



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA BANCA EN SU
EMPRESA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
DANIEL HILARIO LOPEZ GARCIA



MEXICO, D. F.

FEBRERO 1995

FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

**Este proyecto fue realizado bajo la dirección del
Ing. José Luis Olmedo Talavera y del
Ing. Adolfo Millán Nájera.**

**Con respeto a mis padres, Gloria e Hilario,
quienes me lo dieron todo.**

AGRADECIMIENTOS

Al Ing. José Luis Olmedo Talavera, director de este trabajo de tesis, por su valiosa ayuda y por su inmejorable aportación de sugerencias y consejos durante la realización del proyecto.

Al Ing. Adolfo A. Millán Nájera, codirector del proyecto, quien siempre mostró un gran interés y quien siempre estuvo dispuesto a brindar parte de su valioso tiempo al análisis y crítica de este trabajo.

A mis padres Gloria e Hilario quienes siempre han sido un ejemplo a seguir y a quienes debo todo lo que soy y todos los logros en mi vida.

A mis hermanos: Angel, Lourdes, Guadalupe y David con quienes compartí toda una inolvidable infancia.

A mi compañera, amiga y esposa, Lourdes, por su apoyo, comprensión y cariño.

A la familia Carreón Romero por su invaluable apoyo en momentos tan difíciles.

A todos mis amigos con quienes compartí y comparto grandes momentos de cultura, aventura y diversión.

A UNISYS de México por su apoyo incondicional.

A la Universidad Nacional Autónoma de México, forjadora de grandes profesionistas.

ÍNDICE

Página

I INTRODUCCIÓN

1.1 ¿Qué es un sistema?	1-1
1.2 ¿Qué es información?	1-2
1.3 Sistemas de información	1-2
1.4 ¿Qué es un banco ?	1-2
1.5 La información y la toma de decisiones	1-2

II SITUACIÓN ACTUAL DE LA BANCA

II.1 Estructura del sistema financiero mexicano	
II.1.1 El sistema bancario	2-1
II.1.2 Organizaciones auxiliares de crédito	2-1
II.1.3 Intermediarios en el mercado de valores	2-1
II.1.4 El sistema asegurador y el de finanzas	2-1
II.1.5 Autoridades financieras	2-2
II.1.6 Características generales del sistema financiero	2-2
II.2 Tendencias	
II.2.1 Servicio al cliente	2-4
II.3 Tecnología	
II.3.1 Centro de cómputo bancario	2-5
II.4 Competencia bancaria	2-7

III ESQUEMA GENERAL DE UNA SOLUCIÓN BANCARIA

III.1 Sistema en un equipo central	
III.1.1 Objetivos del SFB	3-1
III.1.2 Concepto de núcleo	3-1
III.1.3 Características del SFB	3-2
III.1.4 Elementos de hardware	3-3
III.1.5 Elementos de software	3-3
III.1.6 Funcionalidades relevantes	3-4
III.2 Sistema en sucursales	
III.2.1 Aplicación de caja	3-6

	<i>Página</i>
III.3 Medios de comunicación	
III.3.1 Medios conductores	3-9
III.3.2 Medios radiados	3-11
III.4 Sistema en su empresa	
III.4.1 Alcance del proyecto	3-13
IV FASE DE INVESTIGACIÓN	
IV.1 Objetivos, metas y restricciones	
IV.1.1 Objetivos del sistema BANCEMP	4-1
IV.1.2 Metas	4-1
IV.1.3 Restricciones	4-1
IV.2 Funcionamiento del negocio	4-2
IV.3 Requerimientos del usuario	4-2
IV.4 Selección de las herramientas de desarrollo	
IV.4.1 Requerimientos de hardware	4-3
IV.4.2 Requerimientos de software	4-4
IV.5 Justificaciones de las elecciones de hardware y software	4-4
IV.6 Visual Basic	4-5
IV.7 Infoconnect	4-6
V DEFINICIÓN Y DISEÑO	
V.1 Estándares de diseño	
V.1.1 Diccionario de datos	5-1
V.1.2 Convenciones de nomenclatura	5-1
V.1.3 Diseño de pantallas	5-2
V.1.4 Ejemplos de aplicación de estándares	5-5
V.2 Modelo	5-7
V.3 Diseño	
V.3.1 Interfaz con el usuario	5-12
V.3.2 Módulo de administración	5-16

V.3.3 Módulo de comunicaciones	5-20
VI DESARROLLO	
VI.1 Estándares de desarrollo	6-1
VI.2 Descripción de los bloques de lógica	
VI.2.1 Menú principal	6-1
VI.2.2 Menú de administración	6-2
VI.2.3 Ventana de cuentas	6-2
VI.2.4 Ventana de usuarios	6-3
VI.2.5 Ventana de cliente	6-4
VI.2.6 Ventana de servicios	6-5
VI.2.7 Menú de comunicaciones	6-5
VI.2.8 Ventana de conexión	6-6
VI.2.9 Menú de consultas	6-6
VI.2.10 Menú de transferencias	6-7
VI.2.11 Cuentas del cliente	6-7
VI.2.12 Consulta de saldos	6-7
VI.2.13 Consulta de movimientos	6-7
VI.2.14 Consulta de chequeras	6-8
VI.2.15 Consulta de cheques	6-8
VI.2.16 Consulta clientes relacionados	6-8
VI.2.17 Transferencia de fondos	6-9
VI.2.18 Pago de servicios	6-9
VI.2.19 Envío de mensajes	6-9
VI.2.20 Procedimientos y funciones globales	6-9
VI.3 Programación del sistema	
VI.3.1 Lógica asociada al menú principal	6-11
VI.3.2 Lógica asociada al menú de administración	6-12
VI.3.3 Lógica asociada a la ventana de cuentas	6-13
VI.3.4 Lógica asociada a la ventana de usuarios	6-18
VI.3.5 Lógica asociada a la ventana de cliente	6-22
VI.3.6 Lógica asociada a la ventana de servicios	6-24
VI.3.7 Lógica asociada al menú de comunicaciones	6-29
VI.3.8 Lógica asociada a la ventana de conexión	6-32
VI.3.9 Lógica asociada al menú de consultas	6-34
VI.3.10 Lógica asociada al menú de transferencias	6-35
VI.3.11 Lógica asociada a la consulta cuentas del cliente	6-35
VI.3.12 Lógica asociada a la consulta de saldo	6-38
VI.3.13 Lógica asociada a la consulta de movimientos	6-41
VI.3.14 Lógica asociada a la consulta de chequeras	6-44

VI.3.15	Lógica asociada a la consulta de cheques	6-47
VI.3.16	Lógica asociada a la consulta de clientes relacionados	6-50
VI.3.17	Lógica asociada a la transferencia de fondos	6-52
VI.3.18	Lógica asociada al pago de servicios	6-54
VI.3.19	Lógica asociada al envío de mensajes	6-55
VI.3.20	Procedimientos y funciones globales	6-56
VI.4	Procedimientos de seguridad	
VI.4.1	Seguridad local	6-72
VI.4.2	Seguridad central	6-72
VII PRUEBAS DEL SISTEMA.		
VII.1	Métodos de prueba	7-1
VII.1.1	Prueba de caja blanca	7-2
VII.1.2	Prueba de caja negra	7-2
VII.1.3	Prueba de seguridad	7-2
VII.2	Pruebas a BANCEMP	7-3
CONCLUSIONES.		I
BIBLIOGRAFÍA.		III
APÉNDICES.		IV
GLOSARIO.		VII

CAPÍTULO I
INTRODUCCIÓN

I. INTRODUCCIÓN.

El objetivo de este proyecto es proveer a la banca latinoamericana, que cuente para su operación con el Sistema Financiero Bancario (SFB) comercializado por UNISYS Corporation, de un producto que satisfaga las tendencias actuales de servicio al cliente bancario; tendencias que obligan a los bancos a acercar cada vez más los servicios que ofrecen a sus clientes.

Atendiendo a estas necesidades surge el requerimiento de ofrecer un producto que permita a los clientes bancarios realizar algunas de las operaciones que más frecuentemente efectúan sobre sus cuentas desde la comodidad de sus oficinas u hogares. Para ello hemos diseñado el sistema para computadoras personales BANCEMP que, apoyado en algunos dispositivos de comunicaciones, es capaz de conectarse al sistema central bancario a través de la línea telefónica bajo un completo esquema de seguridad.

El sistema BANCEMP ha sido diseñado bajo un estricto esquema de estandarización y flexibilidad, con el propósito de facilitar su mantenimiento y la posible necesidad de ampliar sus funcionalidades y quizás poder conectarlo a un sistema bancario distinto a SFB con un mínimo de modificaciones y manteniendo su estructura original. De igual manera, atendiendo a las tendencias actuales de sistemas para computadoras personales, BANCEMP fue creado para ser ejecutado bajo el ambiente operativo WINDOWS, ofreciendo al cliente una solución que ya le resulte familiar desde el punto de vista visual, y por lo tanto facilitando su aprendizaje y pudiendo utilizar el sistema casi inmediatamente después de instalarlo.

BANCEMP fue escrito usando VISUAL BASIC ®, que por ser un lenguaje simple, pero a la vez muy poderoso, resulta fácil conseguir personal capacitado para dar mantenimiento al sistema.

A lo largo de este documento se hace referencia constantemente a nombres de productos que son una marca registrada y que con el propósito de abreviar se aluden a ellos como sigue:

-Microsoft ® WINDOWS ™ es una marca registrada de Microsoft Corporation y se hace referencia a ella sólo como Windows.

-Microsoft ® VISUAL BASIC ™ es una marca registrada de Microsoft Corporation y se hace referencia a ella sólo como Visual Basic.

-INFOConnect ™ es una marca registrada de Unisys Corporation y se hace referencia a ella sólo como Infoconnect.

I.1 ¿QUÉ ES UN SISTEMA?

Un sistema es una forma de percibir un conjunto de elementos y actividades relacionados entre sí con la finalidad de lograr algún propósito, de tal manera que las relaciones puedan visualizarse y por lo tanto comprenderse.

I.2 ¿QUÉ ES INFORMACIÓN?

Información son *datos* que tienen contexto o significado, y son relevantes para alguien en particular. Por el contrario, *datos* son material no procesado y son la base para producir información.

I.3 SISTEMAS DE INFORMACIÓN.

En virtud de que la información es la base de todas las actividades realizadas en una compañía, deben desarrollarse sistemas para producirla y administrarla. El objetivo de tales sistemas es asegurar la disponibilidad de información exacta y confiable cuando se necesite y presentarla en forma fácilmente aprovechable.

I.4 ¿QUÉ ES UN BANCO?

Un banco es una institución que tiene como objetivo captar dinero y colocarlo en forma de créditos comerciales o créditos a personas físicas con fines lucrativos. Debido a que gran parte de la captación de fondos proviene de los cuentahabientes del banco, se requiere de una serie de servicios ofrecidos por la institución que les resulten atractivos y los conduzcan a incrementar sus ahorros.

I.5 LA INFORMACIÓN Y LA TOMA DE DECISIONES.

La toma de decisiones es un proceso repetitivo que involucra el reconocimiento de la oportunidad o del problema, la obtención de datos, la comprensión del contenido de la información, la búsqueda de alternativas, su evaluación, elección e implementación. Así, para algún cliente de alguna institución financiera puede ser vital el conocer el estado de sus cuentas antes de poder tomar alguna decisión.

Las decisiones administrativas tratan sobre la forma de como emplear los recursos para alcanzar los objetivos de la corporación, y las decisiones estratégicas consideran aspectos a un plazo más largo. La toma de decisiones se sustenta básicamente en la información gerencial y un sistema que ponga al alcance información actualizada de las cuentas bancarias con las que se tiene relación y además que permita cursar transacciones de transferencia de fondos, puede ser una pieza fundamental en la toma de decisiones.

Así, el sistema Banca en su Empresa resulta un producto atractivo tanto para los clientes de una institución financiera como para la misma institución.

CAPÍTULO II

SITUACIÓN ACTUAL DE LA BANCA

II.1 ESTRUCTURA DEL SISTEMA FINANCIERO MEXICANO.

El sistema financiero Mexicano comprende: el sistema bancario, aquel correspondiente a las organizaciones y actividades auxiliares de crédito, el referido a los intermediarios que actúan de manera directa y principal en el mercado de valores, el sistema asegurador y el de finanzas.

II.1.1 EL SISTEMA BANCARIO.

El sistema bancario está integrado formalmente por: el Banco de México; las instituciones de crédito, tanto de banca múltiple como de desarrollo; el Patronato del Ahorro Nacional, cuyas operaciones tienen similitud con las de esas instituciones; los fideicomisos públicos creados por el Gobierno Federal para el fomento económico ya que su objeto es precisamente otorgar crédito, y aquellos constituidos para complementar la actividad del Banco Central en las funciones que la ley le encomienda, caso del fideicomiso en que se opera la cámara de compensación (CECOBAN) o del que proporciona cobertura contra riesgos cambiarios (FICORCA).

II.1.2 ORGANIZACIONES AUXILIARES DE CRÉDITO.

Las organizaciones auxiliares de crédito comprenden a los almacenes generales de depósito, las uniones de crédito, las arrendadoras financieras y empresas de factoraje financiero, en tanto que las actividades auxiliares del crédito se encuentran referidas a la operación de las casas de cambio.

II.1.3 INTERMEDIARIOS EN EL MERCADO DE VALORES.

En el mercado de valores operan de manera directa y principal las bolsas de valores, las instituciones para el depósito de valores, las casas de bolsa, los especialistas bursátiles y las sociedades de inversión, así como aquellas empresas operadoras de estas últimas.

II.1.4 EL SISTEMA ASEGURADOR Y EL DE FINANZAS.

Comprende a las instituciones de seguros y a las sociedades mutualistas de seguros, en tanto que el de finanzas estructura de manera uniforme a las instituciones de finanzas.

Todos estos intermediarios participan en varios y diversos segmentos del mercado financiero, haciéndolo mediante el empleo de los diferentes instrumentos de captación y de financiamiento que la ley les asigna. El desarrollo y complejidad que han alcanzado tanto en el mercado de dinero como el de capitales no permiten ya una división clara y formal referida a la parte o partes del mercado que correspondan a cada tipo de intermediación. Esta característica constituye elemento de amplio significado para avanzar hacia la intermediación universal pues la interrelación ya existe en cuanto a que operaciones realizadas por diversos tipos de intermediarios financieros va conduciendo a una mayor integración y complementación operativa entre los mismos.

II.1.5 AUTORIDADES FINANCIERAS.

La regulación y vigilancia del sistema corresponde a la Secretaría de Hacienda y Crédito Público (SHCP), al Banco de México y a los organismos siguientes: Comisión Nacional Bancaria, Comisión Nacional de Valores y Comisión Nacional de Seguros y Fianzas.

A la SHCP le corresponde regular la estructura del sistema financiero y parte de las operaciones que realizan intermediarios financieros no bancarios. Al Banco de México compete la operación del sistema bancario y ciertas actividades, principalmente crediticias, propias de otros intermediarios financieros como casas de bolsa, arrendadoras financieras, empresas de factoraje financiero o casas de cambio. Las Comisiones antes mencionadas tienen, respecto a los sectores a que están referidas, funciones de inspección y vigilancia, además de que tienen ciertas atribuciones para instrumentar políticas y regular operaciones.

II.1.6 CARACTERÍSTICAS GENERALES DEL SISTEMA FINANCIERO.

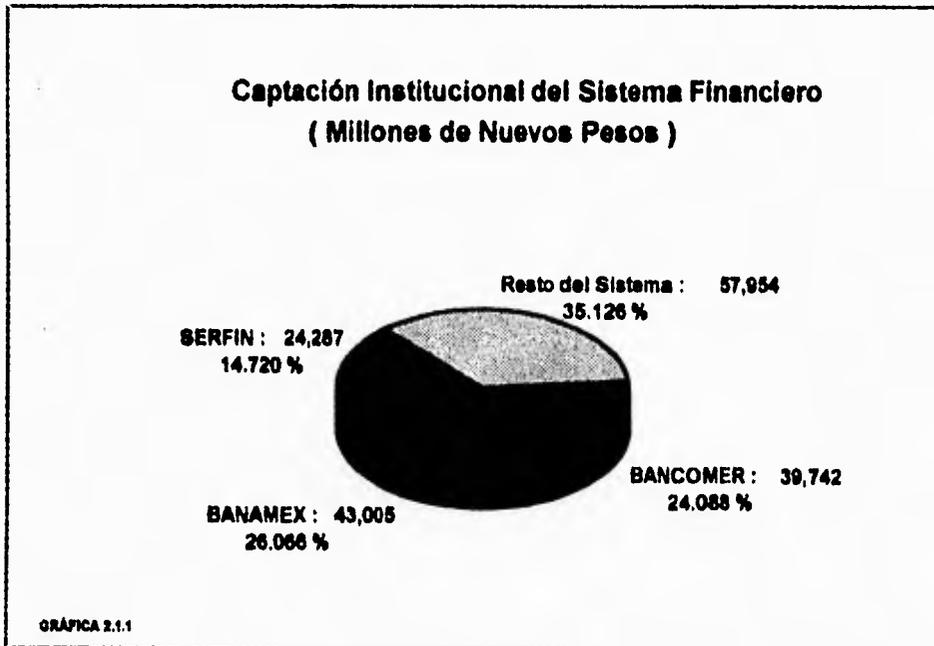
Actualmente en el sistema financiero mexicano destacan ampliamente los bancos y las casas de bolsa a quienes corresponde con creces la mayor participación en el mercado. En él conservan participación mayoritaria las instituciones de crédito, seguidas por las casas de bolsa cuyo desarrollo ha sido intenso en los últimos años, debido en gran medida al crecimiento relevante del mercado de valores el cual también ha sido causa, en ese lapso, de un señalado crecimiento por parte de las sociedades de inversión.

Factor importante en el dinamismo que observa el sistema financiero, respecto al mercado de valores, corresponde a la operación en él de títulos representativos de deuda pública, que hasta hace poco tiempo no eran colocados de manera masiva en el mercado. Otros intermediarios que en los últimos años han tenido también significativa y creciente penetración en el propio mercado son las arrendadoras financieras de empresas de factoraje financiero.

Cabe destacar que en las instituciones de crédito se presenta una importante concentración de los recursos, que la totalidad de ese sistema capta del público, en reducido número de ellas, fenómeno que también puede observarse recientemente en las casas de bolsa.

La gráfica 2.1.1 muestra la concentración a la que nos referimos, tratándose de instituciones de crédito.

Considerando lo anterior, la nueva ley de grupos financieros tiene entre sus propósitos facilitar un más equilibrado crecimiento del sistema financiero sin que ello implique establecer limitaciones inconvenientes en el desarrollo de aquellos integrantes del mismo que mantienen participación mayoritaria en él.



II.2 TENDENCIAS.

Puede preverse que con la celebración del acuerdo de libre comercio entre México, Canadá y Estados Unidos de América se avanza en el proceso tendiente a dar una mayor internacionalización de los servicios financieros.

En las negociaciones del acuerdo se utilizaron tres conceptos básicos: tratamiento nacional, tratamiento recíproco y trato equivalente.

El tratamiento nacional obliga al Estado anfitrión a conceder a las instituciones financieras del país socio un tratamiento similar al que tienen sus propias instituciones financieras. Este principio contrasta con el tratamiento recíproco, en el que el Estado anfitrión otorga a las instituciones financieras del país socio un trato similar al que reciben en su propio país.

La aplicación del tratamiento nacional tiende a favorecer al país más restrictivo en su legislación, donde tanto los bancos nacionales como los extranjeros están sujetos al mismo régimen. Por el contrario, el tratamiento recíproco tiene la ventaja de promover la liberación de la reglamentación respectiva.

Debido a la desproporción de condiciones que afectaba a Canadá, ya fuera por medio del tratamiento nacional o recíproco, su gobierno propuso un criterio compensatorio de negociación: el "Acceso equivalente", bajo el cual Canadá abre sus mercados a la participación de instituciones financieras extranjeras solamente cuando se asegure a las instituciones financieras canadienses un acceso similar e irrestricto a los mercados de otros países signatarios.

La propuesta de Canadá no fue aceptada, con lo cual predominó, en principio, el concepto de tratamiento nacional.

II.2.1 SERVICIO AL CLIENTE.

Hablando de servicios al cliente, la tendencia actual de las instituciones financieras bancarias en el mercado mexicano, es el lograr que el cliente esté "alejado" de las sucursales el mayor tiempo posible, reduciendo sus visitas al banco al mínimo. Es decir, se trata de hacer llegar, de alguna manera, los servicios financieros al cliente.

Es aquí donde este proyecto de tesis puede contribuir al logro de esta meta, llevando hasta la oficina u hogar del cliente los servicios bancarios que requiere realizar sobre sus cuentas, evitándole pérdidas de tiempo y mejorando su relación con el banco, juntos navegando con las tendencias y exigencias del mercado.

II.3 TECNOLOGÍA.

Otro aspecto importante en lo que atañe a la proyección del sistema financiero bancario mexicano es la continuación del desarrollo que en él han tenido la tecnología informática y de telecomunicaciones.

En todo sistema financiero existen mecanismos que permiten que los capitales involucrados en la operación diaria de dicho sistema sean transferidos entre todos sus participantes. Lo que varía en cada sistema financiero es el grado de madurez, eficacia y eficiencia con que funcionan tales mecanismos para la transferencia de dichos recursos.

Dichos mecanismos pueden darse por medio del uso de sistemas tradicionales, efectuando las transferencias con base en documentos y procesos manuales o, lo que es más común actualmente, llevarse a cabo en forma automática e involucrando avanzadas tecnologías de proceso de datos y telecomunicaciones, mismas que permiten lo que hoy se denomina Transferencia Electrónica de Fondos (TEF), la cual hace posible realizar un gran volumen de operaciones con la seguridad y oportunidad requeridas.

En nuestro país, la gran mayoría de las instituciones del sector financiero han sostenido un desarrollo importante en los servicios financieros y de transferencia electrónica de fondos que ofrecen a sus clientes. Asimismo, en los últimos años las instituciones rectoras del sistema financiero mexicano han promovido: el uso común de infraestructuras existentes, el desarrollo de nuevas infraestructuras compartidas y la interconexión de sistemas institucionales; todo ello a fin de diversificar y hacer más eficientes tanto los medios de pago para la transferencia de fondos como los servicios para la transferencia de información entre las instituciones del sector financiero del país.

II.3.1 CENTRO DE CÓMPUTO BANCARIO.

En lo referente a las cámaras de compensación, México cuenta con el denominado Centro de Cómputo Bancario (CECOBAN), fideicomiso de la banca mexicana bajo la administración del Banco de México, que desde hace más de una década se constituyó como la única cámara automatizada de compensación en el país, con operación inicial en la ciudad de México. Actualmente el CECOBAN está a cargo del proyecto cúpula de la banca mexicana para el desarrollo y operación de medios de pago para la Transferencia Electrónica Interbancaria (TEI), y otros servicios interbancarios, que permitan la transferencia electrónica de fondos e información entre instituciones bancarias, a nivel local, regional y nacional. El desarrollo de este proyecto permitirá la creación y utilización extensiva de medios de pago electrónicos que promoverán mayor eficiencia no sólo de las instituciones bancarias sino, en general, de su clientela.

En lo referente a los sistemas de pagos para el cliente final, a la fecha existen en el país tres grandes redes de cajeros automáticos, mismas que en un futuro cercano se interconectarán, a fin de compartir infraestructura instalada y con ello ampliar los servicios proporcionados a los clientes.

Complementando esos sistemas, el Banco de México ha implantado uno más, que permite la transferencia de fondos entre las principales instituciones financieras nacionales, hecho que por sí mismo constituye el primer gran paso en el desarrollo y consolidación de un sistema para transferencias concernientes a operaciones reducidas en número pero de gran valor individual por corresponder a grandes sumas de dinero.

II.4 COMPETENCIA BANCARIA.

Para el 5 de Abril de 1994 la Secretaría de Hacienda y Crédito Público (SHCP) había autorizado la operación de otros nueve intermediarios financieros además de los ya antes autorizados desde 1993. Todo con el propósito de impulsar una mayor competencia interna que permita abaratar el costo del crédito. Sin embargo, algunos funcionarios bancarios y analistas del Grupo de Economistas Asociados (GEA) opinan que la presencia de un mayor número de intermediarios financieros no necesariamente incrementará en corto plazo la competencia ni reducirá los márgenes de intermediación y, en consecuencia, no bajarán las tasas de interés internas.

Con estas nuevas autorizaciones de la SHCP existirán 34 bancos en el país incluyendo City Bank y Banco Obrero. De las 14 nuevas autorizaciones, sólo Banco Inbursa y Banco Interacciones cuentan con cobertura nacional y siete tienen su sede en el Distrito Federal.

En términos teóricos la presencia de un mayor número de bancos debería coadyuvar a incrementar la competencia en el sistema financiero. Sin embargo, la evidencia demuestra que este fenómeno no se presenta en el corto plazo, por lo cual no se prevén en México reducciones del margen financiero, derivadas de un mayor número de intermediarios en el sistema.

Diversos estudios en varios países -como Chile y Estados Unidos- demostraron que la presencia de nuevos bancos no sirvió para aumentar la competencia del sistema financiero, pues los márgenes de intermediación no disminuyeron y no se incrementaron los puntos de venta. Las causas de ello se debieron a que la entrada de nuevos bancos se presentó en pequeña escala y a que los nuevos bancos ingresaron al mercado en periodos de crecimiento.

Para el caso particular de México, GEA prevé la presencia de ambos factores y apunta que la limitación a la banca extranjera en el TLC y el escaso capital disponible de los nuevos bancos nacionales permiten suponer que su participación será insuficiente para estimular la mayor competencia en el mercado. Por lo tanto, no parece ser que hayan reducciones del margen financiero y la captación podría crecer a una tasa real promedio anual de 9.4 % entre 1994 y 1997. Según explica GEA la banca comercial mexicana se ha integrado a los mercados internacionales; No obstante, dicha integración se ha orientado principalmente hacia Estados Unidos, al dejar de lado a los mercados latinoamericanos.

CAPÍTULO III

ESQUEMA GENERAL DE UNA SOLUCIÓN BANCARIA

III.1 SISTEMA EN UN EQUIPO CENTRAL.

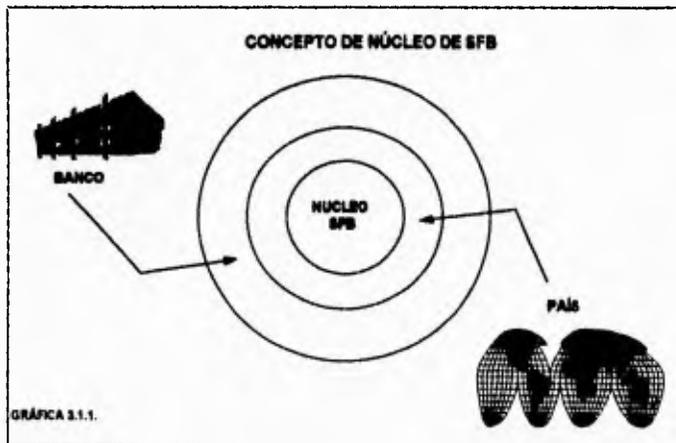
Existe en el mercado una aplicación conocida como Sistema Financiero Bancario (SFB), que es un sistema de información en línea con actualización en tiempo real, que comprende la aplicaciones más críticas de toda institución financiera bancaria.

III.1.1 OBJETIVOS DEL SFB.

El objetivo del producto es proveer a la banca latinoamericana un sistema que, siendo completo, sea la base o núcleo para sus aplicaciones, por lo que debe ser fácilmente adaptable y/o expansible. Estas facilidades de adaptación y expansión son objetivos principales del producto y se manifiestan en formas tales como una minuciosa aplicación de estándares, con orientación hacia el logro de un mejor rendimiento y a facilitar el mantenimiento tanto del software como de la documentación, un diseño modular que permite el crecimiento mediante la adición de nuevos módulos o aplicaciones sin impactar lo ya implantado, un esquema de construcción de cada módulo orientado funcionalmente, que otorga amplias facilidades en las etapas de adaptación y mantenimiento al permitir el análisis del sistema en términos de funciones propias de la institución financiera, en lugar de un análisis de términos técnicos desprovisto de sentido funcional.

III.1.2 CONCEPTO DE NÚCLEO.

El propósito del SFB, como núcleo, es de que éste sea "Nacionalizado", incorporándole aquellas funciones comunes a todas las instituciones del país, como normas de un organismo regulador; y luego, cada usuario pueda "Personalizar" el "Núcleo Nacional" a su medida como lo muestra la gráfica 3.1.1.

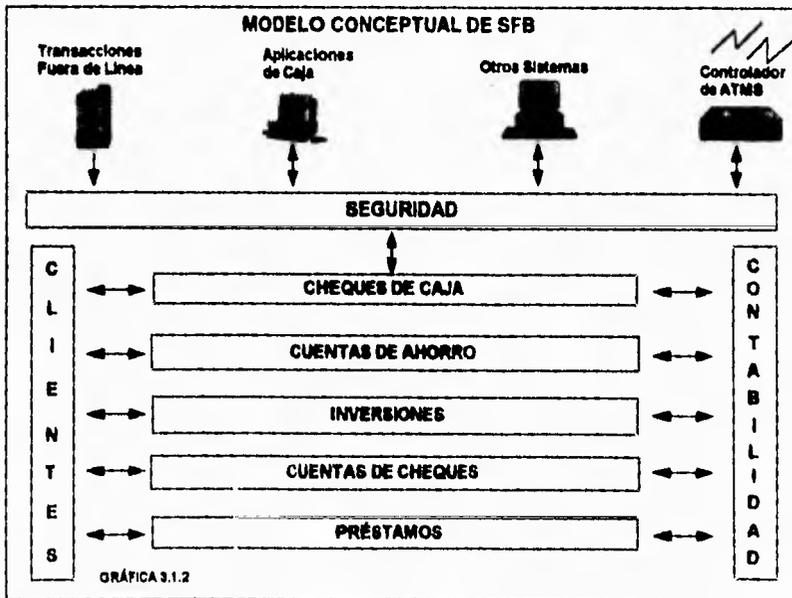


III.1.3 CARACTERÍSTICAS DEL SFB.

Las aplicaciones básicas que definen la arquitectura del SFB son:

- Información de clientes
- Cuentas de cheques
- Cuentas de ahorro
- Inversiones
- Cheques de caja
- Tarjetas y cajeros automáticos
- Control de cajas
- Retenciones y
- Un completo sistema de seguridad.

La integración de toda la información relativa a un cliente y sus cuentas relacionadas permite a la institución conocer, en todo momento, la situación real del cliente frente a ella, a la vez que ofrece a su base usuaria una mayor agilidad en el manejo de sus cuentas. Por su diseño de arquitectura abierta, SFB posibilita integrar información proveniente de otros sistemas en su base de datos y comunicarse con otras aplicaciones y otras bases de información; Es así posible complementar las aplicaciones que forman SFB con otros sistemas (Ver gráfica 3.1.2).



El mercado de SFB comprende bancos comerciales, instituciones de ahorro, de préstamos, sociedades financieras y centros de procesamiento que den servicio a instituciones financieras. SFB entrega una solución efectiva, desde el punto de vista de costos, a los requerimientos de proceso de información de las instituciones financieras, dando soporte a sus objetivos y planes estratégicos, además de que por estar desarrollado para equipos de la serie A de Unisys, cubre las necesidades de instituciones pequeñas, medianas y grandes sin limitaciones de productividad.

III.1.4 ELEMENTOS DE HARDWARE.

SFB ha sido desarrollado para ser ejecutado en equipos de la serie A de Unisys con un mínimo de 12 MB de memoria y 400 MB en disco. Con estaciones de trabajo para actividades administrativas han sido calificados los terminales ET1100 y T27; y las impresoras de las familias APX13xx y EF45xx.

III.1.5 ELEMENTOS DE SOFTWARE.

El SFB ha sido construido utilizando extensamente la herramientas de software más avanzadas, tanto en el ambiente de desarrollo como en el ambiente de ejecución del sistema.

- Ambiente de desarrollo:

Corresponde al ambiente en el cual los sistemas del núcleo son adaptados tanto a las normativas generales del mercado de un país en particular, como a los requerimientos propios de la institución.

- LINC II (*Logic and Information Network Compiler*)

LINC II es un lenguaje de cuarta generación desarrollado por Unisys que genera sistemas a partir de la especificación formal del problema, además es una herramienta de alta productividad que permite adaptar el "Núcleo" de SFB de una manera natural y a un mínimo tiempo.

- Ambiente de ejecución:

Es el ambiente en el cual los sistemas en producción validan y registran las transacciones definidas y actualizan la base de datos.

-DMS II (*Data Management System*). Software de administración de bases de datos de alta confiabilidad y madurez.

-COMS (*Communications Management System*). Software de administración y control de la red de comunicaciones. Esta herramienta permite la comunicación con terminales no "inteligentes" o terminales controlados por otra aplicación.

-MCP (*Master Control Program*). Sistema operativo de máquinas de la serie A de Unisys.

-Controlador de ATM. Manejador de cajeros automáticos independiente de la aplicación.

- Ambiente de explotación:

Es el ambiente en el cual los usuarios finales ejecutan los sistemas en el régimen normal de proceso.

-MARC (*Menu Assisted Resource Control*). Software ambiental que permite efectuar las labores de operación y explotación de sistemas mediante menús.

-WFL (*Work Flow Language*). Permite activar y controlar la ejecución de programas de usuario, utilerías del sistema, funciones del sistema operativo etc.

III.1.6 FUNCIONALIDADES RELEVANTES.

- Procesamiento continuo:

Impide que los procesos masivos periódicos interrumpan la recepción y proceso de transacciones.

- Multimoneda:

La habilidad para manejar cuentas en diversas monedas es necesaria, dadas las exigencias siempre crecientes del mercado, que obliga a las instituciones financieras a manejar nuevos productos y servicios.

- Jornada Adelantada:

Permite el proceso de transacciones con fecha de "Mañana", situación que en la práctica se produce en ambientes en que participan cajeros automáticos y horario de atención al público extendido.

Al permitir el arribo de transacciones de dos días, se elimina la necesidad de cortes bruscos en la atención al público.

- Supervisión de transacciones:

Controla las transacciones efectuadas por los cajeros, detectando incapacidades para pagar o recibir documentos de cierto valor y proporcionando un mecanismo de autorización por parte del supervisor.

- Autorización de transacciones:

Controla y maneja las situaciones de excepción registradas en las cuentas de clientes, que requieren autorización previa a la ejecución de una transacción. Se provee un mecanismo de autorización por parte de los usuarios administrativos capacitados para tal efecto, estableciéndose una comunicación en línea con los cajeros responsables de la transacción.

- Transacciones fecha valor:

Permite el ingreso de movimientos atrasados en cuentas de cheques, cuentas de ahorro, inversiones y préstamos, los cuales son registrados con la fecha actual del sistema, haciendo referencia a la fecha real de su aplicación. Se provee un nuevo cálculo de los saldos disponibles, y en el caso de cuentas de ahorro, se realiza el nuevo cálculo de saldos promedios o saldos mínimos según corresponda.

- Transacciones de aplicación automática:

Es la capacidad de ingresar transacciones en forma adelantada, automatizando su aplicación en forma periódica según parámetros de control, periodo de vigencia y primera fecha de aplicación.

- Verificación de firmas:

SFB provee la capacidad de despliegue de firmas en la aplicación de caja. Para esto hace uso de una base de datos de firmas construida y mantenida con el paquete de software Singbank.

III.2 SISTEMA EN SUCURSALES.

III.2.1 APLICACIÓN DE CAJA.

Existe en el mercado un sistema que permite la ejecución de todas las transacciones financieras de SFB desarrollado con FSA FINESSE (FINANCIAL SYSTEM ARCHITECTURE FINESSE) que es una herramienta de software específica para la automatización de las funciones diarias en las tres áreas primarias de una sucursal financiera:

- Administrativa
- Servicio a clientes y
- Cajeros

FINESSE a sido utilizado por SFB para la definición y programación de la aplicación financiera de caja y plataforma comercial en el ambiente BTOS.

La aplicación permite cursar transacciones de los módulos SFB de:

- Cuentas de cheques
- Cuentas de ahorros
- Inversiones
- Cheques de caja
- Préstamos
- Caja y varios

e incluye las siguientes funcionalidades:

Posibilidad de trabajar fuera de línea.

En caso de que la línea quede fuera de servicio, la aplicación detecta esta condición y continua operando fuera de línea. También se puede trabajar fuera de línea desde el inicio de la sesión y las transacciones, son en ambos casos, almacenadas automáticamente en el archivo de auditoria electrónica para su posterior destanqueo. Además, se dispone de la información necesaria para conservar la mayor parte de la funcionalidad de la aplicación tal y como si estuviera trabajando en línea. Se exceptúan de lo anterior las transacciones que, por diseño, están prohibidas ejecutar fuera de línea, como las consultas de información al HOST, así como también algunas funcionalidades que sólo tienen sentido cuando se está trabajando en línea.

Destanqueo de transacciones efectuadas fuera de línea.

FINESSE provee un esquema de destanqueo automático de transacciones cursadas fuera de línea, efectuado a petición del cajero.

Auditoría electrónica.

La auditoría electrónica es un archivo donde se lleva el registro de todas las transacciones efectuadas con su estado (Aceptadas, rechazadas o pendientes).

Límites de giros y depósitos por cajero.

Localmente se mantienen límites de cajero que pueden ser asociados a tipos de transacciones y producen incapacidad del cajero para pagar o recibir documentos por valores superiores a dichos límites.

Causales válidas.

Para verificar que se ingrese solo causales válidas en aquellas transacciones que así lo requieran, la aplicación consulta un archivo de causales que puede ser actualizado con datos del HOST cuando sea necesario.

Supervisión de transacciones.

La supervisión de transacciones financieras se refiere al manejo y control de las siguientes situaciones de excepción:

- Incapacidad del cajero para pagar o recibir documentos por valores superiores a determinados montos límites.
- Incapacidad del cajero para efectuar ciertas transacciones con causales especiales.
- Incapacidad del cajero para cursar transacciones que requieren supervisión obligatoria.

Cuando se está trabajando fuera de línea, además de las situaciones de excepción anteriores, se supervisan algunas transacciones en situación de saldo insuficiente y orden de no pago. En este caso se consulta, además, el archivo de saldos locales.

Autorización de transacciones.

El esquema de autorización de transacciones financieras se refiere al manejo y control de la situación de excepción que se origina en la incapacidad de la cuenta para cubrir algún documento (Sobregiro) y se aplica sólo cuando se está trabajando en línea. Si se produce la excepción para una transacción que participa en el proceso de autorización, el HOST contesta con un mensaje avisando que ésta entró en dicho proceso, liberando la caja para que pueda seguir cursando otras transacciones mientras recibe otro mensaje indicando si la transacción fue autorizada o rechazada.

Consulta y actualización de saldos locales

La aplicación de caja considera la consulta y actualización de saldos de cuentas pertenecientes a la sucursal desde donde se está cursando la transacción. Cuando se trabaja en línea dichos saldos son actualizados al momento de alguna transacción exitosa. En el caso de que se trabaje fuera de línea, se valida la disponibilidad de fondos suficientes, la no existencia de órdenes de no pago y la no existencia de restricciones en la cuenta; pudiéndose cursar la transacción, previa supervisión de la persona responsable.

Control de secuencia de ejecución de transacciones.

Este esquema controla que las transacciones sean ejecutadas en un cierto orden:

- Ninguna transacción se puede ejecutar antes de la transacción de conexión de usuarios.
- Ninguna transacción se puede ejecutar sin haber iniciado el módulo al que pertenece.
- Ningún módulo se puede iniciar antes que el módulo de caja.
- El módulo de caja no se puede cerrar si existe algún otro abierto.

Despliegue de firmas.

Se cuenta con la capacidad de desplegar firmas que han sido almacenadas empleando el sistema de software SignBank. Para esto, se utiliza un servidor que, dada la cuenta, despliega una a una las firmas asociadas a la cuenta y sus restricciones de pago.

III.3 MEDIOS DE COMUNICACIÓN.

Los medios de transmisión de datos más comúnmente usados en redes de comunicación pueden subdividirse en dos grandes grupos: medios conductores y medios radiados. Entre los medios conductores se incluyen el teléfono, telégrafo alámbrico, cableado privado, cables coaxiales y fibra óptica. Los medios radiados incluyen la radiodifusión, microondas, transmisión infrarroja y satélite.

III.3.1 MEDIOS CONDUCTORES.

Líneas Alámbricas.

Los cables son el medio de transmisión de información más frecuentemente usado, y además son los iniciadores. Mucha de la terminología y tecnología respecto a este medio de comunicación se deriva de la telefonía y de la telegrafía, debido a que la industria de la computación se valía de las redes existentes de teléfono y telégrafo para formar sus propias redes de comunicaciones. Las ventajas de los cables son su fácil disponibilidad y su costo relativamente bajo. Sus desventajas incluyen su susceptibilidad a la distorsión de la señal o error y las velocidades relativamente bajas de transmisión que proveen a largas distancias.

Cable Coaxial.

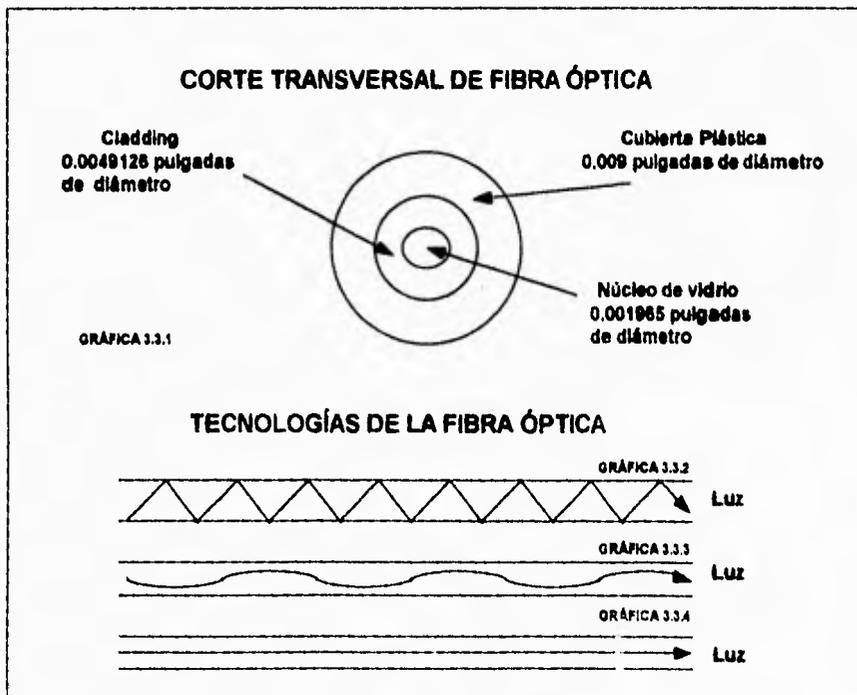
El cable coaxial se utiliza primordialmente en redes de área local (LAN) o entre distancias relativamente cortas, generalmente menos de 16 Km. La mayoría de las redes LAN son privadas y generalmente están restringidas a áreas geográficas pequeñas como un edificio o un grupo de edificios. El cable coaxial también se emplea para conectar terminales con unidades controladoras. Las velocidades de transmisión mayores de 100 Mbps no son inusuales y la velocidad teórica de transmisión del cable coaxial es de más de 400 Mbps. El cable coaxial consiste esencialmente de uno o dos alambres centrales de transmisión rodeados de una capa aislante, una capa metálica protectora y finalmente una capa aislante externa.

Fibra óptica.

La fibra óptica es un medio de comunicación relativamente nuevo usada por compañías telefónicas en cables de larga distancia. También es utilizada por compañías privadas en la implementación de redes locales de comunicación de datos.

A pesar de que existen tres variedades de fibra óptica y de que cada una tiene su propia manera de conducir el impulso luminoso desde el origen hasta el destino, todas ellas tienen la misma forma y mismas características. Una o más fibras de plástico o vidrio se entrelazan para formar el corazón del cable y éste está rodeado por una capa de plástico o vidrio llamada "cladding". El cladding está además cubierto por otra capa de plástico o algún otro material para protección. La gráfica 3.3.1 muestra una sección transversal de un cable de fibra óptica. Las tres variedades de fibras ópticas requieren de una fuente de luz, con láser o diodos emisores de luz (LED), siendo éstos usados más frecuentemente.

La más antigua de las tres tecnologías de fibra óptica usa una fibra multimodo de índice escalonado, en la que sus paredes reflejantes mueven los pulsos luminosos hacia el receptor. La gráfica 3.3.2 ilustra la transmisión multimodo de índice escalonado. Las fibras multimodo de índice graduado actúan refractando el rayo de luz hacia el centro del cable gracias a variaciones de densidad del mismo. La gráfica 3.3.3 ilustra el movimiento de la luz en una fibra multimodo de índice graduado. La última y más rápida de las técnicas de fibras ópticas es la de modo de transmisión único, en la que la luz es guiada por el centro de un corazón extremadamente angosto. Esta técnica se muestra en la gráfica 3.3.4. Las velocidades de transmisión actualmente disponibles van más allá de aproximadamente 1 Gbps.



Uno de los contratiempos de las fibras ópticas es la indisponibilidad de agregar nuevos nodos cuando otros están activos. A pesar de que ahora es relativamente fácil empalmar el cable de fibra óptica para agregar nuevas estaciones, la red o parte de ella debe inhabilitarse para poder preparar el empalme. La fibra óptica es mucho más costosa que el cable de cobre, ya sea coaxial o par trenzado, pero tiene la ventaja de que es mucho más ligera, más pequeña y tiene tasas de error bajas e inmunidad a interferencia ambiental.

III.3.2 MEDIOS RADIADOS.

Microondas.

La velocidad de transmisión de las microondas alcanza hasta 45 Mbps. Debido a que las señales de las microondas viajan en línea recta, el transmisor y el receptor deben estar sobre la misma línea de transmisión (No debe haber obstáculos entre ellos). Debido a la curvatura de la tierra se requiere entonces que no halla entre las estaciones de microondas más de 48 Km de distancia. El ancho de banda total del canal de microondas puede subdividirse en muchos subcanales que se usan para transmisión de voz, transmisión rápida de datos, o ambas.

La transmisión via microondas ofrece rapidez, eficiencia en costos, y facilidad de implementación, sin embargo, tiene la desventaja de poder ser interferida por otras señales de radio. Está también limitada por cuestiones de evitar obstáculos físicos entre estaciones. Además, las transmisiones comerciales son inseguras, pues pueden ser interceptadas por cualquier persona que tenga un receptor sobre la línea de transmisión. Las microondas también pueden ser afectadas por condiciones meteorológicas adversas.

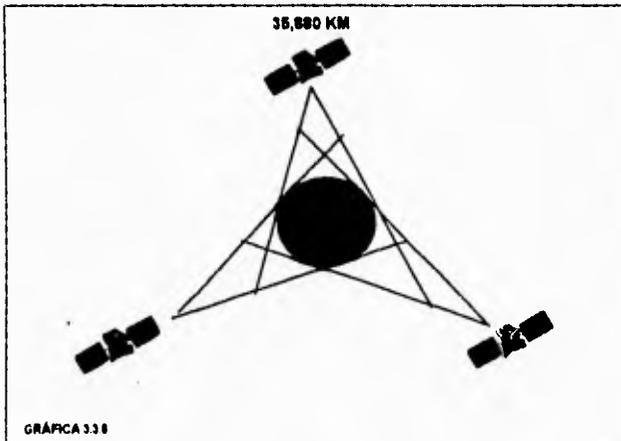
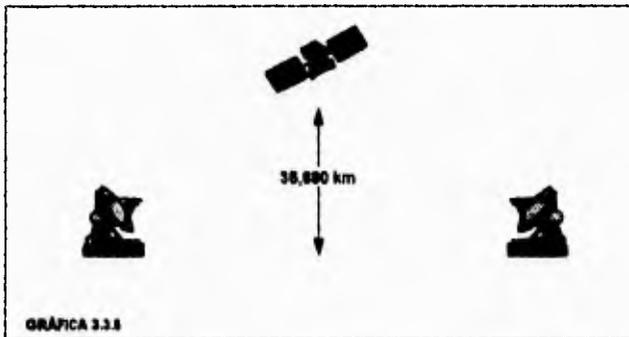
Radio.

La transmisión por radio no sólo emplea las frecuencias típicas de AM y FM, también usa frecuencias de onda corta con un rango total de 500,000 a 108 millones de ciclos por segundo. Las principales aplicaciones de las ondas de radio son para teléfonos celulares, localizadores y redes inalámbricas de área local.

Cuando el radio se utiliza en redes de área local, los cables que conectan cada micro computadora se eliminan y esto facilita la instalación y el movimiento de las estaciones y reduce el problema de conexiones sueltas que resultan de accidentes provocados por personas al caminar o al mover cables.

Satélite.

Igual que la radio transmisión por microondas, la transmisión de datos via satélite emplea ondas de radio VHF (Very High Frecuency) y ambas requieren que las estaciones estén en la línea de transmisión. La diferencia principal entre ellas es la localización de las estaciones. Las microondas sólo emplean estaciones en la tierra, mientras que la de satélite utiliza estaciones en la tierra y estaciones en órbita. Los satélites de comunicaciones comerciales se colocan en una órbita ecuatorial geosincrónica a una altitud de 35,880 Km. Una órbita geosincrónica significa que el satélite permanece en una posición estática relativa a un punto dado en la tierra como se muestra en la gráfica 3.3.5. A esta altitud sólo se requieren tres satélites para tener cualquier punto de la tierra dentro del alcance, como lo muestra la gráfica 3.3.6.



III.4 SISTEMA EN SU EMPRESA.

III.4.1 Alcance del Proyecto.

El propósito de este trabajo de tesis es diseñar y programar un sistema, para computadoras personales, que sea ofrecido por alguna institución bancaria que cuente con la aplicación SFB.

Por medio de este sistema, los clientes del banco podrán efectuar transacciones desde sus oficinas contra SFB, empleando una PC.

El programa permitirá al cliente efectuar operaciones sobre sus cuentas y obtener información desde la base de datos del banco, utilizando su propia computadora personal conectada al equipo central del banco mediante la red telefónica, en un ambiente seguro y controlado.

El sistema podrá ser utilizado tanto por instituciones como por personas físicas e incluirá los siguientes servicios :

- Consultas:
 - Saldo de una cuenta.
 - Movimientos de la cuenta.
 - Estado de un cheque.
 - Estado de una chequera.
 - Cuentas de un cliente.
 - Clientes relacionados con una cuenta.
- Transferencias de fondos entre cuentas del cliente y
- Pago de servicios definidos por el banco.

existiendo la posibilidad de desarrollar nuevos servicios en función de la aparición de nuevos requerimientos.

Al sistema central se podrán conectar un número finito de sistemas BANCEMP simultáneamente, dependiendo de las limitaciones tecnológicas del equipo de comunicaciones conectado al equipo central del banco. Debido a restricciones económicas, en este proyecto de tesis conectaremos sólo un sistema BANCEMP.

Se prevé un procedimiento de distribución de nuevas versiones actualizadas del sistema, mediante avisos al cliente en línea y la entrega personal del software actualizado, previo cumplimiento de requisitos de control y seguridad establecidos por el banco.

CAPÍTULO IV
FASE DE INVESTIGACIÓN

IV.1 OBJETIVOS, METAS Y RESTRICCIONES.

IV.1.1 OBJETIVOS DEL SISTEMA BANCEMP.

El objetivo principal del sistema es el ofrecer una herramienta a clientes de una institución financiera bancaria, para que éstos puedan realizar operaciones sobre sus cuentas sin tener que trasladarse fuera de sus instalaciones o de sus hogares. Para ello se requiere que los clientes cuenten con una computadora personal que podrá conectarse al sistema central del banco (SFB) ejecutando nuestra aplicación: Sistema Banca en su Empresa (BANCEMP).

Se pretende que el sistema que resulte de este trabajo de tesis sea funcional y flexible, pensando en futuros requerimientos del banco o cambios operativos en la institución. Además de que resulte práctico para el cliente del banco y fácil de operar; para esto se emplearán las ventajas gráficas que ofrece el entorno operativo WINDOWS.

Recordando que las operaciones que se realizarán desde BANCEMP involucran intereses monetarios empresariales y/o personales se implementarán estrictos controles de seguridad dentro de nuestro sistema, además de los existentes en SFB.

IV.1.2 METAS.

La principal meta de este sistema es lograr que la institución bancaria obtenga ganancias atractivas resultado de la venta y de las comisiones por uso del sistema, al mismo tiempo que logre la completa satisfacción de sus clientes.

IV.1.3 RESTRICCIONES.

Recordando que nuestro sistema BANCEMP estará conectado al sistema central SFB a través de la línea telefónica, tendremos que considerar el inevitable gasto por HARDWARE y SOFTWARE requerido para las comunicaciones. Por lo que seleccionaremos productos de costo razonable pero sin dejar de lado la requerida confianza del sistema. Otra restricción del sistema es que éste dependerá al 100% del las líneas telefónicas para poder comunicarse con el sistema central.

IV.2 FUNCIONAMIENTO DEL NEGOCIO.

La operación de una institución financiera bancaria comprende muchos rubros pero su finalidad principal es la captación de fondos y la colocación de los mismos en un mercado productivo con el propósito de obtener ganancias. Una parte importante de esta captación de fondos proviene de los depósitos que los clientes del banco hacen en sus cuentas, por lo que se requiere que el banco ofrezca a sus cuentahabientes ganancias por los depósitos y servicios de control y administración de sus cuentas para mantener una captación constante. Dentro de los servicios más comunes que una institución financiera ofrece a sus clientes con respecto a sus cuentas tenemos:

- Consulta de saldos
- Consulta de movimientos de la cuenta.
- Transferencia de fondos entre cuentas.
- Consulta de estado de cheques.
- Consulta de estado de chequeras.
- Pago de servicios.
- Consulta de cuentas del cliente y
- Consulta de clientes relacionados con una cuenta.

Todos estos servicios están disponibles para los clientes en cualquier sucursal del banco, pero se pretende que también estén disponibles desde sus oficinas u hogares.

IV.3 REQUERIMIENTOS DEL USUARIO.

La competencia actual entre instituciones financieras bancarias ya establecidas y la futura aparición de otras tantas, obliga a lo bancos a crear nuevos servicios y a ofrecer nuevas ventajas a sus clientes, con el fin de mantenerse a la vanguardia y siempre tener un buen nivel competitivo en el mercado. Es por eso que el sistema BANCEMP resulta una oferta muy atractiva para los cuentahabientes del banco, si consideramos las ventajas que les ofrece y los problemas que les evita. Así, los ejecutivos del banco requieren de un producto que sea capaz de conectarse a su sistema central en un ambiente seguro y controlado. Dicho producto deberá cumplir con las siguientes características:

- Para ejecutarse en cualquier PC compatible.
- Ser flexible y de fácil mantenimiento.

- Deberá correr dentro del ambiente gráfico de WINDOWS, con el propósito de facilitar su utilización al cliente final y de presentarle un producto con características visuales amigables.
- Ofrecerá al cliente una clave secreta a nivel del PC y la posibilidad de modificarla en cualquier momento.
- Habrá dos niveles de usuarios: Un nivel alto que tendrá acceso a todo el sistema y un nivel bajo que tendrá acceso solo a consultas.

IV.4 SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO.

De acuerdo a los requerimientos obtenidos en la fase anterior, a continuación se enumeran los recursos requeridos para crear el sistema BANCEMP y se explica el porqué de la elección.

IV.4.1 REQUERIMIENTOS DE HARDWARE.

El sistema BANCEMP requiere una computadora personal PC con las siguientes características:

- Procesador 80386 ó superior.
- 4 Mb en memoria RAM como mínimo para WINDOWS.
- Unidad de disco flexible.
- Puerto serial RS232.
- Disco duro con 20 Mb libres para WINDOWS, BANCEMP y software de comunicaciones.
- Tarjeta de video VGA o superior
- Puerto para MOUSE y el MOUSE.

Todas las características antes mencionadas deben cumplirse estrictamente, con el fin de satisfacer los requerimientos del usuario y poder emplear el SOFTWARE requerido para construir una aplicación de buena calidad.

Se requiere además de un MODEM asíncrono Hayes, MultiTech, TrailBlazer o compatible para poder conectarnos con el sistema central del banco (SFB) a través de la línea telefónica.

IV.4.2 REQUERIMIENTOS DE SOFTWARE.

Ambiente de ejecución.

Para poder cubrir las características esperadas del banco se requiere que la computadora personal cuente con el siguiente ambiente de ejecución:

- Sistema operativo MSDOS versión 3.1 ó superior para poder usar:
- WINDOWS versión 3.1 ó superior, que ofrece un ambiente gráfico de trabajo.
- Software de comunicaciones Infoconnect 2.0 ó superior.

Ambiente de desarrollo.

Visual Basic 3.0, además de todo lo indicado en el ambiente de ejecución.

IV.5 JUSTIFICACIONES DE LAS ELECCIONES DE SOFTWARE y HARDWARE.

Se ha elegido VISUAL BASIC por ser un lenguaje simple ya difundido y por las ventajas que ofrece para crear aplicaciones WINDOWS bajo la filosofía de programación conducida por eventos y orientada a objetos.

Infoconnect es un software de comunicaciones que permite lograr independencia del HOST con el que se pretende establecer un enlace (Serie A , Unix) y del protocolo de comunicaciones (Poll Select , TCP/IP).

La elección de emplear WINDOWS se debe a los requerimientos del usuario y a la necesidad de proveer de una interfaz gráfica al cliente final. Debe contarse con la versión 3.1 para poder soportar el uso de Visual Basic 3.0.

Las características de la PC fueron pensadas de acuerdo a lo mínimo necesario para poder manejar el SOFTWARE antes mencionado.

El MODEM debe ser compatible con WINDOWS, de ahí las opciones antes propuestas.

IV.6 VISUAL BASIC.

Visual Basic es un sistema de desarrollo diseñado especialmente para crear aplicaciones gráficas de una forma rápida y sencilla. Incluye una herramienta para diseño de gráficos y un lenguaje de alto nivel: BASIC, que es un lenguaje muy poderoso y muy fácil de aprender y utilizar.

Para crear una aplicación en Visual Basic se crean *ventanas* y sobre ellas se dibujan *controles* (Botones, Cajas, etc.) y a continuación se escribe el código fuente relacionado a cada objeto. Esto es, cada objeto está ligado a un código que permanece inactivo hasta que se dé el suceso que lo activa (Ej. click del MOUSE). Visual Basic permite crear *ventanas* y *controles* sin escribir código. Se dice entonces que Visual Basic es una herramienta con filosofía de *programación conducida por eventos y orientada a objetos*.

Visual Basic también incluye:

- Soporte para intercambio de datos con otras aplicaciones (DDE - Dynamic Data Exchange)
- Soporte para establecer enlaces con WINDOWS y con rutinas escritas en otros lenguajes (DLL - Dynamic Link Libraries)
- Un sistema de ayuda en línea, que ofrece solución a los problemas del programador mientras crea una aplicación.
- Un archivo de iconos para trabajos profesionales.

Cuando se combinan estas características, se dispone de un sistema de desarrollo que permite diseñar aplicaciones sofisticadas rápidamente y además, con una muy buena presentación.

Para crear una aplicación se debe:

Diseñar el medio de comunicación (Interfaz) entre el usuario y la aplicación. Para ello,

- 1- Se crea una *forma* (Ventana).
- 2- Se añade los controles a la *forma*.

El siguiente paso consiste en especificar las propiedades para las *formas* y los *controles*. Por ejemplo, títulos deseados para los objetos, valores iniciales, colores, etc.

Finalmente se une código a la *forma* y a los *controles* para hacer que la aplicación ejecute las tareas deseadas.

Una *forma* es una ventana que sirve de fondo para los *controles* y para los gráficos situados sobre la misma. Se pueden utilizar tantas *formas* como sea necesario y dependiendo de la utilidad que se les dé, éstas serán de diferentes tipos. Así, se puede crear una *forma* para que contenga un gráfico, para visualizar información o para aceptar datos.

Los *controles* son objetos que dibujamos sobre una forma con fines de aceptar o visualizar datos. Para dibujar controles sobre una forma, Visual Basic pone a nuestra disposición un panel de utilidades (Toolbox). Con él, se pueden dibujar controles como botones, cajas de texto, etiquetas, etc.

Cada objeto lleva asociado un conjunto de propiedades, las cuales pueden ser modificadas en tiempo de diseño y/o en tiempo de ejecución.

Para hacer que una aplicación responda a ciertos sucesos, tenemos que unir código a las *formas* y a los *controles*. El código que se une se denomina *procedimiento conducido por un suceso* o simplemente *procedimiento*.

Cuando en una aplicación se utilizan muchas órdenes, Visual Basic permite agruparlas en un menú. Un menú se diseña utilizando la ventana de diseños de menús, evitándonos así, la tarea de escribir el código necesario para tal efecto.

Además, Visual Basic tiene instalada una ayuda en línea que da información acerca de propiedades, sucesos, mensajes de error, sentencias, teclado, depurador, intercambio dinámico de datos y entorno de Visual Basic.

IV.7 INFOCONNECT.

Infoconnect es un software mediante el cual se puede crear una plataforma abierta de comunicaciones, orientada a usuarios, desde el centro de cómputo (SITE); permitiendo comunicar PCs, con ambiente WINDOWS en ejecución, con una amplia variedad de computadoras centrales (Mainframes).

CAPÍTULO V
DEFINICIÓN Y DISEÑO

V.1 ESTÁNDARES DE DISEÑO.

Con la finalidad de mantener determinados niveles de calidad y adaptabilidad del sistema resultante de este trabajo de tesis, hemos fijado los siguientes estándares de desarrollo:

V.1.1 DICCIONARIO DE DATOS.

Se creará un diccionario de datos global para evitar la redundancia e inconsistencias en la definición de datos de uso general en el proyecto, y se obligará a la utilización del mismo para facilitar futuros mantenimientos al sistema. Restringiendo, además, la declaración de variables de uso local al mínimo y solo en casos de absoluta necesidad.

Dicho diccionario será un *módulo* general de Visual Basic y lo llamaremos DICC.BAS.

V.1.2 CONVENCIONES DE NOMENCLATURA.

Formas: Todos los nombres de las *formas*, que representen alguna transacción, estarán limitados a 5 caracteres alfanuméricos con la siguiente estructura:

RRTNN

donde : **RR** es el módulo o aplicación a la que nos conectaremos con SFB.
(Ver apéndice A).

T es el tipo de transacción. (Ver apéndice B) y

NN es un mnemotécnico que identifica a la transacción o un número consecutivo.

Los menús de nuestra aplicación serán de la forma: **MENÚ <identificador>**.
Donde el identificador es una palabra que denota la función.

Controles: Los nombres de los controles tendrán la siguiente estructura:

rrr<Ident1><Ident2>

donde : **rrr** es el tipo de control de que Visual Basic provee (Ver apéndice C)
y los identificadores son mnemónicos de 3 caracteres cada uno.

Datos: Los nombres de datos se formarán con un prefijo, un nombre y un sufijo, según la siguiente lista:

<u>Categoría de dato</u>	<u>Nombre</u>
Dato global TYPE.	GG_<Nombre>[<sufijo>]
Dato global simple.	GD_<Nombre>[<sufijo>]
Dato global contenido constante.	GC_<Nombre>[<sufijo>]
Dato local TYPE.	SG_<Nombre>[<sufijo>]
Dato local simple.	SD_<Nombre>[<sufijo>]
Dato local contenido constante.	SC_<Nombre>[<sufijo>]

Los sufijos son opcionales y se utilizarán para diferenciar instancias del mismo dato. En caso de que el contenido de un dato sea constante se usará el propio dato como nombre.

Funciones y procedimientos: Los nombres de las funciones y procedimientos de uso global se denominarán de la siguiente manera:

GF_<Nombre> para funciones y
GP_<Nombre> para procedimientos,

siendo <Nombre>, en lo posible, de la forma <Prefijo>_<Sufijo> donde

<Prefijo>: indica la acción realizada por la función o el procedimiento y
<Sufijo> : es un mnemotécnico que completa la descripción.

V.1.3 DISEÑO DE PANTALLAS.

Algunas de las consideraciones importantes en el diseño de pantallas son el tamaño de los controles, los colores, los nombres de las etiquetas, etc.; es por eso que a continuación se listan las características del diseño de pantallas :

- Apariencia ordenada y limpia.

- Indicaciones obvias de qué se muestra y de lo que debe ser hecho.
- Ubicación apropiada de la información.
- Lenguaje simple y claro.
- Uso de colores y tonalidades a discreción.
- Los controles de entrada de datos deberán ser blancos y los de salida de datos deberán ser grises.
- Seguirán características de formato y colores de otras aplicaciones WINDOWS en el mercado.

Organización de datos en pantalla.

- La información deberá presentarse en forma directamente utilizable, sin hacer referencia a documentación ni a traducción de códigos.
- Para títulos o etiquetas se utilizarán minúsculas con la primera letra de la oración en mayúscula.
- No se utilizarán abreviaturas en lo posible.

Mensajes.

Los mensajes que sirvan de diálogo entre la aplicación y el cliente final deberán utilizar palabras:

- Cortas, significativas y comunes.
- En lenguaje familiar.
- Interpretables de una sola manera.
- No abreviadas.

Además deberán utilizarse oraciones:

- Afirmativas.
- En voz activa.
- No autoritarias.
- No ofensivas.

Dichos mensajes estarán declarados en un módulo global de Visual Basic con el nombre de **MSG.BAS** y tendrán la siguiente estructura:

MSG_<prefijo><consecutivo> = " mensaje " donde:

<prefijo> será **E** si se trata de un error y

A si se trata de un aviso o una advertencia.

<consecutivo> será un número de identificación del mensaje.

Codificación.

- Utilizar sangrias y alinear los comandos END con sus respectivos asociados.
- Comentar todos los bloques de lógica y líneas de importancia relevante.
- Sólo utilizar saltos de lógica en casos estrictamente necesarios.
- Cuando una misma sección de código tenga que utilizarse dos o más veces, ésta debe ser declarada como una función o un procedimiento general para evitar redundancias y facilitar el mantenimiento futuro del sistema.

Funciones externas.

Si se utilizan funciones externas a nuestra aplicación (Ej. API de WINDOWS), éstas deben ser declaradas en un módulo general Visual Basic para nuestra aplicación denominado **GRAL.BAS**.

V.1.4 EJEMPLOS DE APLICACIÓN DE ESTÁNDARES.

- **ACCSA** es el nombre de una forma donde :

AC corresponde al módulo cheques del sistema SFB (Ver apéndice A),
C indica que se trata de una consulta (Ver apéndice B) y
SA es un mnemónico para "saldo".

Así, este nombre se daría a una transacción de consulta del saldo de una cuenta de cheques.

- **txtCodCli** es el nombre de un control donde :

txt es el tipo de control que Visual Basic provee (Ver apéndice C),
Cod es un mnemónico que indica "código" y
Cli es otro mnemónico que indica "cliente".

Así, este nombre se daría a un control que fuera una caja de texto y que sirviera como entrada o salida del código de identificación de un cliente.

- **GC_FALSO** es el nombre de un dato donde :

GC indica que se trata de una variable global constante (Ver sección V.1.2),
FALSO es un mnemotécnico que indica el propio valor del dato.

Así, este nombre se daría a un dato global constante cuyo valor podría ser : -1, indicando una falsedad.

- **GP_Valida_Número** es el nombre de una procedimiento donde :

GP indica que se trata de un procedimiento global (Ver sección V.1.2),
Valida indica la acción realizada y
Número completa la descripción.

Así, este nombre se daría a una función global validara si la cadena de caracteres ingresada por el usuario se trata de un número.

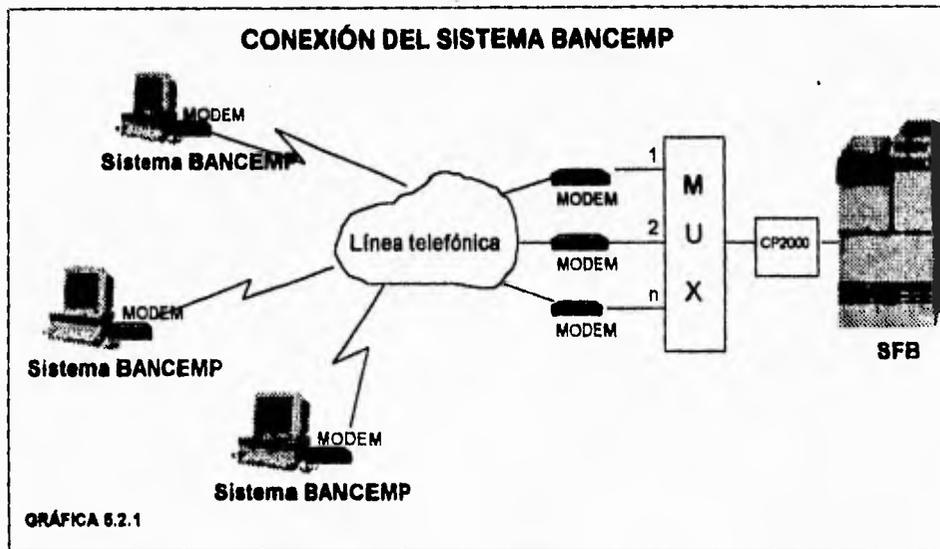
- **MSG_E001** es el nombre de un mensaje donde :

MSG indica que se trata de un mensaje (Ver sección V.1.3),
E indica que se trata de un error y
001 es un número consecutivo identificador del mensaje.

Nótese la utilización de letras mayúsculas y minúsculas en los ejemplos presentados, comparando con la definición de estándares establecidos en este capítulo.

V.2 MODELO.

El sistema BANCEMP permitirá a un cliente de una institución financiera bancaria efectuar operaciones sobre sus cuentas, utilizando una computadora personal conectado al sistema central (SFB) del banco, a través de la línea telefónica y auxiliado de un MODEM como lo muestra la gráfica 5.2.1.



Para permitir la conexión simultánea de varias computadoras personales se empleará un multiplexor y un equipo especializado de comunicaciones conectados al equipo central del banco, logrando de esta manera tener una línea del equipo dedicada a n computadoras personales, donde n depende de las características del multiplexor que se utilice. Así, el número de PC's, que pueden conectarse, podrá ser incrementado considerablemente, asignando más multiplexores a diferentes líneas del equipo central.

Esta arquitectura es sólo una sugerencia que por razones económicas queda fuera del alcance de este proyecto de tesis. Personal especializado en comunicaciones se encargará de definir la arquitectura más apropiada para algún banco que decida adquirir el sistema BANCEMP.

El sistema BANCEMP está dividido en 4 módulos :

- Módulo de seguridad
- Módulo de comunicaciones
- Módulo de administración y
- Módulo de servicios.

El módulo de seguridad tiene como objetivo el restringir el acceso y uso del sistema sólo a personal autorizado, empleando para ello claves de usuario y códigos de acceso, a nivel PC y, además, a nivel del sistema central.

El módulo de administración tiene como objetivo el permitir al usuario administrar el acceso al módulo de comunicaciones y facilitar su uso. Este módulo está compuesto por los siguientes menús:

- Cliente
- Usuarios
- Cuentas y
- Servicios.

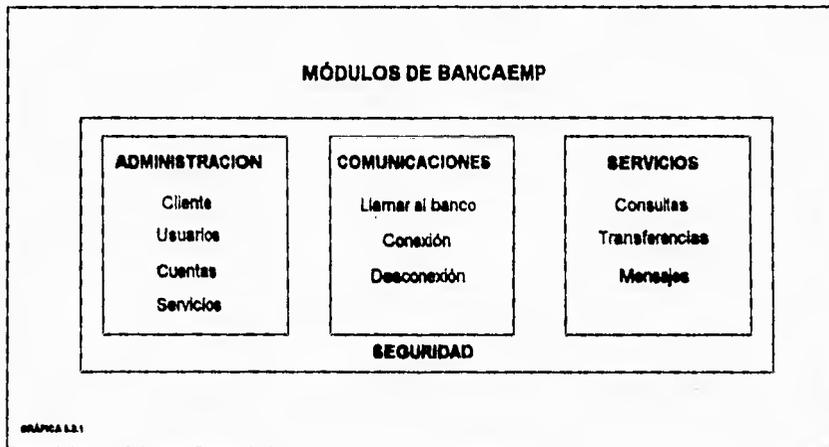
El módulo de comunicaciones permite al usuario conectarse al banco y poder efectuar operaciones sobre sus cuentas. Este módulo está compuesto por los siguientes menús:

- Llamar al banco
- Conexión al sistema central y
- Des conexión

El módulo de servicios permite efectuar operaciones sobre las cuentas del cliente empleando, para ello, los siguientes menús:

- Consultas
- Transferencias y
- Mensajes.

A continuación se presenta un modelo general de los módulos de administración y comunicaciones del sistema BANCEMP.



A continuación se mencionan las funcionalidades de BANCEMP a través de los menús antes mencionados:

Módulo de administración.

- **Menú cliente:** Contiene una transacción que permite modificar el código y nombre del cliente.

- **Menú usuarios:** Contiene 2 transacciones que permiten listar y definir hasta 10 usuarios que tendrán derecho a utilizar el sistema BANCEMP.

- **Menú de cuentas:** Contiene 2 transacciones que permiten listar y definir hasta 10 cuentas con una breve descripción para facilitar su reconocimiento, las más usadas preferentemente. La definición de estas cuentas no limita el uso del sistema, esto es, el usuario puede realizar operaciones sobre otras de sus cuentas, autorizadas por el banco y no definidas en BANCEMP.

- **Menú de servicios:** Contiene 2 transacciones que permiten listar y definir hasta 10 códigos de servicios que se pagan a través del sistema BANCEMP. Pudiendo aún pagar servicios cuyos códigos no estén definidos en el sistema BANCEMP, empleando una relación de códigos que será proporcionada por el banco.

Módulo de comunicaciones.

- **Llamar al banco:** Permite marcar el número telefónico asignado por el banco.
- **Conexión:** Permite conectarse al sistema SFB .
- **Desconexión:** Permite desconectarse del sistema SFB.

Módulo de servicios.

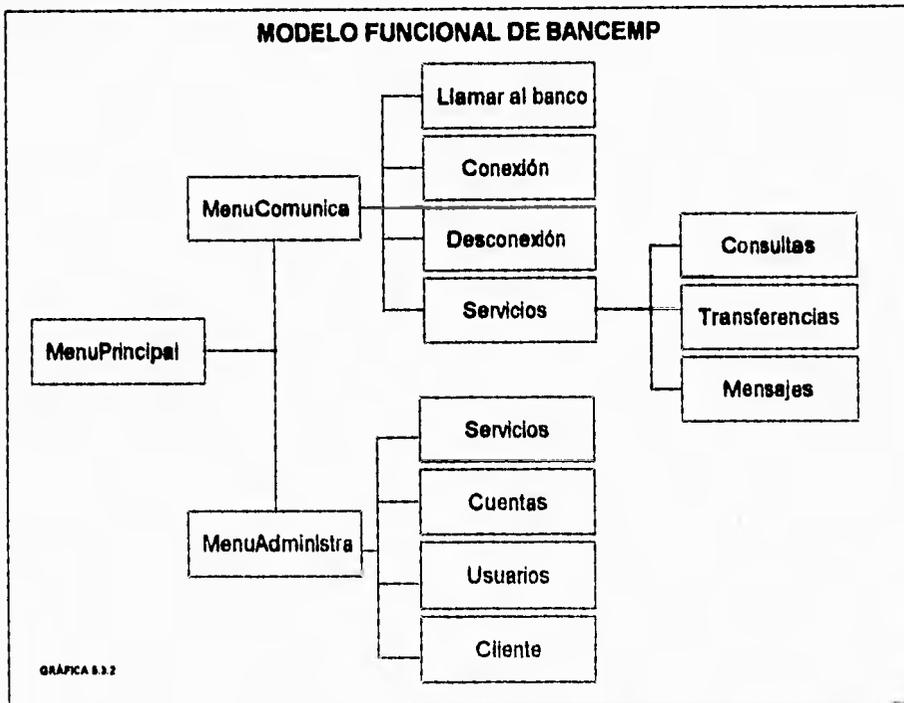
- **Menú de consultas:** Contiene transacciones que permiten al cliente realizar las siguiente consultas:

- Saldo de una cuenta
- Movimientos de la cuenta
- Estado de un cheque
- Estado de una chequera
- Cuentas de un cliente y
- Clientes relacionados con una cuenta.

- **Menú transferencias:** Contiene 2 transacciones mediante las cuales el usuario puede hacer transferencias de fondos entre dos de sus cuentas y efectuar el pago de algún servicio.

- **Menú mensajes:** Contiene 2 transacciones que permiten al usuario enviar y recibir mensajes del banco.

A continuación se presenta un diagrama funcional del sistema BANCEMP.

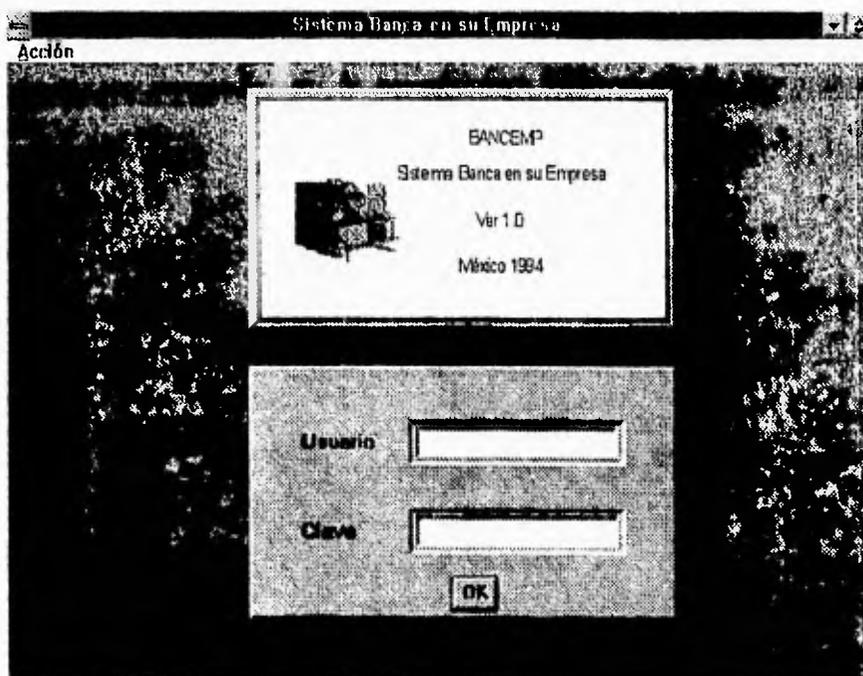


V.3 DISEÑO.

V.3.1 INTERFAZ CON EL USUARIO.

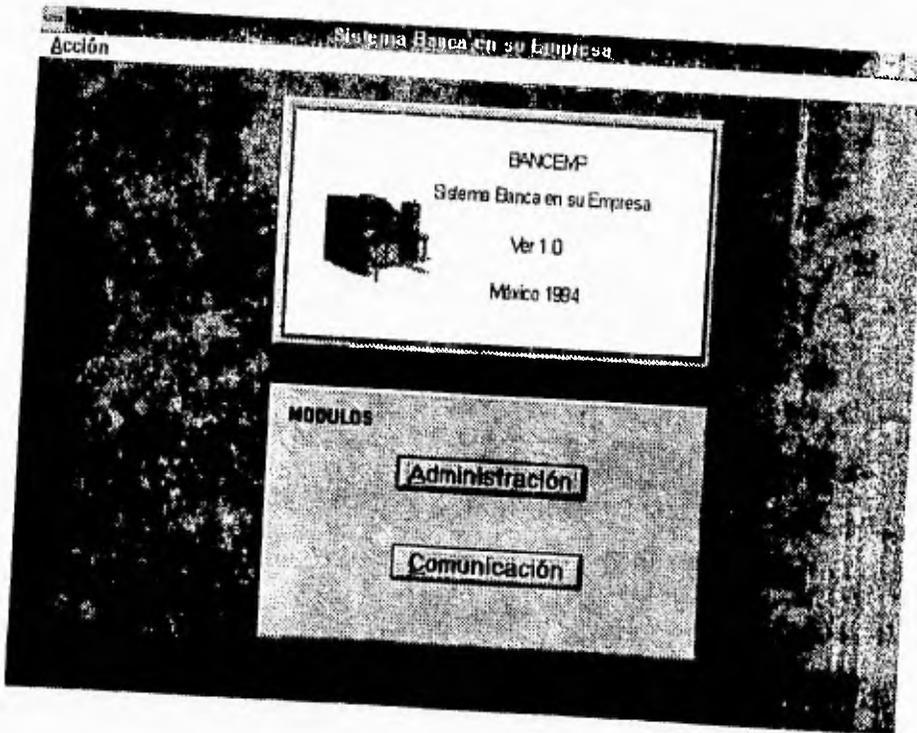
El sistema BANCEMP utiliza una interfaz muy popularizada basada en ventanas y menús, que pueden manipularse desde el teclado de la computadora personal o a través del MOUSE. Esto hace que los usuarios familiarizados con el software WINDOWS encuentren el sistema muy fácil de usar.

Menú Principal.

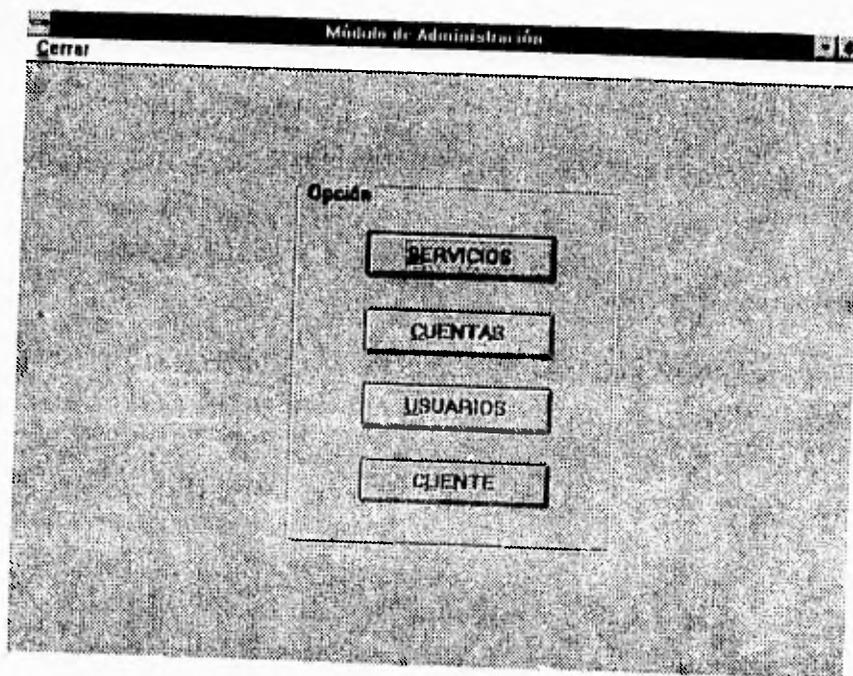


Objetos que componen al menú principal:

1. Una barra de menú con la opción Acción, la cual permite finalizar la ejecución de la aplicación.
2. El logotipo de BANCEMP que presenta datos informativos.
3. Un panel de seguridad pidiendo el nombre del usuario del sistema y su clave secreta. Este panel desaparecerá una vez que el usuario haya ingresado información autorizada de acceso; en caso contrario recibirá un mensaje de error.
4. Un panel de menú que aparecerá una vez que el usuario haya ingresado información autorizada en el panel de seguridad. Dicho panel permite el acceso a los módulos de administración y comunicaciones, como muestra la siguiente gráfica.



Menú de Administración.



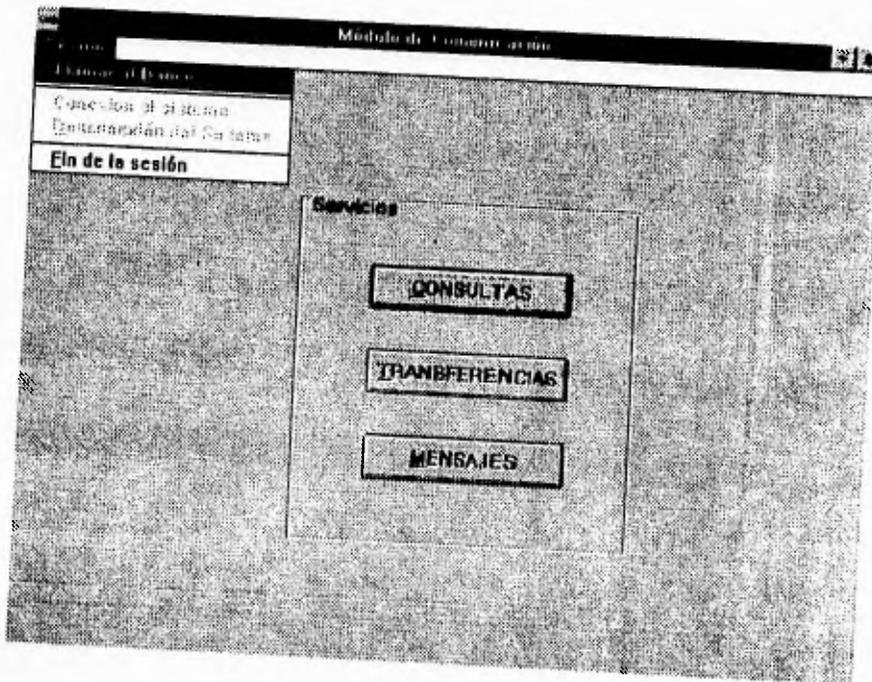
Objetos que forman al menú de administración:

1. Una barra de menú con la opción "Cerrar" que permite salir de la ventana actual y regresar al menú principal.

2. Un panel de menú que solo aparece una vez establecida la comunicación con el sistema central. Este panel permite el acceso a los servicios de:

- Servicios
- Cuentas
- Usuarios y
- Clientes

Menú de Comunicaciones.



Objetos que forman al menú de comunicaciones:

1. Una barra de menú con la opción de sesión que brinda al usuario las siguientes funciones:

- Llamar al banco
- Conectarse al sistema central.
- Desconectarse del sistema central.
- Terminar la sesión de comunicaciones y volver al menú principal.

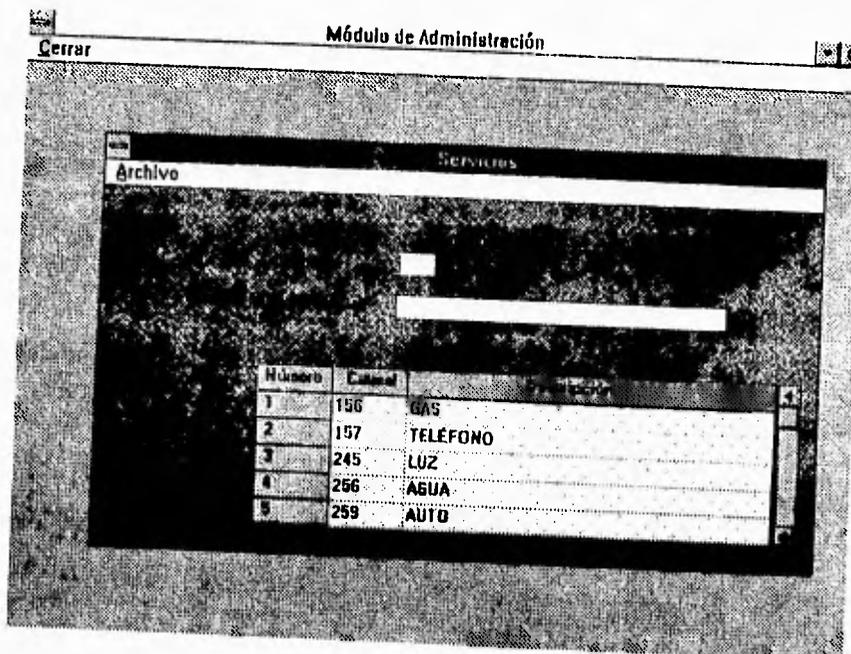
2. Un panel de menú que solo aparece una vez establecida la comunicación con el sistema central. Este panel permite el acceso a los servicios de:

- Consultas
- Transferencias y
- Mensajes.

V.3.2 MÓDULO DE ADMINISTRACIÓN.

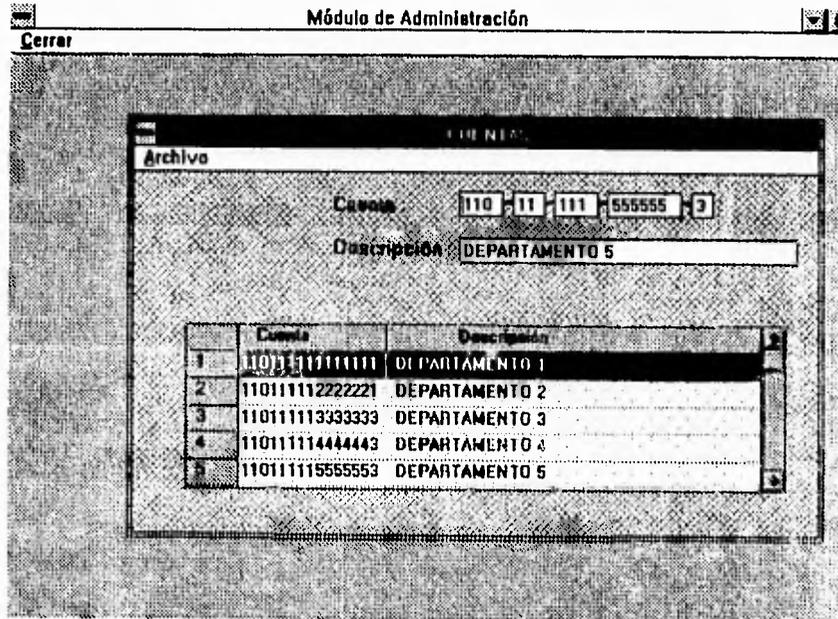
El uso de este módulo permite al usuario administrar el acceso al módulo de comunicaciones y facilitar sus uso. Está compuesto por cuatro ventanas: Servicios, Cuentas, Usuarios y Cliente.

Ventana de Servicios.



Objetos que forman a la ventana de servicios:

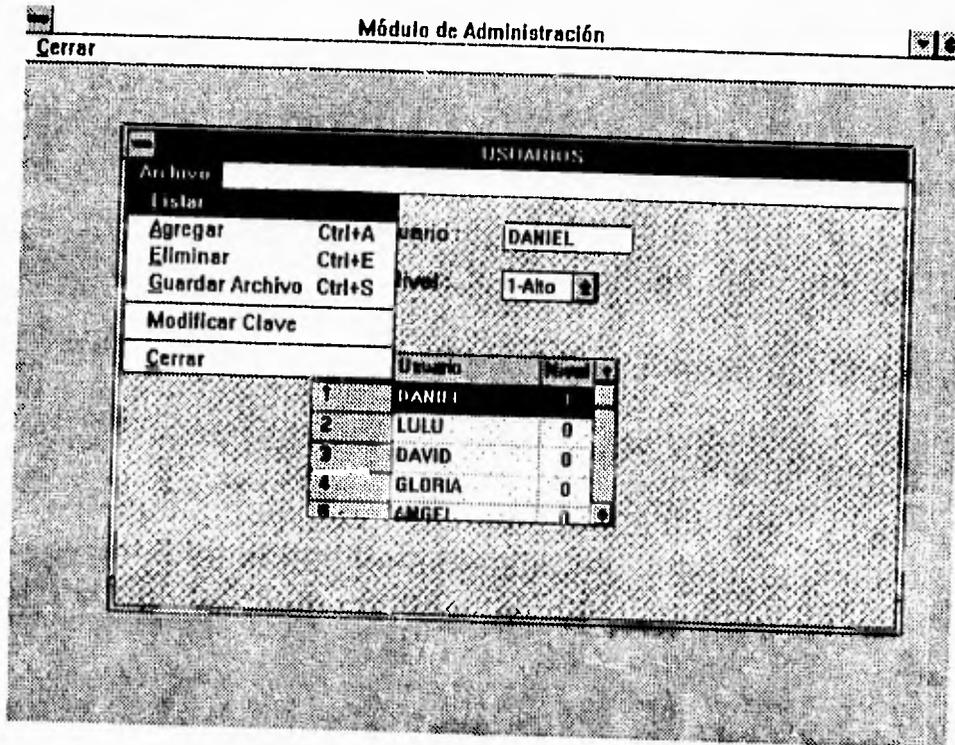
1. Una barra de menú que permite listar, eliminar, agregar y salvar servicios. Además permite cerrar la ventana y volver al menú de administración.
2. Una caja de texto para ingresar el código de algún servicio (Causal) a pagarse. Una lista de estos códigos será provista por el banco.
3. Una caja de texto para ingresar una breve descripción de algún servicio.
4. Una lista que muestra los servicios ya definidos.

Ventana de Cuentas.

Objetos que forman a la ventana de cuentas:

1. Una barra de menú que permite listar, borrar, agregar, salvar e imprimir las cuentas. Además permite cerrar la ventana y volver al menú de administración.
2. Cinco cajas de texto para ingresar el número de cuenta.
3. Una caja de texto para ingresar una breve descripción de la cuenta.
4. Una lista que muestra las cuentas ya definidas.

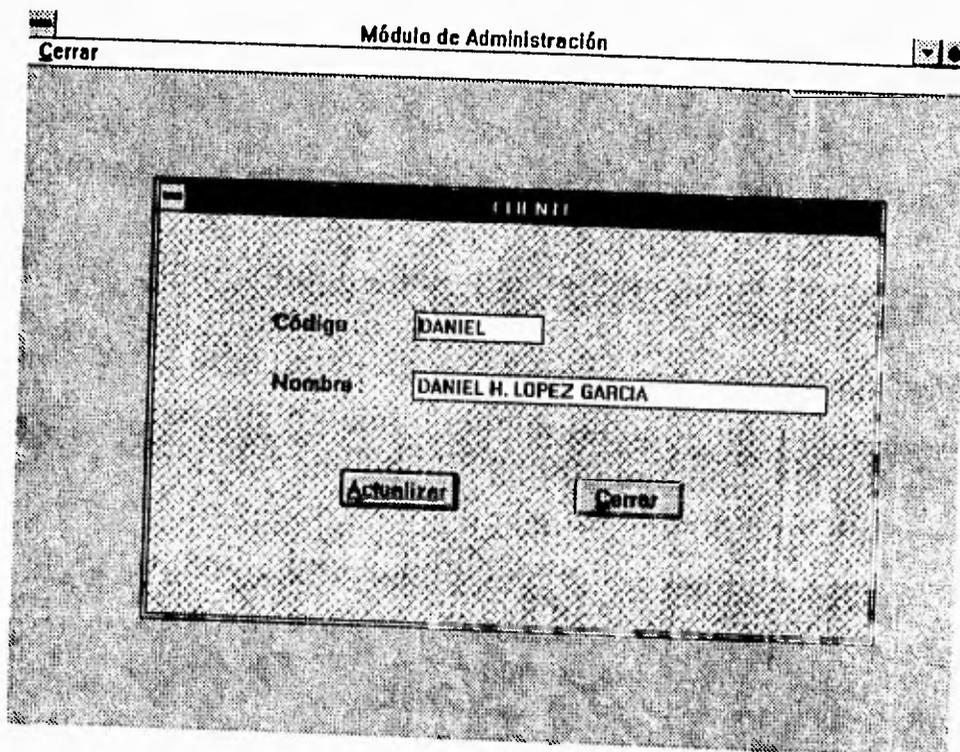
Ventana de Usuarios.



Objetos que forman a la ventana de usuarios:

1. Una barra de menú que permite listar, agregar, eliminar y salvar los usuarios que pueden acceder al módulo de comunicaciones. Además permite modificar la clave secreta del usuario y cerrar la ventana para volver al menú de administración.
2. Una caja de texto para ingresar el código del usuario.
3. Una lista para asignarle un nivel al usuario. El valor 0 (bajo) indica que solo tiene acceso a consultas y derecho a modificar su clave secreta. El valor 1 (alto) indica que tiene acceso a todo el sistema.
4. Una lista que muestra los usuarios ya definidos.

Ventana de Cliente.



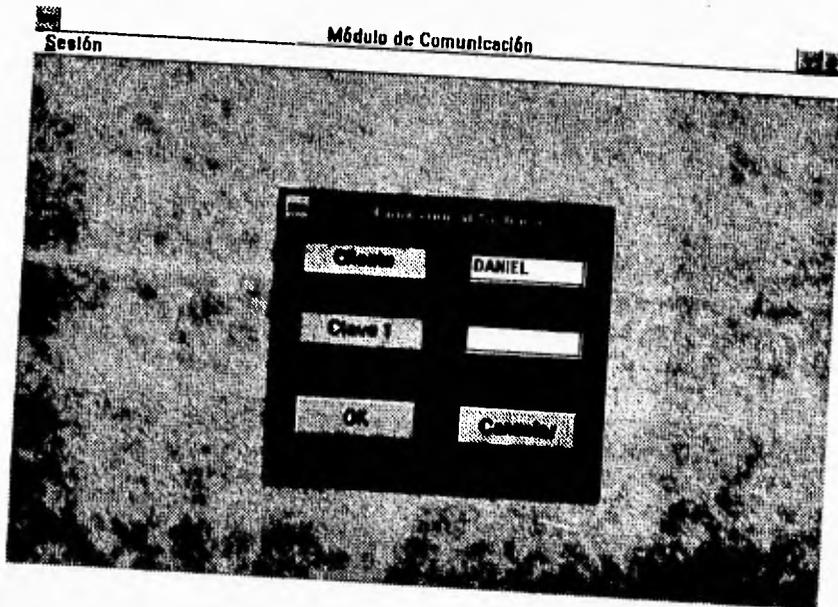
Objetos que forman a la ventana de cliente:

1. Una caja de texto para ingresar el código del cliente.
2. Una caja de texto para ingresar el nombre o razón social del cliente.
3. Un botón que permite actualizar la especificación del código y nombre del cliente
4. Un botón que permite cerrar la ventana y volver al menú de administración.

V.3.3 MÓDULO DE COMUNICACIONES.

El uso de este módulo permite al usuario iniciar la comunicación con el equipo central del banco, el acceso a sus bases de datos y efectuar transacciones.

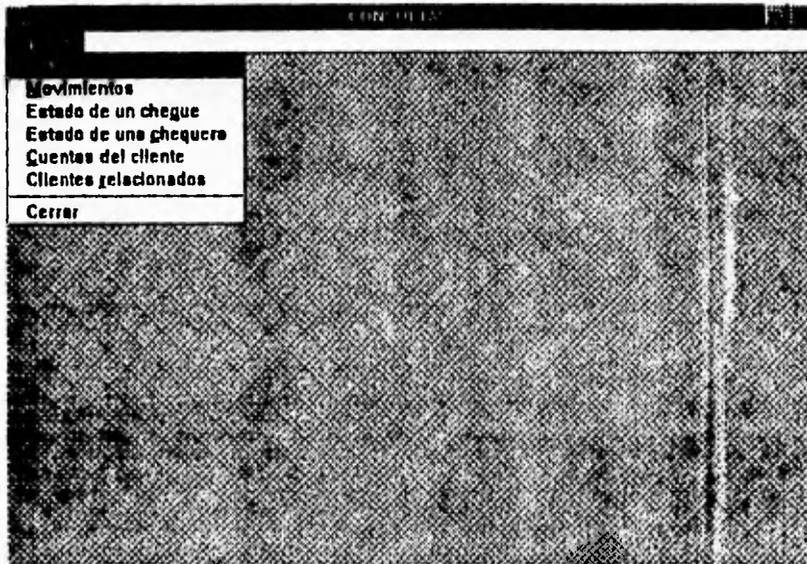
Ventana de Conexión.



Objetos que forman a la ventana de conexión:

1. Una caja de texto para ingresar el código del cliente. Este campo viene con el código definido en el módulo de administración, pero puede ser modificado.
2. Una caja de texto para ingresar la clave secreta asignada por el banco. Una vez usada, ésta se desecha y no puede volver a usarse. El banco proporcionará regularmente al cliente una lista de claves no re-utilizables bajo las políticas de seguridad previamente establecidas.
3. Un botón de confirmación y otro de cancelación de la conexión.

Ventana de Consultas.

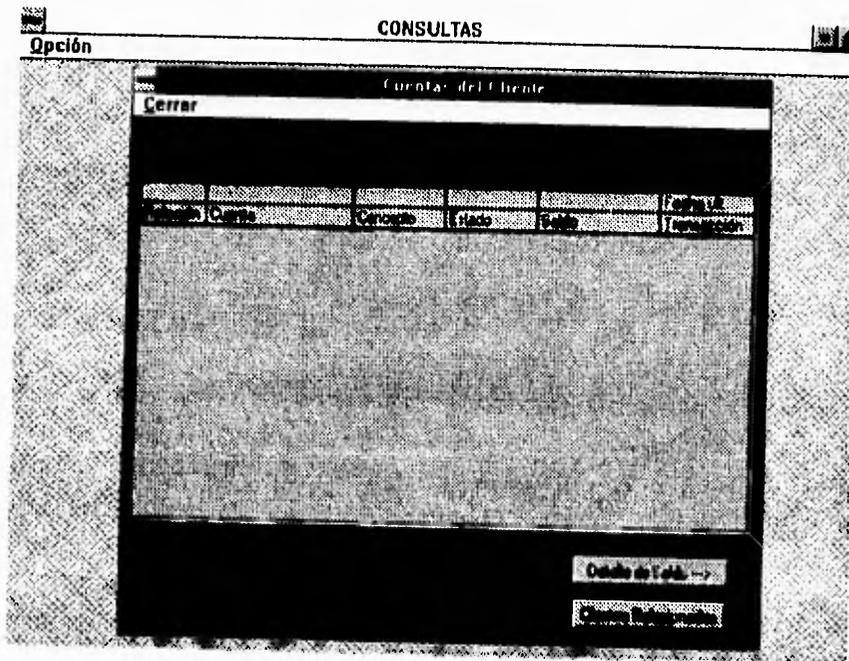


Objetos que forman a la ventana de consultas:

1. Una barra de menú con las siguientes opciones:

- Saldo de una cuenta de cheques
- Movimientos de la cuenta de cheques
- Estado de un cheque
- Estado de una chequera
- Cuentas del cliente
- Clientes relacionados con una cuenta de cheques.

Al ir seleccionando las opciones aparecerán nuevas ventanas con la información que el cliente necesita.

Ventana Cuentas del cliente.

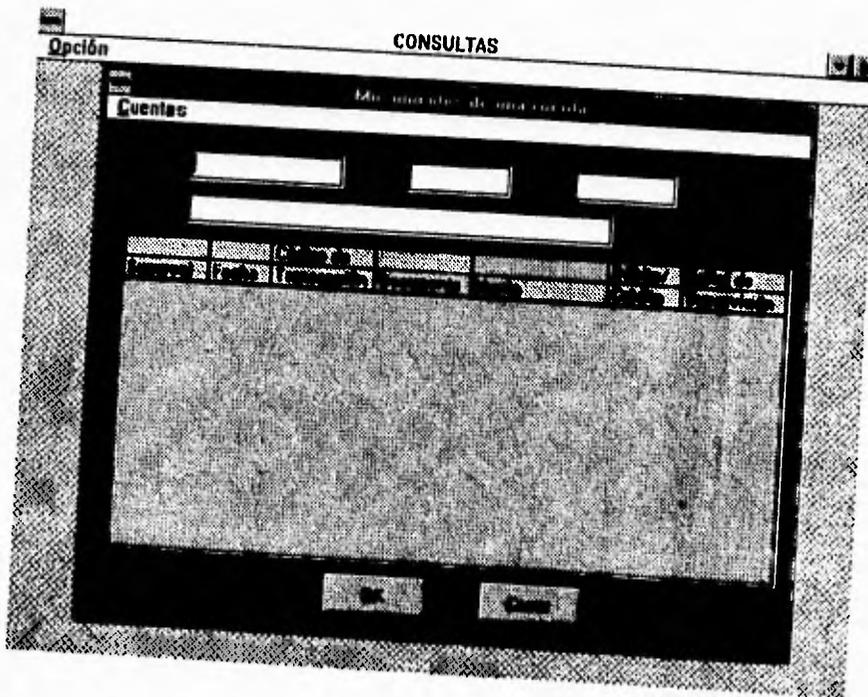
Esta transacción solicita directamente a la base de datos del banco el detalle de las cuentas con las que el cliente tiene relación. Esta transacción no solicita ningún dato de entrada.

Objetos que forman a la ventana de cuentas del cliente

1. Una barra de menú con la opción de cerrar la ventana y volver al menú de consultas.
2. Un cuadro con información de las cuentas.
3. Dos botones que, para facilitar la navegación, permiten pasar directamente a las ventanas de saldos y clientes relacionados si ya hemos seleccionado alguna de las cuentas del cuadro de información.

5. Una caja de texto indicando el tipo de la cuenta de cheques, ya sea ordinaria o de crédito.
6. Una caja de texto que informa del estado de la cuenta. Una cuenta podría estar activa, inactiva, cerrada o bloqueada.
7. Un cuadro proporcionando los diferentes saldos de la cuenta.
8. Una caja de texto indicando el valor del cupo de sobregiro asignado a la cuenta si es que lo hubiese.
9. Una caja de texto señalando la fecha de expiración del cupo de sobregiro
10. Una caja de texto indicando los intereses devengados por saldo sobregirado y
11. Un botón que permite cursar la transacción una vez que se ha asignado el número de cuenta deseado y otro botón que permite cancelar la transacción y volver al menú de consultas.

Ventana Movimientos de una Cuentas de Cheques.



Esta transacción solicita directamente a la base de datos del banco el detalle de los movimientos de una cuenta de cheques para el periodo especificado por las fechas.

Objetos que forman a la ventana de movimientos de una cuenta de cheques:

1. Una barra de menú con la opción de consultar la cuentas previamente definidas en el módulo de administración.
2. Un caja de texto para indicar el número de cuenta asociado a la transacción que se solicita. Este número puede seleccionarse entre las cuentas ya definidas o puede ser el seleccionado en la consulta de cuentas del cliente.
3. Una caja de texto para ingresar la fecha a partir de la cual se requiere el detalle de movimientos.

4. Una caja de texto para indicar hasta cuando queremos el detalle de los movimientos.

5. Una caja de texto en la que el sistema mostrará el nombre de la cuenta.

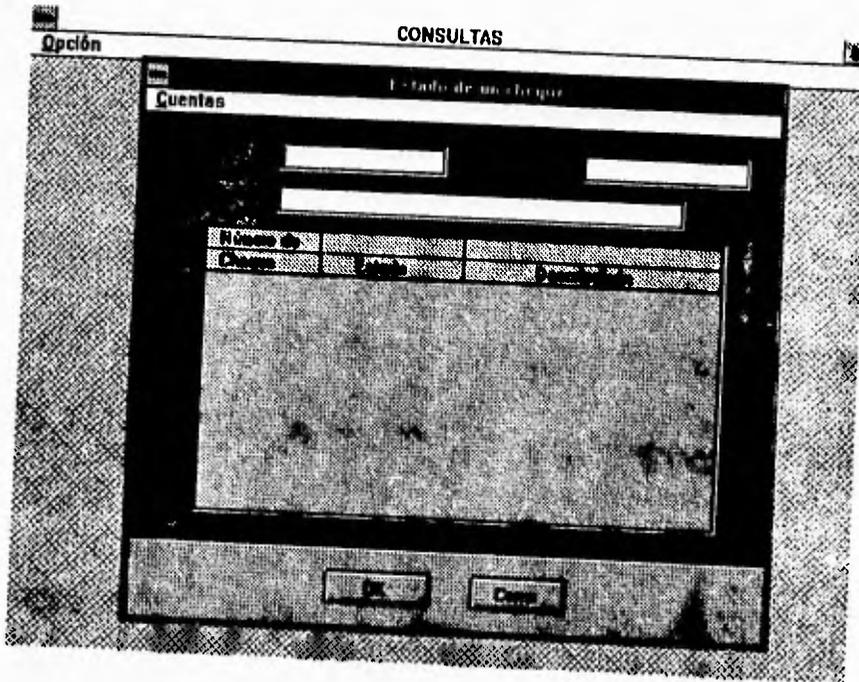
6. Una caja de información donde el sistema desplegará el detalle de los movimientos de la cuenta especificada.

7. Un botón que permite cursar la transacción.

8. Un botón que sirve para cerrar la ventana y volver al menú de consultas.

Las fechas son opcionales y, si el cliente los deja en blanco, el sistema mostrará todos los movimientos realizados sobre la cuenta registrados en la base de datos.

Ventana Estado de un Cheque.



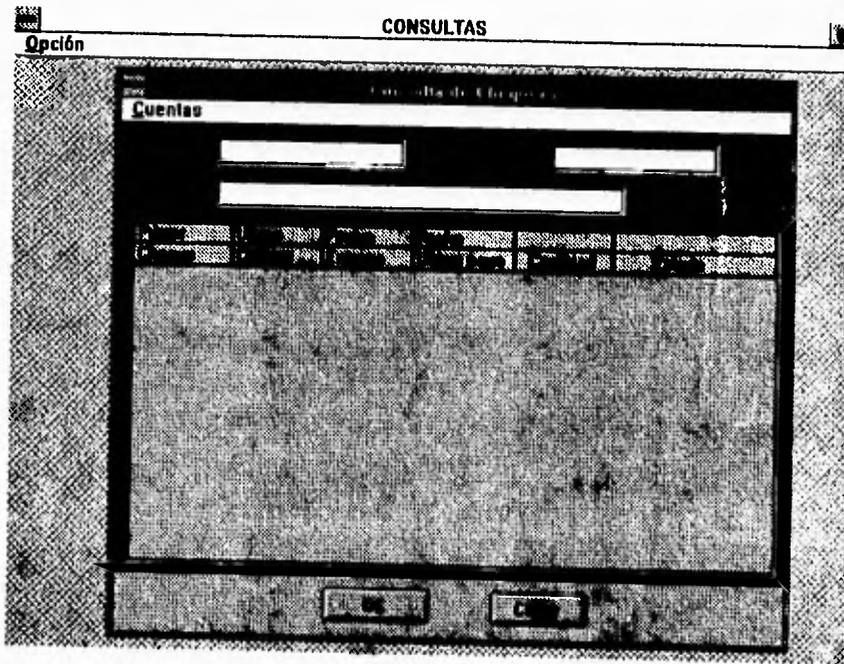
Esta transacción solicita a la base de datos del banco el detalle del estado de un cheque de una de las cuentas del cliente.

Objetos que forman a la ventana estado de un cheque:

1. Una barra de menú con la opción de consultar la cuentas previamente definidas en el módulo de administración.
2. Un caja de texto para indicar el número de cuenta asociado a la transacción que se solicita. Este número puede seleccionarse entre las cuentas ya definidas o puede ser el seleccionado en la consulta de cuentas del cliente.
3. Una caja de texto para ingresar el número del cheque cuyo estado estamos solicitando.

4. Una caja de texto donde el sistema indicará la cuenta de la que se trata.
5. Una caja de información donde el sistema desplegará el detalle del estado del cheque especificado y adicionalmente mostrará el estado de los cheques subsecuentes al indicado.
6. Un botón que permite cursar la transacción.
7. Un botón que sirve para cerrar la ventana y volver al menú de consultas.

Ventana Estado de una Chequera.



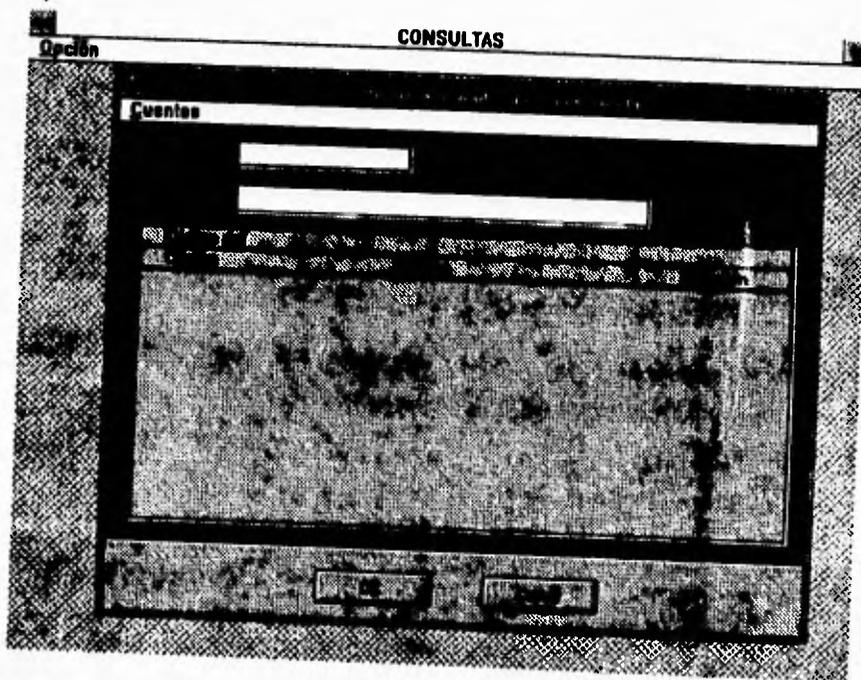
Esta transacción solicita a la base de datos del banco el detalle del estado de una chequera de una de las cuentas del cliente.

Objetos que forman a la ventana de estado de una chequera:

1. Una barra de menú con la opción de consultar la cuentas previamente definidas en el módulo de administración.
2. Un caja de texto para indicar el número de cuenta asociado a la transacción que se solicita. Este número puede seleccionarse entre las cuentas ya definidas o puede ser el seleccionado en la consulta de cuentas del cliente.
3. Una caja de texto para ingresar el número del primer cheque de la chequera cuyo estado estamos solicitando. Si este valor va en ceros, se desplegará el estado de todas las chequeras asociadas a la cuenta especificada.

4. Una caja de texto donde el sistema indicará la cuenta de la que se trata.
5. Una caja de información donde el sistema desplegará el detalle del estado de la chequera y adicionalmente mostrará el estado de las chequeras subsecuentes a la indicada.
6. Un botón que permite cursar la transacción.
7. Un botón que sirve para cerrar la ventana y volver al menú de consultas.

Ventana Clientes Relacionados con una Cuenta.



Esta transacción solicita a la base de datos del banco información de los clientes relacionados con una cuenta en particular.

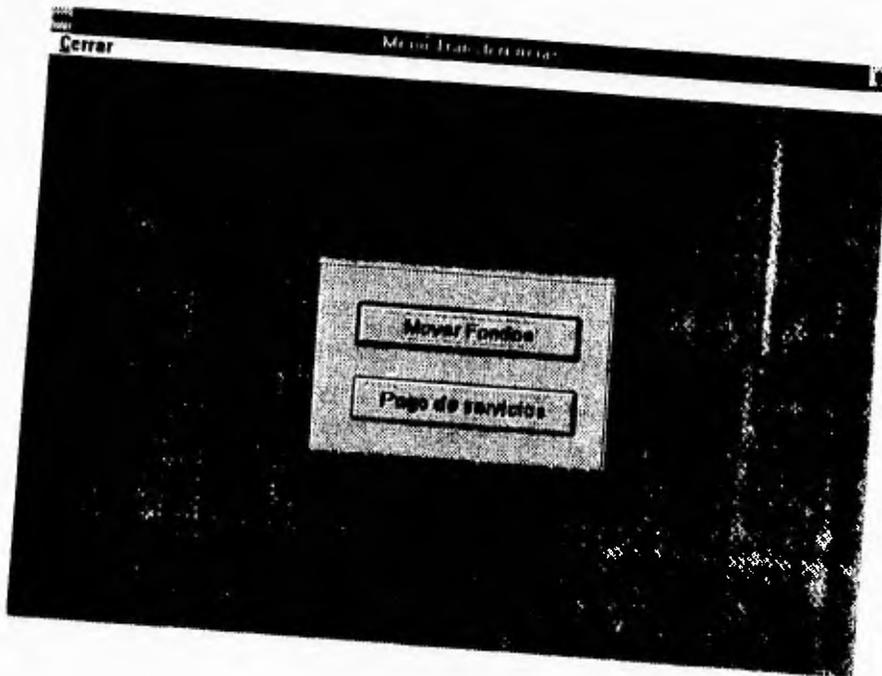
Objetos que forman a la ventana de clientes relacionados:

1. Una barra de menú con la opción de consultar la cuentas previamente definidas en el módulo de administración.
2. Un caja de texto para indicar el número de cuenta asociado a la transacción que se solicita. Este número puede seleccionarse entre las cuentas ya definidas o puede ser el seleccionado en la consulta de cuentas del cliente.
3. Una caja de texto donde el sistema indicará la cuenta de la que se trata.

Definición y diseño

4. Una caja de información donde el sistema desplegará el detalle de los clientes relacionados a la cuenta especificada.
5. Un botón que permite cursar la transacción.
6. Un botón que sirve para cerrar la ventana y volver al menú de consultas.

Ventana Menú de Transferencias.

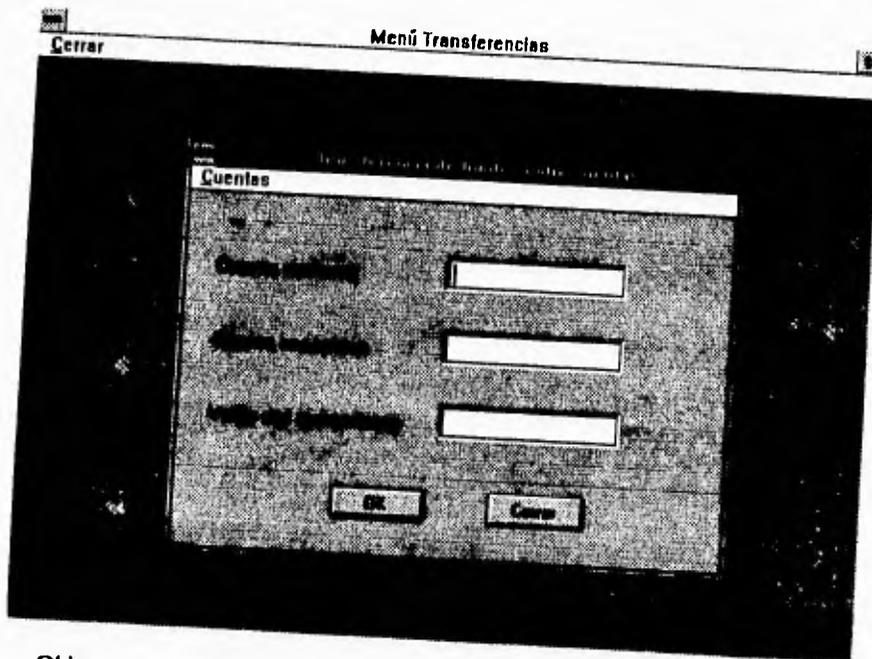


Objetos que forman al menú de transferencias:

1. Una barra de menú con la opción de cerrar la ventana y volver al menú de comunicaciones.
2. Un panel con dos botones con las opciones :
 - 1- Mover fondos y
 - 2- Pago de servicios.

La transacción de mover fondos permite transferir dinero entre dos cuentas. Sólo se requiere indicar la cuenta a debitar, la cuenta a acreditar y el monto a transferir.

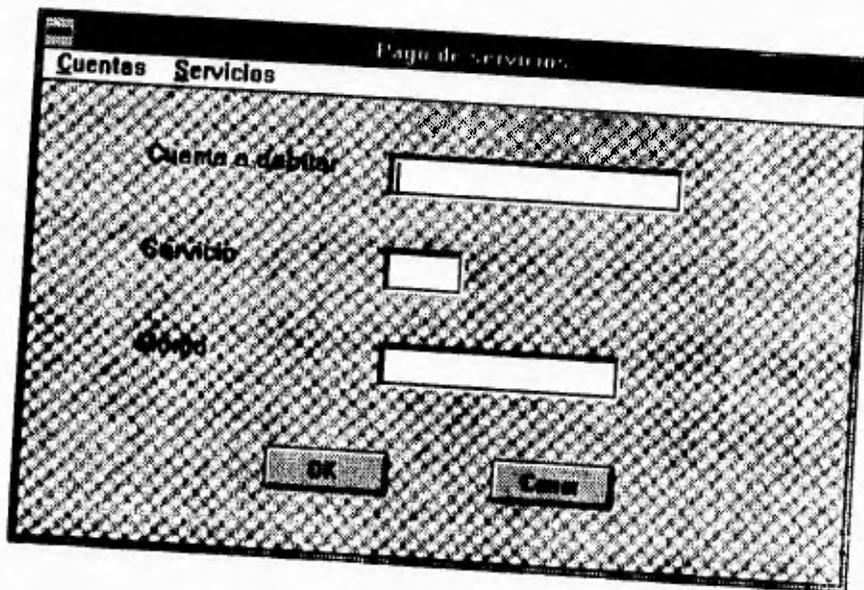
Ventana Transferencia de Fondos.



Objetos que forman a la ventana de transferencia de fondos:

1. Una barra de menú con la opción de consultar las cuentas previamente definidas en el módulo de administración.
2. Una caja de texto para especificar la cuenta a debitar.
3. Una caja de texto para especificar la cuenta a acreditar.
4. Una caja de texto para especificar al monto a transferir.
5. Un botón que permite cursar la transacción y otro que permite cerrar la ventana y volver al menú de transferencias.

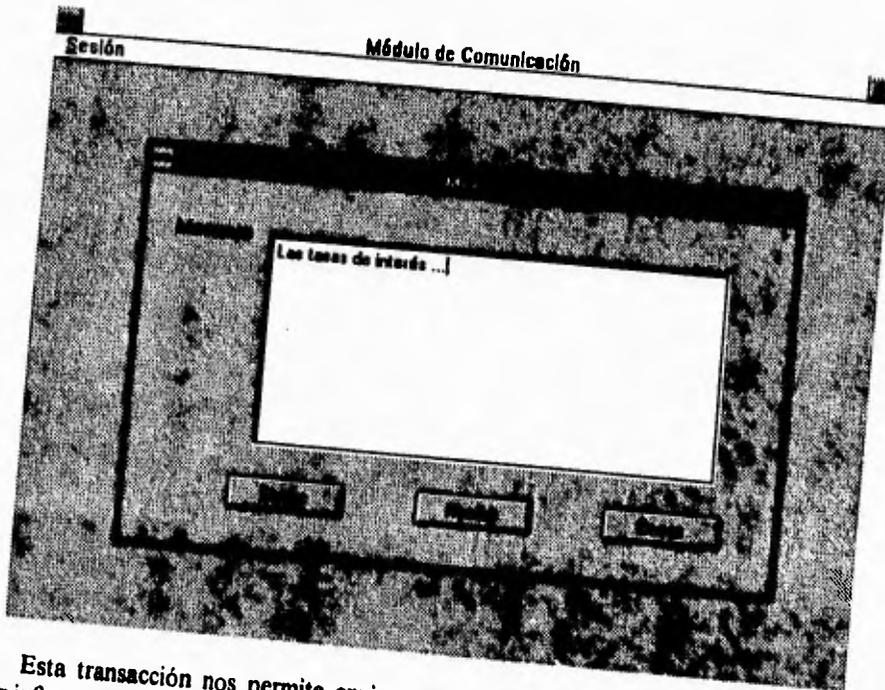
Ventana Pago de Servicios.



Objetos que forman a la ventana de pago de servicios:

1. Una barra de menú con la opción de consultar las cuentas y servicios previamente definidos en el módulo de administración.
2. Una caja de texto para especificar la cuenta a debitar.
3. Una caja de texto para especificar el código del servicio a pagar.
4. Una caja de texto para especificar al monto a pagar.
5. Un botón que permite cursar la transacción y otro que permite cerrar la ventana y volver al menú de transferencias.

Ventana de Mensajes.



Esta transacción nos permite enviar mensajes al ejecutivo de cuenta del cliente y recibir información de índole económica y/o cualquier noticia que tenga el banco.

Objetos que forman a la ventana de mensajes:

1. Una caja de texto donde el cliente puede escribir y/o recibir mensajes.
2. Un botón que permite enviar el mensaje.
3. Un botón que permite recibir mensajes.
4. Un botón para cerrar la ventana y volver al menú de comunicaciones.

CAPÍTULO VI
DESARROLLO

VI.1 ESTÁNDARES DE DESARROLLO.

El definir y el aplicar estándares ayuda a facilitar futuras modificaciones del sistema y mejora la comprensión del código. Consideraremos los siguientes estándares durante la programación del sistema BANCEMP:

- 1- Seguir los siguientes pasos al crear una nueva forma:
 - Abrir la forma.
 - Dibujar sus controles.
 - Definir las propiedades de la forma y los controles.
 - Escribir el código para cada uno de los objetos.

- 2- Evitar el uso de la instrucción GO TO.

- 3- Dividir los programas en 3 grandes secciones y en el siguiente orden:
 - Sección de declaración de variables locales.
 - Sección de validaciones.
 - Sección de proceso principal.

- 4- Usar rutinas globales en caso de que una misma sección de código tenga que usarse más de dos veces.

- 5- Referirse a los estándares descritos en el capítulo V sección 1.

VI.2 DESCRIPCIÓN DE LOS BLOQUES DE LÓGICA.

Recordando que Visual Basic es un lenguaje de programación conducido por eventos y orientado a objetos, el sistema BANCEMP no está compuesto por un único e interminable programa general, sino que está constituido por diferentes bloques de lógica asociados a diferentes objetos que sólo se ejecutan al ocurrir el evento. A continuación se presenta una descripción global de la función de los bloques más importantes asociados a las ventanas del sistema según especificaciones de la fase de investigación y de la fase de diseño.

VI.2.1 MENÚ PRINCIPAL.

Esta pantalla es la bienvenida al sistema y la barrera de seguridad al mismo.

Subrutinas:

*** Sub BotOkSegClick()**

- Valida el número máximo de intentos fallidos de conexión y termina la ejecución del sistema si este límite ha sido excedido.
- Abre el archivo de usuarios autorizados y valida el código de usuario ingresado, enviando un mensaje de error si éste no existe y terminando la ejecución del bloque.
- *Des-encrpta* la clave secreta del archivo de usuarios autorizados y la compara con la clave ingresada, enviando un mensaje de error si no coinciden y terminando la ejecución del bloque.
- Hace invisible el panel de recepción de código de usuario y clave secreta y hace visible el panel de menú de los módulos de comunicación y administración para que el usuario continúe la navegación después de haber pasado todas la validaciones anteriores.
- Retiene el nivel del usuario para futuras validaciones.

VI.2.2 MENÚ DE ADMINISTRACIÓN.

Esta pantalla muestra las distintas opciones del módulo de administración.

Subrutinas:

*** Sub Form_Load()**

- Discrimina, de acuerdo al nivel del usuario, cuales opciones del menú debe habilitar y cuales dejar deshabilitadas.

VI.2.3 VENTANA DE CUENTAS.

Esta pantalla tiene como objetivo dar mantenimiento a las cuentas del cliente que se definen localmente y que después estarán disponibles para su uso en el módulo de comunicaciones.

Subrutinas:

*** Sub OrdAgregar_Click**

- Valida que se ingresen valores numéricos en los campos del número de cuenta.
- Valida que la descripción de las cuentas no se deje sin información.
- Verifica el límite de registros en el archivo local.
- Agrega el nuevo registro la estructura de cuentas en memoria.

*** Sub OrdEliminar_Click**

- Valida que exista la cuenta a borrar en el archivo de cuentas local.
- Borra el registro de la estructura de cuentas en memoria .

*** Sub OrdGuardar_Click**

- Actualiza el archivo local de cuentas con la información de la estructura de cuentas residente en memoria.

*** Sub OrdListar_Click**

- Carga la estructura de cuentas residente en memoria con información obtenida del archivo local de cuentas y la presenta en pantalla.

*** Sub SelRegCue()**

- Copia el registro seleccionado de la lista de cuentas en pantalla a las cajas de texto de entrada de datos.

VI.2.4 VENTANA DE USUARIOS.

Esta pantalla tiene como objetivo dar mantenimiento a los usuarios que tienen acceso al sistema BANCEMP.

Subrutinas:

*** Sub Form_Load()**

- Hace invisible las opciones a las que un usuario con nivel bajo no tiene permitido el acceso.
- Dibuja la lista donde se mostrarán los usuarios.

*** Sub OrdAgregar_Click()**

- Valida que se ingrese información en la caja de texto referente al código de usuario.
- Valida que no se exceda el límite de registros del archivo local de usuarios.
- Valida que no exista el usuario a agregar.
- Agrega el registro a la tabla de usuarios residente en memoria.

*** Sub OrdEliminar_Click()**

- Valida que exista el usuario a borrar.
- Borra registro de la tabla local de usuarios en memoria.

*** Sub OrdGuardar_Click()**

- Actualiza el archivo local de usuarios con la información de la tabla residente en memoria.
- Para un usuario nuevo deja el campo correspondiente a la clave secreta en blancos, teniendo que ingresar una clave secreta la primera vez que se conecte dicho usuario a BANCEMP.

*** Sub OrdListar_Click()**

- Actualiza la tabla local de usuarios con información proveniente del archivo local de usuarios.

*** Sub OrdModificar_()**

- Muestra la pantalla que permite modificar la clave secreta del usuario conectado actualmente en el sistema. Esta pantalla *encripta* la clave antes de guardarla en el archivo local de usuarios.

*** Sub SelRegUsu()**

- Copia la información del registro seleccionado de la lista a las cajas de texto de recepción de información.

VI.2.5 VENTANA DE CLIENTE.

Esta pantalla tiene como propósito actualizar el código y la descripción del cliente del banco y dueño del sistema BANCEMP.

Subrutinas:

*** Sub BotAceptar_Click()**

- Valida que se ingrese información en las cajas de texto asociadas al código y nombre del cliente.
- Actualiza el archivo local de cliente con la información ingresada.

*** Sub Form_Load()**

- Presenta el código y nombre actual del cliente al cargarse la ventana en memoria.

VI.2.6 VENTANA DE SERVICIOS.

Esta pantalla tiene como objetivo dar mantenimiento a los servicios definidos en BANCEMP y que el cliente podrá utilizar en el módulo de comunicaciones al hacer un pago.

Subrutinas:

*** Sub OrdAgregar_Click()**

- Valida que se ingrese un valor numérico en la caja de texto asociada al código del servicio.
- Valida que se ingrese información en la descripción del servicio.
- Valida el número máximo de registros en el archivo de servicios local.
- Agrega el registro a la tabla local de servicios.

*** Sub OrdEliminar_Click()**

- Valida que exista el registro a borrar en la tabla local de servicios.
- Borra el registro de la tabla local de servicios.

*** Sub OrdGuardar_Click()**

- Actualiza el archivo local de servicios con la información de la tabla local de servicios residente en memoria.

*** Sub OrdListar_Click()**

- Actualiza la tabla local de servicios con información proveniente del archivo local de servicios.

*** Sub SelRegSer()**

- Copia el registro seleccionado de la tabla a las cajas de texto de entrada de datos.

VI.2.7 MENÚ DE COMUNICACIONES.

Esta pantalla tiene como objetivo permitir al usuario conectarse al sistema central bancario SFB y poder navegar hacia las pantallas de consultas y transferencias.

Subrutinas:

*** Sub GP_AbreHost()**

- Inicia la comunicación con la computadora central del banco abriendo una sesión de Infoconnect.

- Abre la ventana del sistema SFB definida en COMS si es que está cerrada.

*** Sub GP_Ctas_Predefinidas()**

- Carga las cuentas definidas en el módulo de administración en una tabla temporal para utilizarlas dentro del módulo de comunicaciones.

*** Sub GP_Ser_Predefinidos()**

- Carga los servicios definidos en el módulo de administración en una tabla temporal para utilizarlos dentro del módulo de comunicaciones.

*** Sub GP_Pregunta_Salir()**

- Desconecta a BANCEMP del sistema central bancario SFB si es que estaba conectado y cierra la sesión de Infoconnect perdiéndose la comunicación con el HOST.

- Cierra el menú de comunicaciones y presenta el menú principal.

*** Sub OrdDesconexión_Click()**

- Desconecta a BANCEMP de SFB sin perder la comunicación con el HOST.

VI.2.8 VENTANA DE CONEXIÓN.

Esta pantalla tiene como objetivo conectar al cliente al sistema bancario SFB.

Subrutinas:

*** Sub BotOkConex_Click()**

- Valida que se ingrese información en las cajas de texto.

- Envía el mensaje de requerimiento conexión y recibe una respuesta.

VI.2.9 MENÚ CONSULTAS.

Esta pantalla tiene como objetivo permitir la navegación a las ventanas de consultas del módulo de comunicaciones una vez conectado al sistema SFB.

VI.2.10 MENÚ DE TRANSFERENCIAS.

Esta pantalla tiene como objetivo permitir la navegación a las ventanas de transferencia de fondos entre cuentas y pago de servicios una vez conectado a SFB.

VI.2.11 CUENTAS DEL CLIENTE.

Esta pantalla tiene como objetivo consultar información residente en el equipo central del banco sobre las cuentas de cheques del cliente.

Subrutinas:

*** Sub Form_Load()**

- Dibuja tabla donde se presentará la información.
- Envía requerimiento de consulta al HOST y recibe respuesta.
- Edita y presenta información en pantalla.
- Si se selecciona alguna cuenta, ésta se retiene para utilizarla durante la navegación dentro de BANCEMP.

VI.2.12 CONSULTA DE SALDOS.

Subrutinas:

*** Sub BotOk_Click()**

- Valida que se ingrese un valor numérico en la cuenta.
- Envía requerimiento al HOST y recibe respuesta.
- Edita y presenta información recibida.

VI.2.13 CONSULTA DE MOVIMIENTOS.

Subrutinas:

*** Sub BotOk_Click()**

- Valida que se ingrese un valor numérico en la cuenta y en el rango de fechas.
- Envía requerimiento al HOST y recibe respuesta.
- Edita y presenta información recibida.

VI.2.14 CONSULTA DE CHEQUERAS.

Esta pantalla tiene como objetivo consultar información de las chequeras asociadas a una cuenta del cliente.

Subrutinas:

*** Sub BotOk_Click()**

- Valida que se ingrese un valor numérico en la cuenta y en el número de chequera inicial.
- Envía requerimiento al HOST y recibe respuesta.
- Edita y presenta información recibida.

VI.2.15 CONSULTA DE CHEQUES.

Esta pantalla tiene como objetivo consultar información de los cheques asociados a una chequera de una cuenta del cliente.

Subrutinas:

*** Sub BotOk_Click()**

- Valida que se ingrese un valor numérico en la cuenta y en el número de cheque inicial.
- Envía requerimiento al HOST y recibe respuesta.
- Edita y presenta información recibida.

VI.2.16 CONSULTA CLIENTES RELACIONADOS.

Esta pantalla tiene como objetivo consultar información de los clientes asociados a una cuenta del cliente BANCEMP.

Subrutinas:

*** Sub BotOk_Click()**

- Valida que se ingrese un valor numérico en la cuenta.
- Envía requerimiento al HOST y recibe respuesta.
- Edita y presenta información recibida.

VI.2.17 TRANSFERENCIA DE FONDOS.

Esta pantalla tiene como objetivo permitir al cliente BANCEMP transferir dinero entre dos de sus cuentas de cheques.

Subrutinas:

*** Sub BotOk_Click()**

- Valida que se ingrese un valor numérico en las cuentas.
- Valida que las cuentas sean diferentes.
- Valida el formato del monto a transferir.
- Envía requerimiento al HOST y recibe respuesta.

VI.2.18 PAGO DE SERVICIOS.

El objetivo de esta pantalla es el de permitir al cliente BANCEMP hacer el pago de algún servicio.

Subrutinas:

*** Sub BotOk_Click()**

- Valida que se ingrese un valor numérico en la cuenta en el código de servicio.
- Valida el formato del monto a pagar.
- Envía requerimiento al HOST y recibe respuesta.

VI.2.19 ENVÍO DE MENSAJES.

Esta pantalla tiene como objetivo permitir al cliente BANCEMP enviar algún mensaje al banco y leer mensajes dirigidos a él.

VI.2.20 PROCEDIMIENTOS Y FUNCIONES GLOBALES.

El sistema BANCEMP hace uso de rutinas globales para evitar la repetición de código y facilitar su mantenimiento.

- * GF_Chr_Numero: Valida que un caracter sea numérico o punto.
- * GF_EdFecha\$: Cambia el formato de una fecha de ddmmaa a dd/mm/aa.

- * GF_Formatee\$: Cambia el formato de un monto proveniente del HOST a formato para la PC.
- * GP_Valida_Numero: Valida que solo haya números dentro de una cadena de caracteres.
- * GP_Valida_Usuario: Valida que el usuario especificado exista en el archivo local de usuarios.
- * GF_ErrorCF: Valida el código de mensaje enviado por el HOST en una consulta múltiple (COPY.FROM).
- * GP_ClsHost: Desconecta a BANCEMP de SFB.
- * GP_EnviarMsg: Envía mensaje al HOST con formato dependiente de la transacción de que se trate.
- * GD_FormatHea: Prepara el Header del mensaje a enviar al HOST.
- * GP_OpnHost: Conecta a BANCEMP a SFB.
- * GP_RcvHost: Recibe respuesta de conexión del HOST.
- * GP_RecibirRpta: Recibe mensaje del HOST después de haber cursado alguna transacción.
- * GP_Signo: Edita el signo de un monto recibido del HOST.
- * GF_Descripta_clave: *Des-cripta* la clave de usuario especificada.
- * GF_Encrypta: *Encrypta* la clave de usuario especificada.

Existen subrutinas y procedimientos que por ser tan simples no se describen detalladamente, pero su función cae dentro de lo siguiente:

- Abrir una ventana.
- Cerrar una ventana.
- Inicializar variables.
- Inicializar dimensiones de *Objetos*.

VI.3 PROGRAMACIÓN DEL SISTEMA.

Los bloques de lógica asociado a cada objeto inician con la instrucción SUB y terminan con la instrucción END SUB. Para mayor claridad, estas dos instrucciones están resaltadas en los siguientes listados:

VI.3.1 LÓGICA ASOCIADA AL MENÚ PRINCIPAL.

```
Sub BotAdministración_Click ()
```

```
    MenuAdmin.Show 1
```

```
End Sub
```

```
Sub BotComunicación_Click ()
```

```
    MenuComunica.Show 1
```

```
End Sub
```

```
Sub BotOkSeg_Click ()
```

```
    ' Verifica el nombre del usuario y su clave secreta. Máximo número de intentos conexión
```

```
    If GD_Intentos > GC_MaxInt Then
```

```
        Beep
```

```
        MsgBox MSG_E011, 48, "ERROR"
```

```
        Exit Sub
```

```
    End If
```

```
    GD_UsuEnt = UCase(RTrim$(MenuPrincipal.TxtNomUsu.Text))
```

```
    If MenuPrincipal.TxtClvUsu.Text <> " " Then
```

```
        GD_ClvEnt = UCase(RTrim$(MenuPrincipal.TxtClvUsu.Text))
```

```
    Else
```

```
        GD_ClvEnt = Chr$(0) & Chr$(0) & Chr$(0) & Chr$(0) & Chr$(0)  
                & Chr$(0) & Chr$(0) & Chr$(0) & Chr$(0) & Chr$(0)
```

```
        GD_ClvEnt = GF_Descripta_Clave(GD_ClvEnt)
```

```
    End If
```

```
    Call GP_Valida_Usuario(GD_UsuEnt)
```

```
    If GD_Existe = False Then
```

```
        Beep
```

```
        MsgBox MSG_E011, 48, "ERROR"
```

```
        MenuPrincipal.TxtNomUsu.SetFocus
```

```
        GD_Intentos = GD_Intentos + 1
```

```
        Exit Sub
```

```
    End If
```

```
    ' Des-cripta clave secreta
```

```
    GD_Clave = GF_Descripta_Clave(GD_Clave)
```

```
    ' Valida Clave
```

```
    If GD_Clave <> GD_ClvEnt Then
```

```
        Beep
```

```
        MsgBox MSG_E017, 16, "ERROR"
```

```
        MenuPrincipal.TxtClvUsu.SetFocus
```

```
        GD_Intentos = GD_Intentos + 1
```

```
        Exit Sub
```

```

End If
If Not GD_HayErr Then
  MenuPrincipal.Panel3D3.Visible = False ' Panel de seguridad invisible
  MenuPrincipal.Panel3D1.Visible = True  ' Panel menú visible
  MenuPrincipal.BotAdministración.Enabled = True
  MenuPrincipal.BotComunicación.Enabled = True
  BotAdministración.SetFocus
End If
End Sub

```

```

Sub Form_Unload (Cancel As Integer)
  End
End Sub

```

```

Sub OrdCerrar_Click ()
  End
End Sub

```

VI.3.2 LÓGICA ASOCIADA AL MENÚ DE ADMINISTRACIÓN.

```

Sub BotCliente_Click ()
  DXMCL.Show 1
End Sub

```

```

Sub BotCuentas_Click ()
  AAMCU.Show 1
End Sub

```

```

Sub BotServicios_Click ()
  AAMSE.Show 1
End Sub

```

```

Sub BotUsuarios_Click ()
  DXMUS.Show 1
End Sub

```

```

Sub Form_Load ()
  ' Seguridad nivel de usuario

```

```

If GD_NivEnt = GC_Bajo Then
    MenuAdmin.BotCliente.Enabled = False
    MenuAdmin.BotServicios.Enabled = False
    MenuAdmin.BotCuentas.Enabled = False
End If
End Sub

```

```

Sub OrdCerrar_Click ()
    Unload MenuAdmin
End Sub

```

VI.3.3 LÓGICA ASOCIADA A LA VENTANA DE CUENTAS.

```

Sub Form_Load ()
    ' Inicializa variables globales
    GD_TotRegCue = 0
    Call GP_IniciaGriCuentas(GriCuentas)
    Call OrdListar_Click
End Sub

```

```

Sub GriCuentas_Click ()
    Call SelRegCue
End Sub

```

```

Sub GriCuentas_GotFocus ()
    AAMCU.GriCuentas.HighLight = True
End Sub

```

```

Sub GriCuentas_LostFocus ()
    AAMCU.GriCuentas.HighLight = False
End Sub

```

```

Sub GriCuentas_SelChange ()
    Call SelRegCue
End Sub

```

```

Sub OrdAgregar_Click ()
    GD_Producto = AAMCU.TxtProCue.Text

```

```

GD_Moneda = AAMCU.TxtMonCue.Text
GD_Oficina = AAMCU.TxtOfiCue.Text
GD_Correlativo = AAMCU.TxtNumCue.Text
GD_Digito = AAMCU.TxtDigVer.Text
GD_Cuenta = GD_Producto + GD_Moneda + GD_Oficina + GD_Correlativo + GD_Digito
GD_Nombre = UCase$(RTrim$(AAMCU.TxtNomCue.Text))
' Valida producto cuenta
Call GP_Valida_Numero(GD_Producto)
If GD_Producto = " " Or GD_EsNumero = False Then
    Beep
    MsgBox MSG_E012, 48, "ERROR"
    AAMCU.TxtProCue.SetFocus
    Exit Sub
End If
' Valida moneda cuenta
Call GP_Valida_Numero(GD_Moneda)
If GD_Moneda = " " Or GD_EsNumero = False Then
    Beep
    MsgBox MSG_E012, 48, "ERROR"
    AAMCU.TxtMonCue.SetFocus
    Exit Sub
End If
' Valida sucursal cuenta
Call GP_Valida_Numero(GD_Oficina)
If GD_Oficina = " " Or GD_EsNumero = False Then
    Beep
    MsgBox MSG_E012, 48, "ERROR"
    AAMCU.TxtOfiCue.SetFocus
    Exit Sub
End If
' Valida digito verificador
Call GP_Valida_Numero(GD_Digito)
If GD_Digito = " " Or GD_EsNumero = False Then
    Beep
    MsgBox MSG_E012, 48, "ERROR"
    AAMCU.TxtDigVer.SetFocus
    Exit Sub
End If
' Valida Descripción de la cuenta
If GD_Nombre = " " Then
    Beep
    MsgBox MSG_E006, 48, "ERROR"
    AAMCU.TxtNomCue.SetFocus
    Exit Sub
End If

```

```

' Valida número máximo de registros en la tabla
If GD_TotRegCue >= GC_MaxRegCue Then
    Beep
    MsgBox MSG_E007, 48, "ERROR"
    Exit Sub
End If
' Valida que no exista el registro
For GD_Contador1 = 1 To GC_MaxRegCue
    AAMCU.GriCuentas.Col = 1
    AAMCU.GriCuentas.Row = GD_Contador1
    If AAMCU.GriCuentas.Text = GD_Cuenta Then
        Beep
        MsgBox MSG_E013, 48, "ERROR"
        AAMCU.TxtProCue.SetFocus
        Exit Sub
    End If
Next GD_Contador1
' Agrega a la estructura
GD_TotRegCue = GD_TotRegCue + 1
GD_RegCuenta(GD_TotRegCue).GD_FolCue = Str$(GD_TotRegCue)
GD_RegCuenta(GD_TotRegCue).GD_NumCue = GD_Cuenta
GD_RegCuenta(GD_TotRegCue).GD_NomCue = GD_Nombre
' Actualiza Grid
Call GP_ActualizaGriCue
AAMCU.TxtProCue.SetFocus
End Sub

Sub OrdCerrar_Click ()
    Unload AAMCU
End Sub

Sub OrdEliminar_Click ()
    GD_Producto = AAMCU.TxtProCue.Text
    GD_Moneda = AAMCU.TxtMonCue.Text
    GD_Oficina = AAMCU.TxtOfiCue.Text
    GD_Correlativo = AAMCU.TxtNumCue.Text
    GD_Digito = AAMCU.TxtDigVer.Text
    GD_Cuenta = GD_Producto + GD_Moneda + GD_Oficina + GD_Correlativo + GD_Digito
    ' Valida que exista la cuenta a borrar
    GD_Existe = False
    For GD_Contador1 = 1 To GD_TotRegCue
        AAMCU.GriCuentas.Col = 1
        AAMCU.GriCuentas.Row = GD_Contador1
    
```

```

If AAMCU.GriCuentas.Text = GD_Cuenta Then
  GD_Existe = True
  Exit For
End If
Next GD_Contador1
If GD_Existe = False Then
  Beep
  MsgBox MSG_E014, 48, "ERROR"
  AAMCU.TxtProCue.SetFocus
  Exit Sub
End If
' Borra registro
For GD_Contador2 = GD_Contador1 To GD_TotRegCue - 1
  GD_RegCuenta(GD_Contador2).GD_FolCue =
  Str$(Val(GD_RegCuenta(GD_Contador2 + 1).GD_FolCue) - 1)
  GD_RegCuenta(GD_Contador2).GD_NumCue =
  GD_RegCuenta(GD_Contador2 + 1).GD_NumCue
  GD_RegCuenta(GD_Contador2).GD_NomCue =
  GD_RegCuenta(GD_Contador2 + 1).GD_NomCue
Next GD_Contador2
GD_RegCuenta(GD_TotRegCue).GD_FolCue = " "
GD_RegCuenta(GD_TotRegCue).GD_NumCue = " "
GD_RegCuenta(GD_TotRegCue).GD_NomCue = " "
GD_TotRegCue = GD_TotRegCue - 1
' Actualiza Grid
Call GP_ActualizaGriCue
AAMCU.TxtProCue.SetFocus
End Sub

```

Sub OrdGuardar_Click ()

```

' Inicializa valores
GD_LongReg = Len(GD_RegCuenta(1))
On Error GoTo RutinaError
' Abre el archivo de cuentas
Open GC_ArchCuentas For Random As #1 Len = GD_LongReg
For GD_CuentaRegs = 1 To GC_MaxRegCue
  Put #1, , GD_RegCuenta(GD_CuentaRegs)
Next GD_CuentaRegs
Close #1
SalGuardar:
Exit Sub
RutinaError:
MsgBox MSG_E001, 48, Error
Resume SalGuardar

```

End Sub

Sub OrdListar_Click ()

' Inicializa valores

GD_LongReg = Len(GD_RegCuenta(1))

On Error GoTo RutinaErrorListar

' Abre el archivo de cuentas

Open GC_ArchCuentas For Random As #1 Len = GD_LongReg

GD_NumReg = LOF(1) / GD_LongReg

For GD_CuentaRegs = 1 To GD_NumReg

 Get #1, , GD_RegCuenta(GD_CuentaRegs)

Next GD_CuentaRegs

Close #1

Call GP_ActualizaGriCue

AAMCU.TxtProCue = ""

AAMCU.TxtMonCue = ""

AAMCU.TxtOfiCue = ""

AAMCU.TxtNumCue = ""

AAMCU.TxtDigVer = ""

AAMCU.TxtNomCue = ""

Sallistar:

Exit Sub

RutinaErrorListar:

MsgBox MSG_E001, 48, Error

Resume Sallistar

End Sub

Sub SelRegCue ()

GriCuentas.SelStartCol = 1

GriCuentas.SelEndCol = 2

GriCuentas.SelStartRow = GriCuentas.Row

GriCuentas.SelEndRow = GriCuentas.Row

GriCuentas.HighLight = True

GriCuentas.Col = 1

AAMCU.TxtProCue.Text = Mid\$(GriCuentas.Text, 1, 3)

AAMCU.TxtMonCue.Text = Mid\$(GriCuentas.Text, 4, 2)

AAMCU.TxtOfiCue.Text = Mid\$(GriCuentas.Text, 6, 3)

AAMCU.TxtNumCue.Text = Mid\$(GriCuentas.Text, 9, 6)

AAMCU.TxtDigVer.Text = Mid\$(GriCuentas.Text, 15, 1)

GriCuentas.Col = 2

AAMCU.TxtNomCue.Text = Trim\$(GriCuentas.Text) GriCuentas.Col = 0

End Sub

VI.3.4 LÓGICA ASOCIADA A LA VENTANA DE USUARIOS.**Sub Form_Load ()**

```

' restricciones usuario con nivel bajo
If GD_NivEnt = GC_Bajo Then
  DXMUS.GriUsuarios.Visible = False
  DXMUS.OrdAgregar.Visible = False
  DXMUS.OrdListar.Visible = False
  DXMUS.OrdGuardar.Visible = False
  DXMUS.OrdEliminar.Visible = False
  DXMUS.OrdImprimir.Visible = False
  DXMUS.Label1.Visible = False
  DXMUS.Label3.Visible = False
  DXMUS.ComNivUsu.Visible = False
  DXMUS.TxtCodUsu.Visible = False

```

```
End If
```

```
' Inicializa variables globales
```

```
GD_TotRegUsu = 0
```

```
' Agrega niveles de usuario a la lista
```

```
DXMUS.ComNivUsu.AddItem GC_Niv0
```

```
DXMUS.ComNivUsu.AddItem GC_Niv1
```

```
DXMUS.ComNivUsu.ListIndex = 0
```

```
' Formato del Grid
```

```
Call GP_IniciaGriUsuarios(GriUsuarios)
```

```
Call OrdListar_Click
```

```
End Sub
```

Sub GriUsuarios_Click ()

```
Call SelRegUsu
```

```
End Sub
```

Sub GriUsuarios_GotFocus ()

```
DXMUS.GriUsuarios.HighLight = True
```

```
End Sub
```

Sub GriUsuarios_LostFocus ()

```
DXMUS.GriUsuarios.HighLight = False
```

```
End Sub
```

```

Sub GriUsuarios_SelChange ()
  Call SelRegUsu
End Sub

```

```

Sub OrdAgregar_Click ()
  GD_Usuario = UCase$(RTrim$(TxtCodUsu.Text))
  GD_Nivel = Left$(ComNivUsu.Text, 1)
  ' Valida usuario
  If GD_Usuario = " " Or GD_Usuario = "" Then
    Beep
    MsgBox MSG_E002, 48, "ERROR"
    DXMUS.TxtCodUsu.SetFocus
    Exit Sub
  End If
  ' Valida número máximo de registros en la tabla
  If GD_TotRegUsu >= GC_MaxRegUsu Then
    Beep
    MsgBox MSG_E007, 48, "ERROR"
    Exit Sub
  End If
  ' Valida que no exista el registro
  For GD_Contador1 = 1 To GC_MaxRegUsu
    DXMUS.GriUsuarios.Col = 1
    DXMUS.GriUsuarios.Row = GD_Contador1
    If UCase$(RTrim$(DXMUS.GriUsuarios.Text)) = GD_Usuario Then
      Beep
      MsgBox MSG_E010, 48, "ERROR"
      DXMUS.TxtCodUsu.SetFocus
      Exit Sub
    End If
  Next GD_Contador1
  ' Agrega a la estructura
  GD_TotRegUsu = GD_TotRegUsu + 1
  GD_RegUsuario(GD_TotRegUsu).GD_NumUsu = Str$(GD_TotRegUsu)
  GD_RegUsuario(GD_TotRegUsu).GD_CodUsu = GD_Usuario
  GD_RegUsuario(GD_TotRegUsu).GD_NivUsu = GD_Nivel
  ' Actualiza Grid
  Call GP_ActualizaGriUsu
  DXMUS.TxtCodUsu.SetFocus
End Sub

```

```

Sub OrdCerrar_Click ()
  Unload DXMUS

```

End Sub

Sub OrdEliminar_Click ()

```

GD_Usuario = UCase$(RTrim$(TxtCodUsu.Text))
' Valida que exista el usuario a borrar
GD_Existe = False
For GD_Contador1 = 1 To GD_TotRegUsu
    DXMUS.GriUsuarios.Col = 1
    DXMUS.GriUsuarios.Row = GD_Contador1
    If UCase$(RTrim$(DXMUS.GriUsuarios.Text)) = GD_Usuario Then
        GD_Existe = True
        Exit For
    End If
Next GD_Contador1
If GD_Existe = False Then
    Beep
    MsgBox MSG_E011, 48, "ERROR"
    DXMUS.TxtCodUsu.SetFocus
    Exit Sub
End If
' Borra registro
For GD_Contador2 = GD_Contador1 To GD_TotRegUsu - 1
    GD_RegUsuario(GD_Contador2).GD_NumUsu =
    Str$(Val(GD_RegUsuario(GD_Contador2 + 1).GD_NumUsu) - 1)
    GD_RegUsuario(GD_Contador2).GD_CodUsu =
    GD_RegUsuario(GD_Contador2 + 1).GD_CodUsu
    GD_RegUsuario(GD_Contador2).GD_NivUsu =
    GD_RegUsuario(GD_Contador2 + 1).GD_NivUsu
    GD_RegUsuario(GD_Contador2).GD_ClvUsu =
    GD_RegUsuario(GD_Contador2 + 1).GD_ClvUsu
Next GD_Contador2
GD_RegUsuario(GD_TotRegUsu).GD_NumUsu = ""
GD_RegUsuario(GD_TotRegUsu).GD_CodUsu = ""
GD_RegUsuario(GD_TotRegUsu).GD_NivUsu = ""
GD_RegUsuario(GD_TotRegUsu).GD_ClvUsu = ""
GD_TotRegUsu = GD_TotRegUsu - 1
' Actualiza Grid
Call GP_ActualizaGriUsu
DXMUS.TxtCodUsu.SetFocus
End Sub

```

Sub OrdGuardar_Click ()

' Inicializa valores

```

GD_LongReg = Len(GD_RegUsuario(1))
On Error GoTo RutinaError
' Abre el archivo de usuarios
Open GC_ArchUsuarios For Random As #1 Len = GD_LongReg
For GD_CuentaRegs = 1 To GC_MaxRegUsu
    Put #1, , GD_RegUsuario(GD_CuentaRegs)
Next GD_CuentaRegs
Close #1
SalGuardar:
Exit Sub
RutinaError:
MsgBox MSG_E001, 48, Error
Resume SalGuardar
End Sub

```

```

Sub OrdListar_Click ()
' Inicializa valores
GD_LongReg = Len(GD_RegUsuario(1))
On Error GoTo RutinaErrorListar
' Abre el archivo de usuarios
Open GC_ArchUsuarios For Random As #1 Len = GD_LongReg
GD_NumReg = LOF(1) / GD_LongReg
For GD_CuentaRegs = 1 To GD_NumReg
    Get #1, , GD_RegUsuario(GD_CuentaRegs)
Next GD_CuentaRegs
Close #1
Call GP_ActualizaGriUsu
DXMUS.TxtCodUsu = ""
SalListar:
Exit Sub
RutinaErrorListar:
MsgBox MSG_E001, 48, Error
Resume SalListar
End Sub

```

```

Sub OrdModificar_Click ()
DAMCV.Show 1
End Sub

```

```

Sub SelRegUsu ()
' Selecciona todo el registro
GriUsuarios.SelStartCol = 1

```

```

GriUsuarios.SelEndCol = 2
GriUsuarios.SelStartRow = GriUsuarios.Row
GriUsuarios.SelEndRow = GriUsuarios.Row
GriUsuarios.HighLight = True
'Actualiza cajas de texto
GriUsuarios.Col = 1
DXMUS.TxtCodUsu.Text = GriUsuarios.Text
GriUsuarios.Col = 2
DXMUS.ComNivUsu.ListIndex = Val(Left$(GriUsuarios.Text, 1))
GriUsuarios.Col = 0
End Sub

```

VL3.5 LÓGICA ASOCIADA A LA VENTANA DE CLIENTE.

```

Sub BotAceptar_Click ()
'Inicializa valores
GD_LongReg = Len(GD_RegCliente)
GD_Cliente = UCase$(RTrim$(DXMCL.TxtCodCli.Text))
GD_Nombre = UCase$(RTrim$(DXMCL.TxtNomCli.Text))
'Valida código de cliente
If GD_Cliente = " " Or GD_Cliente = "" Then
    Beep
    MsgBox MSG_E015, 48, "ERROR"
    DXMCL.TxtCodCli.SetFocus
    Exit Sub
End If
'Valida nombre de cliente
If GD_Nombre = " " Or GD_Nombre = "" Then
    Beep
    MsgBox MSG_E016, 48, "ERROR"
    DXMCL.TxtNomCli.SetFocus
    Exit Sub
End If
'Agrega al registro
GD_RegCliente.GD_CodCli = GD_Cliente
GD_RegCliente.GD_NomCli = GD_Nombre
On Error GoTo RutinaError
'Abre el archivo de cliente
Open GC_ArchCliente For Random As #1 Len = GD_LongReg
Put #1, 1, GD_RegCliente
Close #1
DXMCL.TxtCodCli.SetFocus

```

```
SalGuardar:  
  Exit Sub  
RutinaError:  
  MsgBox MSG_E001, 48, Error  
  Resume SalGuardar  
End Sub
```

```
Sub BotCancelar_Click ()  
  Unload DXMCL  
End Sub
```

```
Sub Form_Load ()  
  ' Inicializa valores  
  GD_LongReg = Len(GD_RegCliente)  
  On Error GoTo RutinaErrorListar  
  ' Abre el archivo de cliente  
  Open GC_ArchCliente For Random As #1 Len = GD_LongReg  
  Get #1, 1, GD_RegCliente  
  Close #1  
  ' Presenta valores  
  If GD_RegCliente.GD_CodCli = " " Or GD_RegCliente.GD_NomCli = "" Then  
    DXMCL.TxtCodCli = ""  
    DXMCL.TxtNomCli = ""  
  Else  
    DXMCL.TxtCodCli = RTrim$(GD_RegCliente.GD_CodCli)  
    DXMCL.TxtNomCli = RTrim$(GD_RegCliente.GD_NomCli)  
  End If  
SalListar:  
  Exit Sub  
RutinaErrorListar:  
  MsgBox MSG_E001, 48, Error  
  Resume SalListar  
End Sub
```

VI.3.6 LÓGICA ASOCIADA A LA VENTANA DE SERVICIOS.**Sub Form_Load ()***' Inicializa variables globales*

GD_TotRegSer = 0

,

Call GP_IniciaGriServicios(GriServicios)

Call OrdListar_Click

End Sub**Sub Form_Unload (Cancel As Integer)**

MenuAdmin.Enabled = True

End Sub**Sub GriServicios_Click ()**

Call SelRegSer

End Sub**Sub GriServicios_GotFocus ()**

AAMSE.GriServicios.HighLight = True

End Sub**Sub GriServicios_LostFocus ()**

AAMSE.GriServicios.HighLight = False

End Sub**Sub GriServicios_SelChange ()**

Call SelRegSer

End Sub**Sub OrdAgregar_Click ()**

GD_Causa = RTrim\$(AAMSE.TxtCausal.Text)

GD_Nombre = UCase\$(RTrim\$(AAMSE.TxtNomCau.Text))

' Valida causa

If GD_Causa = " " Or GD_Causa = "" Then

Beep

```

    MsgBox MSG_E005, 48, "ERROR"
    AAMSE.TxtCausal.SetFocus
    Exit Sub
End If
Call GP_Valida_Numero(GD_Causa)
If GD_EsNumero = False Then
    Beep
    MsgBox MSG_E005, 48, "ERROR"
    AAMSE.TxtCausal.SetFocus
    Exit Sub
End If
' Valida Nombre de la causa
If GD_Nombre = " " Or GD_Nombre = "" Then
    Beep
    MsgBox MSG_E006, 48, "ERROR"
    AAMSE.TxtNomCau.SetFocus
    Exit Sub
End If
' Valida número máximo de registros en la tabla
If GD_TotRegSer >= GC_MaxRegSer Then
    Beep
    MsgBox MSG_E007, 48, "ERROR"
    Exit Sub
End If
' Valida que no exista el registro
For GD_Contador1 = 1 To GC_MaxRegSer
    AAMSE.GriServicios.Col = 1
    AAMSE.GriServicios.Row = GD_Contador1
    If RTrim$(AAMSE.GriServicios.Text) = GD_Causa Then
        Beep
        MsgBox MSG_E008, 48, "ERROR"
        AAMSE.TxtCausal.SetFocus
        Exit Sub
    End If
Next GD_Contador1
' Agrega a la estructura
GD_TotRegSer = GD_TotRegSer + 1
GD_RegServicio(GD_TotRegSer).GD_NumCau = Str$(GD_TotRegSer)
GD_RegServicio(GD_TotRegSer).GD_CodCau = GD_Causa
GD_RegServicio(GD_TotRegSer).GD_NomCau = GD_Nombre
' Actualiza Grid
Call GP_ActualizaGriSer
AAMSE.TxtCausal.SetFocus
End Sub

```

Sub OrdCerrar_Click ()

```

AAMSE.TxtCausal = " "
AAMSE.TxtNomCau = " "
AAMSE.TxtCausal.SetFocus
Unload AAMSE

```

End Sub**Sub OrdEliminar_Click ()**

```

GD_Causa = RTrim$(AAMSE.TxtCausal.Text)
' Valida que exista el codigo a borrar
GD_Existe = False
For GD_Contador1 = 1 To GD_TotRegSer
    AAMSE.GriServicios.Col = 1
    AAMSE.GriServicios.Row = GD_Contador1
    If RTrim$(AAMSE.GriServicios.Text) = GD_Causa Then
        GD_Existe = True
        Exit For
    End If
Next GD_Contador1
If GD_Existe = False Then
    Beep
    MsgBox MSG_E009, 48, "ERROR"
    AAMSE.TxtCausal.SetFocus
    Exit Sub

```

End If*' Borra registro*

```

For GD_Contador2 = GD_Contador1 To GD_TotRegSer - 1
    GD_RegServicio(GD_Contador2).GD_NumCau =
    Str$(Val(GD_RegServicio(GD_Contador2 + 1).GD_NumCau) - 1)
    GD_RegServicio(GD_Contador2).GD_CodCau =GD_RegServicio(GD_Contador2+
    1).GD_CodCau
    GD_RegServicio(GD_Contador2).GD_NomCau=GD_RegServicio(GD_Contador2+
    1).GD_NomCau

```

Next GD_Contador2

```

GD_RegServicio(GD_TotRegSer).GD_NumCau = ""

```

```

GD_RegServicio(GD_TotRegSer).GD_CodCau = ""

```

```

GD_RegServicio(GD_TotRegSer).GD_NomCau = ""

```

```

GD_TotRegSer = GD_TotRegSer - 1

```

' Actualiza Grid

```

Call GP_ActualizaGriSer

```

```

AAMSE.TxtCausal.SetFocus

```

```

AAMSE.TxtNomCau.Text = ""

```

End Sub

Sub OrdGuardar_Click ()*' Inicializa valores*

GD_LongReg = Len(GD_RegServicio(1))

On Error GoTo RutinaError

' Abre el archivo de servicios

Open GC_ArchServicios For Random As #1 Len = GD_LongReg

For GD_CuentaRegs = 1 To GC_MaxRegSer

Put #1, , GD_RegServicio(GD_CuentaRegs)

Next GD_CuentaRegs

Close #1

SalGuardar:

Exit Sub

RutinaError:

MsgBox MSG_E001, 48, Error

Resume SalGuardar

End Sub**Sub OrdListar_Click ()***' Inicializa valores*

GD_LongReg = Len(GD_RegServicio(1))

On Error GoTo RutinaErrorListar

' Abre el archivo de servicios

Open GC_ArchServicios For Random As #1 Len = GD_LongReg

GD_NumReg = LOF(1) / GD_LongReg

For GD_CuentaRegs = 1 To GD_NumReg

Get #1, , GD_RegServicio(GD_CuentaRegs)

Next GD_CuentaRegs

Close #1

Call GP_ActualizaGriSer

AAMSE.TxtCausal = ""

AAMSE.TxtNomCau = ""

SalListar:

Exit Sub

RutinaErrorListar:

MsgBox MSG_E001, 48, Error

Resume SalListar

End Sub**Sub SelRegSer ()***' Selecciona todo el registro*

GriServicios.SelStartCol = 1

GriServicios.SelEndCol = 2

GriServicios.SelStartRow = GriServicios.Row

```
GriServicios.SelEndRow = GriServicios.Row  
GriServicios.HighLight = True  
'Actualiza cajas de texto  
GriServicios.Col = 1  
AAMSE.TxtCausal.Text = RTrim$(GriServicios.Text)  
GriServicios.Col = 2  
AAMSE.TxtNomCau.Text = RTrim$(GriServicios.Text)  
GriServicios.Col = 0  
End Sub
```

VI.3.7 LÓGICA ASOCIADA AL MENÚ DE COMUNICACIONES.

Sub Form_Unload (Cancel As Integer)

```

GP_Pregunta_Salir Cancel, "Fin de la Sesión"
End Sub

```

Sub GP_AbreHost ()

```

Dim SD_RHost As String
Dim SD_RHost1 As String
' Inicia comunicación con el SERIE A
GD_Sesion = "1234567890123456789012345678901234567890"
OpenInfoConnect GD_Sesion, ""
GD_Mensaje.GD_Todo = "SFB SFB"
GP_SndHost
Do While True
  GP_RcvHost
  If GD_Error Then
    Screen.MousePointer = 0
    Exit Sub
  End If
  ' Conexión a SFB
  If Len(GD_Respuesta.GD_Todo) > 21 Then
    SD_RHost = Mid$(GD_Respuesta.GD_Todo, 7, 4)
    If SD_RHost = "MARC" Then
      If GD_Mensaje.GD_Todo = "ON SFBDEMO" Then
        MsgBox MSG_E020, 16, "ERROR"
        GD_Error = True
        Exit Sub
      End If
      GD_Mensaje.GD_Todo = "ON SFBDEMO DX430"
      GP_SndHost
    End If
    SD_RHost1 = Mid$(GD_Respuesta.GD_Todo, 1, 6)
    If RHost1 = "DX430T" Then
      Exit Do
    End If
    If GD_TranCod = "STATU" Then
      Exit Do
    End If
  End If
Loop
GD_Mensaje.GD_Todo = "?+I"
GP_SndHost
End Sub

```

Sub GP_Ctas_Predefinidas ()

```

' Carga cuentas predefinidas en el módulo de administración
GD_LongReg = Len(GD_RegCuenta(1))
'
On Error GoTo RutinaError
'
' Abre el archivo de cuentas
Open GC_ArchCuentas For Random As #1 Len = GD_LongReg
'
GD_NumReg = LOF(1) / GD_LongReg
For GD_CuentaRegs = 1 To GD_NumReg
    Get #1, , GD_RegCuenta(GD_CuentaRegs)
    GD_CuePre(GD_CuentaRegs) = GD_RegCuenta(GD_CuentaRegs).GD_NumCue
Next GD_CuentaRegs
Close #1
Salir:
Exit Sub
RutinaError:
MsgBox MSG_E001, 48, Error
Resume Salir
End Sub

```

Sub GP_Pregunta_Salir (Cancel As Integer, Titulo As String)

```

If MsgBox(MSG_E019, 52, Titulo) = GC_BotNo Then
    MenuComunica.OrdConexión.Enabled = True
    Cancel = -1
Else
    Screen.MousePointer = 11
    If GD_Anyconex Then
        GP_ClsHost
    End If
    Screen.MousePointer = 0
    If Not MenuComunica.OrdConexión.Enabled Then
        CloseInfoConnect (GD_Sesion)
    End If
    MenuPrincipal.Panel3D3.Visible = True ' Panel de seguridad visible
    MenuPrincipal.Panel3D1.Visible = False ' Panel menú invisible
    MenuPrincipal.TxtClvUsu = ""
    MenuPrincipal.TxtNomUsu = ""
End If
End Sub

```

Sub GP_Ser_Predefinidos ()

```
' Carga servicios predefinidos en el módulo de administración
```

```
GD_LongReg = Len(GD_RegServicio(1))
```

```
On Error GoTo RutinaErrorListar
```

```
' Abre el archivo de cuentas
```

```
Open GC_ArchServicios For Random As #1 Len = GD_LongReg
```

```
GD_NumReg = LOF(1) / GD_LongReg
```

```
For GD_CuentaRegs = 1 To GD_NumReg
```

```
    Get #1, , GD_RegServicio(GD_CuentaRegs)
```

```
    GD_SerPre(GD_CuentaRegs)=GD_RegServicio(GD_CuentaRegs).GD_CodCa  
    u & Chr(45) & GD_RegServicio(GD_CuentaRegs).GD_NomCau
```

```
Next GD_CuentaRegs
```

```
Close #1
```

```
SalListar:
```

```
Exit Sub
```

```
RutinaErrorListar:
```

```
MsgBox MSG_E001, 48, Error
```

```
Resume SalListar
```

```
End Sub
```

Sub OrdConexión_Click ()

```
DX430.Show 1
```

```
End Sub
```

Sub OrdDesconexión_Click ()

```
If MsgBox(MSG_E022, 52, "Desconexión del Sistema") = GC_BotSi Then
```

```
Screen.MousePointer = 11
```

```
GP_ClsHost
```

```
GD_Anyconex = Flase
```

```
Screen.MousePointer = 0
```

```
MenuComunica.Frame3D1.Visible = False
```

```
MenuComunica.OrdDesconexión.Enabled = False
```

```
MenuComunica.OrdFin.Enabled = True
```

```
End If
```

```
End Sub
```

Sub OrdFin_Click ()

```

If GD_Anyconex = True Then
  Beep
  MsgBox MSG_E023, 16, "ERROR"
  Exit Sub
End If
GD_telefono = GC_Colgar
Print #1, GD_telefono
Close #1
Unload MenuComunica
End Sub

```

Sub OrdLlamar_Click ()

```

Screen.MousePointer = 11
GD_telefono = GC_PulsoTono & GC_Telbanco
Open GC_Puerto For Output As #1
Print #1, GD_telefono
' Ya enlazado al MODEM Del banco
Call GP_AbreHost
MenuComunica.OrdLlamar.Enabled = False
MenuComunica.OrdConexión.Enabled = True
Screen.MousePointer = 0

```

End Sub**VI.3.8 LÓGICA ASOCIADA A LA VENTANA DE CONEXIÓN.****Sub BotCaConex_Click ()**

```

Unload DX430

```

End Sub**Sub BotOkConex_Click ()**

```

GD_CodCli = UCase$(Left$(DX430.TxtCodCli.Text, 10))
GD_Clave = UCase$(Left$(DX430.TxtClvUsu.Text, 10))

```

```

If GD_CodCli = " " Or GD_CodCli = "" Then

```

```

  Beep
  MsgBox MSG_E021, 16, "ERROR"
  DX430.TxtCodCli.SetFocus

```

```

Elseif GD_Clave = " " Or GD_Clave = "" Then

```

```

  Beep
  MsgBox MSG_E003, 16, "ERROR"

```

```

DX430.TxtCivUsu.SetFocus
Else
  Call GP_Deshabilitar
  Call GP_OpnHost
  Call GP_Habilitar
  If Not GD_Error Then
    Unload DX430
    MenuComunica.Frame3D1.Visible = True
    If GD_NivEnt = GC_Bajo Then
      MenuComunica.BotMensajes.Enabled = False
      MenuComunica.BotTransferencias.Enabled = False
    End If
    MenuComunica.OrdConexión.Enabled = False
    MenuComunica.OrdFin.Enabled = False
    MenuComunica.OrdDesconexión.Enabled = True
    GD_AnyConex = True
  End If
End If
End Sub

Sub Form_Load ()
  ' Busca código de cliente predefinido
  GD_LongReg = Len(GD_RegCliente)
  '
  On Error GoTo RutinaErrorListar
  '
  ' Abre el archivo de cliente
  Open GC_ArchCliente For Random As #1 Len = GD_LongReg
  Get #1, 1, GD_RegCliente
  Close #1
  '
  ' Presenta valores
  If GD_RegCliente.GD_CodCli = " " Then
    DX430.TxtCodCli = ""
  Else
    DX430.TxtCodCli = RTrim$(GD_RegCliente.GD_CodCli)
  End If

SalListar:
  Exit Sub
  '
RutinaErrorListar:

```

```
MsgBox MSG_E001, 48, Error  
Resume SalListar  
End Sub
```

```
Sub Form_Unload (Cancel As Integer)  
If Not GD_Error Then  
MenuComunica.Frame3D1.Visible = True  
End If  
End Sub
```

VI.3.9 LÓGICA ASOCIADA AL MENÚ DE CONSULTAS.

```
Sub OrdCerrar_Click ()  
Unload MenuConsultas  
End Sub
```

```
Sub OrdCuentas_Click ()  
BSCCC.Show 1  
End Sub
```

```
Sub OrdCheque_Click ()  
ACCSC.Show 1  
End Sub
```

```
Sub OrdChequera_Click ()  
ACCCH.Show 1  
End Sub
```

```
Sub OrdMovimiento_Click ()  
ACCEC.Show 1  
End Sub
```

```
Sub OrdRelaciones_Click ()  
    BSCRC.Show 1  
End Sub
```

```
Sub OrdSaldo_Click ()  
    ACCBA.Show 1  
End Sub
```

VI.3.10 LÓGICA ASOCIADA AL MENÚ DE TRANSFERENCIAS.

```
Sub BotMovFon_Click ()  
    AA300.Show 1  
End Sub
```

```
Sub BotPagSer_Click ()  
    AJ403.Show 1  
End Sub
```

```
Sub OrdCerrar_Click ()  
    Unload MenuTrans  
End Sub
```

VI.3.11 LÓGICA ASOCIADA A LA VENTANA DE CUENTAS DE CLIENTE.

```
Sub BotDetSal_Click ()  
    BSCCC.Hide  
    ACCBA.Show 1  
End Sub
```

```
Sub Form_Load ()  
,  
    Dim SD_Fila As Integer  
    Dim SD_Indice As Integer  
,  
    Call GP_Iniciar_GriCuentas
```

```

GD_Concepto = "00"
,
GD_OtroMsg = True
SD_Fila = 0
Do While GD_OtroMsg
    GP_EnviarMsg ("BSCCC")
    GP_RecibirRpta ("BSCCC")
    If GD_Error Then
        Call GP_Habilitar
        Exit Sub
    End If
    For SD_Indice = 1 To UBound(BSCCCResp.LINEACF)
        If Left$(BSCCCResp.LINEACF(SD_Indice).SD_CtaEdi, 3) = " " Then
            Exit For
        End If
        Call GP_Signo("S", BSCCCResp.LINEACF(SD_Indice).FILLER02,
            BSCCCResp.LINEACF(SD_Indice).BSaEfeHoy)
        GD_Elemento = Chr$(9) + GF_Traduc$(ComTipRel,
            BSCCCResp.LINEACF(SD_Indice).SCOTIRECC) + Chr$(9) +
            BSCCCResp.LINEACF(SD_Indice).SD_CtaEdi
        GD_Elemento = GD_Elemento + Chr$(9) +
            BSCCCResp.LINEACF(SD_Indice).BNoTipCue + Chr$(9) +
            BSCCCResp.LINEACF(SD_Indice).ANoEstCue
        GD_Elemento = GD_Elemento + Chr$(9) +
            GF_Formatee$(BSCCCResp.LINEACF(SD_Indice).FILLER02 +
            BSCCCResp.LINEACF(SD_Indice).BSaEfeHoy,
            "##,###,###,##0.00") + Chr$(9) +
            GF_EdFecha$(BSCCCResp.LINEACF(SD_Indice).BFEULTTRA)
        BSCCC.GriCuentas.AddItem GD_Elemento
    Next SD_Indice
Loop
End Sub

Sub GP_Iniciar_GriCuentas ()
,
    BSCCC.GriCuentas.Cols = 7
    BSCCC.GriCuentas.Rows = 3
,
    BSCCC.GriCuentas.RowHeight(2) = 1      'Fila Oculta
,
    BSCCC.GriCuentas.ColWidth(0) = 1       'Oculta (mula)
    BSCCC.GriCuentas.ColWidth(1) = 730    'Relación
    BSCCC.GriCuentas.ColWidth(2) = 1660   'Cuenta
    BSCCC.GriCuentas.ColWidth(3) = 1050   'Concepto

```

```

BSCCC.GriCuentas.ColWidth(4) = 1050 ' Estado
BSCCC.GriCuentas.ColWidth(5) = 1410 ' Saldo
BSCCC.GriCuentas.ColWidth(6) = 1050 ' Fecha Ult. Trans.
BSCCC.GriCuentas.ColAlignment(5) = 1
BSCCC.GriCuentas.ColAlignment(6) = 1

```

```

BSCCC.GriCuentas.Row = 0
BSCCC.GriCuentas.Col = 6
BSCCC.GriCuentas.Text = "Fecha ult."
BSCCC.GriCuentas.Row = 1
BSCCC.GriCuentas.Col = 1
BSCCC.GriCuentas.Text = "Relación"
BSCCC.GriCuentas.Col = 2
BSCCC.GriCuentas.Text = "Cuenta"
BSCCC.GriCuentas.Col = 3
BSCCC.GriCuentas.Text = "Concepto"
BSCCC.GriCuentas.Col = 4
BSCCC.GriCuentas.Text = "Estado"
BSCCC.GriCuentas.Col = 5
BSCCC.GriCuentas.Text = "Saldo"
BSCCC.GriCuentas.Col = 6
BSCCC.GriCuentas.Text = "Transacción"

```

```

BSCCC.GriCuentas.FixedRows = 2

```

End Sub

Sub GriCuentas_Click ()

```

' Selecciona la cuenta
BSCCC.GriCuentas.Col = 2
GD_Cuenta = Mid$(BSCCC.GriCuentas.Text, 1, 3) +
             Mid$(BSCCC.GriCuentas.Text, 5, 2) +
             Mid$(BSCCC.GriCuentas.Text, 8, 3) +
             Mid$(BSCCC.GriCuentas.Text, 12, 6) +
             Mid$(BSCCC.GriCuentas.Text, 19, 1)
GD_CueNav = GD_Cuenta
GD_CtaEdit = BSCCC.GriCuentas.Text

BSCCC.GriCuentas.Col = 0
BSCCC.GriCuentas.SelStartCol = 0
BSCCC.GriCuentas.SelEndCol = 6
BSCCC.GriCuentas.SelStartRow = BSCCC.GriCuentas.Row
BSCCC.GriCuentas.SelEndRow = BSCCC.GriCuentas.Row
BSCCC.GriCuentas.HighLight = True

```

```
BSCCC.BotDetSal.Visible = True  
BSCCC.BotCliRel.Visible = True  
End Sub
```

```
Sub OrdCerrar_Click ()  
  Unload BSCCC  
End Sub
```

VI.3.12 LÓGICA ASOCIADA A LA CONSULTA DE SALDOS.

```
Sub BotCerrar_Click ()  
  GD_CueNav = ACCBA.TxtNumCue.Text 'Navega  
  Unload ACCBA  
End Sub
```

```
Sub BotOk_Click ()  
  Dim SD_Elem1 As String * 18  
  Dim SD_Elem2 As String * 18  
  '  
  'Valida cuenta  
  GD_Cuenta = ACCBA.TxtNumCue  
  Call GP_Valida_Numero(GD_Cuenta)  
  If GD_EsNumero = False Then  
    Beep  
    MsgBox MSG_E012, 48, "ERROR"  
    ACCBA.TxtNumCue.SetFocus  
    Exit Sub  
  End If  
  '  
  'Envia mensaje al host  
  Call GP_Deshabilitar  
  GP_EnviarMsg ("ACCBA")  
  GP_RecibirRpta ("ACCBA")  
  If GD_ErrorCod Then  
    Call GP_Habilitar  
    Exit Sub  
  End If  
  '  
  'Presenta información  
  ACCBA.TxtNomCue.Text = ACCBAResp.CNO_CUENT  
  ACCBA.TxtNomEst.Text = ACCBAResp.ANo_Estado
```

```

ACCBA.TxtTipCue.Text = ACCBAResp.CGLTIPCUE
ACCBA.TxtVaCupSob.Text = GF_Formatee$(ACCBAResp.CVcCupSob,
    "###,###,##0.00")
ACCBA.TxtFeCupSob.Text = GF_EdFecha$(ACCBAResp.CFeCupSob)
ACCBA.TxtFeUltTra.Text = GF_EdFecha$(ACCBAResp.CFeUltTra)
Call GP_Signo("S", ACCBAResp.S1CInCauAct, ACCBAResp.CInCauAct)
ACCBA.TxtInCauAct.Text = GF_Formatee$(ACCBAResp.S1CInCauAct +
    ACCBAResp.CInCauAct, "##,###,##0.00")

```

```

Call GP_Iniciar_GriACCBA
Call GP_Signo("T", ACCBAResp.S1CSaTotHoy, ACCBAResp.CSaTotHoy)
Call GP_Signo("T", ACCBAResp.S1CMoTotDSi, ACCBAResp.CMoTotDSi)
Call GP_Signo("T", ACCBAResp.S1CSa_Reten, ACCBAResp.CSa_Reten)
Call GP_Signo("T", ACCBAResp.S1CSaCheHoy, ACCBAResp.CSaCheHoy)
Call GP_Signo("T", ACCBAResp.S1CSaCheDSi, ACCBAResp.CSaCheDSi)
Call GP_Signo("T", ACCBAResp.S1CSaATMHoy, ACCBAResp.CSaATMHoy)
Call GP_Signo("T", ACCBAResp.S1CMoATMDSi, ACCBAResp.CMoATMDSi)
Call GP_Signo("T", ACCBAResp.S1CSaEfeHoy, ACCBAResp.CSaEfeHoy)
Call GP_Signo("T", ACCBAResp.S1CMoEfeDSi, ACCBAResp.CMoEfeDSi)
Call GP_Signo("T", ACCBAResp.S1CSa_Prote, ACCBAResp.CSa_Prote)
Call GP_Signo("T", ACCBAResp.S1CSa_Prote1, ACCBAResp.CSa_Prote1)
Call GP_Signo("T", ACCBAResp.S1CSa_Dispo, ACCBAResp.CSa_Dispo)

```

```

GriACCBA.Row = 0
GriACCBA.Col = 1
GriACCBA.Text = " Hoy"
GriACCBA.Col = 2
GriACCBA.Text = " Mañana"

```

```

SD_Elem1 = GF_Formatee$(ACCBAResp.S1CSaTotHoy + ACCBAResp.CSaTotHoy,
    "##,###,###,##0.00")
SD_Elem2 = GF_Formatee$(ACCBAResp.S1CMoTotDSi + ACCBAResp.CMoTotDSi,
    "##,###,###,##0.00")
ACCBA.GriACCBA.AddItem " Saldo Contable" + Chr$(9) + SD_Elem1 + Chr$(9) +
    SD_Elem2, 2

```

```

SD_Elem1 = GF_Formatee$(ACCBAResp.S1CSa_Reten + ACCBAResp.CSa_Reten,
    "##,###,###,##0.00")
ACCBA.GriACCBA.AddItem " Saldo Retenido" + Chr$(9) + SD_Elem1, 3

```

```

SD_Elem1 = GF_Formatee$(ACCBAResp.S1CSaCheHoy + ACCBAResp.CSaCheHoy,
    "##,###,###,##0.00")
SD_Elem2 = GF_Formatee$(ACCBAResp.S1CSaCheDSi + ACCBAResp.CSaCheDSi,
    "##,###,###,##0.00")

```

```

ACCBA.GriACCBA.AddItem " Saldo Cheques Hoy" + Chr$(9) + SD_Elem1 +
    Chr$(9) + SD_Elem2, 4
,
SD_Elem1 = GF_Formatee$(ACCBAResp.SICSaATMHoy +
    ACCBAResp.CSaATMHoy, "##,###,###,##0.00")
SD_Elem2 = GF_Formatee$(ACCBAResp.SICMoATMDSi +
    ACCBAResp.CMoATMDSi, "##,###,###,##0.00")
ACCBA.GriACCBA.AddItem " Depósitos en ATM" + Chr$(9) + SD_Elem1 + Chr$(9) +
    SD_Elem2, 5
,
SD_Elem1 = GF_Formatee$(ACCBAResp.SICSaEfeHoy + ACCBAResp.CSaEfeHoy,
    "##,###,###,##0.00")
SD_Elem2 = GF_Formatee$(ACCBAResp.SICMoEfeDSi + ACCBAResp.CMoEfeDSi,
    "##,###,###,##0.00")
ACCBA.GriACCBA.AddItem " Saldo Efectivo" + Chr$(9) + SD_Elem1 + Chr$(9) +
    SD_Elem2, 6
,
SD_Elem1 = GF_Formatee$(ACCBAResp.SICSa_Prote + ACCBAResp.CSa_Prote,
    "##,###,###,##0.00")
SD_Elem2 = GF_Formatee$(ACCBAResp.SICSa_Protel + ACCBAResp.CSa_Protel,
    "##,###,###,##0.00")
ACCBA.GriACCBA.AddItem " Saldo Bloqueado" + Chr$(9) + SD_Elem1 + Chr$(9) +
    SD_Elem2, 7
,
SD_Elem1 = GF_Formatee$(ACCBAResp.SICSa_Dispo + ACCBAResp.CSa_Dispo,
    "##,###,###,##0.00")
ACCBA.GriACCBA.AddItem " Saldo Disponible" + Chr$(9) + SD_Elem1, 8
End Sub

```

```

Sub Form_Activate ()
    Panel3D2.Refresh
End Sub

```

```

Sub Form_Load ()

```

```

' Carga cuentas predefinidas módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_CuePre)
    Load ACCBA.MenuCuenta(GD_CuentaRegs)
    ACCBA.MenuCuenta(GD_CuentaRegs).Caption = GD_CuePre(GD_CuentaRegs)
    ACCBA.MenuCuenta(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs
' Cuenta navega

```

```
ACCBA.TxtNumCue.Text = GD_CueNav  
End Sub
```

```
Sub GP_Iniciar_GriACCBA ()  
GriACCBA.RowHeight(1) = 1  
GriACCBA.Rows = 2  
,  
GriACCBA.ColWidth(0) = 2100  
GriACCBA.ColWidth(1) = 2005  
GriACCBA.ColWidth(2) = 2105  
GriACCBA.ColWidth(3) = 2105  
GriACCBA.ColAlignment(1) = 1  
GriACCBA.ColAlignment(2) = 1  
End Sub
```

```
Sub MenuCuenta_Click (Index As Integer)  
ACCBA.TxtNumCue.Text = MenuCuenta(Index).Caption  
ACCBA.BotOk.SetFocus  
End Sub
```

VI.3.13 LÓGICA ASOCIADA A LA CONSULTA DE MOVIMIENTOS.

```
Sub BotCerrar_Click ()  
GD_CueNav = ACCEC.TxtNumCue.Text  
Unload ACCEC  
End Sub
```

```
Sub BotOk_Click ()  
Dim SD_Fila As Integer  
Dim SD_Indice As Integer  
,  
GD_Cuenta = ACCEC.TxtNumCue  
GD_FecIni = ACCEC.TxtFecIni  
GD_FecFin = ACCEC.TxtFecFin  
,  
' Valida cuenta  
Call GP_Valida_Numero(GD_Cuenta)  
If GD_EsNumero = False Then  
Beep
```

```

MsgBox MSG_E012, 48, "ERROR"
ACCEC.TxtNumCue.SetFocus
Exit Sub
End If
'
' Valida Fechas
Call GP_Valida_Numero(GD_FecIni)
If GD_EsNumero = False Then
    Beep
    MsgBox MSG_E027, 48, "ERROR"
    ACCEC.TxtFecIni.SetFocus
    Exit Sub
End If
Call GP_Valida_Numero(GD_FecFin)
If GD_EsNumero = False Then
    Beep
    MsgBox MSG_E027, 48, "ERROR"
    ACCEC.TxtFecFin.SetFocus
    Exit Sub
End If
'
' Envía y recibe información
GD_OtroMsg = True
SD_Fila = 0
Do While GD_OtroMsg
    GP_EnviarMsg ("ACCEC")
    GP_RecibirRpta ("ACCEC")
    If GD_Error Then
        Call GP_Habilitar
        Exit Sub
    End If
    ACCEC.TxtNomCue = ACCECResp.CNO_CUENT
    For SD_Indice = 1 To UBound(ACCECResp.LINEACF)
        If Left$(ACCECResp.LINEACF(SD_Indice).DCO_OFICI, 3) = "  " Then
            Exit For
        End If
        GD_Elemento = Chr$(9) + ACCECResp.LINEACF(SD_Indice).DCO_OFICI
        GD_Elemento = GD_Elemento + Chr$(9) +
            GF_EdFecha$(ACCECResp.LINEACF(SD_Indice).TFE_TRANS)
        GD_Elemento = GD_Elemento + Chr$(9) +
            ACCECResp.LINEACF(SD_Indice).DCO_TRANS
        GD_Elemento=GD_Elemento+Chr$(9)+
            ACCECResp.LINEACF(SD_Indice).TNUDOCTRA
        GD_Elemento = GD_Elemento + Chr$(9) +
            GF_Formatee$(ACCECResp.LINEACF(SD_Indice).TVA_MOVIM,

```

```

        "###,###,###,##0.00")
    GD_Elemento = GD_Elemento + Chr$(9) +
        ACCECResp.LINEACF(SD_Indice).DCODEBCRE
    GD_Elemento = GD_Elemento + Chr$(9) +
        ACCECResp.LINEACF(SD_Indice).TSE_SUPER1
    ACCEC.GriCuentas.AddItem GD_Elemento
Next SD_Indice
Loop
End Sub

```

Sub Form_Load ()

```

' Carga cuentas predefinidas módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_CuePre)
    Load ACCEC.MenuCuenta(GD_CuentaRegs)
    ACCEC.MenuCuenta(GD_CuentaRegs).Caption = GD_CuePre(GD_CuentaRegs)
    ACCEC.MenuCuenta(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs

Call GP_Iniciar_GriMovimientos

' Navega Cuenta
ACCEC.TxtNumCue.Text = GD_CueNav
End Sub

```

Sub GP_Iniciar_GriMovimientos ()

```

ACCEC.GriMovimientos.Cols = 8
ACCEC.GriMovimientos.Rows = 3

ACCEC.GriMovimientos.RowHeight(2) = 1 ' Fila Oculta

ACCEC.GriMovimientos.ColWidth(0) = 1 ' Oculta (nula)
ACCEC.GriMovimientos.ColWidth(1) = 945 ' Sucursal
ACCEC.GriMovimientos.ColWidth(2) = 720 ' Fecha
ACCEC.GriMovimientos.ColWidth(3) = 1170 ' Código transacción
ACCEC.GriMovimientos.ColWidth(4) = 1110 ' Número documento
ACCEC.GriMovimientos.ColWidth(5) = 1545 ' Monto
ACCEC.GriMovimientos.ColWidth(6) = 840 ' Código debito / credito
ACCEC.GriMovimientos.ColWidth(7) = 1155 ' Señal de supervisión

ACCEC.GriMovimientos.ColAlignment(2) = 1
ACCEC.GriMovimientos.ColAlignment(5) = 1

```

```
ACCEC.GriMovimientos.ColAlignment(6) = 2
ACCEC.GriMovimientos.ColAlignment(7) = 2
,
ACCEC.GriMovimientos.Row = 0
ACCEC.GriMovimientos.Col = 7
ACCEC.GriMovimientos.Text = "Señal de"
ACCEC.GriMovimientos.Col = 6
ACCEC.GriMovimientos.Text = "Débito/"
ACCEC.GriMovimientos.Col = 3
ACCEC.GriMovimientos.Text = "Código de"
ACCEC.GriMovimientos.Row = 1
ACCEC.GriMovimientos.Col = 1
ACCEC.GriMovimientos.Text = "Sucursal"
ACCEC.GriMovimientos.Col = 2
ACCEC.GriMovimientos.Text = "Fecha"
ACCEC.GriMovimientos.Col = 3
ACCEC.GriMovimientos.Text = "Transacción"
ACCEC.GriMovimientos.Col = 4
ACCEC.GriMovimientos.Text = "Documento"
ACCEC.GriMovimientos.Col = 5
ACCEC.GriMovimientos.Text = "Monto"
ACCEC.GriMovimientos.Col = 6
ACCEC.GriMovimientos.Text = "Crédito"
ACCEC.GriMovimientos.Col = 7
ACCEC.GriMovimientos.Text = "Supervisión"
,
ACCEC.GriMovimientos.FixedRows = 2
End Sub

Sub MenuCuenta_Click (Index As Integer)
    ACCEC.TxtNumCue.Text = MenuCuenta(Index).Caption
    ACCEC.BotOk.SetFocus
End Sub
```

VI.3.14 LÓGICA ASOCIADA A LA CONSULTA DE CHEQUERAS.

```
Sub BotCerrar_Click ()
    GD_CueNav = ACCCH.TxtNumCue.Text
    Unload ACCCH
End Sub
```

Sub BotOk_Click ()

```

Dim SD_Fila As Integer
Dim SD_Indice As Integer
GD_Cuenta = ACCCH.TxtNumCue
GD_NumChe = ACCCH.TxtPriChe
'
' Valida cuenta
Call GP_Valida_Numero(GD_Cuenta)
If GD_EsNumero = False Then
    Beep
    MsgBox MSG_E012, 48, "ERROR"
    ACCCH.TxtNumCue.SetFocus
    Exit Sub
End If
'
' Valida cheque
Call GP_Valida_Numero(GD_NumChe)
If GD_EsNumero = False Then
    Beep
    MsgBox MSG_E028, 48, "ERROR"
    ACCCH.TxtNumCue.SetFocus
    Exit Sub
End If
'
' Envía y recibe información
GD_OtroMsg = True
SD_Fila = 0
Do While GD_OtroMsg
    GP_EnviarMsg ("ACCCH")
    GP_RecibirRpta ("ACCCH")
    If GD_Error Then
        Call GP_Habilitar
        Exit Sub
    End If
    ACCCH.TxtNomCue = ACCCHResp.CNO_CUENT
    For SD_Indice = 1 To UBound(ACCCHResp.LINEACF)
        If Left$(ACCCHResp.LINEACF(SD_Indice).CNUPRIDOC, 3) = " " Then
            Exit For
        End If
        GD_Elemento = Chr$(9) + ACCCHResp.LINEACF(SD_Indice).CNUPRIDOC
        GD_Elemento = Chr$(9) + ACCCHResp.LINEACF(SD_Indice).CNUULTCHE
        GD_Elemento = GD_Elemento + Chr$(9) +
            GF_EdFecha$(ACCCHResp.LINEACF(SD_Indice).CFEENTCHE)
        GD_Elemento = GD_Elemento + Chr$(9) +
            GF_EdFecha$(ACCCHResp.LINEACF(SD_Indice).CFEULTPAG)
    Next SD_Indice

```

```

GD_Elemento = GD_Elemento + Chr$(9) +
    ACCCHResp.LINEACF(SD_Indice).CCN_CHEQU
GD_Elemento = GD_Elemento + Chr$(9) +
    ACCCHResp.LINEACF(SD_Indice).CNO_ESTAD
ACCCH.GriCuentas.AddItem GD_Elemento
Next SD_Indice
Loop
End Sub

Sub Form_Load ()
' Carga cuentas predefinidas módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_CuePre)
    Load ACCCH.MenuCuenta(GD_CuentaRegs)
    ACCCH.MenuCuenta(GD_CuentaRegs).Caption = GD_CuePre(GD_CuentaRegs)
    ACCCH.MenuCuenta(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs
'
Call GP_Iniciar_Grichequeras
'
' Navega Cuenta
ACCCH.TxtNumCue.Text = GD_CueNav
End Sub

```

Sub GP_Iniciar_Grichequeras ()

```

ACCCH.GriChequeras.Cols = 7
ACCCH.GriChequeras.Rows = 3
'
ACCCH.GriChequeras.RowHeight(2) = 1    ' Fila Oculta
'
ACCCH.GriChequeras.ColWidth(0) = 1    ' Oculta (nula)
ACCCH.GriChequeras.ColWidth(1) = 1110 ' Primer cheque
ACCCH.GriChequeras.ColWidth(2) = 1020 ' Ultimo Cheque
ACCCH.GriChequeras.ColWidth(3) = 1020 ' Fecha
ACCCH.GriChequeras.ColWidth(4) = 1155 ' Fecha último pago
ACCCH.GriChequeras.ColWidth(5) = 1095 ' Cantidad
ACCCH.GriChequeras.ColWidth(6) = 1860 ' Estado
'
ACCCH.GriChequeras.ColAlignment(1) = 1
ACCCH.GriChequeras.ColAlignment(2) = 1
ACCCH.GriChequeras.ColAlignment(3) = 1
ACCCH.GriChequeras.ColAlignment(4) = 1

```

```

ACCCH.GriChequeras.ColAlignment(5) = 1
,
ACCCH.GriChequeras.Row = 0
ACCCH.GriChequeras.Col = 1
ACCCH.GriChequeras.Text = "Primer"
ACCCH.GriChequeras.Col = 2
ACCCH.GriChequeras.Text = "Ultimo"
ACCCH.GriChequeras.Col = 3
ACCCH.GriChequeras.Text = "Fecha"
ACCCH.GriChequeras.Col = 4
ACCCH.GriChequeras.Text = "Fecha"
,
ACCCH.GriChequeras.Row = 1
ACCCH.GriChequeras.Col = 1
ACCCH.GriChequeras.Text = "Cheque"
ACCCH.GriChequeras.Col = 2
ACCCH.GriChequeras.Text = "Cheque"
ACCCH.GriChequeras.Col = 3
ACCCH.GriChequeras.Text = "Entrega"
ACCCH.GriChequeras.Col = 4
ACCCH.GriChequeras.Text = "Ultimo pago"
ACCCH.GriChequeras.Col = 5
ACCCH.GriChequeras.Text = " Cantidad"
ACCCH.GriChequeras.Col = 6
ACCCH.GriChequeras.Text = " Estado"
,
ACCCH.GriChequeras.FixedRows = 2
End Sub

Sub MenuCuenta_Click (Index As Integer)
    ACCCH.TxtNumCue.Text = MenuCuenta(Index).Caption
    ACCCH.BotOk.SetFocus
End Sub

```

VI.3.15 LÓGICA ASOCIADA A LA CONSULTA DE CHEQUES.

```

Sub BotCerrar_Click ()
    GD_NavCue = ACCSC.TxtNumCue.Text
    Unload ACCSC
End Sub

```

Sub BotOk_Click ()

```

Dim SD_Fila As Integer
Dim SD_Indice As Integer
GD_Cuenta = ACCSC.TxtNumCue
GD_NumChe = ACCSC.TxtNumChe
'
' Valida cuenta
Call GP_Valida_Numero(GD_Cuenta)
If GD_EsNumero = False Then
    Beep
    MsgBox MSG_E012, 48, "ERROR"
    ACCSC.TxtNumCue.SetFocus
    Exit Sub
End If
'
' Valida cheque
Call GP_Valida_Numero(GD_NumChe)
If GD_EsNumero = False Then
    Beep
    MsgBox MSG_E028, 48, "ERROR"
    ACCSC.TxtNumChe.SetFocus
    Exit Sub
End If
'
' Envía y recibe información
GD_OtroMsg = True
SD_Fila = 0
Do While GD_OtroMsg
    GP_EnviarMsg ("ACCSC")
    GP_RecibirRpta ("ACCSC")
    If GD_Error Then
        Call GP_Habilitar
        Exit Sub
    End If
    ACCSC.TxtNomCue = ACCSCResp.CNO_CUENT
    For SD_Indice = 1 To UBound(ACCSCResp.LINEACF)
        If Left$(ACCSCResp.LINEACF(SD_Indice).CNUCHEQUE, 3) = " " Then
            Exit For
        End If
        GD_Elemento = Chr$(9) + ACCSCResp.LINEACF(SD_Indice).CNUCHEQUE
        GD_Elemento = GD_Elemento + Chr$(9) +
            ACCSCResp.LINEACF(SD_Indice).CCOESTCHE
        GD_Elemento = GD_Elemento + Chr$(9) +
            ACCSCResp.LINEACF(SD_Indice).CNOESTCHE
        ACCSC.GriCuentas.AddItem GD_Elemento
    Next SD_Indice
Loop

```

```

Next SD_Indice
Loop
End Sub

```

Sub Form_Load ()

```

' Carga cuentas predefinidas módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_CuePre)
  Load ACCSC.MenuCuenta(GD_CuentaRegs)
  ACCSC.MenuCuenta(GD_CuentaRegs).Caption = GD_CuePre(GD_CuentaRegs)
  ACCSC.MenuCuenta(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs

Call GP_Iniciar_GriCheques

' Navega Cuenta
ACCSC.TxtNumCue.Text = GD_CueNav
End Sub

```

Sub GP_Iniciar_GriCheques ()

```

ACCSC.GriCheques.Cols = 4
ACCSC.GriCheques.Rows = 3
ACCSC.GriCheques.RowHeight(2) = 1      ' Fila Oculta
ACCSC.GriCheques.ColWidth(0) = 1      ' Oculta (mula)
ACCSC.GriCheques.ColWidth(1) = 1305   ' Número de cheque
ACCSC.GriCheques.ColWidth(2) = 1635   ' Código de estado
ACCSC.GriCheques.ColWidth(3) = 2925   ' Descripción
ACCSC.GriCheques.ColAlignment(1) = 1
ACCSC.GriCheques.ColAlignment(2) = 2
ACCSC.GriCheques.Row = 0
ACCSC.GriCheques.Col = 1
ACCSC.GriCheques.Text = " Número de"
ACCSC.GriCheques.Row = 1
ACCSC.GriCheques.Col = 1
ACCSC.GriCheques.Text = " Cheque"
ACCSC.GriCheques.Col = 2
ACCSC.GriCheques.Text = " Estado"
ACCSC.GriCheques.Col = 3
ACCSC.GriCheques.Text = " Descripción"
ACCSC.GriCheques.FixedRows = 2
End Sub

```

```

Sub MenuCuenta_Click (Index As Integer)
    ACCSC.TxtNumCue.Text = MenuCuenta(Index).Caption
    ACCSC.BotOk.SetFocus
End Sub

```

VI.3.16 LÓGICA ASOCIADA A LA CONSULTA DE CLIENTES RELACIONADOS.

```

Sub BotCerrar_Click ()
    Unload BSCRC
End Sub

```

```

Sub BotOk_Click ()
    Dim SD_Fila As Integer
    Dim SD_Indice As Integer
    ,
    ' Valida cuenta
    GD_Cuenta = BSCRC.TxtNumCue
    Call GP_Valida_Numero(GD_Cuenta)
    If GD_EsNumero = False Then
        Beep
        MsgBox MSG_E012, 48, "ERROR"
        BSCRC.TxtNumCue.SetFocus
        Exit Sub
    End If
    ,
    ' Recibe información
    GD_Concepto = "00"
    ,
    GD_OtroMsg = True
    SD_Fila = 0
    Do While GD_OtroMsg
        GP_EnviarMsg ("BSCRC")
        GP_RecibirRpta ("BSRCC")
        If GD_Error Then
            Call GP_Habilitar
            Exit Sub
        End If
        For SD_Indice = 1 To UBound(BSCRCResp.LINEACF)
            If Left$(BSCRCResp.LINEACF(SD_Indice).SCO_IDENT, 3) = " " Then
                Exit For
            End If
        Next SD_Indice
    Loop

```

```

GD_Elemento = Chr$(9) + BSCRResp.LINEACF(SD_Indice).SCO_IDENT +
              Chr$(9) + BSCRResp.LINEACF(SD_Indice).SCOTIRECC
GD_Elemento = GD_Elemento + Chr$(9) +
              BSCRResp.LINEACF(SD_Indice).SNO_CLIEN
BSRCC.GriClientes.AddItem GD_Elemento
Next SD_Indice
Loop
End Sub

```

Sub Form_Load ()

```

' Carga cuentas predefinidas módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_CuePre)
  Load BSCR.MenuCuenta(GD_CuentaRegs)
  BSCR.MenuCuenta(GD_CuentaRegs).Caption = GD_CuePre(GD_CuentaRegs)
  BSCR.MenuCuenta(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs

Call GP_Iniciar_GriClientes

' Navega Cuenta
BSCR.TxtNumCue.Text = GD_CueNav
End Sub

```

Sub GP_Iniciar_GriClientes ()

```

BSCR.GriClientes.Cols = 4
BSCR.GriClientes.Rows = 3
BSCR.GriClientes.RowHeight(2) = 1      ' Fila Oculta
BSCR.GriClientes.ColWidth(0) = 1      ' Oculta (mula)
BSCR.GriClientes.ColWidth(1) = 1830   ' Código del cliente
BSCR.GriClientes.ColWidth(2) = 3825   ' Nombre del cliente
BSCR.GriClientes.ColWidth(3) = 1770   ' Relación
BSCR.GriClientes.ColAlignment(1) = 1
BSCR.GriClientes.ColAlignment(2) = 2
BSCR.GriClientes.Row = 0
BSCR.GriClientes.Col = 1
BSCR.GriClientes.Text = " Código del"
BSCR.GriClientes.Row = 1
BSCR.GriClientes.Col = 1
BSCR.GriClientes.Text = " Cliente"
BSCR.GriClientes.Col = 2
BSCR.GriClientes.Text = " Nombre"
BSCR.GriClientes.Col = 3

```

```

BSCRC.GriClientes.Text = "    Relación"
BSCRC.GriClientes.FixedRows = 2
End Sub

```

```

Sub MenuCuenta_Click (Index As Integer)
    BSCRC.TxtNumCue.Text = MenuCuenta(Index).Caption
    BSCRC.BotOk.SetFocus
End Sub

```

VI.3.17 LÓGICA ASOCIADA A LA TRANSFERENCIA DE FONDOS.

```

Sub BotCerrar_Click ()
    Unload AA300
End Sub

```

```

Sub BotOk_Click ()
    GD_Cuenta1 = AA300.TxtCtaEmi.Text
    GD_Cuenta2 = AA300.TxtCtaRec.Text
    GD_ValMov = AA300.TxtValMov.Text
    '
    ' Valida cuentas
    Call GP_Valida_Numero(GD_Cuenta1)
    If (GD_EsNumero = False) Or (Len(GD_Cuenta1) <> 15) Then
        Beep
        MsgBox MSG_E012, 48, "ERROR"
        AA300.TxtCtaEmi.SetFocus
        Exit Sub
    End If
    '
    Call GP_Valida_Numero(GD_Cuenta2)
    If (GD_EsNumero = False) Or (Len(GD_Cuenta2) <> 15) Then
        Beep
        MsgBox MSG_E012, 48, "ERROR"
        AA300.TxtCtaRec.SetFocus
        Exit Sub
    End If
    '
    ' Cuentas deben ser diferentes
    If GD_Cuenta1 = GD_Cuenta2 Then
        Beep
        MsgBox MSG_E025, 48, "ERROR"
        AA300.TxtCtaEmi.SetFocus
    End If

```

```

Exit Sub
End If
'
' Valida monto
Call GP_Valida_Monto(AA300.TxtValMov)
If GD_HayErr = True Then Exit Sub
'
' Envia mensaje al host
Call GP_Deshabilitar
GP_EnviarMsg ("AA300")
GP_RecibirRpta ("AA300")
If GD_Error Then
    Call GP_Habilitar
    Exit Sub
End If
End Sub

Sub Form_Load ()
'
' Carga cuentas predefinidas módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_CuePre)
    Load AA300.MenuCuenta(GD_CuentaRegs)
    AA300.MenuCuenta(GD_CuentaRegs).Caption = GD_CuePre(GD_CuentaRegs)
    AA300.MenuCuenta(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs
End Sub

Sub MenuCuenta_Click (Index As Integer)
Static SD_Turno As Integer
'
If SD_Turno Then
    AA300.TxtCtaRec.Text = MenuCuenta(Index).Caption
    AA300.TxtValMov.SetFocus
    SD_Turno = False
Else
    AA300.TxtCtaEmi.Text = MenuCuenta(Index).Caption
    AA300.TxtCtaRec.SetFocus
    SD_Turno = True
End If
End Sub

```

VI.3.18 LÓGICA ASOCIADA AL PAGO DE SERVICIOS.**Sub BotCerrar_Click ()**

```
GD_CueNav = AJ403.TxtNumCue.Text
Unload AJ403
```

```
End Sub
```

Sub BotOk_Click ()

```
GD_Cuenta = AJ403.TxtNumCue.Text
GD_Causa = AJ403.TxtCodSer.Text
GD_ValMov = AJ403.TxtValMov.Text
```

```
' Valida cuenta
```

```
Call GP_Valida_Numero(GD_Cuenta)
```

```
If GD_EsNumero = False Then
```

```
    Beep
```

```
    MsgBox MSG_E012, 48, "ERROR"
```

```
    AJ403.TxtNumCue.SetFocus
```

```
    Exit Sub
```

```
End If
```

```
' Valida código servicio
```

```
Call GP_Valida_Numero(GD_Causa)
```

```
If GD_EsNumero = False Then
```

```
    Beep
```

```
    MsgBox MSG_E015, 48, "ERROR"
```

```
    AJ403.TxtCodSer.SetFocus
```

```
    Exit Sub
```

```
End If
```

```
' Valida monto
```

```
Call GP_Valida_Monto(AJ403.TxtValMov)
```

```
If GD_HayErr = True Then Exit Sub
```

```
' Envía mensaje al host
```

```
Call GP_Deshabilitar
```

```
GP_EnviarMsg ("AJ403")
```

```
GP_RecibirRpta ("AJ403")
```

```
If GD_Error Then
```

```
    Call GP_Habilitar
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

Sub Form_Load ()

```

' Carga cuentas predefinidas módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_CuePre)
  Load AJ403.MenuCuenta(GD_CuentaRegs)
  AJ403.MenuCuenta(GD_CuentaRegs).Caption = GD_CuePre(GD_CuentaRegs)
  AJ403.MenuCuenta(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs

' Carga códigos de servicios predefinidos en el módulo de admin.
For GD_CuentaRegs = 1 To UBound(GD_SerPre)
  Load AJ403.MenuServicio(GD_CuentaRegs)
  AJ403.MenuServicio(GD_CuentaRegs).Caption = GD_SerPre(GD_CuentaRegs)
  AJ403.MenuServicio(GD_CuentaRegs).Visible = True
Next GD_CuentaRegs

' Navega Cuenta
AJ403.TxtNumCue = GD_CueNav
End Sub

```

Sub MenuCuenta_Click (index As Integer)

```

AJ403.TxtNumCue.Text = MenuCuenta(index).Caption
AJ403.TxtCodSer.SetFocus
End Sub

```

Sub MenuServicio_Click (index As Integer)

```

AJ403.TxtCodSer.Text = Left$(AJ403.MenuServicio(index).Caption, 3)
AJ403.TxtValMov.SetFocus
End Sub

```

VI.3.19 LÓGICA ASOCIADA AL ENVÍO DE MENSAJES.**Sub BotCerrar_Click ()**

```

Unload AATME
End Sub

```

Sub BotEnvMsg_Click: ()

```

GD_Bandera1 = 1

```

```

' Envía mensaje al host

```

```

Call GP_Deshabilitar
GP_EnviarMsg ("AATME")
GP_RecibirRpta ("AATME")
If GD_Error Then
    Call GP_Habilitar
    Exit Sub
End If
End Sub

```

```

Sub BotRecMsg_Click ()
    GD_Bandera1 = 2
    ' Envía mensaje al host
    Call GP_Deshabilitar
    GP_EnviarMsg ("AATME")
    GP_RecibirRpta ("AATME")
    If GD_Error Then
        Call GP_Habilitar
        Exit Sub
    End If
End Sub

```

VI.3.20 PROCEDIMIENTOS Y FUNCIONES GLOBALES.

Mensajes.

```

' *****
' *   Mensajes de error propios de BANCEMP   *
' *****
Global Const MSG_E001 = " Error al abrir el Archivo "
Global Const MSG_E002 = " Debe especificar Usuario "
Global Const MSG_E003 = " Debe especificar Clave "
Global Const MSG_E004 = " Registro ya existe "
Global Const MSG_E005 = " Código de causa incorrecto "
Global Const MSG_E006 = " Descripción Incorrecta "
Global Const MSG_E007 = " Numero Máximo de registros excedido "
Global Const MSG_E008 = " Causal ya Existe "
Global Const MSG_E009 = " Causal No Existe "
Global Const MSG_E010 = " Usuario ya Existe "
Global Const MSG_E011 = " Usuario No Existe "
Global Const MSG_E012 = " Número de cuenta incorrecto "
Global Const MSG_E013 = " Número de cuenta ya existe "
Global Const MSG_E014 = " Número de cuenta No existe "

```

Global Const MSG_E015 = " Código incorrecto"
 Global Const MSG_E016 = " Nombre incorrecto"
 Global Const MSG_E017 = " Clave Incorrecta"
 Global Const MSG_E018 = " Time Out"
 Global Const MSG_E019 = " ¿Desea finalizar la sesión ?"
 Global Const MSG_E020 = " SFB no está disponible"
 Global Const MSG_E021 = " Debe especificar Cliente"
 Global Const MSG_E022 = " ¿Desea desconectarse del sistema?"
 Global Const MSG_E023 = " Desconectarse del sistema antes"
 Global Const MSG_E024 = " Intentos de conexión excedido"
 Global Const MSG_E025 = " Cuentas deben ser diferentes"
 Global Const MSG_E026 = " Monto contiene caracteres inválidos"
 Global Const MSG_E027 = " Fecha Incorrecta"
 Global Const MSG_E028 = " Número Incorrecto"

Validaciones y ediciones.

Function GF_Chr_Numero (GD_Caracter As String)

```

GF_Chr_Numero = True
,
If (Asc(GD_Caracter) < 48) Or (Asc(GD_Caracter) > 57) Then ' No es número
  If Asc(GD_Caracter) < 46 Then ' No es punto
    GF_Chr_Numero = False
  End If
End If
End Function
  
```

Function GF_EdFecha\$ (Fecha As String)

```

GF_EdFecha$ = Mid$(Fecha, 1, 2) + "/" + Mid$(Fecha, 3, 2) + "/" + Mid$(Fecha, 5, 2)
End Function
  
```

Function GF_Formatee\$ (Valor As String, Formato As String)

```

' Esta función debe usarse sólo para datos
' de SFB que se requiere poner en pantalla.
If UNIX Then
  If InStr(Valor, ".") Or InStr(Valor, ",") Then
    GF_Formatee$ = Valor
  Else
    GF_Formatee$ = Format$(Valor, Formato)
  End If
End If
  
```

```

If GC_SERIEA Then
  If InStr(Formato, ".") Then
    GF_Formatee$ = Format$(GF_ValorComa(Valor) / 100, Formato)
  Else
    GF_Formatee$ = Format$(GF_ValorComa(Valor), Formato)
  End If
End If
End Function

```

Function GF_ValorComa (S As String) As Double

```

Do While True
  J = InStr(S, ".")
  If J = 0 Then
    Exit Do
  Else
    Mid$(S, J, 1) = " "
  End If
Loop
I = InStr(S, ",")
If I = 0 Then
  Else
    Mid$(S, I, 1) = "."
  End If
GF_ValorComa = Val(S)
End Function

```

Sub GP_Valida_Monto (GD_Monto As Control)

```

Dim SD_NumCar As Integer
Dim SD_Caracter As String * 1
'
' Para edición
Dim SD_LonCmp As Integer
Dim SD_PosPto As Integer
Dim SD_MonFin As String
Dim SD_Entero As String
Dim SD_Decimal As String
'
' Inicia variables
SD_ConPto = 0
SD_LonCmp = GD_Monto.MaxLength ' Longitud del monto
GD_Monto = Trim$(GD_Monto) ' Elimina blancos
SD_PosPto = InStr(GD_Monto, ".") ' Posición del punto
SD_MonFin = GD_Monto ' Inicia monto final

```

```

GD_HayErr = False
'
' Valida sólo caracteres numéricos y .
For SD_NumCar = 1 To Len(GD_Monto)
  SD_Caracter = Mid$(GD_Monto, SD_NumCar, 1)
  If Not GF_Chr_Numero(SD_Caracter) Then
    Beep
    MsgBox MSG_E026, 46, "ERROR"
    GD_Monto.SetFocus
    GD_HayErr = True
    Exit Sub
  End If
Next SD_NumCar
'
' Sólo debe haber un punto decimal
SD_Decimal = Mid$(GD_Monto, SD_PosPto + 1, Len(GD_Monto) - SD_PosPto)
If (InStr(SD_Decimal, ".")) <> 0 Then
  Beep
  MsgBox MSG_E026, 46, "ERROR"
  GD_Monto.SetFocus
  GD_HayErr = True
  Exit Sub
End If
'
' Edita monto
If Len(GD_Monto) = 0 Then
  GD_Monto.Text = "0.00"
  Exit Sub
End If
'
If SD_PosPto <> 0 Then
'
' Separa entero
SD_LonCmp = SD_PosPto - 1
SD_Decimal = "." + Mid$(GD_Monto, SD_PosPto + 1, 2)
SD_Entero = Mid$(GD_Monto, 1, SD_PosPto - 1)
If Len(SD_Entero) = 0 Then
  SD_Entero = "0"
  SD_LonCmp = 1
End If
Else
SD_Decimal = ".00"
If Len(GD_Monto) = 0 Then
  SD_Entero = "0"
  SD_LonCmp = 1

```

```

Else
    SD_Entero = GD_Monto
    SD_LonCmp = Len(GD_Monto)
End If
End If
'
SD_MonFin = SD_Entero & SD_Decimal
GD_Monto.Text = SD_MonFin
End Sub

```

```

Sub GP_Valida_Numero (ByVal GD_Cadena As String)
    Dim SD_Cuenta As Integer
    Dim SD_Longitud As Integer
    Dim SD_Caracter As String * 1
    '
    SD_Longitud = Len(GD_Cadena)
    GD_EsNumero = True
    For SD_Cuenta = 1 To SD_Longitud
        SD_Caracter = Mid$(GD_Cadena, SD_Cuenta, 1)
        If (Asc(SD_Caracter) < 48) Or (Asc(SD_Caracter) > 57) Then
            GD_EsNumero = False
        End If
    Next SD_Cuenta
End Sub

```

```

Sub GP_Valida_Usuario (GD_UsuEnt As String)
    '
    ' Inicializa valores
    GD_LongReg = Len(GD_RegUsuario(1))
    '
    On Error GoTo RutinaErrorValida
    '
    ' Abre el archivo de usuarios
    Open GC_ArchUsuarios For Random As #1 Len = GD_LongReg
    '
    GD_NumReg = LOF(1) / GD_LongReg
    '
    GD_Existe = False
    For GD_CuentaRegs = 1 To GD_NumReg
        Get #1, , GD_RegUsuario(GD_CuentaRegs)
        If UCase$(RTrim$(GD_RegUsuario(GD_CuentaRegs).GD_CodUsu)) =
            GD_UsuEnt Then
            GD_Existe = True
        End If
    Next GD_CuentaRegs
End Sub

```

```

GD_Clave = UCase$(RTrim$(GD_RegUsuario(GD_CuentaRegs).GD_ClvUsu))
GD_NivEnt = Left$(GD_RegUsuario(GD_CuentaRegs).GD_NivUsu, 1)
Exit For
End If
Next GD_CuentaRegs
Close #1
SalValida:
Exit Sub
RutinaErrorValida:
MsgBox MSG_E001, 48, Error
GD_HayErr = True
Resume SalValida
End Sub

```

Variables y constantes globales.

```

'
' *****
' *      Declaración de datos constantes      *
' *****
'
Global Const GC_Cm = 567           ' Mide 1 cm
Global Const GC_5mm = 283         ' Mide 1/2 cm
Global Const GC_15mm = 850        ' Mide 1,1/2 Cm
Global Const GC_2cm = 1134        ' Mide 2 cm
Global Const GC_MaxInt = 3         ' Número máximo de intentos de conexión al sistema
Global Const GC_ArchServicios = "C:\BANCEMP\SERVICIO.DAT" ' Nombre del
                                                                    archivo de servicios
Global Const GC_ArchUsuarios = "C:\BANCEMP\USUARIO.DAT" ' Nombre del
                                                                    archivo de usuarios
Global Const GC_ArchCuentas = "C:\BANCEMP\CUENTA.DAT"   ' Nombre del
                                                                    archivo de cuentas
Global Const GC_ArchCliente = "C:\BANCEMP\CLIENTE.DAT" ' Nombre del archivo
                                                                    de cliente
'
Global Const GC_MaxRegSer = 10     ' Máximo de registros de servicios
Global Const GC_MaxRegUsu = 10    ' Máximo de registros de usuarios
Global Const GC_MaxRegCue = 10    ' Máximo de registros de cuentas
Global Const GC_LonCau = 3        ' Longitud causal
'
Global Const GC_Niv0 = "0-Bajo"    ' Nivel de usuario
Global Const GC_Niv1 = "1-Alto"   ' Nivel de usuario
Global Const GC_Bajo = "0"
Global Const GC_Alto = "1"

```

```
Global Const GC_Bajo = "0"
Global Const GC_Alto = "1"
```

```
' Botones presionados en una MSG Box
```

```
Global Const GC_BotOk = 1
Global Const GC_BotCancel = 2
Global Const GC_BotAborta = 3
Global Const GC_BotReintenta = 4
Global Const GC_BotIgnora = 5
Global Const GC_BotSi = 6
Global Const GC_BotNo = 7
```

```
' Conceptos
```

```
Global Const GC_CtaCte = 1
Global Const GC_CtaAho = 2
```

```
*****
```

```
' *            Declaración de variables simples            *
```

```
*****
```

```
Global GD_Concepto        As String * 3
Global GD_Usuario         As String        ' Código de usuario
Global GD_Cliente         As String        ' Código de cliente
Global GD_Clave           As String        ' Clave secreta de usuario
Global GD_Causa           As String * 3        ' Código de causa
Global GD_Nombre          As String * 35     ' Nombre general
Global GD_Nivel           As String * 1        ' Nivel de seguridad del usuario
Global GD_Cuenta          As String * 15     ' Número de cuenta
Global GD_Cuenta1         As String * 15
Global GD_Cuenta2         As String * 15
Global GD_CueNav          As String * 15     ' Numero de cuenta para navegación
Global GD_Producto        As String * 3        ' Producto
Global GD_Moneda          As String * 2        ' Moneda
Global GD_Oficina         As String * 3        ' Sucursal
Global GD_Correlativo     As String * 6        ' Número cuenta
Global GD_Digito          As String * 1        ' Dígito verificador
Global GD_UsuEnt          As String        ' Código usuario de entrada al sistema
Global GD_NivEnt          As String * 1        ' Nivel de entrada al sistema
Global GD_ClvEnt          As String        ' Clave secreta de entrada al sistema
Global GD_ClvNue          As String        ' Nueva clave secreta
Global GD_ClvAnt          As String        ' Clave anterior
Global GD_Telefono        As String        ' Teléfono a marcar
Global GD_FecIni          As String * 6        ' Fecha inicial
Global GD_FecFin          As String * 6        ' Fecha final
Global GD_NumChe          As String * 8        ' Número de cheque
Global GD_Valmov          As String * 11      ' Importe
```

```

Global GD_NumDoc      As String * 8      ' Documento
Global GD_Texto      As String * 80
'
' Contadores y totales
Global GD_Contador1  As Integer        ' Contador de uso general
Global GD_Contador2  As Integer        ' Contador de uso general
Global GD_CuentaRegs  As Integer        ' Cuenta registros
Global GD_NumReg      As Integer        ' Número de registros
Global GD_TotRegSer   As Integer        ' Total de registros de servicios
Global GD_TotRegUsu   As Integer        ' Total de registros de usuarios
Global GD_TotRegCue   As Integer        ' Total de registros de cuentas
Global GD_Intentos    As Integer        ' Intentos de conexión al sistema
'
' Banderas y señales
Global GD_Bandera1    As Integer        ' Bandera de uso general
Global GD_Encontrado  As Integer        ' Señal de encontrado
Global GD_EsNumero    As Integer        ' Señal cadena si es número
Global GD_Existe      As Integer        ' Señal de existencia
Global GD_HayErr      As Integer        ' Señal de error
'
' Archivos
Global GD_LongReg     As Integer        ' Longitud de un registro
'
' *****
' *      Declaración de variables TYPE      *
' *****
'
Type GG_Registro_Servicio      ' Registro del archivo de servicios
    GD_NumCau      As String * 3
    GD_CodCau      As String * 3
    GD_NomCau      As String * 35
End Type
'
Global GD_RegServicio(1 To GC_MaxRegSer) As GG_Registro_Servicio
Global GD_SerPre(1 To GC_MaxRegSer)     As String * 15 'Mantiene servicios
                                          definidos en el módulo de administración
'
Type GG_Registro_Usuario      ' Registro del archivo de usuarios
    GD_NumUsu      As String * 3
    GD_CodUsu      As String * 10
    GD_NivUsu      As String * 1
    GD_ClvUsu      As String * 10
End Type
'
Global GD_RegUsuario(1 To GC_MaxRegUsu)  As GG_Registro_Usuario

```

```

Type GG_Registro_Cuenta          ' Registro del archivo de cuentas
  GD_FolCue      As String * 3
  GD_NumCue      As String * 15
  GD_NomCue      As String * 35
End Type
'
Global GD_RegCuenta(1 To GC_MaxRegCue) As GG_Registro_Cuenta
Global GD_CuePre(1 To GC_MaxRegCue)   As String * 15 ' Mantiene cuentas definidas
                                         en el módulo de administración
'
Type GG_Registro_cliente          ' Registro del archivo de cliente
  GD_CodCli      As String * 10
  GD_NomCli      As String * 35
End Type
'
Global GD_RegCliente              As GG_Registro_cliente

```

Comunicaciones.

```

'
' *****
' * Comunicación *
' *****
'
' Constantes
Global Const GC_ErcTimeOut = 2
Global Const GC_ErcRcvError = 3
Global Const GC_SERIREA = True
Global Const GC_UNIX = False
'
' Variables
Global GD_Error      As Integer      ' Hubo error
Global GD_ErrorCF    As Integer      ' Error Copy From
Global GD_AnyConex   As Integer      ' Hay conexión
Global GD_OtroMsg    As Integer      ' Señal hay más mensajes
Global GD_Elemento   As String       ' Elemento
Global GD_CtaEdit    As String * 19  ' Cuenta editada
Global GD_TransCod   As String * 5   ' Transacción
Global GD_ErrorMsg   As String * 50  ' Mensaje de error SFB
Global GD_Sesion     As String * 40  ' Sesión infoconnect

```

' Componentes del HEADER de LINC

```
Global Ispec      As String * 5
Global TranNo    As String * 6
Global InputDate As String * 7
Global ActMth    As String * 4
Global Accion    As String * 3
Global Const BlancosHea = "    " ' 8 blancos
Global HeaderLINC As String * 23
Global Header    As String
```

' Registro de respuesta

```
Type GG_Registro_Respuesta
  GD_Todo As String * 1920
End Type
```

```
Global GD_Respuesta As GG_Registro_Respuesta
Global GD_Mensaje As GG_Registro_Respuesta
Global GD_LongResp As Integer
```

' Para uso del MODEM

```
Global Const GC_PulsoTono = "ATDP" ' ATDP teléfono de pulsos y ATDT teléfono de tonos
Global Const GC_Telbanco = "2081800" ' Teléfono del banco
Global Const GC_Colgar = "ATH" ' Colgar teléfono
Global Const GC_Puerto = "COM2" ' Puerto de comunicaciones
```

' Funciones

Function GF_ErrorCF (MEN_CODPA As String, MEN_TEXTO As String)

```
  If (Mid$(MEN_CODPA, 2, 4) = "E312") Then
    GD_OtroMsg = True
    GD_Error = False
    GD_ErrorCF = False
  ElseIf (Mid$(MEN_CODPA, 2, 4) = "E313") Then
    GD_OtroMsg = False
    GD_Error = False
    GD_ErrorCF = False
  ElseIf (Mid$(MEN_CODPA, 2, 4) <> "E000") Then
    MsgBox MEN_TEXTO, 16, "ERROR"
    GD_ErrorCod = True
    GD_ErrorCF = True
  End If
```

End Function

Function GF_Traduc (Combo As Control,Codigo As String) As String
Dim SD_Len As Integer

```

SD_Len = Len(Codigo)
For SD_Incr = 0 To Combo.ListCount - 1
    If Codigo = Left$(Combo.List(SD_Incr), SD_Len) Then
        GF_Traduc = Right$(Combo.List(SD_Incr), 50)
        Exit For
    End If
Next SD_Incr
End Function

```

Sub GP_Carga_Vars ()

```

TranNo = "000000"
GD_Usuario = "      "
GD_Clave = "      "
GD_Terminal = "ET11253 "
GD_AnyConex = False
End Sub

```

Sub GP_ClsHost ()

```

TranNo = Format$(Val(TranNo) + 1, "000000")
InputDate = Format$(Now, "ddmmmyy")
ActMth = Format$(Now, "yyymm")
GD_Mensaje.GD_Todo = "DX440T" + TranNo + InputDate + ActMth + "      " +
                    GD_Clave ' 14 blancos despues de ActMth

GP_SndHost
GP_RcvHost
End Sub

```

Sub GP_Deshabilitar ()

```

Screen.MousePointer = 0
End Sub

```

Sub GP_FormatHea (ISPEC2 As String, Accion2 As String)

```

TranNo = Format$(Val(TranNo) + 1, "000000") ' Incrementa número de transacción
InputDate = Format$(Now, "ddmmmyy")
ActMth = Format$(Now, "yyymm")
HeaderLINC = ISPEC2 + "T" + TranNo + InputDate + ActMth
Header = HeaderLINC + Accion2 + BlancosHea
End Sub

```

```

Sub GP_Habilitar ()
  Screen.MousePointer = 0
End Sub

```

```

Sub GP_Signo (Tipo As String, Sig As String, Monto As String)
  Select Case Right$(Monto, 1)
    Case "{"
      Mid$(Sig, 1, 1) = "+"
      Mid$(Monto, Len(Monto), 1) = "0"
    Case "A" To "I"
      Mid$(Sig, 1, 1) = "+"
      Mid$(Monto, Len(Monto), 1) = Chr$(Asc(Right$(Monto, 1)) - 16)
    Case "}"
      Mid$(Sig, 1, 1) = "-"
      Mid$(Monto, Len(Monto), 1) = "0"
    Case "J" To "R"
      Mid$(Sig, 1, 1) = "-"
      If Tipo = "S" Then
        Mid$(Monto, Len(Monto), 1) = Chr$(131 - Asc(Right$(Monto, 1)))
      Else
        Mid$(Monto, Len(Monto), 1) = Chr$(Asc(Right$(Monto, 1)) - 25)
      End If
  End Select
End Sub

```

```

Sub GP_SndHost ()
  GD_Error = False
  WriteHost GD_Sesion, GD_Mensaje.GD_Todo
End Sub

```

Inicializaciones.

```

Sub GP_ActualizaGriCue ()
  GD_TotRegCue = 0
  For GD_Contador1 = 1 To GC_MaxRegCue
    AAMCU.GriCuentas.Col = 0
    AAMCU.GriCuentas.Row = GD_Contador1
    AAMCU.GriCuentas.Text = GD_RegCuenta(GD_Contador1).GD_FolCue
  Next GD_Contador1
End Sub

```

```

AAMCU.GriCuentas.Col = 1
AAMCU.GriCuentas.Row = GD_Contador1
AAMCU.GriCuentas.Text = GD_RegCuenta(GD_Contador1).GD_NumCue
'
' Actualiza número de registros con información
GD_Cuenta = AAMCU.GriCuentas.Text
If GD_Cuenta <> " " Then
    GD_TotRegCue = GD_TotRegCue + 1
End If
'
AAMCU.GriCuentas.Col = 2
AAMCU.GriCuentas.Row = GD_Contador1
AAMCU.GriCuentas.Text = " " + GD_RegCuenta(GD_Contador1).GD_NomCue
Next GD_Contador1
End Sub

```

Sub GP_ActualizaGriSer ()

```

GD_TotRegSer = 0
For GD_Contador1 = 1 To GC_MaxRegSer
    AAMSE.GriServicios.Col = 0
    AAMSE.GriServicios.Row = GD_Contador1
    AAMSE.GriServicios.Text = GD_RegServicio(GD_Contador1).GD_NumCau
'
    AAMSE.GriServicios.Col = 1
    AAMSE.GriServicios.Row = GD_Contador1
    AAMSE.GriServicios.Text = GD_RegServicio(GD_Contador1).GD_CodCau
'
    ' Actualiza número de registros con información
    GD_Causa = AAMSE.GriServicios.Text
    If GD_Causa <> " " Then
        GD_TotRegSer = GD_TotRegSer + 1
    End If
    AAMSE.GriServicios.Col = 2
    AAMSE.GriServicios.Row = GD_Contador1
    AAMSE.GriServicios.Text = GD_RegServicio(GD_Contador1).GD_NomCau
Next GD_Contador1
End Sub

```

Sub GP_ActualizaGriUsu ()

```

GD_TotRegUsu = 0
For GD_Contador1 = 1 To GC_MaxRegUsu
    DXMUS.GriUsuarios.Col = 0
    DXMUS.GriUsuarios.Row = GD_Contador1

```

```

DXMUS.GriUsuarios.Text = GD_RegUsuario(GD_Contador1).GD_NumUsu
'
DXMUS.GriUsuarios.Col = 1
DXMUS.GriUsuarios.Row = GD_Contador1
DXMUS.GriUsuarios.Text = GD_RegUsuario(GD_Contador1).GD_CodUsu
'
' Actualiza número de registros con información
GD_Usuario = DXMUS.GriUsuarios.Text
If GD_Usuario <> " " And GD_Usuario <> "" Then
    GD_TotRegUsu = GD_TotRegUsu + 1
End If
'
DXMUS.GriUsuarios.Col = 2
DXMUS.GriUsuarios.Row = GD_Contador1
DXMUS.GriUsuarios.Text = GD_RegUsuario(GD_Contador1).GD_NivUsu
Next GD_Contador1
End Sub

```

Sub GP_IniciaGriCuentas (GriCuentas As Control)

```

' Ajusta dimensiones del Grid
AAMCU.GriCuentas.Cols = 3
AAMCU.GriCuentas.Rows = 11
' Titulos
AAMCU.GriCuentas.Row = 0
AAMCU.GriCuentas.Col = 2
AAMCU.GriCuentas.Text = " Descripción"
AAMCU.GriCuentas.Col = 1
AAMCU.GriCuentas.Text = " Cuenta"
AAMCU.GriCuentas.Col = 0
AAMCU.GriCuentas.Text = " "
' Altura de renglones
AAMCU.GriCuentas.RowHeight(0) = GC_5mm
AAMCU.GriCuentas.RowHeight(1) = GC_5mm
AAMCU.GriCuentas.RowHeight(2) = GC_5mm
AAMCU.GriCuentas.RowHeight(3) = GC_5mm
AAMCU.GriCuentas.RowHeight(4) = GC_5mm
AAMCU.GriCuentas.RowHeight(5) = GC_5mm
AAMCU.GriCuentas.RowHeight(6) = GC_5mm
AAMCU.GriCuentas.RowHeight(7) = GC_5mm
AAMCU.GriCuentas.RowHeight(8) = GC_5mm
AAMCU.GriCuentas.RowHeight(9) = GC_5mm
AAMCU.GriCuentas.RowHeight(10) = GC_5mm
' Ancho de columnas

```

```

AAMCU.GriCuentas.ColWidth(0) = 285
AAMCU.GriCuentas.ColWidth(1) = 1725
AAMCU.GriCuentas.ColWidth(2) = 3810
End Sub

```

Sub GP_IniciaGriServicios (GriServicios As Control)

```

'
' Ajusta dimensiones del Grid
AAMSE.GriServicios.Cols = 3
AAMSE.GriServicios.Rows = 11
'
' Titulos
AAMSE.GriServicios.Row = 0
AAMSE.GriServicios.Col = 2
AAMSE.GriServicios.Text = "                Descripción"
AAMSE.GriServicios.Col = 1
AAMSE.GriServicios.Text = " Causal"
AAMSE.GriServicios.Col = 0
AAMSE.GriServicios.Text = " Número"
'
' Altura de renglones
AAMSE.GriServicios.RowHeight(0) = GC_5mm
AAMSE.GriServicios.RowHeight(1) = GC_5mm
AAMSE.GriServicios.RowHeight(2) = GC_5mm
AAMSE.GriServicios.RowHeight(3) = GC_5mm
AAMSE.GriServicios.RowHeight(4) = GC_5mm
AAMSE.GriServicios.RowHeight(5) = GC_5mm
AAMSE.GriServicios.RowHeight(6) = GC_5mm
AAMSE.GriServicios.RowHeight(7) = GC_5mm
AAMSE.GriServicios.RowHeight(8) = GC_5mm
AAMSE.GriServicios.RowHeight(9) = GC_5mm
AAMSE.GriServicios.RowHeight(10) = GC_5mm
' Ancho de columnas
AAMSE.GriServicios.ColWidth(0) = 825
AAMSE.GriServicios.ColWidth(1) = 825
AAMSE.GriServicios.ColWidth(2) = 3585
AAMSE.GriServicios.ColAlignment(1) = 2
End Sub

```

Sub GP_IniciaGriUsuarios (GriUsuarios As Control)

```
' Ajusta dimensiones del Grid
```

```
DXMUS.GriUsuarios.Cols = 3
```

```
DXMUS.GriUsuarios.Rows = 11
```

```
,
```

```
' Titulos
```

```
DXMUS.GriUsuarios.Row = 0
```

```
DXMUS.GriUsuarios.Col = 2
```

```
DXMUS.GriUsuarios.Text = "Nivel"
```

```
DXMUS.GriUsuarios.Col = 1
```

```
DXMUS.GriUsuarios.Text = "Usuario"
```

```
DXMUS.GriUsuarios.Col = 0
```

```
DXMUS.GriUsuarios.Text = "Número"
```

```
,
```

```
' Altura de renglones
```

```
DXMUS.GriUsuarios.RowHeight(0) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(1) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(2) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(3) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(4) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(5) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(6) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(7) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(8) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(9) = GC_5mm
```

```
DXMUS.GriUsuarios.RowHeight(10) = GC_5mm
```

```
,
```

```
' Ancho de columnas
```

```
DXMUS.GriUsuarios.ColWidth(0) = 810
```

```
DXMUS.GriUsuarios.ColWidth(1) = 1500
```

```
DXMUS.GriUsuarios.ColWidth(2) = 525
```

```
,
```

```
DXMUS.GriUsuarios.ColAlignment(2) = 2
```

```
End Sub
```

VI.4 PROCEDIMIENTOS DE SEGURIDAD.

La seguridad está basada en la conjunción de dos elementos: La seguridad local en el sistema BANCEMP y la seguridad central del sistema SFB.

VI.4.1 SEGURIDAD LOCAL.

La seguridad local está controlada por una ventana inicial de acceso al sistema, donde debe ingresarse el código de un usuario junto con su clave secreta. Este usuario debe existir dentro del catálogo de usuarios definido en el módulo de administración del sistema. Como medida preventiva existe un número máximo de intentos continuos fallidos de conexión, que al sobrepasarse, el sistema queda completamente bloqueado y solo podrá reactivarse con la ayuda de algún supervisor.

Los usuarios pueden pertenecer a una de dos categorías: nivel alto (1) o nivel bajo (0). Los usuarios de nivel bajo sólo pueden hacer consultas dentro del módulo de comunicaciones y únicamente tienen derecho a modificar su clave secreta dentro del módulo de administración; quedando completamente imposibilitados para realizar cualquier otra función dentro del sistema.

Los usuarios de nivel alto tienen total libertad para realizar cualquier función dentro de los dos módulos del sistema BANCEMP, incluyendo cambiar de nivel, eliminar o incluir a otro usuario.

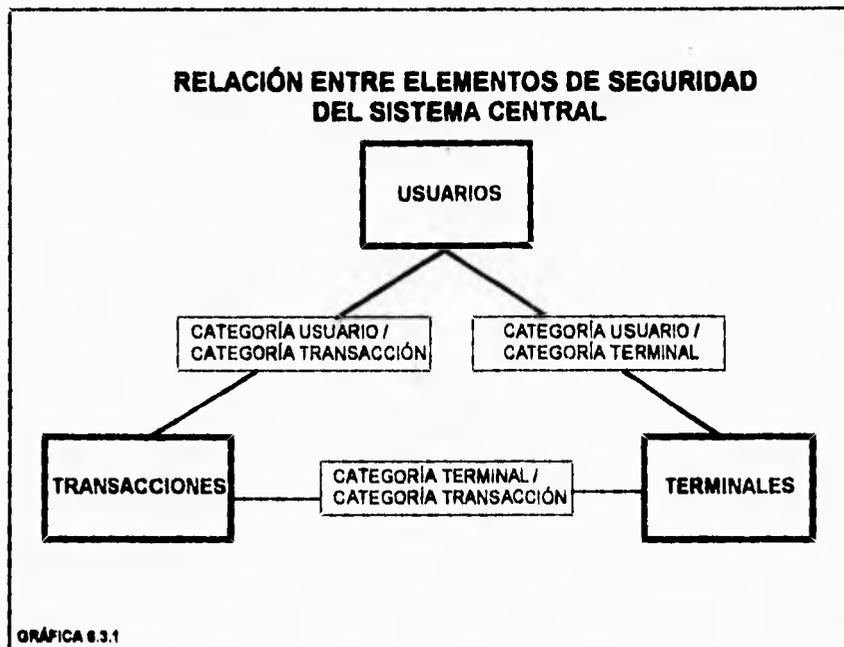
Con respecto a las claves secretas, éstas son creadas por el propio usuario y son almacenadas después de haber pasado por un proceso de *encriptación*.

VI.4.2 SEGURIDAD CENTRAL.

La seguridad en el sistema central está basada en la unificación de tres elementos :

- Usuarios SFB
- Transacciones y
- Terminales

Dicha unificación se establece en base a crear relaciones entre *categorías* de estos elementos. Así, cada usuario pertenece a una *categoría* de usuario, cada transacción pertenece a una *categoría* de transacción y cada terminal pertenece a una *categoría* de terminal. De tal manera que al establecer relaciones entre dichas categorías el resultado será que cierta transacción válida dentro de SFB sólo podrá ejecutarse por cierto(s) usuario(s) válido(s) en cierta(s) terminal(es) válida(s). Ver gráfica 6.3.1.



Acceso al sistema.

Para que un usuario BANCEMP pueda ingresar a sus transacciones dentro del sistema SFB debe cumplirse lo siguiente :

1- Se debe haber ejecutado la transacción de conexión de terminales BANCEMP. El usuario SFB utilizado es, en este caso, el código del cliente. Además, se debe ingresar una palabra clave que es proporcionada por el banco y una vez utilizada no puede volver a usarse. El banco proporcionará periódicamente una lista de claves válidas al cliente, bajo los esquemas de seguridad pactados. Estas claves deben utilizarse en el orden consecutivo en el que aparecen y el banco sólo estará en posibilidad de indicarle al cliente la última clave usada en caso de que éste lo solicite.

3- La terminal BANCEMP debe estar definida en el catálogo de terminales SFB.

4- La transacción debe estar definida en el catálogo de transacciones SFB.

5- Debe existir una categoría de usuario BANCEMP y ésta debe estar relacionada con la categoría a la que pertenece la transacción.

6- Debe existir una categoría de terminales BANCEMP y ésta debe estar relacionada con la categoría de usuarios BANCEMP.

7- La categoría de terminal BANCEMP debe estar relacionada con la categoría a la que pertenece la transacción; ya sea BANCEMP u otra, si es que existieran transacciones compartidas con otros usuarios.

8- Las transacciones son capaces de distinguir su origen (i.e. desde que tipo de terminal están siendo ejecutadas). Por lo tanto la terminal debe estar definida con un tipo de terminal equivalente al que la transacción supone como origen.

Las validaciones de seguridad para usuario, terminal y transacción se realizan en forma similar a la seguridad sobre los demás usuarios de SFB. Pero debido a que el usuario BANCEMP es, además, un cliente, se realiza una validación sobre las cuentas con las que tiene alguna relación. De esta manera es posible establecer las cuentas sobre las que puede operar el cliente y que tipo de transacción puede realizar sobre ellas.

Atributos de Usuario.

Además de la categoría de usuario BANCEMP que debe definirse en el sistema central, se define para cada usuario su capacidad para realizar adiciones, modificaciones o eliminaciones en transacciones administrativas de mantenimiento, si es que tuviera acceso a alguna de ellas.

Atributos de transacciones.

Además de la categoría de transacciones, se define para cada transacción su aptitud para recibir adiciones, modificaciones o eliminaciones, si corresponde.

Atributos de terminales.

Además de la categoría BANCEMP que debe definirse en el sistema central para nuestras terminales, se especifica para cada una de ellas su capacidad para realizar desde allí adiciones, modificaciones o eliminaciones sobre transacciones administrativas de mantenimiento, si es que tuviera acceso a alguna de ellas.

Además, se define el número máximo de intentos fallidos continuos de conexión que son permitidos antes de provocar su deshabilitación. También se puede asociar un tiempo de espera de desconexión en minutos, el cual establece el tiempo máximo que puede transcurrir entre cualquier par de transacciones ejecutadas desde esa terminal, sin asumir que ésta debe realizar nuevamente la transacción de conexión al sistema.

CAPÍTULO VII

PRUEBAS DEL SISTEMA

VII. PRUEBAS DEL SISTEMA.

El desarrollo de sistemas de software involucra una serie de actividades de producción en las que las posibilidades de error humano son muy grandes. Éstos pueden empezar a darse desde el primer momento del proceso en el que los objetivos se pueden especificar en forma errónea, así como los errores que aparecen en los pasos posteriores de diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse en forma perfecta, el desarrollo de sistemas debe ir acompañado de una actividad que garantice la calidad.

Existen una serie de reglas que sirven como objetivos de prueba:

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

Si la prueba se lleva a cabo con éxito, descubrirá errores en el software, como ventaja secundaria la prueba demuestra hasta que punto las funciones del software parecen funcionar de acuerdo a las especificaciones originales y hasta que punto parecen alcanzar los requerimientos del usuario. Además, los datos que se van recogiendo a medida que se realiza la prueba proporcionan una buena indicación de la confiabilidad del software y de alguna manera indican la calidad del producto.

VII.1 MÉTODOS DE PRUEBA.

Cualquier producto de ingeniería puede ser probado de una de dos formas : 1) Conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es operativa ("*Prueba de caja negra*"). 2) Conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que "todas las piezas encajan", es decir, que la operación interna se ajusta a las especificaciones ("*Prueba de caja blanca*").

VII.1.1 PRUEBA DE CAJA BLANCA.

La prueba de caja blanca es un método de diseño de casos de prueba mediante los cuales:

- Se garantice que se ejecuten por lo menos una vez todos los caminos independientes de cada módulo.
- Se ejecuten todas las decisiones lógicas en sus caras verdadera y falsa.
- Se ejecuten todos los bucles en sus límites.
- Se ingrese a las estructuras de datos internas para asegurar su validez.

VII.1.2 PRUEBA DE CAJA NEGRA.

Los métodos de la caja negra se centran en los requerimientos funcionales del software y permiten al ingeniero de sistemas, derivar conjuntos de condiciones de entrada que ejerciten todos los requerimientos funcionales de un programa. La prueba de la caja negra es un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de la caja blanca.

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

- Funciones incorrectas o ausentes.
- Errores de interfaz
- Errores en estructuras de datos o en bases de datos externas.
- Errores de iniciación y terminación.

VII.1.3 PRUEBA DE SEGURIDAD.

Cualquier sistema basado en computadora que maneje información sensible o lleve a cabo acciones que puedan perjudicar o beneficiar a los individuos es objeto de pretensiones impropias o ilegales. La prueba de seguridad es para verificar que los mecanismos de protección incorporados en el sistema lo protegerán de accesos no autorizados.

VII.2 PRUEBAS A BANCEMP.***Pruebas de caja blanca.***

Visual Basic es un lenguaje orientado a objetos y conducido por eventos por lo que es muy fácil aplicar pruebas de caja blanca a cada uno de los bloques de lógica asociada a cada objeto. A continuación se presentan algunos ejemplos de caso de prueba asignando valores a las variables utilizadas por algunas de las rutinas presentadas en el capítulo 6.

Caso 1) Ventana de usuarios del módulo de administración.

Valores de entrada	Respuesta esperada	Respuesta BANCEMP
Usuario: Ana Nivel: 0 Acción: Eliminar	Usuario no existe	Usuario no existe
Usuario: David Nivel: 0 Acción: Agregar	Usuario ya existe	Usuario ya existe
Usuario: Gloria Nivel: 0 Acción: Agregar	Usuario agregado	Usuario agregado

Caso 2) Ventana de cuentas del módulo de administración.

Valores de entrada	Respuesta esperada	Respuesta BANCEMP
Cuenta: 1111111111111111 Descripción: Depto 1 Acción: Agregar	Cuenta ya existe	Cuenta ya existe
Cuenta: 1A11111111111111 Descripción: Depto 1 Acción: Agregar	Número de cuenta incorrecto	Número de cuenta incorrecto
Cuenta: 1111111111111111 Descripción: Depto 1 Acción: Eliminar	Cuenta eliminada	Cuenta eliminada

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

Caso 3) Ventana de consulta de saldos del módulo de comunicaciones.

Valores de entrada	Respuesta esperada	Respuesta BANCEMP
Cuenta: 999MM999999999	Número de cuenta incorrecto	Número de cuenta incorrecto
Cuenta: 999999999999999	Cuenta no existe	Cuenta no existe
Cuenta: 10001020000034	Saldos en pantalla	Saldos en pantalla

Caso 4) Ventana de pago de servicios del módulo de comunicaciones.

Valores de entrada	Respuesta esperada	Respuesta BANCEMP
Cuenta: 999MM999999999 Servicio: 999 Monto: 100.00	Número de cuenta incorrecto	Número de cuenta incorrecto
Cuenta: 100010020000034 Servicio: 800 Monto: 100.00	Código de servicio no existe	Código de servicio no existe
Cuenta: 100010020000034 Servicio: 801 Monto: 100.00	Transacción procesada	Transacción procesada

Pruebas de caja negra.

El principal requerimiento del sistema BANCEMP es el de poder realizar ciertas transacciones desde el PC. La prueba de caja negra consiste en comprobar que todas las transacciones definidas en BANCEMP son atendidas por el sistema central SFB y tomar los tiempos de respuesta.

Transacción	Atendida por SFB	Tiempo de respuesta (s)
Cuentas del cliente	Si	3
Consulta de saldos	Si	2.3
Movimientos de una cuenta	Si	3.2
Estado de un cheque	Si	3
Consulta de chequeras	Si	3

Clientes relacionados	Si	2
Transferencia de fondos	Si	2
Pago de servicios	Si	2
Mensajes	No	-

Pruebas de seguridad.

Dado que BANCEMP está diseñado para ingresar a un sistema bancario, es muy importante asegurar la integridad de la información y permitir el acceso sólo a personas autorizadas. Es por eso que dividiremos las pruebas de seguridad en dos partes: 1) Seguridad local y 2) Seguridad central.

1) Seguridad local:

Condiciones	Resultado
Intentar ingresar al sistema BANCEMP con un usuario no autorizado.	Mensaje de error. No ingresa al sistema.
Intentar ingresar al sistema BANCEMP con un usuario no autorizado 3 veces consecutivas.	El sistema queda bloqueado.
Ingresar al sistema con un código de usuario de nivel bajo e intentar consultar otros usuarios.	No lo permite. Sólo permite modificar su clave secreta dentro del módulo de administración.
Ingresar al sistema con un código de usuario de nivel bajo e intentar entrar al menú de transferencias en el módulo de comunicaciones.	No lo permite. Sólo permite entrar al menú de consultas.

2) Seguridad central:

Condiciones	Resultado
Intentar conectarse a SFB sin haber definido la terminal.	Mensaje de error. No ingresa al sistema.
Definir la terminal e intentar conexión con una clave de cliente errónea.	Mensaje de error. No ingresa al sistema.
Intentar cursar alguna transacción BANCEMP del módulo de comunicaciones sin haberla definido en SFB.	Mensaje de error: Transacción no definida.
Definir las transacciones BANCEMP en SFB y cursar alguna de ellas con cuentas ajenas.	Mensaje de error: Transacción no autorizada.
Cursar alguna transacción con cuentas propias.	Se ejecutan exitosamente.
Permanecer conectado a SFB por 10 minutos sin cursar ninguna transacción. Después intentar hacer una consulta.	Mensaje de error: Usuario desconectado.

NOTAS:

1- Para las pruebas se utilizó el sistema bancario SFB versión 3.1.0 instalado en un equipo A7 de UNISYS de México.

2- Dicha versión del sistema no tiene implementada la transacción de mensajes.

3- El sistema BANCEMP se instaló en una PC 386 a 33 Mhz.

CONCLUSIONES

CONCLUSIONES.

El obtener información a todos los niveles dentro de una empresa constituye un factor primordial dentro del proceso de toma de decisiones. En este sentido el sistema banca en su empresa **BANCEMP** podría considerarse como una fuente adicional de información muy importante que requiere la persona encargada de tomar la decisión apropiada en el momento oportuno.

Dado los problemas económicos por los que atraviesa el país y los costos que involucra el sistema **BANCEMP**, éste quizás sea accesible sólo a empresas, en un corto plazo; pero con el tiempo, con las mejoras económicas y con los avances tecnológicos, el sistema **BANCEMP** estará en muchos hogares beneficiando también a particulares.

El éxito de **BANCEMP** depende en gran medida de la infraestructura de comunicaciones y de la confiabilidad de las líneas telefónicas. En nuestro caso, la infraestructura de comunicaciones dependerá de los requerimientos y necesidades del banco que decida adquirir el sistema; quedando el problema de las líneas telefónicas públicas que, en nuestro país, son de confiabilidad media.

El sistema bancario **SFB** es un producto registrado de **UNISYS Corporation** y tiene una base instalada en varios países de Latinoamérica incluyendo a México. Esto implica que ya existe un mercado potencial para el sistema **BANCEMP**, que también será un producto registrado de **UNISYS Corporation**. Además, el sistema **SFB** pronto podrá ser instalado en equipos de plataforma abierta (Serie U de **UNISYS**); y dado que **BANCEMP** utiliza **infoconnect** - software de comunicaciones que también maneja el protocolo **TCP/IP** -, el mercado de **BANCEMP** crecerá aún más.

BANCEMP cuenta ahora con un número fijo de transacciones y, para poder agregarle más, tendríamos que hacer pequeñas modificaciones a las transacciones equivalentes dentro del sistema central **SFB**. Esto podría implicar el tener que contar con personal capacitado en el sistema **SFB** y por lo tanto que conociera el ambiente de desarrollo **LINC**; y por otra parte, personal capacitado en **Visual Basic**. Aún así, ambos sistemas están diseñados de tal forma que dichas modificaciones no son complicadas y por ende el personal no tiene que ser especializado.

Se logró el objetivo principal del proyecto: lograr la comunicación entre el sistema **BANCEMP** y el sistema central bancario **SFB**, quedando para un futuro proyecto la implantación de toda una infraestructura de telecomunicaciones para lograr la comunicación simultánea de varios sistemas **BANCEMP** con **SFB**.

El haber seguido la metodología LSA para la realización de este trabajo dio como resultado un producto flexible, estandarizado, de fácil mantenimiento y apegado a los requerimientos del usuario.

El sistema BANCEMP, objeto de esta tesis, es el primer esfuerzo en el desarrollo del sistema banca en su empresa bajo ambiente windows en UNISYS de México y por tanto seguirá siendo objeto de mejoras basadas en las tendencias del mercado y en futuras experiencias.

BIBLIOGRAFÍA

BORJA MARTÍNEZ , Francisco
El Nuevo Sistema Financiero Mexicano
Fondo de Cultura Económica
México 1991

STAMPER, David A.
Business Data Communications
Benjamin/Cumming
Reswood, California 1991

UNISYS CORPORATION
Sistema Financiero Bancario (SFB)
Manual de Capacidades 3.1.1
Chile 1991

CEBALLOS SIERRA, Francisco Javier
Microsoft ® VISUAL BASIC™ para aplicaciones Windows™
RA-MA
España 1993

HOLZNER, Steven
VISUAL BASIC™ FOR WINDOWS™
BRADY Publishing
USA 1993

UNISYS and ICC Corporations
INFOCONNECT™ INTERCOM® User Reference Guide
USA 1992

CHRISTIE, LINDA
Enciclopedia de términos de microcomputación
Prentice Hall
México, 1986

APÉNDICE A

Módulos del SFB

Código

AA
AC
AH
BS
DX
FF
MD

Significado

Sistema central de apoyo
Cuentas de cheques
Cuentas de ahorros
Sistema de clientes
Apoyo al sistema
Inversiones
Mesa de dinero

APÉNDICE B

Tipos de transacciones

<i>Código</i>	<i>Significado</i>
A	Archivo de mantenimiento
C	Consulta
M	Archivo maestro
T	Transacción

APÉNDICE C

Tipos de controles Visual Basic

<i>Código</i>	<i>Significado</i>
Ani	Animación
Arc	Lista de archivos
Chk	Caja de selección
Dat	Control de datos
Dir	Lista de directorios
Drv	Lista de unidades de disco
Frm	Marco
Gri	Matriz
Hsb	Barra horizontal desplazamiento
Img	Imagen
Lin	Línea
Opt	Opción
Pic	Dibujo
Shp	Figura
Tim	Temporizador
Vsb	Barra vertical desplazamiento

GLOSARIO

A

A.M. (Amplitud Modulada)

API de Windows (Application Program Interface) Interfaz de programación de aplicaciones Windows cuya característica principal son las funciones y los mensajes Windows. Estas funciones son el corazón de las aplicaciones Windows y hay más de 600 dispuestas a ser llamadas. Los mensajes son utilizados por Windows para permitir que dos o más aplicaciones se comuniquen entre sí y con el propio sistema Windows. Se dice que las aplicaciones Windows son conducidas por mensajes o sucesos.

ATM (Automatic Teller Machine) Cajero automático. Dispositivo electrónico utilizado por instituciones financieras para acercar algunos de sus servicios, tales como retiros de efectivo o depósitos, a sus clientes las 24 Hrs del día.

B

BANCEMP Sistema desarrollado como objetivo principal de este trabajo de tesis.

BIT Abreviatura de dígito binario. Unidad de información igual a una decisión binaria o a uno de dos posibles estados como uno o cero.

C

Causal Código asociado a una transacción del sistema SFB para poder identificarla fácilmente y poder hacer el asiento contable apropiadamente.

CECOBAN (Centro de Cómputo Bancario) Fideicomiso de la banca mexicana bajo la administración del Banco de México, que desde hace más de 5 años se constituyó como la única cámara automatizada de compensación del país.

COMS (Communication Management System) Software de administración y control de la red de comunicaciones en equipos de la Serie A de UNISYS.

Control En Visual Basic son las entidades gráficas que se pueden diseñar y colocar en una forma.

D

DLL (Dynamic Link Library) Librería dinámica que permite que las aplicaciones Windows compartan código y recursos. Una DLL es en realidad un programa ejecutable que contiene funciones de Windows que pueden ser utilizadas por todas las aplicaciones.

DMS (Data Management System) Software de administración de bases de datos de alta confiabilidad y madurez.

E

Emulación Proceso por el cual algunas computadoras pueden correr programas que no estén escritos específicamente para ellas, debido a que simulan las operaciones de otras computadoras.

F

F.M. (Frecuencia Modulada)

Forma Es una ventana Windows especialmente diseñada desde Visual Basic.

FSA FINESSE (Financial System Architecture) Herramienta de software específica para la automatización de las funciones diarias de una sucursal financiera.

G

Gbps Unidad de velocidad de transmisión de datos que especifica mil millones de bits por segundo.

H

Hardware Circuitos eléctricos y dispositivos electromecánicos que constituyen a un sistema de computación.

Host Computadora que controla las comunicaciones en una red y que administra la base de datos.

I

Ícono Representación gráfica de diversos elementos dentro de Windows, como unidades de disco, aplicaciones y documentos.

L

LED (Light Emitting Diode) Diodo emisor de luz. Semiconductor diseñado para emitir luz cuando se polariza en forma directa (cuando la corriente lo atraviesa).

LAN (Local Area Network) Red de área local. Tecnología para conectar varias computadoras que estén a unos cuantos cientos de metros de distancia entre sí.

LINC (Logic and Information Network Compiler) Es un ambiente de desarrollo que provee toda la tecnología para crear, operar y evolucionar sistemas de información.

M

Mb Unidad de cantidad de información que expresa millones de bits.

Mbps Unidad de velocidad de transmisión de información que expresa millones de bits por segundo.

Mainframe Computadora de gran escala que puede manejar y atender múltiples tareas simultáneamente.

MARC (Menu Assisted Resource Control) Software ambiental que permite efectuar las labores de operación y explotación de sistemas mediante menús en computadoras de la Serie A de UNISYS.

MCP (Master Control Program) Sistema operativo de máquinas de la serie A de UNISYS.

MODEM Dispositivo de comunicación que permite a una computadora transmitir información a través de una línea telefónica.

MOUSE Dispositivo de entrada de posicionamiento relativo con que se controla el cursor en el monitor de una microcomputadora personal.

O

Objeto En Visual Basic los objetos son ambas formas y controles.

P

PC (Personal Computer) Abreviatura de computadora personal.

R

RAM (Random Access Memory) Memoria de acceso aleatorio. Arreglo de almacenamiento interno de datos a partir del cual puede recuperarse información a una velocidad que es relativamente independiente de la localización de la información almacenada.

S

SFB (Sistema Financiero Bancario) Software aplicativo desarrollado por UNISYS Corporation especialmente para instituciones financieras latinoamericanas.

SITE Lugar donde se localiza un *Mainframe* y se centraliza el procesamiento de datos.

Software Programas y rutinas que indican a la computadora qué hacer y cuando hacerlo. También denota la *documentación*: manuales, diagramas e instrucciones del operador.

T

TEF (Transferencia electrónica de fondos) Mecanismos automatizados de transferencia de capitales entre los participantes del sistema financiero.

Transacción Datos introducidos a la computadora que son procesados y que al finalizar obtenemos registros actualizados o alguna información

V

Ventana Es un área rectangular de la pantalla en la cual se presenta una aplicación o documento. Las ventanas pueden abrirse, cerrarse y moverse, y la mayoría de ellas también pueden cambiar de tamaño. Además es posible reducirlas a un icono.

W

Windows Ambiente operativo para microcomputadoras con la característica principal de ser una interfaz de usuario gráfica.

WFL (Work Flow Language) Lenguaje que permite activar y controlar la ejecución de programas de usuario, utilerías del sistema, funciones del sistema, etc., dentro de las computadoras de la Serie A de UNISYS Corporation.