHILDUPDAT
E_LOC,MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadViewKey
&&view key in not
updatable
IF
MESSAGEBOX(C_KEYFLDNOUPD
ATE_LOC,MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadChildKey
&&bad child key -- different
data type from parent
IF
MESSAGEBOX(C_BADCHILDKEY_
LOC,MB_Q_YESNO) # MB_A_YES

RETURN
ENDIF
APPEND BLANK
OTHERWISE
* Need to check
if Key field is updatable

IUpdatableParentKey = .T.
IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND:

ATC("."+m.cPapaKey+"",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0

cPapaKey = ""

IUpdatableParentKey = .F.
ENDIF

```

```

* Check for
expression in cPapaKey if so, then
do not update record with keyvalue
IF
ATC("."+m.cPapaKey)#0 OR
ATC("."+m.cPapaKey)#0

cPapaKey = ""
ENDIF

nSaveSess =
SET("DATASESSION")
oAddRec =
CREATE("GridAddRec")
IF
TYPE("m.oAddRec") # "O"

=MESSAGEBOX(C_NOO
BJ_LOC)

RETURN
ENDIF

oAddRec.ChildPrimaryKey
= m.IPrimeKey

oAddRec.UpdatableParent
Key = m.IUpdatableParentKey

oAddRec.NoSendUpdates
= m.INoSendParentUpdates

oAddRec.KeyField =
m.cPapaKey

oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey),"")

oAddRec.runaddform()
IF
oAddRec.AddOption =
OPT_CANCEL

RETURN
ENDIF

SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)

IF
TYPE("oAddRec.KeyValue") = "C"

oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
ENDIF

* Add record to
parent table

nSaveRec =
RECNO()
IF
INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
IF
EMPTY(m.cPapaKey)

```

```

APPEND BLANK IN
(m.cPapaAlias)
ELSE

INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
ENDIF
ENDIF
nSaveRec2 =
RECNO()

* Add child
record
IF
INLIST(oAddRec.AddOption,OPT_A
DD_CHILD,OPT_ADD_BOTH)
* Need
to check
GO
m.nSaveRec
IF
EMPTY(m.cChildKey) OR
TYPE("m.cChildKey")#TYPE('oAddR
ec.KeyValue')

APPEND BLANK IN
(m.cChildAlias)
ELSE

INSERT INTO
(m.cChildAlias) ((m.cChildKey))
VALUES(oAddRec.KeyValue)
ENDIF
GO
m.nSaveRec2
ENDIF
ENDCASE

this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode =
THIS.Parent.EditMode
this.Parent.topfile = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.

*****
thisform.FE_IN_PA1.value=date()
thisform.FE_FL_PA1.value=date()
thisform.IMP_COB_PA1.value=0.00
thisform.FECH_PAG1.value=date()

```



```

ENDPROC
PROCEDURE cmdDelete.Click
*botón borra

* Tipo de cuadro de diálogo
* 4 + 32 + 256
* 4 = Botones SI y No
* 32 = Icono de PREGUNTA
* 256 = el segundo botón es el de
default

* Respuestas
* 6 = SI
* 7 = NO

IF MESSAGEBOX('¿ Desea eliminar
este registro?', 4+32+256, 'Confirme
por favor...') = 6
Delete
IF
THIS.Parent.updaterows()
&&success
    * Success
    IF !EOF()
        Skip 1
    ENDIF
    IF EOF() AND
        Skip -1
    ENDIF
    regactual=recno()
    ENDIF
    thisform.LockScreen = .T.
    this.Parent.buttonrefresh()
    this.Parent.navrefresh()
    thisform.LockScreen = .F.
&&.....
..... registro en bitácora
SELECT procesos
APPEND blank
REPLACE tarea_bp with "Eliminar
de pagos de agentes"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT pagos
&&.....
.. cerrar procesos
ENDIF

IF THISFORM.ShowWindow = 2
    Activate Window
    (THISFORM.Name)
ENDIF

ENDPROC
PROCEDURE cmdEdit.Click
*botón edita
#DEFINE C_NOUPDATE_LOC
    "You cannot edit because
the view(s) selected does not send
updates."

LOCAL
!NoSendParentUpdates,!NoSendChildUpdates
LOCAL aTablesUsed,nTablesUsed,i

** Reverting record
IF THIS.Parent.EditMode

```

```

thisform.Text4.VALUE=SP
ACE(7)
&&*****<<<<<<
<<<<<<<

IF THIS.Parent.usedataenv
SELECT
(THIS.Parent.OldAlias)
IF
CURSORGETPROP("Buffering")>1

=TableRevert(.T.)
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)

SELECT
(THIS.Parent.GridAlias)
IF
CURSORGETPROP("Buffering")>1

=TableRevert(.T.)
ENDIF
ELSE
DIMENSION
aTablesUsed[1]
m.nTablesUsed
= AUSED(aTablesUsed)
FOR i = 1 TO
m.nTablesUsed
IF
CURSORGETPROP("Buffering",aTablesUsed[m,i,1])>1

=TableRevert(.T.,aTablesUsed[m,i,1])
ENDIF
ENDIF

* Go back to original place
SELECT
(THIS.Parent.OldAlias)
IF RECCOUNT() <
THIS.Parent.oldrec &&added record
at EOF()
GO TOP
ELSE
GO
THIS.Parent.oldrec
ENDIF
ELSE
thisform.Text4.VALUE="MODIFICAR"
&&*****<<<<<<
<<<<<<<

* Check to see if view
allows updates
IF
CURSORGETPROP("SourceType")#
3 AND
!CURSORGETPROP("offline") AND;

!CURSORGETPROP("SendUpdates")

!NoSendParentUpdates =
.T.

```

```

ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)
AND ;

CURSORGETPROP("SourceType",THIS.Parent.GridAlias)#3 AND ;

!CURSORGETPROP("offline",THIS.Parent.GridAlias) AND ;

!CURSORGETPROP("SendUpdates",THIS.Parent.GridAlias)

!NoSendChildUpdates= .T.
ENDIF

IF
(m.!NoSendChildUpdates AND
m.!NoSendParentUpdates) OR;

(EMPTY(THIS.Parent.GridAlias) AND
m.!NoSendParentUpdates)

=MESSAGEBOX(C_NOUPDATE_LOC,
DATE_LOC)
RETURN
ENDIF

this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
ENDIF

** Editing record
this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.

ENDPROC
Enabled = .F.
Height = 21
Left = 490
TabIndex = 11
Top = 90
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40

```

TabIndex = 9
 ForeColor = 255,255,255
 Name = "Embossedlabel1"
 Enabled = .F.
 Height = 21
 Left = 580
 TabIndex = 12
 Top = 90
 Visible = .F.
 Width = 80
 Name = "Text2"
 Caption = "Grupo"
 Height = 13
 Left = 580
 Top = 80
 Visible = .F.
 Width = 40
 TabIndex = 7
 Name = "Embossedlabel2"
 Enabled = .F.
 Height = 21
 Left = 680
 TabIndex = 13
 Top = 40
 Width = 80
 Name = "Text3"
 Caption = "Fecha"
 Height = 13
 Left = 647
 Top = 43
 Visible = .T.
 Width = 40
 TabIndex = 10
 ForeColor = 255,255,255
 Name = "Embossedlabel3"
 Enabled = .F.
 Height = 21
 Left = 680
 TabIndex = 15
 Top = 90
 Visible = .F.
 Width = 80
 Name = "Text4"
 Caption = "clave de usuario"
 Height = 13
 Left = 489
 Top = 80
 Visible = .F.
 Width = 40
 TabIndex = 6
 Name = "Embossedlabel4"
 Enabled = .F.
 Height = 21
 Left = 560
 TabIndex = 14
 Top = 10
 Visible = .T.
 Width = 200
 Name = "Text5"
 Picture = keybrd02.ico
 Stretch = 1
 Height = 40
 Left = 40
 Top = 20
 Width = 40
 Name = "Image1"
 AutoSize = .T.
 Caption = "Teléfono por consultar"
 Height = 16
 Left = 80

Top = 120
 Visible = .T.
 Width = 111
 TabIndex = 8
 ForeColor = 255,255,255
 Name = "Embossedlabel5"
 Top = 220
 Left = 100
 Height = 60
 Width = 81
 WordWrap = .T.
 Picture = checkmrk.ico
 Caption = ""
 StatusBarText = "Depuración de la bitácora de procesos."
 TabIndex = 2
 Name = "Command1"
 PROCEDURE LostFocus
 * Señalar
 thisform.Label3.forecolor=rgb(255,255,255)
 ENDPROC
 PROCEDURE GotFocus
 * Señalar
 thisform.Label3.forecolor=rgb(255,255,128)
 ENDPROC
 PROCEDURE Click
 * Genera reporte

 SELECT cobros
 SET filter to
 telef_co=alltrim(thisform.Text7.value)
 COUNT to hay
 IF hay <>0
 thisform.grid1.visible=t.
 thisform.refresh
 &&.....
 .. registra en bitácora
 SELECT procesos
 APPEND blank
 REPLACE tarea_bp with "Verifica pago tel:
 "+alltrim(thisform.Text7.value)
 REPLACE usua_bp with usuario
 REPLACE fecha_bp with date()
 REPLACE hora_bp with time()
 SELECT cobros
 &&.....
 .. cerrar procesos

 ELSE
 && no hay datos
 MESSAGEBOX("El teléfono solicitado no está registrado.
 ¡Verifique por favor! 0+16+0,'Atención")
 ENDIF
 thisform.refresh

 ENDPROC
 PROCEDURE MouseMove
 LPARAMETERS nButton, nShift, nXCoord, nYCoord
 this.setfocus
 ENDPROC
 BackStyle = 0
 Caption = "Despliega consulta"

Height = 21
 Left = 87
 Top = 284
 Width = 120
 TabIndex = 16
 ForeColor = 255,255,255
 Name = "Label3"
 ColumnCount = 6
 DeleteMark = .F.
 GridLines = 3
 GridLineWidth = 1
 HeaderHeight = 20
 Height = 200
 Left = 260
 RecordMark = .F.
 RecordSource = "cobros"
 ScrollBars = 2
 TabIndex = 17
 Top = 140
 Width = 421
 ForeColor = 0,0,0
 BackColor = 255,255,255
 GridLineColor = 192,192,192
 Name = "Grid1"
 Column1.FontBold = .F.
 Column1.FontName = "Arial"
 Column1.FontSize = 8
 Column1.Alignment = 0
 Column1.ControlSource = "cobros.fech_cob"
 Column1.Width = 79
 Column1.DynamicBackColor = ""
 Column1.ForeColor = 0,0,0
 Column1.BackColor = 255,255,255
 Column1.Name = "Column1"
 Column2.FontBold = .F.
 Column2.FontName = "Arial"
 Column2.FontSize = 8
 Column2.Alignment = 0
 Column2.ControlSource = "cobros.via_cobr"
 Column2.Width = 41
 Column2.DynamicBackColor = ""
 Column2.ForeColor = 0,0,0
 Column2.BackColor = 255,255,255
 Column2.Name = "Column2"
 Column3.FontBold = .F.
 Column3.FontName = "Arial"
 Column3.FontSize = 8
 Column3.Alignment = 0
 Column3.ControlSource = "cobros.est_cobr"
 Column3.Width = 66
 Column3.DynamicBackColor = ""
 Column3.ForeColor = 0,0,0
 Column3.BackColor = 255,255,255
 Column3.Name = "Column3"
 Column4.FontBold = .F.
 Column4.FontName = "Arial"
 Column4.FontSize = 8
 Column4.Alignment = 0
 Column4.ControlSource = "cobros.imp_cobr"
 Column4.Width = 60
 Column4.DynamicBackColor = ""
 Column4.ForeColor = 0,0,0
 Column4.BackColor = 255,255,255
 Column4.Name = "Column4"
 Column5.FontBold = .F.
 Column5.FontName = "Arial"

```

Column5.FontSize = 8
Column5.Alignment = 0
Column5.ControlSource =
"cobros.fe_re_co"
Column5.Width = 76
Column5.DynamicBackColor = ""
Column5.ForeColor = 0,0,0
Column5.BackColor = 255,255,255
Column5.Name = "Column5"
Column6.FontBold = .F.
Column6.FontName = "Arial"
Column6.FontSize = 8
Column6.Alignment = 0
Column6.ControlSource =
"cobros.telef_co"
Column6.Width = 68
Column6.DynamicBackColor = ""
Column6.ForeColor = 0,0,0
Column6.BackColor = 255,255,255
Column6.Name = "Column6"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Fecha cobro"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Via"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Estación"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255

```

```

Name = "Text1"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Importe"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Recepción"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
FontBold = .F.
FontName = "Arial"
FontSize = 10
Alignment = 2
Caption = "Teléfono"
ForeColor = 0,0,0
BackColor = 192,192,192
Name = "Header1"
FontBold = .F.
FontName = "Arial"
FontSize = 8
Alignment = 0
BorderStyle = 0
Margin = 0
ForeColor = 0,0,0
BackColor = 255,255,255
Name = "Text1"
Height = 21
Left = 80
TabIndex = 1
Top = 140
Width = 121
BackColor = 255,255,255
DisabledBackColor = 192,192,192
Name = "Text7"
PROCEDURE GotFocus
*Al tomar el foco quitar el resumen
thisform.grid1.visible=.F.

ENDPROC
PROCEDURE Valid
* validar que tenga 10 digitos
IF len(trim(this.value))<>10
MESSAGEBOX('El número de
teléfono solicitado debe tener 10

```

```

digitos, incluyendo la clave de larga
distancia. ¡Verifique por
favor!,0+16+0,'Atención')
RETURN .f.
ELSE
RETURN .t.
ENDIF

ENDPROC
*****
C:\COBRATEL\CPT0320.SCX
Left = 76
Top = 253
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 29
Top = 12
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor2"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL@. Reporte
diario de caja."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.text11.value=date()

SELECT cobros
ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Reporte diario de caja"
Height = 30
Left = 100
Top = 40
Width = 370
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.

```

```

Picture = arw09ft.ico
Caption = "Menú de depuración"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0300
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "¿ De qué fecha desea el
reporte ?"
Height = 21
Left = 210
Top = 120
Width = 190
ForeColor = 255,255,255
Name = "Label1"
Height = 21
Left = 240
StatusBarText = "Fecha inicial de la
consulta"
Top = 140
Width = 101
Name = "Text11"
PROCEDURE LostFocus
* Igualar fecha final
thisform.Label11.forecolor=rgb(0,0,0)

ENDPROC
PROCEDURE GotFocus
thisform.Label11.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Valid
* Validar que la fecha no sea mayor
a la del sistema
IF this.value > date() then
MESSAGEBOX('La fecha que ha
proporcionado no es válida.
¡Verifique por favor!',0+16+0,'Error.')
RETURN .f.
ELSE
RETURN .t.
ENDIF

ENDPROC
Top = 200
Left = 240
Height = 80
Width = 61
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""

```

```

StatusBarText = "Depuración de la
bitácora de procesos."
TabIndex = 9
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton depurar
*if MESSAGEBOX('¿ Ya realizó el
respaldo de su Base de Datos?',
4+32+256, 'Confirme por favor...') = 6
then
&& si realizó el respaldo

COUNT for fech_cob =
thisform.text11.value TO hay
IF hay <>0 then
thisform.progreso.visible=.t.
FOR i=1 to 100 STEP 2
thisform.progreso.value=i
WAIT " timeout .1
NEXT i
thisform.progreso.visible=.f.
REPORT form diacaja for fech_cob
= thisform.text11.value preview
* messagebox('Bitácora de procesos
depurada',0+64+0,'Operación
finalizada')
SELECT procesos && Registra en
bitácora de procesos, la operación
efectuada
APPEND blank
REPLACE tarea_bp with "Reporte
diario de
caja"+dtoc(thisform.text11.value)
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT cobros && cierra
procesos.....
..... cerrar procesos
*else
&& no realizó el respaldo
* messagebox('Realice el respaldo
de su Base de Datos en
[Utilerias/Respaldo de
Archivos]',0+64+0,'Depuración
cancelada')
*endif
ELSE
MESSAGEBOX('No hay cobros en
esta fecha. Verifique por
favor!',0+64+0,'Atención')
ENDIF

ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0

```

```

Caption = "Generar reporte"
Height = 21
Left = 330
Top = 220
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 70
Visible = .F.
Width = 60
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 519
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 58
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.

```

```

Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
Picture = depu1.ico
Stretch = 1
BackStyle = 0
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 316
Left = 126
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"

*****
C:\COBRATEL\CPT0330.SCX
Left = 76
Top = 253
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 29
Top = 12
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor2"
Left = 154
Top = 14
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor1"
Left = 280
Top = 16
Width = 90
Height = 90
Alias = "meses"
Database = dbCOBRATEL.dbc
CursorSource = "meses"
Name = "CURSOR3"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = T.
Picture = fondo.bmp
Caption = "COBRATEL®. Reporte estadístico."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=usuario

```

```

thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

*thisform.mes.value=year(date())
thisform.anyo.value=year(date())

SELECT cobros
ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Reporte estadístico"
Height = 30
Left = 100
Top = 40
Width = 370
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lit.ico
Caption = "Menú de depuración"
StatusBarText = "Regresar al menú de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0300
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "¿ De qué fecha desea el reporte ?"
Height = 21
Left = 230
Top = 100
Width = 240
ForeColor = 255,255,255
Name = "Label1"
Top = 220
Left = 260
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Depuración de la bitácora de procesos."
TabIndex = 9
Name = "Command1"

```

```

PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,255,255)
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton depurar
*if MESSAGEBOX('¿ Ya realizó el respaldo de su Base de Datos?', 4+32+256, 'Confirme por favor...') = 6 then
&& si realizó el respaldo

COUNT for month(fech_cob) =
thisform.Text6.value .and.
year(fech_cob) = thisform.anyo.value
TO hay
IF hay <>0 then
thisform.progreso.visible=.t.
FOR i=1 to 100 STEP 2
thisform.progreso.value=i
WAIT " timeout .1
NEXT i
thisform.progreso.visible=.f.
REPORT form estadístico for
month(fech_cob) =
thisform.Text6.value .and.
year(fech_cob) = thisform.anyo.value
preview
* messagebox('Bitácora de procesos depurada',0+64+0,'Operación finalizada')
SELECT procesos && Registra en bitácora de procesos, la operación efectuada
APPEND blank
REPLACE tarea_bp with "Reporte estadístico: "+
alltrim(STR(thisform.anyo.value))+
"/"+
alltrim(STR(thisform.Text6.value))
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT cobros && cierra
procesos.....
..... cerrar procesos
*else
&& no realizó el respaldo
* messagebox('Realice el respaldo de su Base de Datos en [Utilerías/Respaldo de Archivos]',0+64+0,'Depuración cancelada')
*endif
ELSE
MESSAGEBOX('No hay cobros en esta fecha. Verifique por favor!',0+64+0,'Atención')
ENDIF

ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord

```

```

this.setfocus
ENDPROC
BackStyle = 0
Caption = "Generar reporte"
Height = 21
Left = 350
Top = 240
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 70
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 519
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 58
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560

```

```

TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
Picture = depu1.ico
Stretch = 1
BackStyle = 0
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 336
Left = 126
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"
ColumnCount = 0
ColumnWidths = ""
RowSourceType = 6
RowSource = "meses.mes"
FirstElement = 1
Height = 21
Left = 220
NumberOfElements = 0
Style = 2
Top = 160
Width = 141
Name = "Mes"
PROCEDURE InteractiveChange
* Cuando escoja un mes...
thisform.Text6.value=meses.numme
s
ENDPROC
Height = 21
KeyboardHighValue = (year(date()))
KeyboardLowValue = 1996
Left = 400
SpinnerHighValue = (year(date()))
SpinnerLowValue = 1996.00
Top = 160
Width = 81
Name = "Anyo"
Enabled = .F.
Height = 21
Left = 220
TabIndex = 25
Top = 180
Visible = .F.
Width = 80
Name = "Text6"
BackStyle = 0
Caption = "Mes"
Height = 21
Left = 220
Top = 140
Width = 30
ForeColor = 255,255,255
Name = "Label2"
BackStyle = 0

```

```

Caption = "Año"
Height = 21
Left = 400
Top = 140
Width = 30
ForeColor = 255,255,255
Name = "Label4"

```

```

*****
C:\COBRATEL\CPOT0340.SCX
Left = 76
Top = 253
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 29
Top = 12
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor2"
Left = 154
Top = 14
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor1"
Left = 290
Top = 15
Width = 90
Height = 90
Alias = "est_cobro"
Database = dbCOBRATEL.dbc
CursorSource = "est_cobro"
Name = "Cursor3"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Reporte
por vía de cobro."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.Text6.value=date()
thisform.Text7.value=date()

SELECT cobros
ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Reporte por vía de cobro"
Height = 30
Left = 100
Top = 40
Width = 370

```

```

TabIndex = 6
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú de depuración"
StatusBarText = "Regresar al menú de bitácoras."
TabIndex = 5
Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setFocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0300
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "¿ De qué fecha desea el reporte ?"
Height = 21
Left = 230
Top = 100
Width = 240
TabIndex = 7
ForeColor = 255,255,255
Name = "Label1"
Top = 220
Left = 260
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Depuración de la bitácora de procesos."
TabIndex = 4
Name = "Command1"
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,255,255)
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton depurar
if MESSAGEBOX('¿ Ya realizó el respaldo de su Base de Datos?', 4+32+256, 'Confirme por favor...') = 6
then

```

```

&& si realizó el respaldo

COUNT for
via_cobr=thisform.Text8.value .and.
fech_cob>=thisform.Text6.value .and.
fech_cob<=thisform.Text7.value TO
hay
IF hay <>0 then
thisform.progreso.visible=.t.
FOR i=1 to 100 STEP 2
thisform.progreso.value=i
WAIT " timeout .1
NEXT i
thisform.progreso.visible=.f.
REPORT form viacobro for
via_cobr=thisform.Text8.value .and.
fech_cob>=thisform.Text6.value .and.
fech_cob<=thisform.Text7.value
preview
SELECT procesos && Registra en bitácora de procesos, la operación efectuada
APPEND blank
REPLACE tarea_bp with "Rpt via cobro: "+ est_cobro.nomb_ec
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT cobros && cierra
procesos.....
..... cerrar procesos
ELSE
MESSAGEBOX('No hay cobros de esta via en este periodo de fechas. Verifique por favor!',0+64+0,'Atención')
ENDIF

ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setFocus
ENDPROC
BackStyle = 0
Caption = "Generar reporte"
Height = 21
Left = 350
Top = 240
Width = 120
TabIndex = 8
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 13
Top = 70
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 519
Top = 14
Visible = .T.
Width = 40

```

```

TabIndex = 11
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 14
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 58
Visible = .F.
Width = 40
TabIndex = 10
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 15
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 43
Visible = .T.
Width = 40
TabIndex = 12
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 17
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 16
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 9
Name = "Embossedlabel4"
Picture = folder04.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 336
Left = 126
Height = 11

```

```

Width = 411
Visible = .F.
TabIndex = 18
Name = "Progreso"
BackStyle = 0
Caption = "Fecha inicial"
Height = 21
Left = 300
Top = 140
Width = 80
TabIndex = 19
ForeColor = 255,255,255
Name = "Label2"
BackStyle = 0
Caption = "Fecha final"
Height = 21
Left = 400
Top = 140
Width = 80
TabIndex = 21
ForeColor = 255,255,255
Name = "Label4"
Enabled = .T.
Height = 21
Left = 300
TabIndex = 2
Top = 160
Width = 80
Name = "Text6"
PROCEDURE Valid
* Validar que la fecha sea correcta
IF this.value > date()
RETURN .f.
MESSAGEBOX('La fecha
proporcionada no es válida. Verifique
por favor!',0+64+0,'Atención')
ELSE
RETURN .t.
ENDIF
ENDPROC
PROCEDURE LostFocus
* Al dar una fecha válida

thisform.Text7.value=this.value
ENDPROC
Enabled = .T.
Height = 21
Left = 400
TabIndex = 3
Top = 160
Width = 80
Name = "Text7"
PROCEDURE Valid
* Validar que la fecha sea correcta
IF this.value > date() .or.
this.value<thisform.Text6.value
RETURN .f.
MESSAGEBOX('La fecha
proporcionada no es válida. Verifique
por favor!',0+64+0,'Atención')
ELSE
RETURN .t.
ENDIF

ENDPROC
BackStyle = 0
Caption = "Vía de cobro"
Height = 21
Left = 140
Top = 140

```

```

Width = 80
TabIndex = 20
ForeColor = 255,255,255
Name = "Label5"
Enabled = .F.
Height = 21
Left = 500
TabIndex = 3
Top = 160
Visible = .F.
Width = 80
Name = "Text8"
PROCEDURE Valid
* Validar que la fecha sea correcta
IF this.value > date() .or.
this.value<thisform.Text6.value
RETURN .f.
MESSAGEBOX('La fecha
proporcionada no es válida. Verifique
por favor!',0+64+0,'Atención')
ELSE
RETURN .t.
ENDIF

ENDPROC
ColumnCount = 2
ColumnWidths = "97,49"
RowSourceType = 6
RowSource =
"est_cobro.nomb_ec.via_ec"
FirstElement = 1
Height = 21
Left = 140
NumberOfElements = 0
Style = 2
TabIndex = 1
Top = 160
Width = 140
Name = "Combo1"
PROCEDURE InteractiveChange
* Al seleccionar alguno
thisform.Text8.value=est_cobro.via_
ec
ENDPROC

*****
C:\COBRATEL\CPT0400.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL@.
Mantenimiento a catálogos"
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Mantenimiento a
catálogos"
Height = 30
Left = 100
Top = 40
Width = 290
TabIndex = 1
ForeColor = 255,255,255

```

```

Name = "Label10"
AutoSize = .F.
Top = 340
Left = 660
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09it.ico
Caption = "Menú principal"
StatusBarText = "Regresar al menú
principal."
TabIndex = 9
Name = "Command10"
PROCEDURE Click
* Boton Salir
thisform.release
DO form.menuprin
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 140
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = misc25.ico
Caption = ""
StatusBarText = "Mantenimiento al
catálogo de agentes."
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton Transferencia de archivos
thisform.release
DO form.cpt0410
ENDPROC
AutoSize = .F.
Top = 220
Left = 60
Height = 59
Width = 80
WordWrap = .T.
Picture = monitr01.ico
Caption = ""

```



```

StatusBarText = "Mantenimiento al
catálogo de estaciones de cobro."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Procesos especiales
thisform.release
DO form cpt0420
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
Top = 140
Left = 380
Height = 61
Width = 81
WordWrap = .T.
Picture = misc27.ico
DownPicture =
Caption = ""
StatusBarText = "Mantenimiento al
catálogo de usuarios."
TabIndex = 5
ToolTipText = ""
Name = "Command3"
PROCEDURE LostFocus
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton Generación de reportes
thisform.release
DO form cpt0430
ENDPROC
Top = 220
Left = 380
Height = 60
Width = 81
WordWrap = T
Picture = misc29.ico
Caption = ""
StatusBarText = "Mantenimiento al
catálogo de grupos."
TabIndex = 6
ToolTipText = ""
Name = "Command4"
PROCEDURE LostFocus

```

```

thisform.Label4.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label4.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Mantenimiento de catálogos
thisform.release
DO form cpt0440
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
BackStyle = 0
Caption = "Agentes"
Height = 21
Left = 150
Top = 160
Width = 160
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Estaciones de cobro"
Height = 21
Left = 150
Top = 240
Width = 160
ForeColor = 255,255,255
Name = "Label2"
BackStyle = 0
Caption = "Usuarios"
Height = 21
Left = 470
Top = 160
Width = 160
ForeColor = 255,255,255
Name = "Label3"
BackStyle = 0
Caption = "Grupos"
Height = 21
Left = 470
Top = 240
Width = 160
ForeColor = 255,255,255
Name = "Label4"
Picture = crdfle11.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

```

```

*****
C:\COBRATEL\CPT0410.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Alias = "agentes"
BufferModeOverride = 5
ORDER = "vi_co_ag"
Database = dbCOBRATEL.dbc

```

```

CursorSource = "agentes"
Name = "Cursor1"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor2"
ScaleMode = 3
Height = 574
Width = 793
DoCreate = .T.
ShowTips = .T.
AutoCenter = .T.
Picture = fondo.bmp
Caption = "COBRATEL®.
Mantenimiento al catálogo de
agentes."
Closable = .T.
Visible = .F.
WindowType = 0
WindowState = 2
LockScreen = .F.
Name = "Form1"
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Name = "Shape4"
layoutsty.Left = 4004
layoutsty.Visible = .F.
layoutsty.Name = "layoutsty"
Label1.Caption = "Catálogo de
agentes"
Label1.Left = 100
Label1.Top = 30
Label1.Width = 235
Label1.ForeColor = 255,255,255
Label1.Name = "Label1"
Shape1.Top = 389
Shape1.Left = 20
Shape1.Width = 700
Shape1.Name = "Shape1"
Shape2.Top = 390
Shape2.Left = 20
Shape2.Width = 700
Shape2.Name = "Shape2"
Shape3.Top = 389
Shape3.Left = 20
Shape3.Width = 700
Shape3.Name = "Shape3"
Shape4.Top = 390
Shape4.Left = 20
Shape4.Width = 700
Shape4.Name = "Shape4"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
GO top
regactual=recno()

ENDPROC
Caption = "Via de cobro"
Height = 13
Left = 182
Top = 164

```

```

Visible = .T.
Width = 78
ForeColor = 255,255,255
Name = "LBLVI_CO_AG1"
Alignment = 1
ControlSource = "agentes.vi_co_ag"
Enabled = .F.
Left = 320
Top = 160
Visible = .T.
Width = 48
Name = "VI_CO_AG1"
PROCEDURE Valid
*VALIDAR CLAVE

IF !EMPTY(THIS.VALUE)
MIREC=RECNO()
GO top
COUNT for vi_co_ag=this.value to
hay
GO RECORD MIREC

IF hay > 1 && ya habia una clave
como esta
MESSAGEBOX('Esta clave ya
esta registrada. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Vía de localización"
Height = 13
Left = 182
Top = 212
Visible = .T.
Width = 98
ForeColor = 255,255,255
Name = "LBLVI_LO_AG1"
ControlSource = "agentes.vi_lo_ag"
Enabled = .F.
InputMask = "XX"
Left = 320
Top = 208
Visible = .T.
Width = 30
Name = "VI_LO_AG1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Responsable"
Height = 13
Left = 182
Top = 236
Visible = .T.
Width = 78
ForeColor = 255,255,255
Name = "LBLRESP_AGEN1"
ControlSource =
"agentes.resp_agen"
Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXX"
Left = 320
Top = 232
Visible = .T.
Width = 150
Name = "RESP_AGEN1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Teléfono del responsable"
Height = 13
```

```

Left = 182
Top = 260
Visible = .T.
Width = 128
ForeColor = 255,255,255
Name = "LBLTE_RE_AG1"
ControlSource = "agentes.te_re_ag"
Enabled = .F.
InputMask = "XXXXXXXXXX"
Left = 320
Top = 256
Visible = .T.
Width = 90
Name = "TE_RE_AG1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Baja del agente"
Height = 13
Left = 182
Top = 284
Visible = .T.
Width = 78
ForeColor = 255,255,255
Name = "LBLBAJA_AG1"
ControlSource = "agentes.baja_ag"
Enabled = .F.
InputMask = "X"
Left = 320
Top = 280
Visible = .T.
Width = 20
Name = "BAJA_AG1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Top = 400
Left = 140
Visible = .T.
TabIndex = 18
oldTalk = OFF
viewkey = ("")
ParentKey =
ViewType = 3
GridAlias =
topfile = .T.
endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
```

```

oldreprocess = 0
previewmode = .F.
previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Name = "cmdPrev"
cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros específicos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzcclose.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = ..\archivos
de
programaldevstudio\wfp\wizards\wizb
mpslwznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."

```

```

cmdDelete.ToolTipText = "Elimina"
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programaldevstudio\wfp\wizards\wizb
mpslwzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
        Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
        Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdTop.Click
*boton primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click
*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*boton búsqueda
thisform.Navegar.visible=.F.
thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*botón imprimir
#DEFINE C_MAKEREPO_LOC
"Could
not locate a report to print. Create
new one?"

```

```

#DEFINE C_NOOPEN_LOC
"Error opening table.
Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC
"Pick report."

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

* Handling for Private data sessions
#IF 0
        IF m.nSaveSess # 1
                SET
                DATASESSION TO 1
                SELECT 0
                IF
                !EMPTY(m.cSaveData)
                        OPEN
                DATABASE (m.cSaveData)
                ENDF
                IF
                USED(m.cSaveAlias)

                SELECT (m.cSaveAlias)
                IF
                CURSORGETPROP("SourceName")
                #m.cSaveSource

                cDiffSource =
                CURSORGETPROP("SourceName")

                USE IN (m.cSaveAlias)

                SELECT 0
                ENDF
                ENDF
                IF
                EMPTY(ALIAS())
                        USE
                (m.cSaveSource) AGAIN ALIAS
                (m.cSaveAlias) SHARED
                IF
                EMPTY(ALIAS())

                =MESSAGEBOX(C_NOO
                PEN_LOC)

                RETURN
                ENDF
                ENDF
#ENDIF

IF FILE(m.cRepName)
        REPORT FORM
        (m.cRepName) PREVIEW NOWAIT

```

```

ELSE
    m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
    IF !EMPTY(m.cRepName)
        IF
FILE(m.cRepName)
            * User
pressed Open button
                REPORT FORM
(m.cRepName) PREVIEW NOWAIT
                ELSE
            * User
pressed New button
                DO
HOME()+"wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
                ENDIF
            ENDIF
        ENDIF
    * Private data session code
    #IF 0
        IF !EMPTY(cDiffSource)
            USE
(m.cDiffSource) IN 0
            ENDIF
            SET DATASESSION TO
m.nSaveSess
            SELECT (m.cSaveAlias)
        #ENDIF
    IF THISFORM.ShowWindow = 2
        Activate Window
(THISFORM.Name)
    ENDIF
    ENDPROC
PROCEDURE cmdExit.Click
*botón salir
thisform.Release
DO form cpt0400 &&menu de
mAntenimiento de catálogos
    ENDPROC
PROCEDURE cmdAdd.Click
*botón agrega
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
#DEFINE MB_A_YES
6
#DEFINE
C_KEYFLDNOUPDATE_LOC
"The field relating the grid's
view to the parent data source is not
updatable."+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
"The fields

```

```

relating the parent and child tables
are not the same data type."+;
"Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
"The child data source is a
view and does not send updates."+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_NOOBJ_LOC
"Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
"You cannot add a new
record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
"You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."
LOCAL
oSearchDlg,oAddRec,cChildAlias,c
PapaAlias,i,IPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,IBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
IBadChildKey,IUpdatableParentKey,I
NoSendParentUpdates,I,NoSendChil
dUpdates
DO CASE
CASE THIS.Parent.EditMode
&&.....
..... registro en bitácora
SELECT procesos
APPEND blank
IF
ALLTRIM(THISFORM.Text4.VALUE)
="MODIFICAR"
REPLACE tarea_bp with
"Cambio al catálogo de agentes"
ELSE
REPLACE tarea_bp with "Alta al
catálogo de agentes"
ENDIF
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT agentes
&&.....
.. cerrar procesos
thisform.Text4.VALUE=SP
ACE(7)
&&.....<<<<<<<<
<<<<<<<<
** Code for saving record
this.Parent.updaterows()
CASE EMPTY(THIS.Parent.GridRef)
&& not using
Grid object
thisform.Text4.VALUE="A
GREGAR"

```

```

&&.....<<<<<<<<
<<<<<<<<
** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IF
CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;
!CURSORGETPROP("Sen
dUpdates")
=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
APPEND BLANK
OTHERWISE
** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IPrimeKey = .F.
&&is child key primary
index
IF
!EMPTY(THIS.Parent.oDataRelation)
oRel =
EVAL(THIS.Parent.oDataRelation)
ENDIF
DO CASE
CASE TYPE("m.oRel") =
"O"
&&data environment
WITH oRel
cPapaAlias = .ParentAlias
cPapaKey =
.RelationalExpr
cChildAlias = .ChildAlias
cChildKey = .ChildOrder
ENDWITH
oRel = .NULL.
&&reset
CASE
!EMPTY(THIS.Parent.viewkey)
&&using views
cPapaAlias =
ALIAS()
cPapaKey =
THIS.Parent.ParentKey

```

```

        cChildAlias =
THIS.Parent.GridAlias
        cChildKey =
THIS.Parent.viewkey

        * Need to check
if Key field is updatable
        IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

                ATC(" "+m.cChildKey+"",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0

        IBadViewKey = .T.
                ENDIF
        OTHERWISE
        cPapaAlias =
ALIAS()
        cPapaKey =
RELATION(1)
        cChildAlias =
THIS.Parent.GridAlias
        cChildKey =
ORDER(m.cChildAlias)
        ENDCASE

        * Test to see if both keys
are of the same data type
        SELECT
(THIS.Parent.GridAlias)
        cGridKeyType = ""

        IF
EMPTY(THIS.Parent.viewkey) AND
!EMPTY(m.cChildKey)
                &&regular table used
                * Get the Child
Key field since only index tag is here
                * if these is an
expression, then default to blank.
                cTagName = ""
                FOR i = 1 TO
TagCount("")
                        IF
UPPER(TAG(m.i)) ==
UPPER(m.cChildKey)

                                cTagName = KEY(m.i)

                                IPrimeKey =
PRIMARY(m.i)

                                EXIT
                                ENDIF
                        ENDFOR

                * Check if we
have an index expression here.
                IF
ATC("(",m.cTagName)#0 OR
ATC("+",m.cTagName)#0

                        cChildKey = ""
                        ELSE
                                cChildKey = m.cTagName

```

```

        cGridKeyType =
TYPE(m.cChildKey)
                ENDIF
        ENDF

        IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cPapaAlias)

        INoSendParentUpdates =
.T.
                ENDIF
        IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cChildAlias)

        INoSendChildUpdates = .T.
        ENDF

        * Check if have two views
and both do not send updates
        IF
m.INoSendParentUpdates AND
m.INoSendChildUpdates

                =MESSAGEBOX(C_NOUP
DATE_LOC)
                RETURN
        ENDF

        SELECT
(THIS.Parent.OldAlias )
        IF
!EMPTY(m.cGridKeyType) AND
m.cGridKeyType #
TYPE(m.cPapaKey)
                IBadChildKey =
.T.
                ENDF

        DO CASE
        CASE
m.INoSendParentUpdates AND
(m.IBadViewKey OR
m.IBadChildKey)

                =MESSAGEBOX(C_NOUP
DATE_LOC)
                RETURN
        CASE m.IPrimeKey AND
m.INoSendParentUpdates

                =MESSAGEBOX(C_NOUP
DATE2_LOC)
                RETURN
        CASE
m.INoSendChildUpdates

```

```

                &&child view does not
send updates
                IF
MESSAGEBOX(C_NOCHILDUPDAT
E_LOC,MB_Q_YESNO) #
MB_A_YES

                RETURN
                ENDF
                APPEND BLANK
                CASE m.IBadViewKey
                &&view key in not
updatable
                IF
MESSAGEBOX(C_KEYFLDNOUPD
ATE_LOC,MB_Q_YESNO) #
MB_A_YES

                RETURN
                ENDF
                APPEND BLANK
                CASE m.IBadChildKey
                &&bad child key -- different
data type from parent
                IF
MESSAGEBOX(C_BADCHILDKEY_
LOC,MB_Q_YESNO) # MB_A_YES

                RETURN
                ENDF
                APPEND BLANK
        OTHERWISE
                * Need to check
if Key field is updatable

                IUpdatableParentKey = .T.
                IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND;

                        ATC(" "+m.cPapaKey+"",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0
                        *
                cPapaKey = ""

                IUpdatableParentKey = .F.
                ENDF

                * Check for
expression in cPapaKey if so, then
do not update record with keyvalue
                IF
ATC("(",m.cPapaKey)#0 OR
ATC("+",m.cPapaKey)#0

                        cPapaKey = ""
                        ENDF

                nSaveSess =
SET("DATASESSION")
                oAddRec =
CREATE("GridAddRec")
                IF
TYPE("m.oAddRec") # "O"

                        =MESSAGEBOX(C_NOO
BJ_LOC)

```

```

RETURN
ENDIF
oAddRec.ChildPrimaryKey
= m.IPrimeKey
oAddRec.UpdatableParent
Key = m.IUpdatableParentKey
oAddRec.NoSendUpdates
= m.INoSendParentUpdates
oAddRec.KeyField =
m.cPapaKey
oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey),"")
oAddRec.runaddform()
IF
oAddRec.AddOption =
OPT_CANCEL
RETURN
ENDIF
SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)
IF
TYPE("oAddRec.KeyValue") = "C"
oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
ENDIF
* Add record to
parent table
nSaveRec =
RECNO()
IF
INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
IF
EMPTY(m.cPapaKey)
APPEND BLANK IN
(m.cPapaAlias)
ELSE
INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
ENDIF
ENDIF
nSaveRec2 =
RECNO()
* Add child
record
IF
INLIST(oAddRec.AddOption,OPT_A
DD_CHILD,OPT_ADD_BOTH)
* Need
to check

```

```

GO
m.nSaveRec
IF
EMPTY(m.cChildKey) OR
TYPE("m.cChildKey")#TYPE("oAddr
ec.KeyValue")
APPEND BLANK IN
(m.cChildAlias)
ELSE
INSERT INTO
(m.cChildAlias) ((m.cChildKey))
VALUES(oAddRec.KeyValue)
ENDIF
GO
m.nSaveRec2
ENDIF
ENDCASE
ENDCASE
this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode =
THIS.Parent.EditMode
this.Parent.topfile = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.
thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
thisform.nom_agen1.enabled=.F.
thisform.vi_lo_ag1.enabled=.F.
thisform.resp_agen1.enabled=.F.
thisform.te_re_ag1.enabled=.F.
thisform.baja_ag1.enabled=.F.
ENDPROC
PROCEDURE cmdDelete.Click
*botón borra
* Tipo de cuadro de diálogo
* 4 + 32 + 256
* 4 = Botones Si y No
* 32 = Icono de PREGUNTA
* 256 = el segundo botón es el de
default
* Respuestas
* 6 = SI
* 7 = NO
IF MESSAGEBOX('¿ Desea eliminar
este registro?', 4+32+256, 'Confirme
por favor...') = 6
Delete
IF
THIS.Parent.updaterows()
&&success
* Success
IF EOF()

```

```

Skip 1
ENDIF
IF EOF() AND
!BOF()
Skip -1
ENDIF
regactual=recno()
ENDIF
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.
&&.....
registro en bitácora
SELECT procesos
APPEND blank
REPLACE tarea_bp with "Baja al
catálogo de agentes"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT agentes
&&.....
.. cerrar procesos
ENDIF
IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF
ENDPROC
PROCEDURE cmdEdit.Click
*botón edita
#DEFINE C_NOUPDATE_LOC
"You cannot edit because
the view(s) selected does not send
updates."
LOCAL
INoSendParentUpdates,IInoSendChil
dUpdates
LOCAL aTablesUsed,nTablesUsed,i
** Reverting record
IF THIS.Parent.EditMode
thisform.Text4.VALUE=SP
ACE(7)
&&*****<<<<<<
<<<<<<<
IF THIS.Parent.usedataenv
SELECT
(THIS.Parent.OldAlias)
IF
CURSORGETPROP("Buffering")>1
=TableRevert(.T.)
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)
SELECT
(THIS.Parent.GridAlias)
IF
CURSORGETPROP("Buffering")>1
=TableRevert(.T.)
ENDIF
ENDIF
ENDIF

```

```

ELSE
    DIMENSION
aTablesUsed[1]
    m.nTablesUsed
= AUSED(aTablesUsed)
    FOR i = 1 TO
m.nTablesUsed
        IF
CURSORGETPROP("Buffering",aTablesUsed[m,i,1])>1
            =TableRevert(.T.,aTablesUsed[m,i,1])
        ENDIF
    ENDFOR
ENDIF

* Go back to original place
SELECT
(THIS.Parent.OldAlias)
IF RECCOUNT() <
THIS.Parent.oldrec &&added record
at EOF()
    GO TOP
ELSE
    GO
THIS.Parent.oldrec
ENDIF
ELSE
    thisform.Text4.VALUE="MODIFICAR"
&&*****<<<<<<
<<<<<<<

* Check to see if view
allows updates
IF
CURSORGETPROP("SourceType")#
3 AND
!CURSORGETPROP("offline") AND;
!CURSORGETPROP("SendUpdates")
    INoSendParentUpdates =
T
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)
AND ;
CURSORGETPROP("SourceType",THIS.Parent.GridAlias)#3 AND ;
!CURSORGETPROP("offline",THIS.Parent.GridAlias) AND ;
!CURSORGETPROP("SendUpdates",THIS.Parent.GridAlias)
    INoSendChildUpdates=.T.
ENDIF

IF
(m.INoSendChildUpdates AND
m.INoSendParentUpdates) OR;
    (EMPTY(THIS.Parent.GridAlias) AND
m.INoSendParentUpdates)

```

```

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF

this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
    this.Parent.oldrec =
RECNO() &&save record in case
reverting
ENDIF

** Editing record
this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
thisform.VI_CO_AG1.enabled=.F.

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Agente"
Height = 31
Left = 140
Top = 120
Width = 80
TabIndex = 21
ForeColor = 255,255,255
Name = "Label2"
ColumnCount = 2
ColumnWidths = "65,155"
RowSourceType = 6
RowSource =
"agentes.vi_co_ag,nom_agen"
FirstElement = 1
Height = 21
Left = 160
NumberOfElements = 0
Style = 2
TabIndex = 22
Top = 160
Visible = .F.
Width = 20
Name = "Busca"
PROCEDURE InteractiveChange
* Cuando encuentra el elemento
buscado
GO record recno()
regactual=recno()
thisform.busca.visible=.F.
thisform.Navegar.visible=.T.

* Reset from prior
thisform.Refresh()

```

```

ENDPROC
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 90
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 580
TabIndex = 24
Top = 90
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 580
Top = 80
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 647
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 90
Visible = .F.
Width = 80
Name = "Text4"
Caption = "clave de usuario"
Height = 13
Left = 489
Top = 80
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26

```

```

Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Picture = misc25.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

```

```

C:\COBRATEL\CPT0420.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Alias = "est_cobro"
BufferModeOverride = 5
ORDER = "num_ec"
Database = dbCOBRATEL.dbc
CursorSource = "est_cobro"
Name = "Cursor1"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor2"
ScaleMode = 3
Height = 574
Width = 793
DoCreate = .T.
ShowTips = .T.
AutoCenter = .T.
Picture = fondo.bmp
Caption = "Estaciones"
Closable = .T.
Visible = .F.
WindowType = 0
WindowState = 2
LockScreen = .F.
Name = "Form1"
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Name = "Shape4"
layoutsty.Left = 4004
layoutsty.Visible = .F.
layoutsty.Name = "layoutsty"
Label1.Caption = "Catálogo de
estaciones de cobro"
Label1.Left = 100
Label1.Top = 30
Label1.Width = 375
Label1.ForeColor = 255,255,255
Label1.Name = "Label1"
Shape1.Top = 389
Shape1.Left = 40
Shape1.Width = 700
Shape1.Name = "Shape1"
Shape2.Top = 390
Shape2.Left = 40
Shape2.Width = 700
Shape2.Name = "Shape2"

```

```

Shape3.Top = 389
Shape3.Left = 40
Shape3.Width = 700
Shape3.Name = "Shape3"
Shape4.Top = 390
Shape4.Left = 40
Shape4.Width = 700
Shape4.Name = "Shape4"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

```

```

GO top
regactual=recno()

```

```

ENDPROC
Caption = "Número"
Left = 119
Top = 154
Visible = .T.
Width = 61
ForeColor = 255,255,255
Name = "LBLNUM_EC1"
Alignment = 1
ControlSource = "est_cobro.num_ec"
Enabled = .F.
Left = 240
Top = 150
Visible = .T.
Width = 48
Name = "NUM_EC1"
PROCEDURE Valid
* VALIDAR CLAVE

```

```

IF !EMPTY(THIS.VALUE)
MIREC=RECNO()
GO top
COUNT for num_ec=this.value to
hay
GO RECORD MIREC

IF hay > 1 && ya había una clave
como esta
MESSAGEBOX('Esta clave ya
esta registrada. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .F. &&no puede dejar el
campo
ELSE
thisform.via_ec1.enabled=.T.
thisform.nomb_ec1.enabled=.T.
thisform.sigla_ec1.enabled=.T.
thisform.direc_ec1.enabled=.T.
thisform.resp_ec1.enabled=.T.
thisform.tel_mod_ec1.enabled=.T.
thisform.tel_re_ec1.enabled=.T.
RETURN .T. &&pasa al siguiente
campo
ENDIF
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .F.
ENDIF
ENDPROC

```

```

Caption = "Vía de cobro"
Height = 13
Left = 119
Top = 178
Visible = .T.
Width = 81
ForeColor = 255,255,255
Name = "LBLVIA_EC1"
Alignment = 1
ControlSource = "est_cobro.via_ec"
Enabled = .F.
Left = 240
Top = 174
Visible = .T.
Width = 48
Name = "VIA_EC1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF
ENDPROC
Caption = "Nombre de estación"
Height = 13
Left = 119
Top = 202
Visible = .T.
Width = 101
ForeColor = 255,255,255
Name = "LBLNOMB_EC1"
ControlSource =
"est_cobro.nomb_ec"
Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXX"
Left = 240
Top = 198
Visible = .T.
Width = 150
Name = "NOMB_EC1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .f.
ENDIF
ENDPROC
Caption = "Siglas"
Left = 119
Top = 226
Visible = .T.
Width = 61
ForeColor = 255,255,255
Name = "LBLSIGLA_EC1"
ControlSource = "est_cobro.sigla_ec"
Enabled = .F.
InputMask = "XXX"
Left = 240
Top = 222

```



```

Visible = .T.
Width = 40
Name = "SIGLA_EC1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ¡Verifique por
favor!",0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Dirección"
Left = 119
Top = 250
Visible = .T.
Width = 61
ForeColor = 255,255,255
Name = "LBLDIREC_EC1"
ControlSource =
"est_cobro.direc_ec"
Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXX"
Left = 240
Top = 246
Visible = .T.
Width = 150
Name = "DIREC_EC1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ¡Verifique por
favor!",0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Responsable"
Height = 13
Left = 119
Top = 274
Visible = .T.
Width = 81
ForeColor = 255,255,255
Name = "LBLRESP_EC1"
ControlSource = "est_cobro.resp_ec"
Enabled = F
InputMask =
"XXXXXXXXXXXXXXXXXXXX"
Left = 240
Top = 270
Visible = .T.
Width = 150
Name = "RESP_EC1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ¡Verifique por
favor!",0+16+0,'Error.')
```

```

RETURN .f.

```

```

ENDIF

ENDPROC
Caption = "Tel. módem"
Left = 119
Top = 298
Visible = .T.
Width = 61
ForeColor = 255,255,255
Name = "LBLTEL_MOD_EC1"
ControlSource =
"est_cobro.tel_mod_ec"
Enabled = .F.
InputMask = "XXXXXXXXXXXX"
Left = 240
Top = 294
Visible = .T.
Width = 90
Name = "TEL_MOD_EC1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ¡Verifique por
favor!",0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Tel. responsable"
Height = 13
Left = 119
Top = 322
Visible = .T.
Width = 91
ForeColor = 255,255,255
Name = "LBLTEL_RE_EC1"
ControlSource =
"est_cobro.tel_re_ec"
Enabled = .F.
InputMask = "XXXXXXXXXXXX"
Left = 240
Top = 318
Visible = .T.
Width = 90
Name = "TEL_RE_EC1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ¡Verifique por
favor!",0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Top = 400
Left = 140
Visible = .T.
TabIndex = 18
oldTalk = OFF
viewkey = ("")
ParentKey =
ViewType = 3
GridAlias =
topfile = .T.

```

```

endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
oldreprocess = 0
previewmode = .F.
previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Name = "cmdPrev"
cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros especificos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclose.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = .\archivos
de
programa\devstudio\vfplwizards\wizb
mpslwznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"

```

```

cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programa\devstudio\wfp\wizards\wizb
mps\wzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
    Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
    Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdTop.Click
*botón primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click
*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*botón búsqueda
thisform.Navegar.visible=.F.
thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*botón imprimir

```

```

*#DEFINE C_MAKEREPO_LOC
    "Could
not locate a report to print. Create
new one?"
*#DEFINE C_NOOPEN_LOC
    "Error opening table.
Unable to print report."
*#DEFINE
C_GETFILEPROMPT_LOC
    "Pick report:"

*LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

*cSaveAlias = ALIAS()
*cSaveSource =
CURSORGETPROP("SourceName")
*cSaveData =
CURSORGETPROP("Database")
*cDiffSource = ""
*cRepName =
LEFT(ALIAS(),8)+".FRX"
*nSaveSess =
SET("DATASESSION")

* Handling for Private data sessions
*#IF 0
*
    IF m.nSaveSess # 1
*
        SET
DATASESSION TO 1
*
        SELECT 0
*
        IF
!EMPTY(m.cSaveData)
*
            OPEN
DATABASE (m.cSaveData)
*
            ENDIF
*
            IF
USED(m.cSaveAlias)
*
                SELECT (m.cSaveAlias)
*
                    IF
CURSORGETPROP("SourceName")
*#m.cSaveSource
*
                        cDiffSource =
CURSORGETPROP("SourceName")
*
                            USE IN (m.cSaveAlias)
*
                                SELECT 0
*
                                    ENDIF
*
                                        ENDIF
*
                                            IF
EMPTY(ALIAS())
*
                                                USE
(m.cSaveSource) AGAIN ALIAS
(m.cSaveAlias) SHARED
*
                                                    IF
EMPTY(ALIAS())
*
                                                        =MESSAGEBOX(C_NOO
PEN_LOC)
*
                                                            RETURN
*
                                                                ENDIF
*
                                                                    ENDIF
*
                                                                        ENDIF
*#ENDIF

```

```

*IF FILE(m.cRepName)
*
    REPORT FORM
(m.cRepName) PREVIEW NOWAIT
*ELSE
*
    m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
*
    IF !EMPTY(m.cRepName)
*
        IF
FILE(m.cRepName)
*
            * User
pressed Open button
*
                REPORT FORM
(m.cRepName) PREVIEW NOWAIT
*
                    ELSE
*
                        * User
pressed New button
*
                            DO
HOME()+\wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
*
                                ENDIF
*
                                    ENDIF
*#ENDIF

* Private data session code
*#IF 0
*
    IF !EMPTY(cDiffSource)
*
        USE
(m.cDiffSource) IN 0
*
            ENDIF
*
                SET DATASESSION TO
m.nSaveSess
*
                    SELECT (m.cSaveAlias)
*#ENDIF

*IF THISFORM.ShowWindow = 2
*
    Activate Window
(THISFORM.Name)
*ENDIF

REPORT form estacion preview

ENDPROC
PROCEDURE cmdExit.Click
*botón salir
thisform.Release
DO form cpt0400 &&menu de
mantenimiento de catálogos

ENDPROC
PROCEDURE cmdAdd.Click
*botón agrega
*#DEFINE OPT_CANCEL
0
*#DEFINE OPT_ADD_PARENT
1
*#DEFINE OPT_ADD_CHILD
2
*#DEFINE OPT_ADD_BOTH
3
*#DEFINE MB_Q_YESNO
36
*#DEFINE MB_A_YES
6

```

```

#DEFINE
C_KEYFLDNOUPDATE_LOC
    "The field relating the grid's
view to the parent data source is not
updatable. "+;
    "Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
    "The fields
relating the parent and child tables
are not the same data type. "+;
    "Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
    "The child data source is a
view and does not send updates. "+;
    "Do you just want to add a
new record to the parent table?"
#DEFINE C_NOOBJ_LOC
    "Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
    "You cannot add a new
record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
    "You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."

LOCAL
oSearchDlog,oAddRec,cChildAlias,c
PapaAlias,i,IPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,IBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
IBadChildKey,IUpdatableParentKey,I
NoSendParentUpdates,I NoSendChil
dUpdates

DO CASE
CASE THIS.Parent.EditMode
&&.....
..... registro en bitácora
    SELECT procesos
    APPEND blank
    IF
ALLTRIM(THISFORM.Text4.VALUE)
="MODIFICAR"
        REPLACE tarea_bp with
"Cambio al catálogo de Est. cobro"
    ELSE
        REPLACE tarea_bp with "Alta al
catálogo de Est. cobro"
    ENDIF
    REPLACE usua_bp with usuario
    REPLACE fecha_bp with date()
    REPLACE hora_bp with time()
    SELECT est_cobro
&&.....
.. cerrar procesos
    thisform.Text4.VALUE=SP
ACE(7)
&&.....<<<<<<
<<<<<<<<

```

```

** Code for saving record
this.Parent.updaterows()

CASE EMPTY(THIS.Parent.GridRef)
    && not using
Grid object
    thisform.Text4.VALUE="A
GREGAR"
&&.....<<<<<<
<<<<<<<<

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
    this.Parent.oldrec =
RECNO() &&save record in case
reverting
    IF
CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;
!CURSORGETPROP("Sen
dUpdates")
=MESSAGEBOX(C_NOUP
DATE_LOC)
    RETURN
    ENDIF
    APPEND BLANK

OTHERWISE

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
    this.Parent.oldrec =
RECNO() &&save record in case
reverting
    IPrimeKey = .F.

    &&is child key primary
index

    IF
!EMPTY(THIS.Parent.oDataRelation)
        oRel =
EVAL(THIS.Parent.oDataRelation)
    ENDIF

DO CASE
CASE TYPE("m.oRel") =
"O"
    &&data environment
        WITH oRel

        cPapaAlias = .ParentAlias

        cPapaKey =
.RelationalExpr

        cChildAlias = .ChildAlias

        cChildKey = .ChildOrder
    ENDWITH

```

```

oRel = .NULL.
&&reset
CASE
!EMPTY(THIS.Parent.viewkey)
    &&using views
        cPapaAlias =
ALIAS()
        cPapaKey =
THIS.Parent.ParentKey
        cChildAlias =
THIS.Parent.GridAlias
        cChildKey =
THIS.Parent.viewkey

        * Need to check
if Key field is updatable
        IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

            ATC(" "+m.cChildKey+"",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0

            IBadViewKey = .T.
            ENDIF
        OTHERWISE
            cPapaAlias =
ALIAS()
            cPapaKey =
RELATION(1)
            cChildAlias =
THIS.Parent.GridAlias
            cChildKey =
ORDER(m.cChildAlias)
            ENDCASE

        * Test to see if both keys
are of the same data type
        SELECT
(THIS.Parent.GridAlias)
cGridKeyType = ""

        IF
EMPTY(THIS.Parent.viewkey) AND
!EMPTY(m.cChildKey)
            &&regular table used
            * Get the Child
Key field since only index tag is here
            * if these is an
expression, then default to blank.
            cTagName = ""
            FOR i = 1 TO
TagCount("")
                IF
UPPER(TAG(m.i)) ==
UPPER(m.cChildKey)

                    cTagName = KEY(m.i)

                IPrimeKey =
PRIMARY(m.i)
            ENDIF
        EXIT
    ENDFOR
ENDFOR

```

```

* Check if we
have an index expression here.
IF
ATC("("m.cTagName)#0 OR
ATC("+",m.cTagName)#0

cChildKey = ""
ELSE

cChildKey = m.cTagName

cGridKeyType =
TYPE(m.cChildKey)
ENDIF
ENDIF

IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cPapaAlias)

INoSendParentUpdates =
.T.
ENDIF
IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

!CURSORGETPROP("SendUpdates
",m.cChildAlias)

INoSendChildUpdates = .T.
ENDIF

* Check if have two views
and both do not send updates
IF
m.INoSendParentUpdates AND
m.INoSendChildUpdates

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF

SELECT
(THIS.Parent.OldAlias)
IF
!EMPTY(m.cGridKeyType) AND
m.cGridKeyType #
TYPE(m.cPapaKey)
!BadChildKey =
.T.
ENDIF

DO CASE
CASE
m.INoSendParentUpdates AND
(m.IBadViewKey OR
m.IBadChildKey)

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN

```

```

CASE m.IPrimeKey AND
m.INoSendParentUpdates

=MESSAGEBOX(C_NOUP
DATE2_LOC)
RETURN

CASE
m.INoSendChildUpdates
&&child view does not
send updates
IF
MESSAGEBOX(C_NOCHILDUPDAT
E_LOC,MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadViewKey
&&view key in not
updatable
IF
MESSAGEBOX(C_KEYFLDNOUPD
ATE_LOC,MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadChildKey
&&bad child key -- different
data type from parent
IF
MESSAGEBOX(C_BADCHILDKEY_
LOC,MB_Q_YESNO) # MB_A_YES

RETURN
ENDIF
APPEND BLANK
OTHERWISE
* Need to check
if Key field is updatable

IUpdatableParentKey = .T.
IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND;

ATC("("m.cPapaKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0

cPapaKey = ""

IUpdatableParentKey = .F.
ENDIF

* Check for
expression in cPapaKey if so, then
do not update record with keyvalue
IF
ATC("("m.cPapaKey)#0 OR
ATC("+",m.cPapaKey)#0

cPapaKey = ""
ENDIF

```

```

nSaveSess =
SET("DATASESSION")
oAddRec =
CREATE("GridAddRec")
IF
TYPE("m.oAddRec") # "O"

=MESSAGEBOX(C_NOO
BJ_LOC)

RETURN
ENDIF

oAddRec.ChildPrimaryKey
= m.IPrimeKey

oAddRec.UpdatableParent
Key = m.IUpdatableParentKey

oAddRec.NoSendUpdates
= m.INoSendParentUpdates

oAddRec.KeyField =
m.cPapaKey

oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey),"")

oAddRec.runaddform()
IF
oAddRec.AddOption =
OPT_CANCEL

RETURN
ENDIF

SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)

IF
TYPE("oAddRec.KeyValue") = "C"

oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
ENDIF

* Add record to
parent table
nSaveRec =
RECNO()
IF
!INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
IF
EMPTY(m.cPapaKey)

APPEND BLANK IN
(m.cPapaAlias)
ELSE

INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
ENDIF
ENDIF

```

```

nSaveRec2 =
RECNO()
record
  * Add child
  IF
  INLIST(oAddRec.AddOption,OPT_A
  DD_CHILD,OPT_ADD_BOTH)
  to check
    * Need
  GO
  m.nSaveRec
  IF
  EMPTY(m.cChildKey) OR
  TYPE("m.cChildKey")#TYPE('oAddR
  ec.KeyValue')
  APPEND BLANK IN
  (m.cChildAlias)
  ELSE
  INSERT INTO
  (m.cChildAlias)((m.cChildKey))
  VALUES(oAddRec.KeyValue)
  ENDIF
  GO
  m.nSaveRec2
  ENDIF
  ENDCASE
  ENDCASE

```

```

this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode =
THIS.Parent.EditMode
this.Parent.topfile = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = F

```

```

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
thisform.via_ec1.enabled=.F.
thisform.nomb_ec1.enabled=.F.
thisform.sigla_ec1.enabled=.F.
thisform.direc_ec1.enabled=.F.
thisform.resp_ec1.enabled=.F.
thisform.tel_mod_ec1.enabled=.F.
thisform.tel_re_ec1.enabled=.F.

```

```

ENDPROC
PROCEDURE cmdDelete.Click
*botón borra

```

```

* Tipo de cuadro de diálogo
* 4 + 32 + 256
* 4 = Botones SI y No
* 32 = Icono de PREGUNTA
* 256 = el segundo botón es el de
default

```

```

* Respuestas
* 6 = SI
* 7 = NO
IF MESSAGEBOX('¿ Desea eliminar
este registro?', 4+32+256, 'Confirme
por favor...') = 6
  Delete
  IF
  THIS.Parent.updaterows()
  &&success
  * Success
  IF !EOF()
  Skip 1
  ENDIF
  IF EOF() AND
  !BOF()
  Skip -1
  ENDIF
  regactual=recno()
  ENDIF
  thisform.LockScreen = .T.
  this.Parent.buttonrefresh()
  this.Parent.navrefresh()
  thisform.LockScreen = .F.
  &&.....
  ..... registro en bitácora
  SELECT procesos
  APPEND blank
  REPLACE tarea_bp with "Baja al
  catálogo de Est.cobro"
  REPLACE usua_bp with usuario
  REPLACE fecha_bp with date()
  REPLACE hora_bp with time()
  SELECT est_cobro
  &&.....
  .. cerrar procesos
  ENDIF
  IF THISFORM.ShowWindow = 2
  Activate Window
  (THISFORM.Name)
  ENDIF
  ENDPROC
  PROCEDURE cmdEdit.Click
  *botón edita
  #DEFINE C_NOUPDATE_LOG
  "You cannot edit because
  the view(s) selected does not send
  updates."
  LOCAL
  INoSendParentUpdates,INoSendChil
  dUpdates
  LOCAL aTablesUsed,nTablesUsed,i
  ** Reverting record
  IF THIS.Parent.EditMode
  thisform.Text4.VALUE=SP
  ACE(7)
  &&.....<<<<<<
  <<<<<<<
  IF THIS.Parent.usedataenv
  SELECT
  (THIS.Parent.OldAlias)
  IF
  CURSORGETPROP("Buffering")>1

```

```

=TableRevert(.T.)
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)
SELECT
(THIS.Parent.GridAlias)
IF
CURSORGETPROP("Buffering")>1
=TableRevert(.T.)
ENDIF
ENDIF
ELSE
DIMENSION
aTablesUsed[1]
m.nTablesUsed
= AUSED(aTablesUsed)
FOR i = 1 TO
m.nTablesUsed
IF
CURSORGETPROP("Buffering",aTa
blesUsed[m,i,1])>1
=TableRevert(.T.,aTablesU
sed[m,i,1])
ENDIF
ENDIF
ENDIF
* Go back to original place
SELECT
(THIS.Parent.OldAlias)
IF RECCOUNT() <
THIS.Parent.oldrec &&added record
at EOF()
GO TOP
ELSE
GO
THIS.Parent.oldrec
ENDIF
ELSE
thisform.Text4.VALUE="M
ODIFICAR"
&&.....<<<<<<
<<<<<<<
* Check to see if view
allows updates
IF
CURSORGETPROP("SourceType")#
3 AND
!CURSORGETPROP("offline") AND;
!CURSORGETPROP("Sen
dUpdates")
INoSendParentUpdates =
.T.
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)
AND ;
CURSORGETPROP("SourceType",T
HIS.Parent.GridAlias)#3 AND ;
!CURSORGETPROP("offline",THIS.
Parent.GridAlias) AND ;

```

```
!CURSORGETPROP("SendUpdates  
".THIS.Parent.GridAlias)
```

```
INoSendChildUpdates= .T.  
ENDIF
```

```
IF
```

```
(m INoSendChildUpdates AND  
m.!NoSendParentUpdates) OR:
```

```
(EMPTY(THIS.Parent.Grid  
Alias) AND  
m.!NoSendParentUpdates)
```

```
=MESSAGEBOX(C_NOUP  
DATE_LOC)
```

```
RETURN
```

```
ENDIF
```

```
this.Parent.OldAlias =  
ALIAS() &&save alias in case  
reverting
```

```
this.Parent.oldrec =  
RECNO() &&save record in case  
reverting  
ENDIF
```

```
** Editing record
```

```
this.Parent.EditMode =  
!THIS.Parent.EditMod.  
this.Parent.AddMode = F.  
thisform.LockScreen = T.  
this.Parent.buttonrefresh()  
this.Parent.navrefresh()  
thisform.LockScreen = F.
```

```
thisform.Text1.enabled=.F.  
thisform.text2.enabled=.F.  
thisform.text3.enabled=.F.  
thisform.Text4.enabled= F.  
thisform.text5.enabled= F.  
thisform.NUM_EC1.enabled= F.
```

```
ENDPROC
```

```
FontBold = T.  
FontSize = 14  
BackStyle = 0  
Caption = "Estación"  
Height = 31  
Left = 88  
Top = 108  
Width = 80  
TabIndex = 21  
ForeColor = 255,255,255  
Name = "Label2"  
Enabled = .F.  
Height = 21  
Left = 490  
TabIndex = 23  
Top = 80  
Visible = F.  
Width = 80  
Name = "Text1"  
Caption = "Usuario"  
Height = 13  
Left = 520  
Top = 14
```

```
Visible = T.  
Width = 40  
TabIndex = 15  
ForeColor = 255,255,255  
Name = "Embossedlabel1"  
Enabled = .F.  
Height = 21  
Left = 590  
TabIndex = 24  
Top = 80  
Visible = .F.  
Width = 80  
Name = "Text2"  
Caption = "Grupo"  
Height = 13  
Left = 590  
Top = 68  
Visible = F.  
Width = 40  
TabIndex = 14  
Name = "Embossedlabel2"  
Enabled = .F.  
Height = 21  
Left = 680  
TabIndex = 25  
Top = 40  
Width = 80  
Name = "Text3"  
Caption = "Fecha"  
Height = 13  
Left = 646  
Top = 43  
Visible = .T.  
Width = 40  
TabIndex = 16  
ForeColor = 255,255,255  
Name = "Embossedlabel3"  
Enabled = F.  
Height = 21  
Left = 680  
TabIndex = 26  
Top = 80  
Visible = .F.  
Width = 80  
Name = "Text4"  
Enabled = .F.  
Height = 21  
Left = 560  
TabIndex = 26  
Top = 10  
Visible = .T.  
Width = 200  
Name = "Text5"  
Caption = "Clave de usuario"  
Height = 13  
Left = 490  
Top = 68  
Visible = F.  
Width = 90  
TabIndex = 14  
Name = "Embossedlabel4"  
ColumnCount = 2  
ColumnWidths = "61,207"  
RowSourceType = 6  
RowSource =  
"est_cobro.num_ec,nomb_ec"  
FirstElement = 1  
Height = 21  
Left = 88  
NumberOfElements = 0
```

```
Style = 2  
TabIndex = 22  
Top = 148  
Visible = F.  
Width = 20  
Name = "Busca"  
PROCEDURE InteractiveChange  
* Cuando encuentra el elemento  
buscado  
GO record recno()  
regactual=recno()  
thisform.busca.visible=.F.  
thisform.Navegar.visible=.T.
```

```
* Reset from prior  
thisform.Refresh()
```

```
ENDPROC
```

```
Picture = monitr01.ico  
Stretch = 1  
BackStyle = 0  
Height = 40  
Left = 40  
Top = 20  
Width = 40  
Name = "Image1"
```

```
*****
```

```
C:\COBRATEL\CPT0430.SCX  
Left = 1  
Top = 220  
Width = 520  
Height = 200  
Name = "Dataenvironment"  
Alias = "usuarios"  
BufferModeOverride = 5  
ORDER = "cve_usua"  
Database = dbCOBRATEL.dbc  
CursorSource = "usuarios"  
Name = "Cursor1"  
Left = 150  
Top = 20  
Width = 90  
Height = 90  
Alias = "procesos"  
Database = dbCOBRATEL.dbc  
CursorSource = "procesos"  
Name = "Cursor2"  
Alias = "grupos"  
Database = dbCOBRATEL.dbc  
CursorSource = "grupos"  
Name = "CURSOR3"  
ScaleMode = 3  
Height = 574  
Width = 793  
DoCreate = .T.  
ShowTips = .T.  
AutoCenter = .T.  
Picture = fondo.bmp  
Caption = "COBRATEL®  
Mantenimiento al catálogo de  
usuarios."  
Closable = .T.  
Visible = F.  
WindowType = 0  
WindowState = 2  
LockScreen = .F.  
Name = "Form1"  
layoutsty.Shape1 Name = "Shape1"
```

```

layoutsty Shape2.Name = "Shape2"
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Name = "Shape4"
layoutsty.Left = 4004
layoutsty.Visible = F
layoutsty.TabIndex = 20
layoutsty.Name = "layoutsty"
Label1.Caption = "Catalogo de usuarios"
Label1.Left = 100
Label1.Top = 30
Label1.Width = 249
Label1.TabIndex = 21
Label1.ForeColor = 255.255.255
Label1.Name = "Label1"
Shape1.Top = 389
Shape1.Left = 22
Shape1.Width = 700
Shape1.Name = "Shape1"
Shape2.Top = 390
Shape2.Left = 22
Shape2.Width = 700
Shape2.Name = "Shape2"
Shape3.Top = 389
Shape3.Left = 23
Shape3.Width = 700
Shape3.Name = "Shape3"
Shape4.Top = 390
Shape4.Left = 23
Shape4.Width = 700
Shape4.Name = "Shape4"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

```

```
SELECT USUARIOS
```

```
GO top
regactual=recno()
```

```

ENDPROC
Caption = "Clave"
Height = 13
Left = 110
Top = 146
Visible = T.
Width = 82
TabIndex = 8
ForeColor = 255.255.255
Name = "LBLEVE_USUA1"
ControlSource = "usuarios.cve_usua"
Enabled = F
Format = ""
InputMask = "XXXXXX"
Left = 190
TabIndex = 1
Top = 140
Visible = T
Width = 66
Name = "CVE_USUA1"
PROCEDURE Valid
* VALIDAR CLAVE

```

```

IF !EMPTY(THIS.VALUE)
MIREC=RECNO()
GO top
COUNT for
cve_usua=alltrim(this.value) to hay

```

```
GO RECORD MIREC
```

```

IF hay > 1 && ya habia una clave
como esta
MESSAGEBOX("Esta clave ya
esta registrada. ;Verifique por
favor".0+16+0,'Error.')
```

```

RETURN .F. &&no puede dejar el
campo
ELSE
thisform.pass_usu1.enabled=.f.
thisform.gpo_usu1.enabled=.T.
thisform.nom_usu1.enabled=.T.
thisform.dir_usu1.enabled=.T.
thisform.tel_usu1.enabled=.T.
thisform.vi_co_us1.enabled=.T.
RETURN .T. &&pasa al siguiente
campo
ENDIF
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ;Verifique por
favor".0+16+0,'Error.')
```

```

RETURN .F.
ENDIF

ENDPROC
Caption = "Password"
Height = 13
Left = 110
Top = 170
Visible = .T.
Width = 82
TabIndex = 9
ForeColor = 255.255.255
Name = "LBLPASS_USU1"
ControlSource = "usuarios.pass_usu"
Enabled = .F.
Format = ""
InputMask = "XXXXXX"
Left = 190
TabIndex = 2
Top = 164
Visible = .T.
Width = 60
Name = "PASS_USU1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ;Verifique por
favor".0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Grupo"
Height = 13
Left = 110
Top = 194
Visible = .T.
Width = 82
TabIndex = 10
ForeColor = 255.255.255
Name = "LBLGPO_USU1"
Caption = "Nombre"
Height = 13

```

```

Left = 110
Top = 218
Visible = T.
Width = 82
TabIndex = 11
ForeColor = 255.255.255
Name = "LBLNOM_USU1"
ControlSource = "usuarios.nom_usu"
Enabled = .F.
Format = ""
InputMask =
"XXXXXXXXXXXXXXXXXXXX"
Left = 190
TabIndex = 4
Top = 212
Visible = .T.
Width = 150
Name = "NOM_USU1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ;Verifique por
favor".0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Dirección"
Height = 13
Left = 110
Top = 242
Visible = .T.
Width = 82
TabIndex = 12
ForeColor = 255.255.255
Name = "LBLDIR_USU1"
ControlSource = "usuarios.dir_usu"
Enabled = F.
Format = ""
InputMask =
"XXXXXXXXXXXXXXXXXXXX"
Left = 190
TabIndex = 5
Top = 236
Visible = .T.
Width = 150
Name = "DIR_USU1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ;Verifique por
favor".0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Teléfono"
Height = 13
Left = 110
Top = 266
Visible = .T.
Width = 82
TabIndex = 13
ForeColor = 255.255.255

```

```

Name = "LBLTEL_USU1"
ControlSource = "usuarios.tel_usu"
Enabled = .F.
Format = "!"
InputMask = "XXXXXXXXXX"
Left = 190
TabIndex = 6
Top = 260
Visible = .T.
Width = 90
Name = "TEL_USU1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ¡Verifique por
favor!",0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Caption = "Via de cobro"
Height = 13
Left = 110
Top = 290
Visible = .T.
Width = 82
TabIndex = 14
ForeColor = 255,255,255
Name = "LBLVI_CO_US1"
Alignment = 1
ControlSource = "usuarios.vi_co_us"
Enabled = .F.
Left = 190
TabIndex = 7
Top = 284
Visible = .T.
Width = 48
Name = "VI_CO_US1"
PROCEDURE Valid
*VALIDAR
IF !EMPTY(THIS.VALUE)
RETURN .t.
ELSE
MESSAGEBOX("Este dato no
puede estar vacio. ¡Verifique por
favor!",0+16+0,'Error.')
```

```

RETURN .f.
ENDIF

ENDPROC
Top = 400
Left = 140
Visible = .T.
TabIndex = 19
oldTalk = OFF
viewkey = ("")
ParentKey =
ViewType = 3
GridAlias =
topfile = .T.
endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
oldreprocess = 0
previewmode = .F.
```

```

previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Name = "cmdPrev"
cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros especificos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclose.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = ..\archivos
de
programa\devstudio\vfplwizards\wizb
mps\wznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Name = "cmdDelete"
```

```

cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programa\devstudio\vfplwizards\wizb
mps\wzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdTop.Click
*botón primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click
*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*botón búsqueda
thisform.Navegar.visible=.F.
thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*botón imprime
#DEFINE C_MAKEREPO_LOC
"Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC
```



```

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IF
CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;

!CURSORGETPROP("SendUpdates")
)
=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
APPEND BLANK

OTHERWISE

** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IPrimeKey = .F.
&&is child key primary
index
IF
!EMPTY(THIS.Parent.oDataRelation)
oRel =
EVAL(THIS.Parent.oDataRelation)
ENDIF
DO CASE
CASE TYPE("m.oRel") =
"O" &&data environment
WITH oRel
cPapaAlias = .ParentAlias
cPapaKey =
.RelationalExpr
cChildAlias = .ChildAlias
cChildKey = .ChildOrder
ENDWITH
oRel = .NULL.
&&reset
CASE
!EMPTY(THIS.Parent.viewkey)
&&using views
cPapaAlias =
ALIAS()
cPapaKey =
THIS.Parent.ParentKey
cChildAlias =
THIS.Parent.GridAlias

```

```

cChildKey =
THIS.Parent.viewkey
* Need to check
if Key field is updatable
IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;
ATC(" "+m.cChildKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",",)=0
!BadViewKey = .T.
ENDIF
OTHERWISE
cPapaAlias =
ALIAS()
cPapaKey =
RELATION(1)
cChildAlias =
THIS.Parent.GridAlias
cChildKey =
ORDER(m.cChildAlias)
ENDCASE
* Test to see if both keys
are of the same data type
SELECT
(THIS.Parent.GridAlias)
cGridKeyType = ""
IF
EMPTY(THIS.Parent.viewkey) AND
!EMPTY(m.cChildKey)
&&regular table used
* Get the Child
Key field since only index tag is here
* if these is an
expression, then default to blank.
cTagName = ""
FOR i = 1 TO
TagCount("")
IF
UPPER(TAG(m.i)) ==
UPPER(m.cChildKey)
cTagName = KEY(m.i)
IPrimeKey =
PRIMARY(m.i)
EXIT
ENDIF
ENDFOR
* Check if we
have an index expression here.
IF
ATC("(",m.cTagName)#0 OR
ATC("+",m.cTagName)#0
cChildKey = ""
ELSE
cChildKey = m.cTagName

```

```

cGridKeyType =
TYPE(m.cChildKey)
ENDIF
ENDIF
IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;
!CURSORGETPROP("SendUpdates
",m.cPapaAlias)
.NoSendParentUpdates =
.T.
ENDIF
IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;
!CURSORGETPROP("SendUpdates
",m.cChildAlias)
.NoSendChildUpdates = .T.
ENDIF
* Check if have two views
and both do not send updates
IF
m.NoSendParentUpdates AND
m.NoSendChildUpdates
=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
SELECT
(THIS.Parent.OldAlias )
IF
!EMPTY(m.cGridKeyType) AND
m.cGridKeyType #
TYPE(m.cPapaKey)
!BadChildKey =
.T.
ENDIF
DO CASE
CASE
m.NoSendParentUpdates AND
(m.NoSendViewKey OR
m.NoSendChildKey)
=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
CASE m.IPrimeKey AND
m.NoSendParentUpdates
=MESSAGEBOX(C_NOUP
DATE2_LOC)
RETURN
CASE
m.NoSendChildUpdates
&&child view does not
send updates

```

```

IF
MESSAGEBOX(C_NOCHILDUPDAT
E_LOC.MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadViewKey
&&view key in not
updatable
IF
MESSAGEBOX(C_KEYFLDNOUPD
ATE_LOC.MB_Q_YESNO) #
MB_A_YES

RETURN
ENDIF
APPEND BLANK
CASE m.IBadChildKey
&&bad child key -- different
data type from parent
IF
MESSAGEBOX(C_BADCHILDKEY_
LOC.MB_Q_YESNO) # MB_A_YES

RETURN
ENDIF
APPEND BLANK
OTHERWISE
* Need to check
if Key field is updatable

IUpdatableParentKey = .T.
IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND;

ATC(" "+m.cPapaKey+" ",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0

cPapaKey = ""

IUpdatableParentKey = .F.
ENDIF

* Check for
expression in cPapaKey if so, then
do not update record with keyvalue
IF
ATC(" "+m.cPapaKey)#0 OR
ATC(" "+m.cPapaKey)#0

cPapaKey = ""
ENDIF

nSaveSess =
SET("DATASESSION")
oAddRec =
CREATE("GridAddRec")
IF
TYPE("m.oAddRec") # "O"

=MESSAGEBOX(C_NOO
BJ_LOC)

RETURN

```

```

ENDIF

oAddRec.ChildPrimaryKey
= m.IPrimeKey

oAddRec.UpdatableParent
Key = m.IUpdatableParentKey

oAddRec.NoSendUpdates
= m.INoSendParentUpdates

oAddRec.KeyField =
m.cPapaKey

oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey),"")

oAddRec.runaddform()
IF
oAddRec.AddOption =
OPT_CANCEL

RETURN
ENDIF

SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)

IF
TYPE("oAddRec.KeyValue") = "C"

oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
ENDIF

* Add record to
parent table

nSaveRec =
RECNO()

IF
INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
IF
EMPTY(m.cPapaKey)

APPEND BLANK IN
(m.cPapaAlias)

ELSE

INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
ENDIF
ENDIF

nSaveRec2 =
RECNO()

* Add child
record

IF
INLIST(oAddRec.AddOption,OPT_A
DD_CHILD,OPT_ADD_BOTH)

* Need
to check

GO

m.nSaveRec

```

```

IF
EMPTY(m.cChildKey) OR
TYPE("m.cChildKey")#TYPE('oAddR
ec.KeyValue')

APPEND BLANK IN
(m.cChildAlias)

ELSE

INSERT INTO
(m.cChildAlias) ((m.cChildKey))
VALUES(oAddRec.KeyValue)
ENDIF
GO

m.nSaveRec2

ENDIF
ENDCASE

this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode =
THIS.Parent.EditMode
this.Parent.topfile = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
thisform.pass_usu1.enabled=.F
thisform.gpo_usu1.enabled=.F.
thisform.norm_usu1.enabled=.F.
thisform.dir_usu1.enabled=.F.
thisform.tel_usu1.enabled=.F.
thisform.vi_co_us1.enabled=.F.

ENDPROC
PROCEDURE cmdDelete.Click
*botón borra

* Tipo de cuadro de diálogo
* 4 + 32 + 256
* 4 = Botones Si y No
* 32 = Icono de PREGUNTA
* 256 = el segundo botón es el de
default

* Respuestas
* 6 = SI
* 7 = NO

IF MESSAGEBOX('¿ Desea eliminar
este registro ?, 4+32+256 ,Confirme
por favor...') = 6
Delete
IF
THIS.Parent.updaterows()
&&success

* Success
IF !EOF()
Skip 1

```

```

ENDIF
IF EOF() AND
!BOF()
Skip -1
ENDIF
regactual=recno()
ENDIF
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.
&&.....
..... registro en bitácora
SELECT procesos
APPEND blank
REPLACE tarea_bp with "Baja al
catálogo de usuarios"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT USUARIOS
&&.....
.. cerrar procesos
ENDIF

IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
PROCEDURE cmdEdit.Click
*botón edita
#DEFINE C_NOUPDATE_LOC
"You cannot edit because
the view(s) selected does not send
updates."

LOCAL
INoSendParentUpdates,iNoSendChildUpdates
LOCAL aTablesUsed,nTablesUsed,i

** Reverting record
IF THIS.Parent.EditMode
thisform.Text4.VALUE=SPACE(7)
&&.....<<<<<<
<<<<<<<

IF THIS.Parent.usedataenv
SELECT
(THIS.Parent.OldAlias)
IF
CURSORGETPROP("Buffering")>1

=TableRevert(.T.)
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)

SELECT
(THIS.Parent.GridAlias)
IF
CURSORGETPROP("Buffering")>1

=TableRevert(.T.)
ENDIF
ENDIF
ELSE

```

```

DIMENSION
aTablesUsed[1]
m.nTablesUsed
= AUUSED(aTablesUsed)
FOR i = 1 TO
m.nTablesUsed
IF
CURSORGETPROP("Buffering",aTablesUsed[m,i,1])>1

=TableRevert(.T.,aTablesUsed[m,i,1])
ENDIF
ENDIF

* Go back to original place
SELECT
(THIS.Parent.OldAlias)
IF RECCOUNT() <
THIS.Parent.oldrec &&added record
at EOF()
GO TOP
ELSE
GO
THIS.Parent.oldrec
ENDIF
ELSE
thisform.Text4.VALUE="MODIFICAR"
&&.....<<<<<<
<<<<<<<

* Check to see if view
allows updates
IF
CURSORGETPROP("SourceType")#
3 AND
!CURSORGETPROP("offline") AND;

!CURSORGETPROP("SendUpdates")

INoSendParentUpdates =
.T.
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)
AND ;

CURSORGETPROP("SourceType",THIS.Parent.GridAlias)#3 AND ;

!CURSORGETPROP("offline",THIS.Parent.GridAlias) AND ;

!CURSORGETPROP("SendUpdates",THIS.Parent.GridAlias)

INoSendChildUpdates= .T.
ENDIF

IF
(m.INoSendChildUpdates AND
m.INoSendParentUpdates) OR;

(EMPTY(THIS.Parent.GridAlias) AND
m.INoSendParentUpdates)

```

```

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
ENDIF

** Editing record
this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
thisform.CVE_USUA1.enabled=.F.
thisform.pass_usu1.enabled=.F.

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Usuario"
Height = 31
Left = 80
Top = 100
Width = 80
TabIndex = 22
ForeColor = 255,255,255
Name = "Label2"
ColumnCount = 2
ColumnWidths = "64, 163"
RowSourceType = 6
RowSource =
"usuarios.cve_usua,nom_usu"
FirstElement = 1
Height = 21
Left = 80
NumberOfElements = 0
Style = 2
TabIndex = 23
Top = 140
Visible = .F.
Width = 20
Name = "Busca"
PROCEDURE InteractiveChange
* Cuando encuentra el elemento
buscado
GO record recno()
regactual=recno()
thisform.busca.visible=.F.
thisform.Navegar.visible=.T.

```

```
* Reset from prior  
thisform.Refresh()
```

```
ENDPROC  
Enabled = .F.  
Height = 21  
Left = 480  
TabIndex = 24  
Top = 80  
Visible = .F.  
Width = 80  
Name = "Text1"  
Caption = "Usuario"  
Height = 13  
Left = 520  
Top = 13  
Visible = .T.  
Width = 40  
TabIndex = 17  
ForeColor = 255,255,255  
Name = "Embossedlabel1"  
Enabled = .F.  
Height = 21  
Left = 590  
TabIndex = 25  
Top = 80  
Visible = .F.  
Width = 80  
Name = "Text2"  
Caption = "Grupo"  
Height = 13  
Left = 590  
Top = 68  
Visible = .F.  
Width = 40  
TabIndex = 15  
Name = "Embossedlabel2"  
Enabled = F.  
Height = 21  
Left = 680  
TabIndex = 26  
Top = 40  
Width = 80  
Name = "Text3"  
Caption = "Fecha"  
Height = 13  
Left = 647  
Top = 44  
Visible = .T.  
Width = 40  
TabIndex = 18  
ForeColor = 255,255,255  
Name = "Embossedlabel3"  
Enabled = F  
Height = 21  
Left = 680  
TabIndex = 27  
Top = 80  
Visible = F  
Width = 80  
Name = "Text4"  
Enabled = F  
Height = 21  
Left = 560  
TabIndex = 28  
Top = 10  
Visible = .T.  
Width = 200  
Name = "Text5"  
Caption = "Clave de usuario"
```

```
Height = 13  
Left = 480  
Top = 68  
Visible = .F.  
Width = 90  
TabIndex = 16  
Name = "Embossedlabel4"  
ColumnCount = 2  
ColumnWidths = "70,160"  
RowSourceType = 6  
RowSource =  
"grupos.cve_gru,nom_gru"  
ControlSource = "Usuarios.Gpo_usu"  
FirstElement = 1  
Height = 22  
Left = 190  
NumberOfElements = 0  
Style = 2  
TabIndex = 3  
Top = 188  
Width = 61  
Name = "GPO_USU1"  
Picture = misc27.ico  
Stretch = 1  
BackStyle = 1  
Height = 40  
Left = 40  
Top = 20  
Width = 40  
Name = "Image1"  
***** *  
C:\COBRATEL\CPT0440.SCX  
Left = 1  
Top = 220  
Width = 520  
Height = 200  
Name = "Dataenvironment"  
Alias = "grupos"  
BufferModeOverride = 5  
ORDER = "cve_gru"  
Database = dbCOBRATEL.dbc  
CursorSource = "grupos"  
Name = "Cursor1"  
Left = 150  
Top = 20  
Width = 90  
Height = 90  
Alias = "procesos"  
Database = dbCOBRATEL.dbc  
CursorSource = "procesos"  
Name = "Cursor2"  
ScaleMode = 3  
Height = 574  
Width = 793  
DoCreate = .T.  
ShowTips = .T.  
AutoCenter = .T.  
  
Picture = fondo.bmp  
Caption = "COBRATEL@.  
Mantenimiento al catálogo de  
grupos."  
Visible = .F.  
WindowState = 2  
LockScreen = .F.  
Name = "Form1"  
layoutsty.Shape1.Name = "Shape1"  
layoutsty.Shape2.Name = "Shape2"  
layoutsty.Shape3.Name = "Shape3"  
layoutsty.Shape4.Name = "Shape4"
```

```
layoutsty.Left = 4004  
layoutsty.Visible = .F.  
layoutsty.Name = "layoutsty"  
Label1.Caption = "Catálogo de  
grupos"  
Label1.Left = 100  
Label1.Top = 30  
Label1.Width = 229  
Label1.ForeColor = 255,255,255  
Label1.Name = "Label1"  
Shape1.Top = 393  
Shape1.Left = 10  
Shape1.Width = 700  
Shape1.Name = "Shape1"  
Shape2.Top = 394  
Shape2.Left = 10  
Shape2.Width = 700  
Shape2.Name = "Shape2"  
Shape3.Top = 393  
Shape3.Left = 10  
Shape3.Width = 700  
Shape3.Name = "Shape3"  
Shape4.Top = 394  
Shape4.Left = 10  
Shape4.Width = 700  
Shape4.Name = "Shape4"  
PROCEDURE Activate  
* Al iniciar...  
thisform.Text1.value=usuario  
thisform.text2.value=grupo  
thisform.text3.value=date()  
thisform.text5.value=nombre  
GO top  
regactual=recno()  
  
ENDPROC  
Caption = "Clave"  
Height = 13  
Left = 125  
Top = 144  
Visible = .T.  
Width = 28  
ForeColor = 255,255,255  
Name = "LBLCVE_GRU1"  
ControlSource = "grupos.cve_gru"  
Enabled = .F.  
Format = ""  
InputMask = "XXX"  
Left = 160  
Top = 140  
Visible = .T.  
Width = 40  
Name = "CVE_GRU1"  
PROCEDURE Valid  
* VALIDAR CLAVE  
  
IF !EMPTY(THIS.VALUE)  
MIREC=RECNO()  
GO top  
COUNT for  
cve_gru=alltrim(this.value) to hay  
GO RECORD MIREC  
  
IF hay > 1 && ya habia una clave  
como esta  
MESSAGEBOX('Esta clave ya  
esta registrada. ¡Verifique por  
favor!,0+16+0,'Error.')
```

```
RETURN .F. &&no puede dejar el  
campo
```

```

ELSE
  thisform.m011.enabled=.T.
  thisform.m021.enabled=.T.
  thisform.m031.enabled=.T.
  thisform.m041.enabled=.T.
  thisform.m051.enabled=.T.
  thisform.m061.enabled=.T.
  RETURN .T. &&pasa al siguiente
campo
ENDIF
ELSE
  MESSAGEBOX("Este dato no
puede estar vacio. ;Verifique por
favor!.0+16+0.'Error.').
  RETURN .F.
ENDIF

ENDPROC
Caption = "Transferencia de
archivos"
Height = 13
Left = 120
Top = 200
Visible = .T.
Width = 144
ForeColor = 255,255,255
Name = "LBLM011"
Top = 200
Left = 260
Width = 18
AutoSize = .F.
Caption = ""
ControlSource = "grupos.m01"
Enabled = .F.
Visible = .T.
Name = "M011"
Caption = "Procesos especiales"
Height = 13
Left = 120
Top = 240
Visible = .T.
Width = 144
ForeColor = 255,255,255
Name = "LBLM021"
Top = 240
Left = 260
Width = 18
AutoSize = .T.
Caption = ""
ControlSource = "grupos.m02"
Enabled = .F.
Visible = .T.
Name = "M021"
Caption = "Generación de reportes"
Height = 13
Left = 120
Top = 280
Visible = .T.
Width = 144
ForeColor = 255,255,255
Name = "LBLM031"
Top = 280
Left = 260
Width = 18
AutoSize = .T.
Caption = ""
ControlSource = "grupos.m03"
Enabled = .F.
Visible = .T.
Name = "M031"

```

```

Caption = "Mantenimiento de
catálogos"
Height = 13
Left = 380
Top = 200
Visible = .T.
Width = 144
ForeColor = 255,255,255
Name = "LBLM041"
Top = 200
Left = 520
Width = 18
AutoSize = .T.
Caption = ""
ControlSource = "grupos.m04"
Enabled = .F.
Visible = .T.
Name = "M041"
Caption = "Utillerias"
Height = 13
Left = 380
Top = 240
Visible = .T.
Width = 144
ForeColor = 255,255,255
Name = "LBLM051"
Top = 240
Left = 520
Width = 18
AutoSize = .T.
Caption = ""
ControlSource = "grupos.m05"
Enabled = .F.
Visible = .T.
Name = "M051"
Caption = "Bitácoras"
Height = 13
Left = 380
Top = 280
Visible = .T.
Width = 144
ForeColor = 255,255,255
Name = "LBLM061"
Top = 280
Left = 520
Width = 18
AutoSize = .T.
Caption = ""
ControlSource = "grupos.m06"
Enabled = .F.
Visible = .T.
Name = "M061"
Height = 21
Left = 490
TabIndex = 23
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.

```

```

Height = 21
Left = 590
TabIndex = 24
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 646
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Top = 400
Left = 140
Visible = .T.
TabIndex = 18
oldTalk = OFF
viewkey = ("")
ParentKey =
ViewType = 3
GridAlias =
topfile = .T.
endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
oldreprocess = 0
previewmode = .F.
previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Name = "cmdPrev"

```

```

cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primer"
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros específicos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.
cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclos.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = .\archivos
de
programa\devstudio\wfp\wizards\wizb
mpslwznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = .\archivos
de
programa\devstudio\wfp\wizards\wizb
mpslwzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS parent.nworkarea)

```

```

IF !BOF()
    Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
    Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdTop.Click
*boton primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()
this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click
*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*boton búsqueda
thisform.Navegar.visible=.F.
thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*botón imprime
#DEFINE C_MAKEREPO_LOC
"Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC
"Error opening table.
Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC
"Pick report:"

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")

```

```

cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

* Handling for Private data sessions
#IF 0
    IF m.nSaveSess # 1
        SET
        DATASESSION TO 1
        SELECT 0
        IF
        !EMPTY(m.cSaveData)
            OPEN
            DATABASE (m.cSaveData)
            ENDIF
            IF
            USED(m.cSaveAlias)
                SELECT (m.cSaveAlias)
                IF
                CURSORGETPROP("SourceName")
                #m.cSaveSource
                    cDiffSource =
                    CURSORGETPROP("SourceName")
                    USE IN (m.cSaveAlias)
                    SELECT 0
                    ENDIF
                    ENDIF
                    IF
                    EMPTY(ALIAS())
                        USE
                        (m.cSaveSource) AGAIN ALIAS
                        (m.cSaveAlias) SHARED
                        IF
                        EMPTY(ALIAS())
                            =MESSAGEBOX(C_NOO
                            PEN_LOC)
                            RETURN
                            ENDIF
                            ENDIF
                            #ENDIF
                            IF FILE(m.cRepName)
                                REPORT FORM
                                (m.cRepName) PREVIEW NOWAIT
                                ELSE
                                    m.cRepName =
                                    GETFILE("frx",C_GETFILEPROMPT
                                    _LOC,,"")
                                    IF !EMPTY(m.cRepName)
                                        FILE(m.cRepName)
                                            * User
                                            pressed Open button
                                            REPORT FORM
                                            (m.cRepName) PREVIEW NOWAIT
                                            ELSE
                                                * User
                                                pressed New button

```

```

DO
HOME()+"wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
    ENDIF
ENDIF
* Private data session code
#IF 0
    IF !EMPTY(cDiffSource)
        USE
(m.cDiffSource) IN 0
    ENDIF
SET DATASESSION TO
m.nSaveSess
SELECT (m.cSaveAlias)
#ENDIF
IF THISFORM.ShowWindow = 2
    Activate Window
(THISFORM.Name)
ENDIF
ENDPROC
PROCEDURE cmdExit.Click
*botón salir
thisform.Release
DO form cpt0400 &&menu de
mAntenimiento de catálogos
ENDPROC
PROCEDURE cmdAdd.Click
*botón agrega
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
#DEFINE MB_A_YES
6
#DEFINE
C_KEYFLDNOUPDATE_LOC
"The field relating the grid's
view to the parent data source is not
updatable. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
"The fields
relating the parent and child tables
are not the same data type. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
"The child data source is a
view and does not send updates. "+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_NOOBJ_LOC
"Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
"You cannot add a new

```

```

record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
"You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."
LOCAL
oSearchDlg,oAddRec,cChildAlias,c
PapaAlias,i,IPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,IBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
IBadChildKey,IUpdatableParentKey,I
NoSendParentUpdates,I NoSendChil
dUpdates
DO CASE
CASE THIS.Parent.EditMode
&&.....
..... registro en bitácora
SELECT procesos
APPEND blank
IF
ALLTRIM(THISFORM.Text4.VALUE)
="MODIFICAR"
REPLACE tarea_bp with
"Cambio al catálogo de grupos"
ELSE
REPLACE tarea_bp with "Alta al
catálogo de grupos"
ENDIF
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT GRUPOS
&&.....
.. cerrar procesos
thisform.Text4.VALUE=SP
ACE(7)
&&.....<<<<<<
<<<<<<
** Code for saving record
this.Parent.updaterows()
CASE EMPTY(THIS.Parent.GridRef)
&& not using
Grid object
thisform.Text4.VALUE="A
GREGAR"
&&.....<<<<<<
<<<<<<
** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IF
CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;

```

```

!CURSORGETPROP("Sen
dUpdates")
=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF
APPEND BLANK
OTHERWISE
** Code for adding record
this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
IPrimeKey = .F.
&&is child key primary
index
IF
!EMPTY(THIS.Parent.oDataRelation)
oRel =
EVAL(THIS.Parent.oDataRelation)
ENDIF
DO CASE
CASE TYPE("m.oRel") =
&&data environment
WITH oRel
cPapaAlias = .ParentAlias
cPapaKey =
.RelationalExpr
cChildAlias = .ChildAlias
cChildKey = .ChildOrder
ENDWITH
oRel = .NULL.
&&reset
CASE
!EMPTY(THIS.Parent.viewkey)
&&using views
cPapaAlias =
ALIAS()
cPapaKey =
THIS.Parent.ParentKey
cChildAlias =
THIS.Parent.GridAlias
cChildKey =
THIS.Parent.viewkey
* Need to check
if Key field is updatable
IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;
ATC(" "+m.cChildKey+",",",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0

```



```

        IBadViewKey = .T.
        ENDIF
    OTHERWISE
        cPapaAlias =
    ALIAS()
        cPapaKey =
    RELATION(1)
        cChildAlias =
    THIS.Parent.GridAlias
        cChildKey =
    ORDER(m.cChildAlias)
    ENDCASE

    * Test to see if both keys
    are of the same data type
    SELECT
    (THIS.Parent.GridAlias)
    cGridKeyType = ""

    IF
    EMPTY(THIS.Parent.viewkey) AND
    !EMPTY(m.cChildKey)
        &&regular table used
        * Get the Child
        Key field since only index tag is here
        * if these is an
        expression, then default to blank.
        cTagName = ""
        FOR i = 1 TO
    TagCount("")
        IF
    UPPER(TAG(m.i)) ==
    UPPER(m.cChildKey)

        cTagName = KEY(m.i)

    IPrimeKey =
    PRIMARY(m.i)

    EXIT
    ENDIF
    ENDFOR

    * Check if we
    have an index expression here.
    IF
    ATC("",m.cTagName)#0 OR
    ATC("+",m.cTagName)#0

        cChildKey = ""
    ELSE

        cChildKey = m.cTagName

        cGridKeyType =
    TYPE(m.cChildKey)
    ENDIF
    ENDIF

    IF
    CURSORGETPROP("SourceType",
    m.cPapaAlias)#3 AND
    !CURSORGETPROP("offline",m.cPa
    paAlias) AND ;

    !CURSORGETPROP("SendUpdates
    ",m.cPapaAlias)

```

```

        !NoSendParentUpdates =
    .T.
    ENDIF
    IF
    CURSORGETPROP("SourceType",
    m.cChildAlias)#3 AND
    !CURSORGETPROP("offline",m.cCh
    ildAlias) AND ;

    !CURSORGETPROP("SendUpdates
    ",m.cChildAlias)

        !NoSendChildUpdates = .T.
    ENDIF

    * Check if have two views
    and both do not send updates
    IF
    m.NoSendParentUpdates AND
    m.NoSendChildUpdates

        =MESSAGEBOX(C_NOUP
    DATE_LOC)
    RETURN
    ENDIF

    SELECT
    (THIS.Parent.OldAlias )
    IF
    !EMPTY(m.cGridKeyType) AND
    m.cGridKeyType #
    TYPE(m.cPapaKey)
        IBadChildKey =
    .T.
    ENDIF

    DO CASE
    CASE
    m.NoSendParentUpdates AND
    (m.IBadViewKey OR
    m.IBadChildKey)

        =MESSAGEBOX(C_NOUP
    DATE_LOC)
    RETURN
    CASE m.IPrimeKey AND
    m.NoSendParentUpdates

        =MESSAGEBOX(C_NOUP
    DATE2_LOC)
    RETURN
    CASE
    m.NoSendChildUpdates
    &&child view does not
    send updates
    IF
    MESSAGEBOX(C_NOCHILDUPDAT
    E_LOC,MB_Q_YESNO) #
    MB_A_YES

        RETURN
    ENDIF
    APPEND BLANK
    CASE m.IBadViewKey
    &&view key in not
    updatable
    IF
    MESSAGEBOX(C_KEYFLDNOUPD

```

```

    ATE_LOC,MB_Q_YESNO) #
    MB_A_YES

    RETURN
    ENDIF
    APPEND BLANK
    CASE m.IBadChildKey
    &&bad child key -- different
    data type from parent
    IF
    MESSAGEBOX(C_BADCHILDKEY_
    LOC,MB_Q_YESNO) # MB_A_YES

    RETURN
    ENDIF
    APPEND BLANK
    OTHERWISE
        * Need to check
        if Key field is updatable

        !UpdatableParentKey = .T.
        IF
        CURSORGETPROP("SourceType",
        m.cPapaAlias)#3 AND
        !CURSORGETPROP("offline",m.cPa
        paAlias) AND;

            ATC(", "+m.cPapaKey+"",
            "+CURSORGETPROP("UpdatableFi
            eldList",m.cPapaAlias)+",")=0
            *
            cPapaKey = ""

            !UpdatableParentKey = .F.
        ENDIF

        * Check for
        expression in cPapaKey if so, then
        do not update record with keyvalue
        IF
        ATC("",m.cPapaKey)#0 OR
        ATC("+",m.cPapaKey)#0

            cPapaKey = ""
        ENDIF

        nSaveSess =
    SET("DATASESSION")
        oAddRec =
    CREATE("GridAddRec")
        IF
        TYPE("m.oAddRec") # "O"

            =MESSAGEBOX(C_NOO
            BJ_LOC)
        RETURN
        ENDIF

        oAddRec.ChildPrimaryKey
    = m.IPrimeKey

        oAddRec.UpdatableParent
    Key = m.IUpdatableParentKey

        oAddRec.NoSendUpdates
    = m.NoSendParentUpdates

        oAddRec.KeyField =
    m.cPapaKey

```

```

oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey),"")

oAddRec.runaddform()
IF
oAddRec.AddOption =
OPT_CANCEL

RETURN
ENDIF

SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)

IF
TYPE("oAddRec.KeyValue") = "C"

oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
ENDIF

* Add record to
parent table
nSaveRec =
RECNO()
IF
INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
IF
EMPTY(m.cPapaKey)

APPEND BLANK IN
(m.cPapaAlias)
ELSE

INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
ENDIF
ENDIF
nSaveRec2 =
RECNO()

* Add child
record
IF
INLIST(oAddRec.AddOption,OPT_A
DD_CHILD,OPT_ADD_BOTH)
* Need
to check
GO
m.nSaveRec
IF
EMPTY(m.cChildKey) OR
TYPE("m.cChildKey")#TYPE("oAddR
ec.KeyValue")

APPEND BLANK IN
(m.cChildAlias)
ELSE

INSERT INTO
(m.cChildAlias) ((m.cChildKey))
VALUES(oAddRec.KeyValue)
ENDIF

```

```

m.nSaveRec2
GO
ENDIF
ENDCASE
ENDCASE

this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode =
THIS.Parent.EditMode
this.Parent.topfile = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
thisform.m011.enabled=.F.
thisform.m021.enabled=.F.
thisform.m031.enabled=.F.
thisform.m041.enabled=.F.
thisform.m051.enabled=.F.
thisform.m061.enabled=.F.

ENDPROC
PROCEDURE cmdDelete.Click
*botón borra

* Tipo de cuadro de diálogo
* 4 + 32 + 256
* 4 = Botones Si y No
* 32 = Icono de PREGUNTA
* 256 = el segundo botón es el de
default

* Respuestas
* 6 = SI
* 7 = NO

IF MESSAGEBOX('¿ Desea eliminar
este registro?', 4+32+256, 'Confirme
por favor...') = 6
Delete
IF
THIS.Parent.updaterows()
&&success
* Success
IF !EOF()
Skip 1
ENDIF
IF EOF() AND
Skip -1
ENDIF
regactual=recno()
ENDIF
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

&&.....
registro en bitácora
SELECT procesos

```

```

APPEND blank
REPLACE tarea_bp with "Baja al
catálogo de grupos"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT GRUPOS
&&.....
.. cerrar procesos
ENDIF

IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
PROCEDURE cmdEdit.Click
*botón edita
#DEFINE C_NOUPDATE_LOC
"You cannot edit because
the view(s) selected does not send
updates."

LOCAL
INoSendParentUpdates,INoSendChil
dUpdates
LOCAL aTablesUsed,nTablesUsed,i

** Reverting record
IF THIS.Parent.EditMode
thisform.Text4.VALUE=SP
ACE(7)
&&.....<<<<<<
<<<<<<<

IF THIS.Parent.usedataenv
SELECT
(THIS.Parent.OldAlias)
IF
CURSORGETPROP("Buffering")>1

=TableRevert(.T.)
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)

SELECT
(THIS.Parent.GridAlias)
IF
CURSORGETPROP("Buffering")>1

=TableRevert(.T.)
ENDIF
ELSE
DIMENSION
aTablesUsed[1]
m.nTablesUsed
= AUSED(aTablesUsed)
FOR i = 1 TO
m.nTablesUsed
IF
CURSORGETPROP("Buffering",aTa
blesUsed[m,i,1])>1

=TableRevert(.T.,aTablesU
sed[m,i,1])
ENDIF
ENDFOR

```

```

ENDIF

* Go back to original place
SELECT
(THIS.Parent.OldAlias)
IF RECCOUNT() <
THIS.Parent.oldrec &&added record
at EOF()
GO TOP
ELSE
GO
THIS.Parent.oldrec
ENDIF
ELSE
thisform.Text4.VALUE="M
ODIFICAR"
&&*****<<<<<<
<<<<<<<<

* Check to see if view
allows updates
IF
CURSORGETPROP("SourceType")#
3 AND
!CURSORGETPROP("offline") AND;

!CURSORGETPROP("Sen
dUpdates")

INoSendParentUpdates =
.T.
ENDIF
IF
!EMPTY(THIS.Parent.GridAlias)
AND ;

CURSORGETPROP("SourceType",T
HIS.Parent.GridAlias)#3 AND ;

!CURSORGETPROP("offline",THIS.
Parent.GridAlias) AND ;

!CURSORGETPROP("SendUpdates
",THIS.Parent.GridAlias)

INoSendChildUpdates=.T.
ENDIF

IF
(m INoSendChildUpdates AND
m.INoSendParentUpdates) OR;

(EMPTY(THIS.Parent.Grid
Alias) AND
m.INoSendParentUpdates)

=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
ENDIF

this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
this.Parent.oldrec =
RECNO() &&save record in case
reverting
ENDIF

** Editing record

```

```

this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
thisform.CVE_GRU1.enabled=.F.

ENDPROC
ColumnCount = 0
ColumnWidths = ""
RowSourceType = 6
RowSource = "grupos.cve_gru"
FirstElement = 1
Height = 21
Left = 100
NumberOfElements = 0
Style = 2
TabIndex = 22
Top = 140
Visible = .F.
Width = 20
Name = "Busca"
PROCEDURE InteractiveChange
* Cuando encuentra el elemento
buscado
GO record recno()
regactual=recno()
thisform.busca.visible=.F.
thisform.Navegar.visible=.T.

* Reset from prior
thisform.Refresh()

ENDPROC
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 67
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Nombre"
Height = 13
Left = 220
Top = 145
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "Embossedlabel5"
ControlSource = "grupos.nom_gru"
Enabled = .F.

```

```

Format = "!"
Height = 22
InputMask =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX"
Left = 280
Top = 140
Visible = .T.
Width = 220
Name = "NOM_GRU1"
PROCEDURE Valid
* VALIDAR CLAVE

IF !EMPTY(THIS.VALUE)
MIREC=RECNO()
GO top
COUNT for
cve_gru=alltrim(this.value) to hay
GO RECORD MIREC

IF hay > 1 && ya habia una clave
como esta
MESSAGEBOX('Esta clave ya
esta registrada. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .F. &&no puede dejar el
campo
ELSE
thisform.m011.enabled=.T.
thisform.m021.enabled=.T.
thisform.m031.enabled=.T.
thisform.m041.enabled=.T.
thisform.m051.enabled=.T.
thisform.m061.enabled=.T.
RETURN .T. &&pasa al siguiente
campo
ENDIF
ELSE
MESSAGEBOX('Este dato no
puede estar vacío. ¡Verifique por
favor!',0+16+0,'Error.')
```

```

RETURN .F.
ENDIF

ENDPROC
Picture = misc29.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

*****
C:\COBRATEL\CPT0500.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú de
utillerias."
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0

```

```

Caption = "Utilerias"
Height = 30
Left = 100
Top = 40
Width = 100
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú principal"
StatusBarText = "Regresar al menú principal."
TabIndex = 9
Name = "Command10"
PROCEDURE Click
* Boton Salir
thisform.release
DO form menuprin
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 20
Left = 480
Height = 41
Width = 81
WordWrap = .T.
Caption = ""
StatusBarText = "Mantenimiento de parámetros."
TabIndex = 3
ToolTipText = ""
Visible = .F.
Name = "Command1"
PROCEDURE LostFocus
thisform.Label11.forecolor=rgb(255,255,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label11.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton mantenimiento de parametros
thisform.release
DO form cpt0510
ENDPROC

```

```

Top = 140
Left = 80
Height = 60
Width = 81
WordWrap = .T.
Picture = depura2.ico
Caption = ""
StatusBarText = "Depuración de bitácoras de procesos."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,255,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton depuracion de bitacora de procesos
thisform.release
DO form cpt0500B
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus

ENDPROC
BackStyle = 0
Caption = "Mantenimiento de parámetros"
Height = 21
Left = 570
Top = 30
Visible = .F.
Width = 180
ForeColor = 255,255,255
Name = "Label1"
WordWrap = .T.
BackStyle = 0
Caption = "Depuración"
Height = 30
Left = 170
Top = 160
Width = 90
ForeColor = 255,255,255
Name = "Label2"
Top = 220
Left = 80
Height = 60
Width = 81
WordWrap = .T.
Picture = bckup.ico
Caption = ""
StatusBarText = "Respaldo de archivos."
TabIndex = 4
ToolTipText = ""
Name = "Command6"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton respaldo de archivos
thisform.release
DO form cpt0560
ENDPROC
PROCEDURE GotFocus
thisform.label6.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
thisform.label6.forecolor=rgb(255,255,255)
ENDPROC
BackStyle = 0
Caption = "Respaldo de archivos"
Height = 21
Left = 170
Top = 240
Width = 200
ForeColor = 255,255,255
Name = "Label6"
Top = 140
Left = 380
Height = 60
Width = 81
WordWrap = .T.
Picture = resto.ico
Caption = ""
StatusBarText = "Restauración de respaldos."
TabIndex = 4
ToolTipText = ""
Name = "Command7"
PROCEDURE LostFocus
thisform.label7.forecolor=rgb(255,255,255)
ENDPROC
PROCEDURE GotFocus
thisform.label7.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton restauracion de respaldos
thisform.release
DO form cpt0570
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift, nXCoord, nYCoord
this.setfocus

ENDPROC
BackStyle = 0
Caption = "Restauración de respaldos"
Height = 21
Left = 470
Top = 160
Width = 200
ForeColor = 255,255,255
Name = "Label7"
Top = 220
Left = 380
Height = 60
Width = 81
WordWrap = .T.
Picture = key04.ico
Caption = ""

```

```

StatusBarText = "Cambio de
password."
TabIndex = 4
ToolTipText = ""
Name = "Command8"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton cambio de password
thisform.release
DO form cpt0580
ENDPROC
PROCEDURE GotFocus
thisform.label8.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.label8.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Cambio de password"
Height = 21
Left = 470
Top = 240
Width = 200
ForeColor = 255,255,255
Name = "Label8"
Picture = wrench.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

*****
C:\COBRATEL\CPT0500B.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú de
depuración."
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Depuración de bitácoras y
tablas"
Height = 30
Left = 100
Top = 40
Width = 360
TabIndex = 1

```

```

ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú utilerías"
StatusBarText = "Regresar al menú
principal."
TabIndex = 9
Name = "Command10"
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0500
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 160
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = depu1.ico
Caption = ""
StatusBarText = "Depuración de
bitácoras de procesos."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton depuracion de bitacora de
procesos
thisform.release
DO form cpt0520
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
Top = 240
Left = 60
Height = 60
Width = 81
WordWrap = .T.

```

```

Picture = depu2.ico
DownPicture =
Caption = ""
StatusBarText = "Depuración de
bitácoras de transmisiones."
TabIndex = 5
ToolTipText = ""
Name = "Command3"
PROCEDURE LostFocus
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton depuracion de bitacoras de
transmisiones
thisform.release
DO form cpt0530
ENDPROC
Top = 160
Left = 380
Height = 60
Width = 81
WordWrap = .T.
Picture = depu3.ico
Caption = ""
StatusBarText = "Depuración de
tabla de pagos."
TabIndex = 6
ToolTipText = ""
Name = "Command4"
PROCEDURE LostFocus
thisform.Label4.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label4.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton depuracion de tabla de
pagos
thisform.release
DO form cpt0540
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
WordWrap = .T.
BackStyle = 0
Caption = "Depuración de bitácoras
de procesos"
Height = 30
Left = 150
Top = 176
Width = 150
ForeColor = 255,255,255
Name = "Label2"

```

```

WordWrap = .T.
BackStyle = 0
Caption = "Depuración de bitácoras
de transmisiones"
Height = 30
Left = 150
Top = 254
Width = 140
ForeColor = 255,255,255
Name = "Label3"
BackStyle = 0
Caption = "Depuración de tabla de
pagos"
Height = 21
Left = 470
Top = 185
Width = 160
ForeColor = 255,255,255
Name = "Label4"
Top = 240
Left = 380
Height = 60
Width = 81
WordWrap = .T.
Picture = depu4.ico
Caption = ""
StatusBarText = "Depuración de
tabla histórica de cobros."
TabIndex = 3
ToolTipText = ""
Name = "Command5"
PROCEDURE Click
* Boton depuracion de tabla de
cobros
thisform.release
DO form cpt0550
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE GotFocus
thisform.Label5.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label5.forecolor=rgb(255,25
5,255)
ENDPROC
WordWrap = .T.
BackStyle = 0
Caption = "Depuración de tabla de
cobros"
Height = 30
Left = 470
Top = 254
Width = 160
ForeColor = 255,255,255
Name = "Label5"
Picture = depura2.ico
Stretch = 1
BackStyle = 1
Height = 42
Left = 40
Top = 20
Width = 42
Name = "Image1"

```

```

*****
C:\COBRATELICPT0510.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Alias = "parametros"
BufferModeOverride = 5
ORDER = "cve_para"
Database = dbCOBRATEL.dbc
CursorSource = "parametros"
Name = "Cursor1"
Left = 150
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor2"
ScaleMode = 3
Height = 574
Width = 793
DoCreate = .T.
ShowTips = .T.
AutoCenter = .T.
Picture = fondo.bmp
Caption = "PARAMETROS"
Closable = .T.
Visible = .F.
WindowType = 0
WindowState = 2
LockScreen = .F.
Name = "Form1"
layoutsty.Shape1.Name = "Shape1"
layoutsty.Shape2.Name = "Shape2"
layoutsty.Shape3.Name = "Shape3"
layoutsty.Shape4.Name = "Shape4"
layoutsty.Left = 4004
layoutsty.Visible = .F.
layoutsty.Name = "layoutsty"
Label1.FontUnderline = .F.
Label1.Caption = "Parámetros del
sistema"
Label1.Height = 34
Label1.Left = 120
Label1.Top = 30
Label1.Width = 275
Label1.ForeColor = 255,255,255
Label1.Name = "Label1"
Shape1.Top = 391
Shape1.Left = 25
Shape1.Width = 700
Shape1.Name = "Shape1"
Shape2.Top = 392
Shape2.Left = 25
Shape2.Width = 700
Shape2.Name = "Shape2"
Shape3.Top = 391
Shape3.Left = 27
Shape3.Width = 700
Shape3.Name = "Shape3"
Shape4.Top = 392
Shape4.Left = 27
Shape4.Width = 700
Shape4.Name = "Shape4"
PROCEDURE Activate
* Al iniciar...

```

```

thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre
GO top
regactual=recno()

ENDPROC
FontUnderline = .F.
Caption = "Cve_para:"
Left = 240
Top = 122
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLCVE_PARA1"
ControlSource =
"parametros.cve_para"
Enabled = .F.
InputMask = "XXX"
Left = 310
Top = 118
Visible = .T.
Width = 40
Name = "CVE_PARA1"
FontUnderline = .F.
Caption = "Val_par1:"
Left = 240
Top = 146
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLVAL_PAR11"
Alignment = 3
ControlSource =
"parametros.val_par1"
Enabled = .F.
Left = 310
Top = 142
Visible = .T.
Width = 62
Name = "VAL_PAR11"
FontUnderline = .F.
Caption = "Val_par2:"
Left = 240
Top = 170
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLVAL_PAR21"
ControlSource =
"parametros.val_par2"
Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXXXXX"
Left = 310
Top = 166
Visible = .T.
Width = 150
Name = "VAL_PAR21"
FontUnderline = .F.
Caption = "Val_par3:"
Left = 240
Top = 194
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLVAL_PAR31"
ControlSource =
"parametros.val_par3"

```

```

Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X"
Left = 310
Top = 190
Visible = .T.
Width = 185
Name = "VAL_PAR31"
FontUnderline = .F.
Caption = "Val_par4:"
Left = 240
Top = 218
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLVAL_PAR41"
ControlSource =
"parametros.val_par4"
Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX"
Left = 310
Top = 214
Visible = .T.
Width = 220
Name = "VAL_PAR41"
FontUnderline = .F.
Caption = "Val_par5:"
Left = 240
Top = 242
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLVAL_PAR51"
ControlSource =
"parametros.val_par5"
Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX"
Left = 310
Top = 238
Visible = .T.
Width = 220
Name = "VAL_PAR51"
FontUnderline = .F.
Caption = "Val_par6:"
Left = 240
Top = 266
Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLVAL_PAR61"
ControlSource =
"parametros.val_par6"
Enabled = .F.
InputMask =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX"
Left = 310
Top = 262
Visible = .T.
Width = 220
Name = "VAL_PAR61"
FontUnderline = .F.
Caption = "Conc_par:"
Left = 240
Top = 290

```

```

Visible = .T.
Width = 50
ForeColor = 255,255,255
Name = "LBLCONC_PAR1"
Left = 310
ReadOnly = .T.
Top = 286
Visible = .T.
ForeColor = 128,128,128
BackColor = 255,255,255
ControlSource =
"parametros.conc_par"
Name = "CONC_PAR1"
Top = 400
Left = 140
Visible = .T.
TabIndex = 18
oldTalk = OFF
viewkey = ("")
ParentKey =
ViewType = 3
GridAlias =
topfile = .T.
endfile = .T.
AddMode = .F.
nworkarea = 3
oldSetDelete = OFF
oldreprocess = 0
previewmode = .F.
previewinit = .F.
usedataenv = .F.
EditMode = .F.
GridRef =
oldSetFields = OFF
oldMultiLocks = ON
oldrefresh = 0
oldBuffering = 1
Name = "Navegar"
cmdPrev.Picture = wzback.bmp
cmdPrev.Enabled = .T.
cmdPrev.StatusBarText = "Ir al
registro anterior."
cmdPrev.ToolTipText = "Anterior"
cmdPrev.Name = "cmdPrev"
cmdNext.Picture = wznext.bmp
cmdNext.Enabled = .T.
cmdNext.StatusBarText = "Ir al
siguiente registro."
cmdNext.ToolTipText = "Siguiente"
cmdNext.Name = "cmdNext"
cmdTop.Picture = wztop.bmp
cmdTop.Enabled = .T.
cmdTop.StatusBarText = "Ir al primer
registro."
cmdTop.ToolTipText = "Primero"
cmdTop.Name = "cmdTop"
cmdEnd.Picture = wzend.bmp
cmdEnd.Enabled = .T.
cmdEnd.StatusBarText = "Ir al último
registro."
cmdEnd.ToolTipText = "Ultimo"
cmdEnd.Name = "cmdEnd"
cmdFind.Picture = wzlocate.bmp
cmdFind.Enabled = .T.
cmdFind.StatusBarText = "Buscar
registros específicos."
cmdFind.ToolTipText = "Buscar"
cmdFind.Name = "cmdFind"
cmdPrint.Picture = wzprint.bmp
cmdPrint.Enabled = .T.

```

```

cmdPrint.StatusBarText = "Imprime
un reporte."
cmdPrint.ToolTipText = "Imprimir"
cmdPrint.Name = "cmdPrint"
cmdExit.Picture = wzclos.bmp
cmdExit.Enabled = .T.
cmdExit.StatusBarText = "Salir del
formulario actual y regresar al menú
de Mantenimiento de Catálogos."
cmdExit.ToolTipText = "Regresar"
cmdExit.Name = "cmdExit"
cmdAdd.Picture = wznew.bmp
cmdAdd.DownPicture = ..\archivos
de
programa\devstudio\wfp\wizards\wizb
mp\wznew.bmp
cmdAdd.Enabled = .T.
cmdAdd.StatusBarText = "Agrega un
nuevo registro / Salva las
modificaciones."
cmdAdd.ToolTipText = "Agrega /
Salva"
cmdAdd.Name = "cmdAdd"
cmdDelete.Picture = wzdelete.bmp
cmdDelete.Enabled = .T.
cmdDelete.StatusBarText = "Elimina
el registro actual."
cmdDelete.ToolTipText = "Elimina"
cmdDelete.Name = "cmdDelete"
cmdEdit.Picture = wzedit.bmp
cmdEdit.DownPicture = ..\archivos
de
programa\devstudio\wfp\wizards\wizb
mp\wzedit.bmp
cmdEdit.Enabled = .T.
cmdEdit.StatusBarText = "Modifica el
registro actual / Revierte las
modificaciones realizadas."
cmdEdit.ToolTipText = "Modifica /
Revierte"
cmdEdit.Name = "cmdEdit"
PROCEDURE cmdPrev.Click
*botón registro previo
SELECT (THIS.parent.nworkarea)
IF !BOF()
Skip -1
ENDIF
regactual=recno()
this.Parent.topfile = BOF()
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdNext.Click
*botón registro siguiente
SELECT (THIS.parent.nworkarea)
IF !EOF()
Skip 1
ENDIF
regactual=recno()
this.Parent.endfile = EOF()
this.Parent.topfile = BOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdTop.Click
*boton primer registro
SELECT (THIS.parent.nworkarea)
LOCATE
regactual=recno()

```

```

this.Parent.topfile = .T.
this.Parent.endfile = EOF()
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdEnd.Click
*botón registro último
SELECT (THIS.parent.nworkarea)
GO BOTTOM
regactual=recno()
this.Parent.topfile = .F.
this.Parent.endfile = .T.
this.Parent.navrefresh()

ENDPROC
PROCEDURE cmdFind.Click
*boton búsqueda
thisform.Navegar.visible=.F.
thisform.busca.visible=.T.

ENDPROC
PROCEDURE cmdPrint.Click
*botón imprime
#DEFINE C_MAKEREPO_LOC
"Could
not locate a report to print. Create
new one?"
#DEFINE C_NOOPEN_LOC
"Error opening table.
Unable to print report."
#DEFINE
C_GETFILEPROMPT_LOC
"Pick report:"

LOCAL
cRepName,nSaveSess,cSaveAlias,c
SaveSource,cSaveData

cSaveAlias = ALIAS()
cSaveSource =
CURSORGETPROP("SourceName")
cSaveData =
CURSORGETPROP("Database")
cDiffSource = ""
cRepName =
LEFT(ALIAS(),8)+".FRX"
nSaveSess =
SET("DATASESSION")

* Handling for Private data sessions
#IF 0
IF m.nSaveSess # 1
SET
DATASESSION TO 1
SELECT 0
IF
!EMPTY(m.cSaveData)
OPEN
DATABASE (m.cSaveData)
ENDIF
IF
USED(m.cSaveAlias)
SELECT (m.cSaveAlias)
IF
CURSORGETPROP("SourceName")
#m.cSaveSource

```

```

cDiffSource =
CURSORGETPROP("SourceName")

USE IN (m.cSaveAlias)

SELECT 0
ENDIF
ENDIF
IF
EMPTY(ALIAS())
USE
(m.cSaveSource) AGAIN ALIAS
(m.cSaveAlias) SHARED
IF
EMPTY(ALIAS())
=MESSAGEBOX(C_NOO
PEN_LOC)

RETURN
ENDIF
ENDIF
#ENDIF

IF FILE(m.cRepName)
REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
m.cRepName =
GETFILE("frx",C_GETFILEPROMPT
_LOC,"",1)
IF !EMPTY(m.cRepName)
IF
FILE(m.cRepName)
* User
pressed Open button

REPORT FORM
(m.cRepName) PREVIEW NOWAIT
ELSE
* User
pressed New button
DO
HOME()+".wizards\wzreport.app"
WITH ALIAS(), "AUTOREPORT"
ENDIF
ENDIF
ENDIF

* Private data session code
#IF 0
IF !EMPTY(cDiffSource)
USE
(m.cDiffSource) IN 0
ENDIF
SET DATASESSION TO
m.nSaveSess
SELECT (m.cSaveAlias)
#ENDIF

IF THISFORM.ShowWindow = 2
Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
PROCEDURE cmdExit.Click
*botón salir

```

```

thisform.Release
DO form cpt0500 &&menu de
UTILERIAS

ENDPROC
PROCEDURE cmdAdd.Click
*botón agrega
#DEFINE OPT_CANCEL
0
#DEFINE OPT_ADD_PARENT
1
#DEFINE OPT_ADD_CHILD
2
#DEFINE OPT_ADD_BOTH
3
#DEFINE MB_Q_YESNO
36
#DEFINE MB_A_YES
6

#DEFINE
C_KEYFLDNOUPDATE_LOC
"The field relating the grid's
view to the parent data source is not
updatable."+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_BADCHILDKEY_LOC
"The fields
relating the parent and child tables
are not the same data type."+;
"Do you just want to add a
new record to the parent table?"
#DEFINE
C_NOCHILDUPDATE_LOC
"The child data source is a
view and does not send updates."+;
"Do you just want to add a
new record to the parent table?"
#DEFINE C_NOOBJ_LOC
"Failed create the Add
Record form class. Check or reinstall
your WIZSTYLE.VCX file."
#DEFINE C_NOUPDATE_LOC
"You cannot add a new
record because the view(s) selected
does not send updates."
#DEFINE C_NOUPDATE2_LOC
"You cannot add a new
record because the view(s) selected
does not send updates and the child
data source has a primary key."

LOCAL
oSearchDlog,oAddRec,cChildAlias,c
PapaAlias,i,lPrimeKey
LOCAL
cPapaKey,cChildKey,nSaveSess,oR
el,cTagName,lBadViewKey,nSaveRe
c,nSaveRec2
LOCAL
lBadChildKey,lUpdatableParentKey,l
NoSendParentUpdates,lNoSendChil
dUpdates

DO CASE
CASE THIS.Parent.EditMode
&&.....
..... registro en bitácora
SELECT procesos
APPEND blank

```



```

IF
ALLTRIM(THISFORM.Text4.VALUE)
="MODIFICAR"
    REPLACE tarea_bp with
"Cambio al catálogo de parámetros"
ELSE
    REPLACE tarea_bp with "Alta al
catálogo de parámetros"
ENDIF
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
SELECT PARAMETROS
&&.....
.. cerrar procesos
    thisform.Text4.VALUE=SP
ACE(7)
&&*****<<<<<<
<<<<<<

    ** Code for saving record
    this.Parent.updaterows()

CASE EMPTY(THIS.Parent.GridRef)
    && not using
Grid object
    thisform.Text4.VALUE="A
GREGAR"
&&*****<<<<<<
<<<<<<

    ** Code for adding record
    this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
    this.Parent.oldrec =
RECNO() &&save record in case
reverting
    IF
CURSORGETPROP("SourceType")#
3 AND ;
!CURSORGETPROP("offline") AND ;

    !CURSORGETPROP("Sen
dUpdates")

        =MESSAGEBOX(C_NOUP
DATE_LOC)
            RETURN
        ENDIF
        APPEND BLANK

OTHERWISE

    ** Code for adding record
    this.Parent.OldAlias =
ALIAS() &&save alias in case
reverting
    this.Parent.oldrec =
RECNO() &&save record in case
reverting
    IPrimeKey = .F.

    &&is child key primary
index

    IF
!EMPTY(THIS.Parent.oDataRelation)

```

```

oRel =
EVAL(THIS.Parent.oDataRelation)
ENDIF

DO CASE
CASE TYPE("m.oRel") =
"O"
    &&data environment
    WITH oRel

    cPapaAlias = .ParentAlias

    cPapaKey =
.RelationalExpr

    cChildAlias = .ChildAlias

    cChildKey = .ChildOrder
    ENDWITH
    oRel = .NULL.

    &&reset
    CASE
!EMPTY(THIS.Parent.viewkey)
    &&using views
        cPapaAlias =
ALIAS()
        cPapaKey =
THIS.Parent.ParentKey
        cChildAlias =
THIS.Parent.GridAlias
        cChildKey =
THIS.Parent.viewkey

        * Need to check
if Key field is updatable
        IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

            ATC(" "+m.cChildKey+" ",",
"+CURSORGETPROP("UpdatableFi
eldList",m.cChildAlias)+",")=0

            IBadViewKey = .T.
            ENDIF
        OTHERWISE
            cPapaAlias =
ALIAS()
            cPapaKey =
RELATION(1)
            cChildAlias =
THIS.Parent.GridAlias
            cChildKey =
ORDER(m.cChildAlias)
        ENDCASE

        * Test to see if both keys
are of the same data type
        SELECT
(THIS.Parent.GridAlias)
        cGridKeyType = ""

        IF
EMPTY(THIS.Parent.viewkey) AND
!EMPTY(m.cChildKey)
            &&regular table used
            * Get the Child
Key field since only index tag is here

```

```

* if these is an
expression, then default to blank.
cTagName = ""
FOR i = 1 TO
TagCount("")
    IF
UPPER(TAG(m.i)) ==
UPPER(m.cChildKey)

        cTagName = KEY(m.i)

        IPrimeKey =
PRIMARY(m.i)

        EXIT
    ENDIF

    * Check if we
have an index expression here.
    IF
ATC("(",m.cTagName)#0 OR
ATC("+",m.cTagName)#0

        cChildKey = ""
        ELSE
        cChildKey = m.cTagName

        cGridKeyType =
TYPE(m.cChildKey)
        ENDIF

    ENDIF

    IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND ;

        !CURSORGETPROP("SendUpdates
",m.cPapaAlias)

        INoSendParentUpdates =
.T.
        ENDIF
        IF
CURSORGETPROP("SourceType",
m.cChildAlias)#3 AND
!CURSORGETPROP("offline",m.cCh
ildAlias) AND ;

        !CURSORGETPROP("SendUpdates
",m.cChildAlias)

        INoSendChildUpdates = .T.
        ENDIF

        * Check if have two views
and both do not send updates
        IF
m.INoSendParentUpdates AND
m.INoSendChildUpdates

            =MESSAGEBOX(C_NOUP
DATE_LOC)
            RETURN
        ENDIF

```

```

SELECT
(THIS.Parent.OldAlias )
IF
!EMPTY(m.cGridKeyType) AND
m.cGridKeyType #
TYPE(m.cPapaKey)
IBadChildKey =
.T.
ENDIF
DO CASE
CASE
m.INoSendParentUpdates AND
(m.IBadViewKey OR
m.IBadChildKey)
=MESSAGEBOX(C_NOUP
DATE_LOC)
RETURN
CASE m.IPrimeKey AND
m.INoSendParentUpdates
=MESSAGEBOX(C_NOUP
DATE2_LOC)
RETURN
CASE
m.INoSendChildUpdates
&&child view does not
send updates
IF
MESSAGEBOX(C_NOCHILDUPDAT
E_LOC,MB_Q_YESNO) #
MB_A_YES
RETURN
ENDIF
APPEND BLANK
CASE m.IBadViewKey
&&view key in not
updatable
IF
MESSAGEBOX(C_KEYFLDNOUPD
ATE_LOC,MB_Q_YESNO) #
MB_A_YES
RETURN
ENDIF
APPEND BLANK
CASE m.IBadChildKey
&&bad child key -- different
data type from parent
IF
MESSAGEBOX(C_BADCHILDKEY_
LOC,MB_Q_YESNO) # MB_A_YES
RETURN
ENDIF
APPEND BLANK
OTHERWISE
* Need to check
if Key field is updatable
IUpdatableParentKey = .T.
IF
CURSORGETPROP("SourceType",
m.cPapaAlias)#3 AND
!CURSORGETPROP("offline",m.cPa
paAlias) AND;
ATC(" "+m.cPapaKey+"",",",

```

```

"+CURSORGETPROP("UpdatableFi
eldList",m.cPapaAlias)+",")=0
cPapaKey = ""
IUpdatableParentKey = .F.
ENDIF
* Check for
expression in cPapaKey if so, then
do not update record with keyvalue
IF
ATC("(",m.cPapaKey)#0 OR
ATC("+",m.cPapaKey)#0
cPapaKey = ""
ENDIF
nSaveSess =
SET("DATASESSION")
oAddRec =
CREATE("GridAddRec")
IF
TYPE("m.oAddRec") # "O"
=MESSAGEBOX(C_NOO
BJ_LOC)
RETURN
ENDIF
oAddRec.ChildPrimaryKey
= m.IPrimeKey
oAddRec.UpdatableParent
Key = m.IUpdatableParentKey
oAddRec.NoSendUpdates
= m.INoSendParentUpdates
oAddRec.KeyField =
m.cPapaKey
oAddRec.KeyValue =
IIF(!EMPTY(m.cPapaKey),
EVAL(m.cPapaKey),"")
oAddRec.runaddform()
IF
oAddRec.AddOption =
OPT_CANCEL
RETURN
ENDIF
SET
DATASESSION TO nSaveSess
SELECT
(THIS.Parent.OldAlias)
IF
TYPE("oAddRec.KeyValue") = "C"
oAddRec.KeyValue =
TRIM(oAddRec.KeyValue)
ENDIF
* Add record to
parent table

```

```

nSaveRec =
RECNO()
IF
INLIST(oAddRec.AddOption,OPT_A
DD_PARENT,OPT_ADD_BOTH)
IF
EMPTY(m.cPapaKey)
APPEND BLANK IN
(m.cPapaAlias)
ELSE
INSERT INTO
(m.cPapaAlias)
((oAddRec.KeyField))
VALUES(oAddRec.KeyValue)
ENDIF
ENDIF
nSaveRec2 =
RECNO()
* Add child
record
IF
INLIST(oAddRec.AddOption,OPT_A
DD_CHILD,OPT_ADD_BOTH)
* Need
to check
GO
m.nSaveRec
IF
EMPTY(m.cChildKey) OR
TYPE("m.cChildKey")#TYPE('oAddR
ec.KeyValue')
APPEND BLANK IN
(m.cChildAlias)
ELSE
INSERT INTO
(m.cChildAlias) ((m.cChildKey))
VALUES(oAddRec.KeyValue)
ENDIF
GO
m.nSaveRec2
ENDIF
ENDCASE
ENDCASE
this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode =
THIS.Parent.EditMode
this.Parent.topfile = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.
thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
*Thisform.nom_agen1.enabled=.F.
*Thisform.vi_lo_ag1.enabled=.F.
*Thisform.resp_agen1.enabled=.F.
*Thisform.te_re_ag1.enabled=.F.

```

```

*Thisform.baja_ag1.enabled=.F.

ENDPROC
PROCEDURE cmdDelete.Click
*botón borra

* Tipo de cuadro de diálogo
* 4 + 32 + 256
* 4 = Botones SI y No
* 32 = Icono de PREGUNTA
* 256 = el segundo botón es el de default

* Respuestas
* 6 = SI
* 7 = NO

IF MESSAGEBOX('¿ Desea eliminar este registro?', 4+32+256, 'Confirme por favor...') = 6
    Delete
    IF
THIS.Parent.updaterows()
&&success
        * Success
        IF !EOF()
            Skip 1
        ENDIF
        IF EOF() AND
!BOF()
            Skip -1
        ENDIF
        regactual=recno()
        ENDIF
        thisform.LockScreen = T
        this.Parent.buttonrefresh()
        this.Parent.navrefresh()
        thisform.LockScreen = .F.
&&.....
..... registro en bitácora
        SELECT procesos
        APPEND blank
        REPLACE tarea_bp with "Baja al catálogo de parámetros"
        REPLACE usua_bp with usuario
        REPLACE fecha_bp with date()
        REPLACE hora_bp with time()
        SELECT PARAMETROS
&&.....
. cerrar procesos
ENDIF

IF THISFORM.ShowWindow = 2
    Activate Window
(THISFORM.Name)
ENDIF

ENDPROC
PROCEDURE cmdEdit.Click
*botón edita
#DEFINE C_NOUPDATE_LOC
    "You cannot edit because the view(s) selected does not send updates."

LOCAL
INoSendParentUpdates,INoSendChildUpdates

```

```

LOCAL aTablesUsed,nTablesUsed,i

** Reverting record
IF THIS.Parent.EditMode
    thisform.Text4.VALUE=SPACE(7)
    &&*****<<<<<<
    <<<<<<<

        IF THIS.Parent.usedataenv
            SELECT
            (THIS.Parent.OldAlias)
            IF
CURSORGETPROP("Buffering")>1

                =TableRevert(.T.)
                ENDIF
            IF
!EMPTY(THIS.Parent.GridAlias)

                SELECT
                (THIS.Parent.GridAlias)
                IF
CURSORGETPROP("Buffering")>1

                    =TableRevert(.T.)
                    ENDIF
                ELSE
                    DIMENSION
                    aTablesUsed[1]
                    m.nTablesUsed
                    = AUUSED(aTablesUsed)
                    FOR i = 1 TO
                    m.nTablesUsed
                        IF
CURSORGETPROP("Buffering",aTablesUsed[m,i,1])>1

                            =TableRevert(.T.,aTablesUsed[m,i,1])
                            ENDIF
                        ENDFOR
                    ENDIF
                * Go back to original place
                SELECT
                (THIS.Parent.OldAlias)
                IF RECCOUNT() <
THIS.Parent.oldrec &&added record at EOF()
                    GO TOP
                ELSE
                    GO
THIS.Parent.oldrec
                ENDIF
            ELSE
                thisform.Text4.VALUE="MODIFICAR"
                &&*****<<<<<<
                <<<<<<<

                    * Check to see if view
                    allows updates
                    IF
CURSORGETPROP("SourceType")#
3 AND
!CURSORGETPROP("offline") AND;

```

```

!CURSORGETPROP("SendUpdates")

                INoSendParentUpdates =
.T.
                ENDIF
            IF
!EMPTY(THIS.Parent.GridAlias)
            AND ;
CURSORGETPROP("SourceType",THIS.Parent.GridAlias)#3 AND ;
!CURSORGETPROP("offline",THIS.Parent.GridAlias) AND ;
!CURSORGETPROP("SendUpdates",THIS.Parent.GridAlias)

                INoSendChildUpdates= .T.
                ENDIF

            IF
(m.INoSendChildUpdates AND
m.INoSendParentUpdates) OR;

                (EMPTY(THIS.Parent.GridAlias) AND
m.INoSendParentUpdates)

                =MESSAGEBOX(C_NOUP
DATE_LOC)
                RETURN
            ENDIF

            this.Parent.OldAlias =
ALIAS() &&save alias in case reverting
            this.Parent.oldrec =
RECNO() &&save record in case reverting
            ENDIF

** Editing record
this.Parent.EditMode =
!THIS.Parent.EditMode
this.Parent.AddMode = .F.
thisform.LockScreen = .T.
this.Parent.buttonrefresh()
this.Parent.navrefresh()
thisform.LockScreen = .F.

thisform.Text1.enabled=.F.
thisform.text2.enabled=.F.
thisform.text3.enabled=.F.
thisform.Text4.enabled=.F.
thisform.text5.enabled=.F.
*Thisform.vi_co_ag1.enabled=.F.

ENDPROC
FontBold = .T.
FontSize = 14
FontUnderline = .F.
BackStyle = 0
Caption = "Parámetros"
Height = 31
Left = 40

```

```

Top = 120
Width = 121
TabIndex = 21
ForeColor = 255,255,255
Name = "Label2"
ColumnCount = 2
ColumnWidths = "70,306"
RowSourceType = 6
RowSource =
"parametros.cve_para_conc_par"
FirstElement = 1
Height = 11
Left = 210
NumberOfElements = 0
Style = 2
TabIndex = 22
Top = 120
Visible = .F.
Width = 20
Name = "Busca"
PROCEDURE InteractiveChange
* Cuando encuentra el elemento
buscado
GO record recno()
regactual=recno()
thisform.busca.visible=.F.
thisform.Navegar.visible=.T.

```

```

* Reset from prior
thisform.Refresh()

```

```

ENDPROC
Enabled = .F.
Height = 21
Left = 520
TabIndex = 23
Top = 90
Visible = .F.
Width = 80
Name = "Text1"
FontUnderline = .F.
Caption = "Usuario"
Height = 13
Left = 520
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 606
TabIndex = 24
Top = 90
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 606
Top = 80
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25

```

```

Top = 40
Width = 80
Name = "Text3"
FontUnderline = .F.
Caption = "Fecha"
Height = 13
Left = 647
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 690
TabIndex = 26
Top = 90
Visible = .F.
Width = 80
Name = "Text4"
Caption = "clave de usuario"
Height = 13
Left = 519
Top = 80
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
.
*****
C:\COBRATEL\CPT0520.SCX
Left = 76
Top = 253
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 10
Top = 20
Width = 90
Height = 90
Alias = "procesos"
Database = dbCOBRATEL.dbc
CursorSource = "procesos"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®.
Depuración de bitácoras de
proceso."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()

```

```
thisform.text5.value=nombre
```

```
thisform.text11.value=date()
thisform.text12.value=date()
```

```

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Depuración de bitácoras
de procesos"
Height = 30
Left = 100
Top = 40
Width = 370
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lit.ico
Caption = "Menú de depuración"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0500B
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Fecha inicial"
Height = 21
Left = 150
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label1"
Height = 21
Left = 150
StatusBarText = "Fecha inicial de la
consulta"
Top = 140
Width = 101
Name = "Text11"
PROCEDURE LostFocus
* Igualar fecha final
thisform.Label1.forecolor=rgb(0,0,0)
thisform.text12.value=this.value
ENDPROC
PROCEDURE GotFocus

```

```

thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Valid
* Validar que la fecha no sea mayor
a la del sistema
IF this.value > date() then
MESSAGEBOX("La fecha que ha
proporcionado no es válida.
¡Verifique por favor!",0+16+0,'Error.')
```

```

RETURN .f.
ELSE
RETURN .t.
ENDIF

ENDPROC
Height = 21
Left = 330
StatusBarText = "Fecha final de la
consulta."
Top = 140
Width = 101
Name = "Text12"
PROCEDURE Valid
* Validar rango de fechas
IF (this.value>=thisform.text11.value)
.and. (this.value<=date()) then
RETURN .t.
ELSE
MESSAGEBOX("La fecha
proporcionada no es válida.
¡Verifique por favor!",0+16+0,'Error.')
```

```

RETURN .f.
ENDIF
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Fecha final"
Height = 21
Left = 330
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label2"
Top = 200
Left = 240
Height = 60
Width = 81
WordWrap = T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Depuración de la
bitácora de procesos."
TabIndex = 9
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton depurar
IF MESSAGEBOX("¿ Ya realizó el
respaldo de su Base de Datos ?",
```

```

4+32+256, 'Confirme por favor...') = 6
then
&& si realizó el respaldo
Delete for fecha_bp >=
thisform.text11.value .and. fecha_bp
<= thisform.text12.value
thisform.progreso.visible=.t.
FOR i=1 to 100 STEP 2
thisform.progreso.value=i
WAIT " timeout .1
NEXT i
thisform.progreso.visible=.f.
MESSAGEBOX("Bitácora de
procesos
depurada",0+64+0,'Operación
finalizada')
* use procesos && Registra en
bitácora de procesos, la operación
efectuada
APPEND blank
REPLACE tarea_bp with
"Depuración bitácora de procesos"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
* use && cierra
procesos.....
..... cerrar procesos
ELSE
&& no realizó el respaldo
MESSAGEBOX("Realice el
respaldo de su Base de Datos en
[Utilerías/Respaldo de
Archivos]",0+64+0,'Depuración
cancelada')
ENDIF

ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Depurar bitácora"
Height = 21
Left = 330
Top = 220
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 70
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 519
Top = 14
```

```

Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 58
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
Picture = depu1.ico
Stretch = 1
BackStyle = 0
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 316
```

```
Left = 126
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"
```

```
*****
```

```
C:\COBRATEL\CPT0530.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 12
Top = 14
Width = 90
Height = 90
Alias = "b_transmision"
Database = dbCOBRATEL.dbc
CursorSource = "b_transmision"
Name = "Cursor2"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®.
Depuración de bitácoras de
transmisiones."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.text11.value=date()
thisform.text12.value=date()
```

```
ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Depuración de bitácoras
de transmisiones"
Height = 30
Left = 100
Top = 40
Width = 400
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú de depuración"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE MouseMove
```

```
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0500B
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Fecha inicial"
Height = 21
Left = 150
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label1"
Height = 21
Left = 150
StatusBarText = "Fecha inicial de la
consulta"
Top = 140
Width = 101
Name = "Text11"
PROCEDURE Valid
* Validar que la fecha no sea mayor
a la del sistema
IF this.value > date() then
MESSAGEBOX("La fecha que ha
proporcionado no es válida.
¡Verifique por favor!,0+16+0,'Error.')
```

```
RETURN .f.
ELSE
RETURN .t.
ENDIF

ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Igualar fecha final
thisform.Label1.forecolor=rgb(0,0,0)
thisform.text12.value=this.value
ENDPROC
Height = 21
Left = 330
StatusBarText = "Fecha final de la
consulta."
Top = 140
Width = 101
Name = "Text12"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Valid
* Validar rango de fechas
```

```
IF (this.value>=thisform.text11.value)
.and. (this.value<=date()) then
RETURN .t.
ELSE
MESSAGEBOX("La fecha
proporcionada no es válida.
¡Verifique por favor!,0+16+0,'Error.')
```

```
RETURN .f.
ENDIF
ENDPROC
BackStyle = 0
Caption = "Fecha final"
Height = 21
Left = 330
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label2"
Top = 200
Left = 240
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Depuración de la
bitácora de procesos."
TabIndex = 9
Name = "Command1"
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton depurar
IF MESSAGEBOX("¿ Ya realizó el
respaldo de su Base de Datos ?",
4+32+256, 'Confirme por favor...') = 6
then
&& si realizó el respaldo
Delete for fech_bt >=
thisform.text11.value .and. fech_bt
<= thisform.text12.value
thisform.progreso.visible=.i.
FOR i=1 to 100 STEP 2
thisform.progreso.value=i
WAIT " timeout .1
NEXT i
thisform.progreso.visible=.f.
MESSAGEBOX('Bitácora de
transmisiones
depurada',0+64+0,'Operación
finalizada')
USE procesos && Registra en
bitácora de procesos, la operación
efectuada
APPEND blank
REPLACE tarea_bp with
"Depuración bitácora de
transmisiones"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
```

```

USE && cierra
procesos.....
..... cerrar procesos
ELSE
&& no realizó el respaldo
MESSAGEBOX('Realice el
respaldo de su Base de Datos en
[Utilerías/Respaldo de
Archivos]',0+64+0,'Depuración
cancelada')
ENDIF

```

```

ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Depurar bitácora"
Height = 21
Left = 330
Top = 220
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 70
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 60
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = F
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13

```

```

Left = 647
Top = 44
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
Picture = depu2.ico
Stretch = 1
BackStyle = 0
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 308
Left = 118
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"

*****
C:\COBRATEL\CPT0540.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 14
Top = 13
Width = 90
Height = 90
Alias = "pagos"
Database = dbCOBRATEL.dbc
CursorSource = "pagos"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL@.
Depuración de tabla de pagos."

```

```

WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.text11.value=date()
thisform.text12.value=date()

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Depuración de tabla de
pagos"
Height = 30
Left = 100
Top = 40
Width = 400
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09itl.ico
Caption = "Menú de depuración"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0500B
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Fecha inicial"
Height = 21
Left = 150
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label1"
Height = 21
Left = 150
StatusBarText = "Fecha inicial de la
consulta"
Top = 140

```

```

Width = 101
Name = "Text11"
PROCEDURE Valid
* Validar que la fecha no sea mayor
a la del sistema
IF this.value > date() then
MESSAGEBOX("La fecha que ha
proporcionado no es válida.
¡Verifique por favor!",0+16+0,'Error.')
```

RETURN .f.

```

ELSE
RETURN .t.
ENDIF

ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Igualar fecha final
thisform.Label1.forecolor=rgb(255,25
5,255)
thisform.text12.value=this.value
ENDPROC
Height = 21
Left = 330
StatusBarText = "Fecha final de la
consulta."
Top = 140
Width = 101
Name = "Text12"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Valid
* Validar rango de fechas
IF (this.value>=thisform.text11.value)
.and. (this.value<=date()) then
RETURN .t.
ELSE
MESSAGEBOX("La fecha
proporcionada no es válida.
¡Verifique por favor!",0+16+0,'Error.')
```

RETURN .f.

```

ENDIF
ENDPROC
BackStyle = 0
Caption = "Fecha final"
Height = 21
Left = 330
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label2"
Top = 200
Left = 240
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Depuración de la
bitácora de procesos."
TabIndex = 9
```

```

Name = "Command1"
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton depurar
IF MESSAGEBOX("¿ Ya realizó el
respaldo de su Base de Datos ?",
4+32+256, 'Confirme por favor...') = 6
then
&& si realizó el respaldo
Delete for fe_fi_pa >=
thisform.text11.value .and. fe_fi_pa
<= thisform.text12.value
thisform.progreso.visible=.t.
FOR i=1 to 100 step 2
thisform.progreso.value=i
WAIT " timeout 0.1
NEXT i
thisform.progreso.visible=.f.
MESSAGEBOX("Tabla de pagos
depurada",0+64+0,'Operación
finalizada')
USE procesos && Registra en
bitácora de procesos, la operación
efectuada
APPEND blank
REPLACE tarea_bp with
"Depuración tabla de pagos"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
USE && cierra
procesos.....
..... cerrar procesos
ELSE
&& no realizó el respaldo
MESSAGEBOX('Realice el
respaldo de su Base de Datos en
[Utilerías/Respaldo de
Archivos]',0+64+0,'Depuración
cancelada')
```

ENDIF

```

ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Depurar tabla"
Height = 21
Left = 330
Top = 220
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
```

```

TabIndex = 23
Top = 70
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 70
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 58
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 44
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 70
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 58
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
```



```

Picture = depu3.ico
Stretch = 1
BackStyle = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 308
Left = 118
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"

```

```

C:\COBRATEL\CPT0550.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 139
Top = 2
Width = 90
Height = 90
Alias = "h_cobros"
Database = dbCOBRATEL.dbc
CursorSource = "h_cobros"
Name = "Cursor2"
Left = 10
Top = 2
Width = 90
Height = 90
Alias = "cobros"
Database = dbCOBRATEL.dbc
CursorSource = "cobros"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®.
Depuración de tabla de cobros."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.text11.value=date()
thisform.text12.value=date()
SELECT cobros
ENDPROC
FontBold = T.
FontSize = 14
BackStyle = 0
Caption = "Depuración de tabla de
cobros"
Height = 30
Left = 100
Top = 40
Width = 420
TabIndex = 1

```

```

ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú de depuración"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0500B
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Fecha inicial"
Height = 21
Left = 150
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label1"
Height = 21
Left = 150
StatusBarText = "Fecha inicial de la
consulta"
Top = 140
Width = 101
Name = "Text11"
PROCEDURE LostFocus
* Igualar fecha final
thisform.Label1.forecolor=rgb(255,25
5,255)
thisform.text12.value=this.value
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Valid
* Validar que la fecha no sea mayor
a la del sistema
IF this.value > date() then
MESSAGEBOX('La fecha que ha
proporcionado no es válida.
¡Verifique por favor!',0+16+0,'Error.
')
RETURN .f.
ELSE
RETURN .t.
ENDIF
ENDPROC

```

```

Height = 21
Left = 330
StatusBarText = "Fecha final de la
consulta."
Top = 140
Width = 101
Name = "Text12"
PROCEDURE Valid
* Validar rango de fechas
IF (this.value>=thisform.text11.value)
.and. (this.value<=date()) then
RETURN .t.
ELSE
MESSAGEBOX('La fecha
proporcionada no es válida.
¡Verifique por favor!',0+16+0,'Error.
')
RETURN .f.
ENDIF
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Fecha final"
Height = 21
Left = 330
Top = 120
Width = 70
ForeColor = 255,255,255
Name = "Label2"
Top = 200
Left = 240
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Depuración de la
bitácora de procesos."
TabIndex = 9
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton depurar
SELECT cobros
IF MESSAGEBOX('¿ Ya realizó el
respaldo de su Base de Datos?',
4+32+256, 'Confirme por favor...') = 6
then
&& si realizó el respaldo
SET filter to fe_re_co >=
thisform.text11.value .and. fe_re_co
<= thisform.text12.value
GO top
DO while .not. eof()
var1=telef_co
var2=fech_cob
var3=via_cobr
var4=est_cobr
var5=imp_cob
var6=fe_re_co

```

```

SELECT h_cobros
APPEND blank
REPLACE telef_hc with var1
REPLACE fech_hc with var2
REPLACE via_co_hc with var3
REPLACE es_co_hc with var4
REPLACE imp_hc with var5
REPLACE fe_rec_hc with var6
SELECT cobros
Skip
ENDDO

Delete all
* pack
SET filter to
thisform.progreso.visible=.t.
FOR i=1 to 100 step 2
  thisform.progreso.value=i
  WAIT " timeout 0.1
NEXT i
thisform.progreso.visible=.f.
MESSAGEBOX("Tabla de cobros
depurada",0+64+0,'Operación
finalizada')
USE procesos && Registra en
bitácora de procesos, la operación
efectuada
APPEND blank
REPLACE tarea_bp with
"Depuración tabla cobros"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
USE && cierra
procesos.....
..... cerrar procesos
ELSE
&& no realizó el respaldo
MESSAGEBOX('Realice el
respaldo de su Base de Datos en
[Utilerías/Respaldo de
Archivos]',0+64+0,'Depuración
cancelada')
ENDIF

ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
BackStyle = 0
Caption = "Depurar tabla"
Height = 21
Left = 330
Top = 220
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23

```

```

Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 643
Top = 44
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 68
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
Picture = depu4.ico

```

```

Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
Top = 300
Left = 110
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"

*****
C:\COBRATELICPT0560.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL@. Respaldo
de archivos."
WindowState = 2
AlwaysOnTop = .F.
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Respaldo de archivos"
Height = 31
Left = 100
Top = 40
Width = 230
ForeColor = 255,255,255
Name = "Label1"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú de utilerías"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir

```

```

thisform.release
DO form cpt0500
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 210
Left = 240
Height = 60
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Respaldar archivos
de bases de datos del sistema."
TabIndex = 9
Name = "Command1"
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton depurar

DO case
CASE
thisform.opciones.Option1.value=1
ruta="A:\"
CASE
thisform.opciones.Option2.value=1
ruta="C:\COBRATEL\respaldos\"
CASE
thisform.opciones.Option3.value=1

ruta="C:\COBRATEL\unidadexterna\"
ENDCASE

IF MESSAGEBOX('¿ Está preparada
su unidad de respaldo ?, 4+32+256,
'Confirme por favor...') = 6 then
&& si está lista la unidad de respaldo
thisform.progreso.visible= T.
thisform.Label2.caption='Base de
datos . AGENTES'
nomarch=ruta+"agentes.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\agentes.dbf to
&nomarch
FOR i=1 to 8
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : ARCHIVOS DE COBRO'

```

```

nomarch=ruta+"arch_cobro.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\arch_cobro.dbf to
&nomarch
FOR i=9 to 17
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : TRANSMISIONES'
nomarch=ruta+"b_transmision.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\b_transmision.dbf to
&nomarch
FOR i=18 to 25
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : COBROS'
nomarch=ruta+"cobros.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\cobros.dbf to
&nomarch
FOR i=26 to 33
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : ESTACIONES DE COBRO'
nomarch=ruta+"est_cobro.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\est_cobro.dbf to
&nomarch
FOR i=34 to 41
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : GRUPOS'
nomarch=ruta+"grupos.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\GRUPOS.dbf to
&nomarch
FOR i=42 to 49
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : HISTORICO DE COBROS'

```

```

nomarch=ruta+"h_cobros.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\h_cobros.dbf to
&nomarch
FOR i=50 to 57
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : PAGOS'
nomarch=ruta+"pagos.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file c:\COBRATEL\pagos.dbf
to &nomarch
FOR i=58 to 65
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : PARAMETROS'
nomarch=ruta+"parametros.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\PARAMETROS.dbf to
&nomarch
FOR i=66 to 73
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : PROCESOS'
nomarch=ruta+"procesos.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\procesos.dbf to
&nomarch
FOR i=74 to 81
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

thisform.Label2.caption='Base de
datos : USUARIOS'
nomarch=ruta+"usuarios.dbf"

thisform.Text6.value=alltrim(nomarch
)
COPY file
c:\COBRATEL\USUARIOS.dbf to
&nomarch
FOR i=82 to 89
thisform.progreso.value=i
WAIT "" timeout .1
NEXT i

USE procesos
APPEND blank

```

```

REPLACE tarea_bp with "Respaldo
DB"
REPLACE usua_bp with usuario
REPLACE fecha_bp with date()
REPLACE hora_bp with time()
USE && cierra
procesos.....
..... cerrar procesos
FOR i=90 to 100
  thisform.progreso.value=i
  WAIT "" timeout .1
NEXT i
thisform.progreso.visible=.f.
thisform.Label2.caption="
MESSAGEBOX('Base de datos
respaldada',0+64+0,'Operación
finalizada')

ELSE
&& no está lista la unidad de
resplado
MESSAGEBOX('Prepare su unidad
de respaldo e inténtelo
nuevamente.',0+64+0,'Respaldo
cancelado')
ENDIF

ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Respaldar archivos"
Height = 21
Left = 330
Top = 230
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Top = 330
Left = 120
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 519
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24

```

```

Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 44
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 26
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 68
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
BackStyle = 0
Caption = ""
Height = 21
Left = 120
Top = 290
Width = 400
Name = "Label2"
Picture = bckup.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
ButtonCount = 3
BackStyle = 0
Value = 1
Height = 70

```

```

Left = 180
Top = 100
Width = 250
Name = "Opciones"
Option1.BackStyle = 0
Option1.Caption = "Drive [A:]"
Option1.Value = 1
Option1.Height = 17
Option1.Left = 5
Option1.Top = 5
Option1.Width = 75
Option1.ForeColor = 255,255,255
Option1.Name = "Option1"
Option2.BackStyle = 0
Option2.Caption = "Drive
C:\COBRATEL\RESPALDOS"
Option2.Height = 17
Option2.Left = 5
Option2.Top = 24
Option2.Width = 225
Option2.ForeColor = 255,255,255
Option2.Name = "Option2"
Option3.BackStyle = 0
Option3.Caption = "Unidad externa
de respaldo"
Option3.Height = 17
Option3.Left = 5
Option3.Top = 43
Option3.Width = 205
Option3.ForeColor = 255,255,255
Option3.Name = "Option3"
Enabled = .F.
Height = 21
Left = 160
TabIndex = 26
Top = 180
Visible = .T.
Width = 400
Name = "Text6"

*****
C:\COBRATEL\CPT0570.SCX
.Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL@.
Restauración de archivos."
WindowState = 2
AlwaysOnTop = .F.
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

ENDPROC
FontBold = .T.
FontSize = 16

```

```

BackStyle = 0
Caption = "Restauración de archivos"
Height = 31
Left = 100
Top = 40
Width = 280
ForeColor = 255,255,255
Name = "Label1"
AutoSize = F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09it.ico
Caption = "Menú de utilerías"
StatusBarText = "Regresar al menú de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setFocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0500
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Top = 220
Left = 220
Height = 60
Width = 81
WordWrap = T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Restauración de
archivos de bases de datos del
sistema."
TabIndex = 9
Name = "Command1"
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton depurar

DO case
CASE
thisform.opciones.Option1.value=1
ruta="A:\\"
CASE
thisform.opciones.Option2.value=1
ruta="C:\COBRATEL\respaldos\"

```

```

CASE
thisform.opciones.Option3.value=1

ruta="C:\COBRATEL\unidadexterna\"
ENDCASE

IF MESSAGEBOX('¿ Están sus
Archivos Respalados en la unidad
de respaldo?', 4+32+256, 'Confirme
por favor...') = 6 then
&& si está lista la unidad de respaldo
IF file('a:\agentes.dbf') and
file('a:\arch_cobro.dbf') and
file('a:\b_transmision.dbf');
and file('a:\cobros.dbf') and
file('a:\est_cobro.dbf') and
file('a:\grupos.dbf');
and file('a:\h_cobros.dbf') and
file('a:\pagos.dbf') and
file('a:\parametros.dbf');
and file('a:\procesos.dbf') and
file('a:\usuarios.dbf') then
&& si se puede resaldar
thisform.progreso.visible=.T.

thisform.Label2.caption='R
estaurando base de datos :
AGENTES'
IF
file('c:\COBRATEL\respaldos\agente
s.dbf')
Delete file
('c:\COBRATEL\respaldos\agentes.d
bf')
ENDIF
COPY file a:\agentes.dbf to
c:\COBRATEL\respaldos\agentes.dbf
FOR i=1 to 8
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='re
staurando base de datos :
ARCHIVOS DE COBRO'
IF
file('c:\COBRATEL\respaldos\arch_c
obro.dbf')
Delete file
('c:\COBRATEL\respaldos\arch_cobr
o.dbf')
ENDIF
COPY file
a:\arch_cobro.dbf to
c:\COBRATEL\respaldos\arch_cobro
.dbf
FOR i=9 to 17
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos :
TRANSMISIONES'
IF
file('c:\COBRATEL\respaldos\b_trans
mision.dbf')
Delete file
('c:\COBRATEL\respaldos\b_transmi
sion.dbf')

```

```

ENDIF
COPY file
a:\b_transmision.dbf to
c:\COBRATEL\respaldos\b_transmi
on.dbf
FOR i=18 to 25
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos :
COBROS'
IF
file('c:\COBRATEL\respaldos\cobros.
dbf')
Delete file
('c:\COBRATEL\respaldos\cobros.dbf
')
ENDIF
COPY file a:\cobros.dbf to
c:\COBRATEL\respaldos\cobros.dbf
FOR i=26 to 33
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos :
ESTACIONES DE COBRO'
IF
file('c:\COBRATEL\respaldos\est_co
bro.dbf')
Delete file
('c:\COBRATEL\respaldos\est_cobro.
dbf')
ENDIF
COPY file a:\est_cobro.dbf
to
c:\COBRATEL\respaldos\est_cobro.d
bf
FOR i=34 to 41
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos :
GRUPOS'
IF
file('c:\COBRATEL\respaldos\grupos.
dbf')
Delete file
('c:\COBRATEL\respaldos\grupos.dbf
')
ENDIF
COPY file a:\GRUPOS.dbf
to
c:\COBRATEL\respaldos\GRUPOS.d
bf
FOR i=42 to 49
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos :
HISTORICO DE COBROS'

```

```

IF
file('c:\COBRATEL\respaldos\h_cobr
os.dbf')
Delete file
('c:\COBRATEL\respaldos\h_cobros.
dbf')
ENDIF
COPY file a:\h_cobros.dbf
to
c:\COBRATEL\respaldos\h_cobros.d
bf
FOR i=50 to 57
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos : PAGOS'
IF
file('c:\COBRATEL\respaldos\pagos.
dbf')
Delete file
('c:\COBRATEL\respaldos\pagos.dbf
')
ENDIF
COPY file a:\pagos.dbf to
c:\COBRATEL\respaldos\pagos.dbf
FOR i=58 to 65
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos :
PARAMETROS'
IF
file('c:\COBRATEL\respaldos\parame
tros.dbf')
Delete file
('c:\COBRATEL\respaldos\parametro
s.dbf')
ENDIF
COPY file
a:\PARAMETROS.dbf to
c:\COBRATEL\respaldos\PARAMET
ROS.dbf
FOR i=66 to 73
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

thisform.Label2.caption='R
estaurando base de datos :
PROCESOS'
IF
file('c:\COBRATEL\respaldos\proces
os.dbf')
Delete file
('c:\COBRATEL\respaldos\procesos.
dbf')
ENDIF
COPY file a:\procesos.dbf
to
c:\COBRATEL\respaldos\procesos.d
bf
FOR i=74 to 81
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

```

```

thisform.Label2.caption='R
estaurando base de datos :
USUARIOS'
IF
file('c:\COBRATEL\respaldos\usuario
s.dbf')
Delete file
('c:\COBRATEL\respaldos\usuarios.d
bf')
ENDIF
COPY file
a:\USUARIOS.dbf to
c:\COBRATEL\respaldos\USUARIOS
.dbf
FOR i=82 to 89
thisform.progreso.value=i
WAIT " timeout .1
NEXT i

USE procesos
APPEND blank
REPLACE tarea_bp with
"Restauración DB"
REPLACE usua_bp with
usuario
REPLACE fecha_bp with
date()
REPLACE hora_bp with
time()
USE && cierra
procesos.....
..... cerrar procesos
FOR i=90 to 100
thisform.progreso.value=i
WAIT " timeout .1
NEXT i
thisform.progreso.visible=.f

thisform.Label2.caption="
MESSAGEBOX('Base de
datos restaurada',0+64+0,'Operación
finalizada')

ELSE
&& no se puede respaldar porque
faltan archivos
MESSAGEBOX('No es posible
realizar la restauración debido a que
el respaldo está incompleto.
¡Proporcione el último respaldo
completo por
favor!',0+16+0,'Restauración
cancelada.')
ENDIF

ELSE
&& no está lista la unidad de
respaldo
MESSAGEBOX('Debe tener sus
Archivos Respaldados para realizar
ésta operación. Inténtelo
después.',0+64+0,'Restauración
cancelada')
ENDIF

ENDPROC
PROCEDURE MouseMove

```

```

LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Restaurar archivos"
Height = 21
Left = 310
Top = 240
Width = 120
ForeColor = 255,255,255
Name = "Label3"
Top = 330
Left = 120
Height = 11
Width = 411
Visible = .F.
Name = "Progreso"
Enabled = .F.
Height = 21
Left = 490
TabIndex = 23
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 14
Visible = .T.
Width = 40
TabIndex = 15
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 590
TabIndex = 24
Top = 80
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 590
Top = 68
Visible = .F.
Width = 40
TabIndex = 14
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 25
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 645
Top = 43
Visible = .T.
Width = 40
TabIndex = 16
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680

```

```

TabIndex = 26
Top = 80
Visible = .F.
Width = 80
Name = "Text4"
Enabled = .F.
Height = 21
Left = 560
TabIndex = 26
Top = 10
Visible = T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 490
Top = 68
Visible = .F.
Width = 90
TabIndex = 14
Name = "Embossedlabel4"
BackStyle = 0
Caption = ""
Height = 21
Left = 120
Top = 300
Width = 400
Name = "Label2"
Picture = resto.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
ButtonCount = 3
BackStyle = 0
Value = 1
Height = 70
Left = 170
Top = 120
Width = 250
Name = "Opciones"
Option1.BackStyle = 0
Option1.Caption = "Drive [A:]"
Option1.Value = 1
Option1.Height = 17
Option1.Left = 5
Option1.Top = 5
Option1.Width = 75
Option1.ForeColor = 255,255,255
Option1.Name = "Option1"
Option2.BackStyle = 0
Option2.Caption = "Drive
C:\COBRATEL\RESPALDOS\"
Option2.Height = 17
Option2.Left = 5
Option2.Top = 24
Option2.Width = 225
Option2.ForeColor = 255,255,255
Option2.Name = "Option2"
Option3.BackStyle = 0
Option3.Caption = "Unidad externa
de respaldo"
Option3.Height = 17
Option3.Left = 5
Option3.Top = 43
Option3.Width = 205
Option3.ForeColor = 255,255,255
Option3.Name = "Option3"

```

```

*****
C:\COBRATEL\CPT0580.SCX
Left = 54
Top = 232
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 10
Top = 20
Width = 90
Height = 90
Alias = "usuarios"
Database = dbCOBRATEL.dbc
CursorSource = "usuarios"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL@. Cambio
de password."
WindowState = 2
AlwaysOnTop = .F.
Name = "Form1"
PROCEDURE Activate
* Al iniciar...
thisform.Text1.value=usuario
thisform.text2.value=grupo
thisform.text3.value=date()
thisform.text5.value=nombre

thisform.label8.visible=.F.
thisform.Text8.visible=.F.
thisform.label9.visible=.F.
thisform.text9.visible=.F.
thisform.Command1.visible=.F.

ENDPROC
FontBold = .T.
FontSize = 16
BackStyle = 0
Caption = "Cambio de password"
Height = 31
Left = 100
Top = 40
Width = 280
TabIndex = 6
ForeColor = 255,255,255
Name = "Label1"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú de utilerías"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 5
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)

```

```

ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0500
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
Enabled = .F.
Height = 21
Left = 490
TabIndex = 15
Top = 80
Visible = .F.
Width = 80
Name = "Text1"
Caption = "Usuario"
Height = 13
Left = 520
Top = 13
Visible = .T.
Width = 40
TabIndex = 9
ForeColor = 255,255,255
Name = "Embossedlabel1"
Enabled = .F.
Height = 21
Left = 588
TabIndex = 16
Top = 78
Visible = .F.
Width = 80
Name = "Text2"
Caption = "Grupo"
Height = 13
Left = 588
Top = 66
Visible = .F.
Width = 40
TabIndex = 8
Name = "Embossedlabel2"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 17
Top = 40
Width = 80
Name = "Text3"
Caption = "Fecha"
Height = 13
Left = 646
Top = 43
Visible = .T.
Width = 40
TabIndex = 14
ForeColor = 255,255,255
Name = "Embossedlabel3"
Enabled = .F.
Height = 21
Left = 680
TabIndex = 19
Top = 80
Visible = .F.

```

```

Width = 80
Name = "Text4"
BackStyle = 1
Enabled = .F.
Height = 21
Left = 560
TabIndex = 18
Top = 10
Visible = .T.
Width = 200
Name = "Text5"
Caption = "Clave de usuario"
Height = 13
Left = 149
Top = 68
Visible = .F.
Width = 90
TabIndex = 7
Name = "Embossedlabel4"
BackStyle = 0
Caption = "Seleccione el usuario"
Height = 21
Left = 160
Top = 140
Width = 140
TabIndex = 20
ForeColor = 255,255,255
Name = "Label2"
ColumnCount = 0
ColumnWidths = ""
RowSourceType = 6
RowSource = "usuarios.nom_usu"
FirstElement = 1
Height = 21
Left = 120
NumberOfElements = 0
Style = 2
TabIndex = 1
Top = 180
Width = 20
Name = "Combo1"
PROCEDURE InteractiveChange
* Al seleccionar un usuario
* posicionarse en su recno()

GO record recno()
thisform.Text6.value=nom_usu
thisform.Text7.value=pass_usu

thisform.label8.visible=.t.
thisform.Text8.visible=.t.
thisform.label9.visible=.t.
thisform.text9.visible=.t.
thisform.Command1.visible=.T.

ENDPROC
Enabled = .F.
Height = 21
Left = 160
TabIndex = 21
Top = 180
Width = 281
Name = "Text6"
Enabled = .F.
Height = 21
Left = 460
TabIndex = 22
Top = 180
Width = 100
Name = "Text7"

```

```

Caption = "Usuario"
Height = 13
Left = 160
Top = 167
Visible = .T.
Width = 40
TabIndex = 12
ForeColor = 255,255,255
Name = "Embossedlabel5"
Caption = "Password actual"
Height = 13
Left = 460
Top = 160
Visible = .T.
Width = 100
TabIndex = 13
ForeColor = 255,255,255
Name = "Embossedlabel6"
Height = 21
Left = 160
TabIndex = 2
Top = 280
Width = 100
PasswordChar = ""
Name = "Text8"
Caption = "Nuevo password"
Height = 13
Left = 160
Top = 260
Visible = .T.
Width = 100
TabIndex = 10
ForeColor = 255,255,255
Name = "label8"
Height = 21
Left = 320
TabIndex = 3
Top = 280
Width = 100
PasswordChar = ""
Name = "Text9"
Caption = "Confirme el nuevo password"
Height = 13
Left = 300
Top = 260
Visible = .T.
Width = 140
TabIndex = 11
ForeColor = 255,255,255
Name = "label9"
AutoSize = .F.
Top = 240
Left = 480
Height = 60
Width = 100
WordWrap = .T.
Picture = checkmrk.ico
Caption = "Asigna password"
StatusBarText = "Asigna el nuevo password al usuario."
TabIndex = 4
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton asigna password

```

```

IF (thisform.Text8.value =
thisform.text9.value) and
!empty(thisform.Text8.value) and
!empty(thisform.text9.value)
REPLACE pass_usu with
alltrim(thisform.Text8.value)

thisform.Text7.value=thisform.Text8.
value
MESSAGEBOX('Aceptado el
cambio de
password.',0+64+0,'Operación
finalizada.')
thisform.Text8.value=""
thisform.text9.value=""
thisform.label8.visible=.F.
thisform.Text8.visible=.F.
thisform.label9.visible=.F.
thisform.text9.visible=.F.
thisform.Command1.visible=.F.
ELSE
MESSAGEBOX('Los passwords
proporcionados no son válidos.
¡Verifique por
favor!',0+64+0,'Operación
cancelada')
ENDIF

ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
Picture = key04.ico
Stretch = 1
Height = 42
Left = 40
Top = 20
Width = 42
Name = "Image1"

*****
C:\COBRATEL\CPT0600.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Bitácoras"
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Bitácoras"
Height = 30
Left = 100
Top = 40
Width = 140
TabIndex = 1
ForeColor = 255,255,255

```



```

Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú principal"
StatusBarText = "Salir del sistema."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form menuprin
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
Top = 140
Left = 160
Height = 60
Width = 81
WordWrap = .T.
Picture = depu1.ico
Caption = ""
StatusBarText = "Consulta de
bitácora de procesos."
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton consulta bitácoras de
procesos
thisform.release
DO form cpt0610
ENDPROC
Top = 220
Left = 160
Height = 60
Width = 81
WordWrap = .T.
Picture = depu2.ico
Caption = ""

```

```

StatusBarText = "Consulta a la
bitácora de transmisiones."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton bitácoras de transmisiones
thisform.release
DO form cpt0620
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
BackStyle = 0
Caption = "Consulta bitácora de
procesos"
Height = 21
Left = 250
Top = 160
Width = 210
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Consulta bitácora de
transmisiones"
Height = 21
Left = 250
Top = 240
Width = 210
ForeColor = 255,255,255
Name = "Label2"
Picture = note14.ico
Stretch = 1
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"

```

```

*****
C:\COBRATEL\CPT0610.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Consulta
bitácora de procesos."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=date()
thisform.text2.value=date()

```

```

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Bitácoras de procesos"
Height = 30
Left = 100
Top = 40
Width = 280
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09lt.ico
Caption = "Menú de bitácoras"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0600
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Fecha inicial"
Height = 21
Left = 150
Top = 150
Width = 70
ForeColor = 255,255,255
Name = "Label1"
Height = 21
Left = 150
StatusBarText = "Fecha inicial de la
consulta"
Top = 170
Width = 101
Name = "Text1"
PROCEDURE LostFocus
* Igualar fecha final
thisform.Label1.forecolor=rgb(0,0,0)
thisform.text2.value=this.value
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
Height = 21
Left = 330

```

```

StatusBarText = "Fecha final de la
consulta."
Top = 170
Width = 101
Name = "Text2"
PROCEDURE Valid
* Validar rango de fechas
IF this.value>=thisform.Text1.value
then
RETURN .t.
ELSE
MESSAGEBOX("El rango de fechas
proporcionado no es válido.
¡Verifique por favor!,0+16+0,'Error.')
```

```

RETURN .f.
ENDIF
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Fecha final"
Height = 21
Left = 330
Top = 150
Width = 70
ForeColor = 255,255,255
Name = "Label2"
Top = 230
Left = 240
Height = 41
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Consultas a la
bitácora de procesos."
TabIndex = 9
Name = "Command1"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
REPORT form procesos for
(fechar_bp>=thisform.Text1.value
.and.
fechar_bp<=thisform.text2.value)
preview
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Consulta de bitácora de
procesos"
Height = 21
Left = 330
```

```

Top = 250
Width = 200
ForeColor = 255,255,255
Name = "Label3"
Picture = depu1.ico
Stretch = 1
BackStyle = 0
Height = 40
Left = 40
Top = 20
Width = 40
Name = "Image1"
BackStyle = 0
Caption = "Proporcione el rango de
fechas que desea consultar"
Height = 21
Left = 150
Top = 110
Width = 310
ForeColor = 255,255,255
Name = "Label4"

```

```

*****
C:\COBRATEL\CPT0620.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Consulta
bitácora de transmisiones."
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* Inicializa periodo de busqueda
thisform.Text1.value=date()
thisform.text2.value=date()

```

```

ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Bitácoras de
transmisiones"
Height = 30
Left = 100
Top = 40
Width = 280
TabIndex = 1
ForeColor = 255,255,255
Name = "Label10"
AutoSize = .F.
Top = 340
Left = 640
Height = 60
Width = 80
WordWrap = .T.
Picture = arw09it.ico
Caption = "Menú de bitácoras"
StatusBarText = "Regresar al menú
de bitácoras."
TabIndex = 9

```

```

Name = "Command10"
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form cpt0600
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
BackStyle = 0
Caption = "Fecha inicial"
Height = 21
Left = 160
Top = 160
Width = 70
ForeColor = 255,255,255
Name = "Label1"
Height = 21
Left = 160
StatusBarText = "Fecha inicial de la
consulta"
Top = 180
Width = 101
Name = "Text1"
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE LostFocus
* Igualar fecha final
thisform.Label1.forecolor=rgb(0,0,0)
thisform.text2.value=this.value
ENDPROC
Height = 21
Left = 340
StatusBarText = "Fecha final de la
consulta."
Top = 180
Width = 101
Name = "Text2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Valid
* Validar rango de fechas
IF this.value>=thisform.Text1.value
then
RETURN .t.
ELSE
MESSAGEBOX("El rango de fechas
proporcionado no es válido.
¡Verifique por favor!,0+16+0,'Error.')
```

```

Caption = "Fecha final"
Height = 21
Left = 340
Top = 160
Width = 70
ForeColor = 255,255,255
Name = "Label2"
Top = 240
Left = 250
Height = 41
Width = 81
WordWrap = .T.
Picture = checkmrk.ico
Caption = ""
StatusBarText = "Consultas a la
bitácora de transmisiones."
TabIndex = 9
Name = "Command1"
PROCEDURE LostFocus
* Señalar
thisform.Label3.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Salir
REPORT form trxs for
(fech_bt>=thisform.Text1.value .and.
fech_bt<=thisform.text2.value)
preview
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
BackStyle = 0
Caption = "Consulta de bitácora de
transmisiones"
Height = 21
Left = 340
Top = 260
Width = 230
ForeColor = 255,255,255
Name = "Label3"
Picture = depu2.ico
Stretch = 1
BackStyle = 0
Height = 42
Left = 40
Top = 20
Width = 42
Name = "Image1"
BackStyle = 0
Caption = "Proporcione el rango de
fechas que desea consultar"
Height = 21
Left = 160
Top = 120
Width = 310
ForeColor = 255,255,255
Name = "Label4"
*****
C:\COBRATEL\CRDFLE11.ICO
*****
C:\COBRATEL\DBC\COBRATEL.DBC
***** * C:\COBRATEL\DEPU1.ICO

```

```

***** * C:\COBRATEL\DEPU2.ICO
***** * C:\COBRATEL\DEPU3.ICO
***** * C:\COBRATEL\DEPU4.ICO
*****
C:\COBRATEL\DEPURA2.ICO
*****
C:\COBRATEL\DIACAJA.FRXL
*****
C:\COBRATEL\ERASE02.ICO
*****
C:\COBRATEL\ESTACION.FRXL
*****
C:\COBRATEL\ESTADISTICO.FRXL
***** * C:\COBRATEL\EYE.ICO
***** * C:\COBRATEL\FILES09.ICO
*****
C:\COBRATEL\FINALIZA.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Salida del
sistema."
WindowState = 2
Name = "Form1"
FontBold = .T.
FontSize = 12
WordWrap = .F.
BackStyle = 0
Caption = "Gracias por utilizar el
Sistema de cobro por terceros"
Height = 30
Left = 180
Top = 190
Width = 410
ForeColor = 255,255,255
Name = "Label1"
FontBold = .T.
FontSize = 24
BackStyle = 0
Caption = "C O B R A T E L ®"
Height = 50
Left = 210
Top = 110
Width = 360
ForeColor = 255,255,255
Name = "Label2"
AutoSize = .F.
Top = 320
Left = 620
Height = 77
Width = 80
WordWrap = .T.
Picture = w95mbx01.ico
Caption = "Salir del sistema"
Name = "Command1"
PROCEDURE Click
* Boton Salir
thisform.release
ENDPROC
*****
C:\COBRATEL\FOLDER04.ICO
***** * C:\COBRATEL\FONDO.BMP
*****
C:\COBRATEL\GRAPH03.ICO
*****
C:\COBRATEL\GRAPH11.ICO

```

```

*****
C:\COBRATEL\GRUPOS.FRXL
***** * C:\COBRATEL\KEY04.ICO
*****
C:\COBRATEL\KEYBRD02.ICO
***** * C:\COBRATEL\KEYS03.ICO
***** * C:\COBRATEL\MAIL03.ICO
***** * C:\COBRATEL\MAIL14.ICO
*****
C:\COBRATEL\MENUPRIN.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 10
Top = 20
Width = 90
Height = 90
Alias = "grupos"
Database = dbCOBRATEL.dbc
CursorSource = "grupos"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Menú
principal"
WindowState = 2
Name = "Form1"
PROCEDURE Activate
* AL INICIALIZARSE VALIDAR
OPCIONES DISPONIBLES POR
GRUPO
* AL QUE PERTENECE EL
USUARIO QUE SE FIRMO

LOCATE FOR cve_gru=grupo
IF FOUND()

thisform.Command1.ENABLED=M01

thisform.Command2.ENABLED=M02

thisform.Command3.ENABLED=M03

thisform.command4.ENABLED=M04

thisform.Command5.ENABLED=M05

thisform.Command6.ENABLED=M06
ENDIF

ENDPROC
FontBold = .T.
FontSize = 20
BackStyle = 0
Caption = "Sistema de cobro por
terceros"
Height = 31
Left = 140
Top = 30
Width = 430
TabIndex = 1

```

```

ForeColor = 255,255,255
Name = "Label10"
AutoSize = .T.
Top = 360
Left = 640
Height = 59
Width = 75
Picture = w95mbx01.ico
Caption = "Salir"
StatusBarText = "Salir del sistema."
TabIndex = 9
Name = "Command10"
PROCEDURE LostFocus
* Señalar
this.forecolor=rgb(0,0,0)
ENDPROC
PROCEDURE GotFocus
* Señalar
this.forecolor=rgb(255,255,128)
ENDPROC
PROCEDURE Click
* Boton Salir
thisform.release
DO form finaliza
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus
ENDPROC
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Menú principal"
Height = 21
Left = 250
Top = 80
Width = 150
TabIndex = 2
ForeColor = 255,255,255
Name = "Label20"
AutoSize = .F.
Top = 140
Left = 60
Height = 59
Width = 80
WordWrap = .T.
Picture = net01.ico
Caption = ""
StatusBarText = "Transferencia de
archivos."
TabIndex = 3
ToolTipText = ""
Name = "Command1"
PROCEDURE LostFocus
thisform.Label1.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label1.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
PROCEDURE Click
* Boton Transferencia de archivos

```

```

thisform.release
valor1=1
valor2=0
DO form cpt0100
ENDPROC
Top = 220
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = vb416mom.bmp
Caption = ""
StatusBarText = "Procesos
especiales."
TabIndex = 4
ToolTipText = ""
Name = "Command2"
PROCEDURE LostFocus
thisform.Label2.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label2.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Procesos especiales
thisform.release
DO form cpt0200
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
Top = 300
Left = 60
Height = 60
Width = 81
WordWrap = .T.
Picture = net06.ico
Caption = ""
StatusBarText = "Generación de
reportes."
TabIndex = 5
ToolTipText = ""
Name = "Command3"
PROCEDURE LostFocus
thisform.Label3.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label3.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton Generación de reportes
thisform.release
DO form cpt0300
ENDPROC
Top = 140
Left = 420

```

```

Height = 60
Width = 81
WordWrap = .T.
Picture = crdfile11.ico
Caption = ""
StatusBarText = "Mantenimiento de
catálogos."
TabIndex = 6
ToolTipText = ""
Name = "Command4"
PROCEDURE LostFocus
thisform.Label4.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label4.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Mantenimiento de catálogos
thisform.release
DO form cpt0400
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
Top = 220
Left = 420
Height = 60
Width = 81
WordWrap = .T.
Picture = wrench.ico
Caption = ""
StatusBarText = "Utillerias."
TabIndex = 7
ToolTipText = ""
Name = "Command5"
PROCEDURE LostFocus
thisform.Label5.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.Label5.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

```

```

ENDPROC
PROCEDURE Click
* Boton Utillerias
thisform.release
DO form cpt0500
ENDPROC
Top = 300
Left = 420
Height = 60
Width = 81
WordWrap = .T.
Picture = note14.ico
Caption = ""
StatusBarText = "Bitácoras."
TabIndex = 8
ToolTipText = ""
Name = "Command6"

```

```

PROCEDURE LostFocus
thisform.label6.forecolor=rgb(255,25
5,255)
ENDPROC
PROCEDURE GotFocus
thisform.label6.forecolor=rgb(255,25
5,128)
ENDPROC
PROCEDURE Click
* Boton Bitacoras
thisform.release
DO form cpt0600
ENDPROC
PROCEDURE MouseMove
LPARAMETERS nButton, nShift,
nXCoord, nYCoord
this.setfocus

ENDPROC
BackStyle = 0
Caption = "Transferencia de
archivos"
Height = 21
Left = 150
Top = 160
Width = 160
ForeColor = 255,255,255
Name = "Label1"
BackStyle = 0
Caption = "Procesos especiales"
Height = 21
Left = 150
Top = 240
Width = 160
ForeColor = 255,255,255
Name = "Label2"
BackStyle = 0
Caption = "Generación de reportes"
Height = 21
Left = 150
Top = 320
Width = 160
ForeColor = 255,255,255
Name = "Label3"
BackStyle = 0
Caption = "Mantenimiento de
catálogos"
Height = 21
Left = 510
Top = 160
Width = 160
ForeColor = 255,255,255
Name = "Label4"
BackStyle = 0
Caption = "Utilerías"
Height = 21
Left = 510
Top = 240
Width = 160
ForeColor = 255,255,255
Name = "Label5"
BackStyle = 0
Caption = "Bitácoras"
Height = 21
Left = 510
Top = 320
Width = 160
ForeColor = 255,255,255
Name = "Label6"
***** * C:\COBRATEL\MISC25.ICO

```

```

***** * C:\COBRATEL\MISC27.ICO
***** * C:\COBRATEL\MISC29.ICO
***** *
C:\COBRATEL\MONITR01.ICO
***** * C:\COBRATEL\NET01.ICO
***** * C:\COBRATEL\NET06.ICO
***** * C:\COBRATEL\NOTE02.ICO
***** * C:\COBRATEL\NOTE03.ICO
***** * C:\COBRATEL\NOTE14.ICO
***** * C:\COBRATEL\PEN07.ICO
***** *
C:\COBRATEL\PENCIL05.ICO
***** *
C:\COBRATEL\PHONE11.ICO
***** *
C:\COBRATEL\PRINCIPAL.SCX
Name = "Dataenvironment"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Sistema
de cobranza por terceros."
WindowState = 2
BackColor = 192,192,192
Name = "principal"
PROCEDURE Activate
* Formulario principal
CD c:\COBRATEL
SET date to ymd
SET century on
SET deleted on
SET talk off

* INICIALIZAR VARIABLES
GLOBALES PUBLICAS
PUBLIC usuario && que CLAVE de
usuario esta activo utilizando el
sistema
PUBLIC grupo && a que grupo
pertenece el usuario activo
PUBLIC nombre && nombre del
usuario que se firmó
PUBLIC no_intentos && cuantos
intentos lleva para ingresar al
sistema
PUBLIC regactual && en que
registro estoy trabajando
PUBLIC valor1 && valor que debe
llevar la op1 en cpt100
PUBLIC valor2 && valor que debe
llevar la op1 en cpt100

usuario = space(7) && clave de
acceso del usuario que utilizar el
sistema
grupo = space(3)

ENDPROC
FontBold = .T.
FontSize = 20
BackStyle = 0
Caption = "C O B R A T E L ®"
DragIcon =
Height = 40
Left = 210

```

```

MousePointer = 0
Top = 240
Width = 310
ForeColor = 255,255,255
Name = "Label1"
FontBold = .T.
FontSize = 14
BackStyle = 0
Caption = "Sistema de cobranza por
terceros"
Height = 30
Left = 200
Top = 280
Width = 330
ForeColor = 255,255,255
Name = "Label2"
Picture = phone11.ico
Stretch = 1
BackStyle = 0
Height = 70
Left = 300
Top = 130
Width = 90
Name = "Image1"
PROCEDURE Click
* Salir del formulario
thisform.release
no_intentos = 1
DO form validar
ENDPROC
***** *
C:\COBRATEL\PROCESOS.FRX
***** * C:\COBRATEL\RESTO.ICO
***** * C:\COBRATEL\TRXS.FRX
***** *
C:\COBRATEL\USUARIOS.FRX
***** *
C:\COBRATEL\VALIDAR.SCX
Left = 1
Top = 220
Width = 520
Height = 200
Name = "Dataenvironment"
Left = 10
Top = 20
Width = 90
Height = 90
Alias = "usuarios"
Database = dbCOBRATEL.dbc
CursorSource = "usuarios"
Name = "Cursor1"
Top = 0
Left = 0
Height = 574
Width = 793
DoCreate = .T.
Picture = fondo.bmp
Caption = "COBRATEL®. Validación
de entrada al sistema."
WindowState = 2
Name = "valida"
FontBold = .T.
FontSize = 12
BackStyle = 0
Caption = "Usuario"
Height = 31
Left = 180
Top = 130
Width = 340
TabIndex = 5

```

```

ForeColor = 255,255,255
Name = "Label1"
FontSize = 14
Height = 30
Left = 220
MaxLength = 15
TabIndex = 1
Top = 150
Width = 221
PasswordChar = ""
Name = "Text1"
AutoSize = .T.
Top = 360
Left = 640
Height = 59
Width = 75
Picture = w95mbx01.ico
Caption = "Salir"
TabIndex = 4
Name = "Command1"
PROCEDURE Click
* Botón Salir
thisform.release
DO form finaliza
ENDPROC
AutoSize = .T.
Top = 360
Left = 520
Height = 59
Width = 88
Picture = checkmrk.ico
Caption = "Validar"
TabIndex = 3
Name = "Command2"
PROCEDURE Click
* Botón Validar

*use usuarios
IF no_intentos<=3
GO top
LOCATE for
cve_usua=alltrim(thisform.Text1.valu
e).and.
pass_usu=alltrim(thisform.text2.value
)
IF found()
usuario = pass_usu
grupo = gpo_usu
nombre = nom_usu
USE
thisform.release
DO form menuprin
ELSE
no_intentos = no_intentos + 1
MESSAGEBOX("¡ Clave de
usuario errónea. Intente nuevamente
por favor! ",0+16+0,'Error.')
ENDIF
ELSE
MESSAGEBOX("¡ Ud. no tiene
acceso al sistema ! ",0+16+0,'Acceso
denegado.')
thisform.release
DO form finaliza
ENDIF

ENDPROC

```

```

FontSize = 14
Height = 30
Left = 220
MaxLength = 15
TabIndex = 2
Top = 229
Width = 221
PasswordChar = ""
Name = "Text2"
FontBold = .T.
FontSize = 12
BackStyle = 0
Caption = "Password"
Height = 31
Left = 180
Top = 209
Width = 340
TabIndex = 6
ForeColor = 255,255,255
Name = "Label2"
FontBold = .T.
FontSize = 20
BackStyle = 0
Caption = "Entrada al sistema"
Height = 31
Left = 200
Top = 40
Width = 340
TabIndex = 5
ForeColor = 255,255,255
Name = "Label3"
Picture = keys03.ico
Stretch = 0
BackStyle = 1
Height = 32
Left = 140
Top = 40
Width = 32
Name = "Image1"
*****
C:\COBRATEL\VB416MOM.BMP
*****
C:\COBRATEL\VIACOBRO.FRX
*****
C:\COBRATEL\W95MBX01.ICO
*****
C:\COBRATEL\WRENCH.ICO
*****
C:\COBRATEL\WZBACK.BMP
*****
C:\COBRATEL\WZCLOSE.BMP
*****
C:\COBRATEL\WZCLOSE.MSK
*****
C:\COBRATEL\WZDELETE.BMP
*****
C:\COBRATEL\WZDELETE.MSK
*****
C:\COBRATEL\WZEDIT.BMP
*****
C:\COBRATEL\WZEDIT.MSK
*****
C:\COBRATEL\WZEND.BMP
*****
C:\COBRATEL\WZLOCATE.BMP
*****
C:\COBRATEL\WZLOCATE.MSK
*****
C:\COBRATEL\WZNEW.BMP
*****
C:\COBRATEL\WZNEW.MSK
*****
C:\COBRATEL\WZNEXT.BMP
*****
C:\COBRATEL\WZPRINT.BMP

```

```

*****
C:\COBRATEL\WZPRINT.MSK
*****
C:\COBRATEL\WZTOP.BMP

```

Esta tesis:

SISTEMA DE COBRANZA POR TERCEROS

Se realizó en conjunto por los alumnos:

Gabriel Osorio Pérez	Facultad de Ingeniería
Jorge Eduardo Villalobos Moreno	Facultad de Ingeniería
Leonardo Luis Arcos	E.N.E.P. Aragón

Autorización de la Facultad de Ingeniería **No. 001/004**, dentro del Programa de Apoyo para la Titulación de la División de Educación Continua, teniéndose que realizar los tramites necesarios de cada alumno para su respectivo plantel.

De acuerdo a la revisión de los profesores de la E.N.E.P. Aragón se realizaron las siguientes correcciones, marcadas con el símbolo * en el cuerpo de la tesis y se indican a continuación.

Capítulo I Marco teórico:

Página 6.- Se elimina el texto "que estarán relacionadas con la preparación táctica del proyecto".

Página 7.-Se cambia el texto "lo que significa que se podía haber logrado uno mayor".

Página 10.- Se anexa el texto de la figura 1.1.3.2 en el texto normal.

Página 12.- Se cambia el texto "Deben comprarse y mantenerse materiales hasta que el ingreso marginal de adquirir y mantenerlos iguales al costo marginal".

Página 23.- Se pone el título correspondiente a la figura 1.1.5.1 Organización de una corporación.

Página 24.- Se anexa la figura 1.1.5.2 Proceso de facturación integro.

Página 28.- Se cambia el texto "Desde entonces se han desarrollado otros lenguajes incluyendo Foxpro".

Página 31.- Se elimina el texto de la figura 1.2.4.2 "dos conjuntos de objetos de distintos tipos" y se cambia el título de la figura.

Página 32.- Se elimina de la figura 1.2.4.3; Relaciones, por lo que también se elimina del título de la figura.

Página 34.- Se cambia el texto "y la subclase (instancia de clases)".

Página 94.- Se cambia de lugar la figura 1.5.1.3

Página 95.- Se eliminan tipos de editores "Ined, vi ed".

Capítulo II Planteamiento del problema y propuesta de solución:

Página 240.- El texto indicado se sube a la parte de Software.

Página 241.- El texto indicado se cambia de lugar.

Capítulo III Desarrollo e implementación del sistema

Página 290.- Se elimina el texto "Los diagramas de estados se refieren a otros modelos".